

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Факультет енергетики і комп'ютерних технологій

ЗАТВЕРДЖУЮ

Зав. каф. “Комп'ютерні науки”

доц. _____ Сергій ШАРОВ

“17” лютого 2025 р.

Пояснювальна записка
до кваліфікаційної роботи здобувача СВО Магістр
(ступінь вищої освіти)

на тему: Адаптивний веб-додаток музичних плагінів з використанням
фреймворків (React, Vue)

52/4КНД.11260551.000000ПЗ

Виконав: здобувач вищої освіти 21МБКН групи
Спеціальності 122 «Комп'ютерні науки»
за ОПП Комп'ютерні науки
(шифр і назва спеціальності та ОПП)

_____ **Іван КЕДА**
(підпис) **прізвище й ініціали**

Керівник, доц. _____ **Микола МІРОШНИЧЕНКО**
(підпис) **прізвище й ініціали**

Нормоконтроль, ст. викл. _____ **Ольга ЗИНОВ'ЄВА**
(підпис) **прізвище й ініціали**

Рецензент, доц. _____ **Ганна ЧЕРНОВА**
(підпис) **прізвище й ініціали**

Запоріжжя – 2025 рік

ТАВРІЙСЬКИЙ ДЕРЖАВНИЙ АГРОТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ДМИТРА МОТОРНОГО

Факультет: Енергетики і комп'ютерних технологій

Кафедра: Комп'ютерні науки

Ступінь вищої освіти: Магістр

Спеціальність: 122 Комп'ютерні науки

ОПП: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

к.пед.н., доц. Сергій ШАРОВ

“04” жовтня 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ**

Кеді Івану Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Адаптивний веб-додаток музичних плагінів з використанням фреймворків (React, Vue)

керівник роботи Мірошніченко Микола Юрійович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від “23” вересня 2024 року № 442-С

2. Термін подання роботи 14 лютого 2025 року

3. Вихідні дані до роботи: дані обстеження об'єкту, статистичні дані, технічне завдання на дипломне проектування, матеріали виробничих практик, нормативні документи, науково-технічна література, електронні ресурси та ін.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

1. Аналіз предметної області та існуючих рішень

2. Специфікація вимог до веб-додатку з продажу музичних плагінів і інструментів

3. Обґрунтування мови та середовища програмування

4. Дослідна експлуатація веб-додатку з продажу музичних плагінів і інструментів

5. Тестування веб-додатку

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)
Діаграма використання, діаграми IDEF0, розкадровка інтерфейсу, структурна схема, візуальні компоненти інтерфейсу, алгоритм роботи, інтерфейс користувача програми

Презентація – 10 слайдів

1. Титульний слайд
2. Предмет, об'єкт, мета, завдання дослідження
3. Порівняльна характеристика аналогів програмного продукту
4. Діаграма варіантів використання
5. Інструментальні засоби
6. Інтерфейс і робота експертної системи
7. Інтерфейс і робота експертної системи
8. Тестування
9. Висновок

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 01 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	12.09.2024	виконано
2	Специфікація вимог до веб-додатку з продажу музичних плагінів і інструментів	06.10.2024	виконано
3	Проектування веб-додатку з продажу музичних плагінів і інструментів	16.10.2024	виконано
4	Обґрунтування інструментальних засобів	05.11.2024	виконано
5	Розробка веб-додатку	10.12.2024	виконано
6	Тестування та налагодження веб-додатку	20.01.2025	виконано
7	Підпис керівником роботи	14.02.2025	виконано
8	Підпис завідувачем кафедри	17.02.2025	виконано

Здобувач вищої освіти

(підпис)

Іван КЕДА

(ім'я та прізвище)

Керівник кваліфікаційної роботи

(підпис)

Микола МІРОШНИЧЕНКО

(ім'я та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра: 84 с., 20 рис., 5 табл., 39 посилань, 1 додаток.

Актуальність: З розвитком інформаційних технологій електронна комерція стала однією з найефективніших сфер економічної діяльності, що надає сучасні можливості підприємствам, потенційним клієнтам, компаніям та організаціям. Електронна комерція охоплює широкий спектр ділових операцій, які можна здійснювати у віртуальному середовищі. Це включає торгівлю товарами та послугами, укладення агентських угод, організацію комерційних представництв, проведення фінансових операцій, надання консалтингових послуг, інжинірингові послуги, купівлю та продаж ліцензій, банківські та страхові послуги та багато інших видів операцій.

Об'єкт дослідження: електронні ресурси для здійснення електронної комерції.

Предмет дослідження: веб-сайт з продажу музичних плагінів і інструментів.

Мета роботи: розробити веб-додаток з продажу музичних плагінів і інструментів.

Задачі дослідження:

- 1.Розглянути основний функціонал веб-додатку з продажу музичних плагінів.
- 2.Проаналізувати існуючі веб-додатки з продажу музичних плагінів.
- 3.Розглянути інструментальні засоби для розробки веб-додатку.
- 4.Створити технічне завдання на розробку веб-додатку з продажу музичних плагінів і інструментів.
- 5.На основі складеного технічного завдання розробити веб-додаток, скласти інструкцію користувача.

Практичне значення дослідження: полягає в розробці веб-додатку, який дозволить продавати музичні пристрої, інструменти та програмні плагіни.

Структура і обсяг роботи: Дипломна робота складається з переліку умовних скорочень, вступу, чотирьох розділів, загальних висновків, списку використаних джерел (39 найменувань). Перший розділ присвячений аналізу специфіки музичних інтернет-магазинів, аналізу аналогічних платформ. Другий розділ присвячений складанню технічного завдання, аналізу інструментальних засобів розробки веб-додатку, опису етапів розробки. Третій розділ присвячено написанню інструкції користувача. Четвертий розділ присвячено дослідній експлуатації та тестуванню розробленого програмного забезпечення. Загальний обсяг роботи складає 84 сторінок.

SUMMARY

Master's Qualification Work: 84 pages, 20 figures, 5 tables, 39 references, 1 appendices.

Relevance: With the development of information technology, e-commerce has become one of the most effective areas of economic activity, providing modern opportunities for businesses, potential clients, companies, and organizations. E-commerce encompasses a wide range of business operations that can be conducted in a virtual environment. This includes the trade of goods and services, the conclusion of agency agreements, the organization of commercial representations, the conduct of financial operations, the provision of consulting services, engineering services, the buying and selling of licenses, banking and insurance services, and many other types of operations.

Object of Research: Electronic resources for conducting e-commerce.

Subject of Research: A website for selling music plugins and instruments.

Purpose of the Work: To develop a web application for selling music plugins and instruments.

Research Tasks:

- 1.To consider the main functionality of the web application for selling music plugins.
- 2.To analyze existing web applications for selling music plugins.
- 3.To examine the tools for developing the web application.
- 4.To create a technical specification for the development of the web application for selling music plugins and instruments.
- 5.Based on the created technical specification, to develop the web application and compile a user manual.

Practical Significance of the Research: Lies in the development of a web application that will allow the sale of musical devices, instruments, and software plugins.

Structure and Volume of the Work: The thesis consists of a list of conditional abbreviations, an introduction, three chapters, general conclusions, and a list of sources used (39 items). The first chapter is dedicated to analyzing the specifics of music online stores and analyzing similar platforms. The second chapter focuses on drafting the technical specification, analyzing the tools for developing the web application, and describing the stages of development. The third chapter is dedicated to writing the user manual. The fourth section is devoted to experimental operation and testing of the developed software. The total volume of the work is 76 pages.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	11
1.1 Напрямки використання веб технологій для електронної комерції	11
1.2 Аналіз існуючих рішень	16
1.3. Технічне завдання на розробку веб-додатку з продажу музичних плагінів	23
3.2 Вимоги до зберігання даних і мов програмування	24
РОЗДІЛ 2. СПЕЦИФІКАЦІЯ ВИМОГ ДО МУЗИЧНОГО ВЕБ-ДОДАТКУ	31
2.1 Концептуальна модель використання музичного веб-додатку	31
2.2. Функціональна модель веб-додатку	37
2.3 Специфікація функціональних та нефункціональних вимог	40
РОЗДІЛ 3. ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ	42
3.1 Загальний огляд технологічного стеку	42
3.2 Обґрунтування вибору мови програмування	45
3.3 Інструменти для розробки та тестування	52
РОЗДІЛ 4. ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ	57
4.1 Особливості програмної реалізації сайту	57
4.2. Інструкція користувача веб-додатку	63
4.3. Тестування веб-додатку	66
ВИСНОВКИ	73
ПЕРЕЛІК ЛІТЕРАТУРИ	75
Додаток А	79

ВСТУП

Актуальність. З розвитком інформаційних технологій електронна комерція стала однією з найефективніших сфер економічної діяльності, що надає сучасні можливості підприємствам, потенційним клієнтам, компаніям та організаціям. Електронна комерція охоплює широкий спектр ділових операцій, які можна здійснювати у віртуальному середовищі. Це включає торгівлю товарами та послугами, укладення агентських угод, організацію комерційних представництв, проведення фінансових операцій, надання консалтингових послуг, інжинірингові послуги, купівлю та продаж ліцензій, банківські та страхові послуги та багато інших видів операцій.

Цифровізація музичної індустрії передбачає використання онлайн-платформ, віртуальних інструментів, аранжування програмного забезпечення, відеоуроків та майстер-класів. Такі ресурси та інструменти дозволяють студентам отримувати доступ до навчальних матеріалів з будь-якої точки світу та створюють можливості експериментувати з різними стилями музики за допомогою цифрових інструментів, таких як синтезатори, електронні барабани та цифрові піаніно.

У музичній індустрії різні веб-сайти та онлайн-платформи використовуються для здійснення транзакцій електронної комерції. Наприклад, веб-сайт для музичного проекту може публікувати вміст, надавати інформацію про історію проекту та анонсувати тури та концерти. Такі сайти стають віртуальними середовищами для взаємодії з меломанами і дозволяють зручно спілкуватися через блоги, форуми або інтегруватися в соціальні мережі.

Тому веб-технології, особливо Інтернет, відіграють ключову роль у розвитку електронної комерції. Інтернет дав змогу розширити цільову аудиторію та продавати товари та послуги. Музичний бізнес зазнає впливу інформаційно-комунікаційних технологій, які розширюють спектр музичних продуктів і послуг.

Об'єкт: веб-додаток для музичної комерції.

Предмет: Адаптивний веб-додаток музичних плагінів.

Мета: розробити веб-додаток з продажу музичних плагінів і інструментів.

Сформована мета, об'єкт і предмет дослідження дозволили поставити такі **завдання**:

1. Розглянути основний функціонал веб-додатку з продажу музичних плагінів.
2. Проаналізувати існуючі веб-додатки з продажу музичних плагінів.
3. Розглянути інструментальні засоби для розробки веб-додатку.
4. Створити технічне завдання на розробку веб-додатку з продажу музичних плагінів і інструментів.
5. На основі складеного технічного завдання розробити веб-додаток, скласти інструкцію користувача.

При реалізації поставлених задач було використано наступні **методи**: теоретичні: аналіз та синтез наукової та спеціальної літератури з теорії баз даних, програмування; емпіричні: спостереження та бесіда з метою визначення специфіки продажу музичних інструментів; практичні: програмна реалізація веб-додатку.

Практичне значення дослідження полягає в розробці веб-додатку, який дозволить продавати музичні пристрої, інструменти та програмні плагіни.

Структура і обсяг роботи. Дипломна робота складається з переліку умовних скорочень, вступу, трьох розділів, загальних висновків, списку використаних джерел (39 найменувань). Перший розділ присвячений аналізу специфіки музичних інтернет-магазинів, аналізу аналогічних платформ. Другий розділ присвячений складанню технічного завдання, аналізу інструментальних засобів розробки веб-додатку, опису етапів розробки. Третій розділ присвячено написанню інструкції користувача. Загальний обсяг роботи складає 84 сторінок.

РОЗДІЛ 1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Напрямки використання веб технологій для електронної комерції

В умовах розвитку інформаційних технологій одним з дієвих напрямків економічної діяльності стала електронна комерція, що відкрила сучасні можливості для бізнесу, потенційних клієнтів, підприємств та організацій. Її використання дозволило змінити традиційні моделі продажу та поведінку споживачів. З одного боку, підприємства малого бізнесу отримали шанс конкурувати з великими компаніями долаючи бар'єри для входу на ринок і розширюючи свою діяльність через віддалену торгівлю. В свою чергу потенційні споживачі отримали доступ до широкого асортименту товарів і послуг, представленого в мережі Інтернет. Там вони можуть обирати продукти, які краще відповідають їхнім потребам.

Поняття електронної комерції сформувалося упродовж декількох десятиліть. Її можна розглядати як нову область знань, що знаходиться на перетині економіки та інформатики [7, с. 287]. Вона досліджує вплив цифрових технологій на організацію, функціонування та розвиток комерційних процесів, а також вивчає інноваційні підходи до ведення бізнесу в глобальному цифровому середовищі.

Електронна комерція має декілька суттєвих недоліків від звичайного ведення комерційної діяльності. По-перше, вона практично не вимагає значних фізичних витрат, оскільки більшість операцій відбувається в автоматизованому режимі завдяки цифровим технологіям. В цих операціях Інтернет-технології відіграє важливу роль, за допомогою яких забезпечується обробка даних і взаємодія між продавцями та покупцями в реальному часі. По-друге, електронна комерція охоплює такі аспекти, як створення ефективної системи реклами для залучення уваги споживачів, надання детальної інформації про товари через цифрові платформи та забезпечення

швидких і безпечних фінансових транзакцій. Також електронна комерція надає можливість здійснювати інформаційну підтримку клієнтів та надання їм необхідної допомоги після придбання продукції. За рахунок обробки даних клієнтів забезпечується аналіз їхніх уподобань, що дозволяє краще розуміти потреби цільової аудиторії та пропонувати персоналізовані пропозиції [23, с. 115].

Науковці вважають електронну комерцію одним з найбільш ефективних механізмів розповсюдження товарів з використанням цифрових засобів та інформування цільової аудиторії про товари та послуги [18, с. 76]. Ця перевага досягається за рахунок того, що електронна комерція передбачає реалізацію товарів та послуг через мережу Інтернет. Водночас, розрахунки за отриману продукцію здійснюються або безпосередньо у пункті видачі товару, або частіше за все через електронні засоби платежу у вигляді безготівкового розрахунку [12, с.97].

Електронна комерція охоплює широкий спектр комерційних операцій, що можуть здійснюватися у віртуальному середовищі. Вона охоплює різні види діяльності, зокрема торгівля товарами й послугами, укладання посередницьких угод, організація комерційного представництва, здійснення фінансових операцій, надання консультативних послуг, інжиніринг, купівля й продаж ліцензій, банківські операції та страхові послуги. Крім того, електронна комерція може містити інші форми промислової чи підприємницької співпраці, наприклад, управління ланцюгами постачання, підтримку стартапів та ін.

Розвиток технологій, розповсюдження цифрових пристроїв, соціальних мереж прискорив зростання можливостей електронної комерції. На думку дослідників, на розвиток цифровізації економічної діяльності значний вплив здійснюють мобільні та хмарні технології, інтернет речей, а також технологія великих даних. Наприклад, частка світового ВВП, що відноситься до використання інтернету речей (промисловий Інтернет), становить 11%, що є доволі високим показником [2, с. 100]. Також реалізація електронної

комерції забезпечується різними цифровими пристроями, що вплинуло на появу різних напрямків електронної комерції. Так, існує M-commerce (комерція, що здійснюється з використанням послуг мобільного зв'язку), T-commerce (комерція, що здійснюється з використанням інтерактивного цифрового телебачення), V-commerce (голосова комерція, що реалізується через голосові команди) [20, с. 24] тощо.

Досить велику роль у поширенні електронної комерції відіграє Інтернет. Відомо, що стрімкий розвиток сучасних технологій трансформував Інтернет у потужну інфраструктуру глобального масштабу. Ця мережа завдяки технології WEB 2.0 [36, с. 27] забезпечує швидку та ефективну взаємодію між різними структурами, надаючи користувачам доступ до безмежного обсягу інформації, яка раніше була важкодоступною або недосяжною. Вона поєднує інформаційні ресурси організацій та data-центри, забезпечує доступ до світових бібліотек, баз даних наукової, державної, правової інформації [1, с. 3] тощо. Ця мережа стала незамінним інструментом для досліджень, навчання, бізнесу та управління; вона стала не лише засобом комунікації, а й динамічним середовищем для здійснення економічної діяльності, що дуже швидко розвивається [14, с. 178].

Частіше за все електронна комерція з використанням мережі Інтернет здійснюється через електронні ресурси або онлайн сервіси, а також через чат-боти. Під електронним ресурсом розуміється сукупність сторінок в електронному вигляді, що розміщуються на Інтернет-сервері, містять гіперпосилання та пов'язані між собою єдиною системою навігації. Інтернет-сервери (або просто сервер) являють собою апаратне та програмне забезпечення, що забезпечує функціонування різноманітних сервісів електронного середовища та надають доступ користувачам мережі Інтернет до цих сервісів. Комерційні електронні ресурси (або веб-ресурси) виконують важливу функцію у розширенні ринків збуту. Особливо це стосується підприємств, що поширюють свою продукцію через Інтернет та через мережу торгових точок. Основною метою таких ресурсів є здійснення процесів

купівлі-продажу та оптимізація взаємодії з клієнтами. В Україні дуже швидко розвивається такий напрям електронної комерції, як Інтернет-торгівля. Це пояснюється появою великої кількості інтернет-магазинів [22, с. 91] та інтернет-представництв підприємств малого та середнього бізнесу. І це не дивно, адже веб-сайт або онлайн сервіс можна розробити за відносно невеликий термін часу. А за наявності хостінгу та сайту послуги або товари підприємства будуть відображені в мережі Інтернет, що вплине на збільшення кількості покупців підприємства.

В контексті електронної комерції електронні ресурси (або веб ресурси) можуть надати користувачу різноманітну інформацію, зокрема про компанію, товари чи послуги, що вона реалізує, можливості для здійснення операцій купівлі-продажу, прайслисти товарів та послуг, функцію пошуку інформації на веб ресурсі. Вони дозволяють оформити замовлення в зручному форматі, отримати детальний опис технічних характеристик товару, подивитися зображення товару, слідкувати за оновленням каталогу продукції відповідно до змін асортименту [14, с. 180]. Також поширеними прикладами електронних ресурсів для здійснення електронної комерції є Інтернет-майданчики, Інтернет-магазини, електронні біржі та Інтернет-аукціони. Вони дозволяють реалізовувати товари та послуги в онлайн-режимі, забезпечуючи простий і безпечний процес оплати за допомогою банківських систем [12, с. 97].

Інформаційно-комунікаційні технології вплинули і на музичну індустрію, що стала одним із провідників цифрових трансформацій, демонструючи прискорені темпи адаптації до інноваційних технологій. За останні два десятиліття технології докорінно змінили підходи до створення, розповсюдження та споживання музичного контенту. Наприклад, сьогодні користувачі мають миттєвий доступ до музики з будь-якого куточка світу, частіше за все безкоштовно. З іншого боку, якщо раніше музиканти отримували основний дохід від продажу дисків із звукозаписом, то з появою мереж для обміну файлів, наприклад Napster, ця модель значно втратила

актуальність. Значні зрушення відбулися у способах збереження та відтворення музики. Замість фізичних пристроїв відтворення звуку зростає кількість цифрових пристроїв, таких як смартфони, планшети та інші гаджети [19, с. 31].

Цифровізація музичної індустрії охоплює використання онлайн-платформ, віртуальних інструментів, програмного забезпечення для аранжування, а також відеоуроків та майстер-класів. Такі ресурси та інструменти забезпечують доступ до навчальних матеріалів із будь-якої точки світу та відкриває можливості для експериментів із різними стилями музики за допомогою цифрових інструментів, таких як синтезатори, електронні барабани або цифрові піаніно. Важливою складовою сучасної музичної індустрії став штучний інтелект. Прикладами можна назвати сервіси Chordify, що застосовують машинне навчання для визначення акордів із будь-якого аудіоджерела; Yousician та Melodics, що інтегрують інтерактивні інструменти для аналізу прогресу майбутнього музиканта [9, с. 4] тощо.

Для реалізації електронної комерції в музичній індустрії використовуються різноманітні сайти або онлайн майданчики. Наприклад, веб-сайти музичних проєктів публікувати свій контент, надавати інформацію про історію проєкту, анонсувати тури та концерти. Такі сайти стають віртуальним середовищем для взаємодії з любителями музики, забезпечуючи зручність спілкування через блоги, форуми або інтеграцію із соціальними мережами. Важливою складовою музикальних веб майданчиків є можливість продажу альбомів, брендированих товарів, квитків на концерти тощо. Крім того, вебресурси виконують маркетингові функції, що містить рекламу, просування та розсилки, спрямовані на розширення аудиторії та підтримку лояльності вже існуючих шанувальників [3, с. 175].

Отже, веб технології, зокрема мережа Інтернет, відіграють важливу роль для розвитку електронної комерції. Завдяки Інтернет можна збільшити цільову аудиторію, продати товари та послуги тощо. Не залишилась осторонь

і музична індустрія, що завдяки інформаційно-комунікаційним технологіям розширила асортимент товарів та послуг музичного спрямування.

1.2 Аналіз існуючих рішень

Для багатьох у сучасному швидко змінюваному світі девізом, мемом стали фрази: «Хочеш перемагати – розвивайся» та більш радикальна «Еволюційнуй або помри». І музична індустрія не є винятком в освоєнні новаторських винаходів, демонструючи чіткі ознаки прискорення у процесі цифрової трансформації. За останні двадцять років технологічні інновації докорінно змінили музичну галузь, принісши і позитивні зміни, і водночас проблеми. Наприклад, користувачі можуть насолоджуватися миттєвим доступом до музики, але при цьому артисти вимушені шукати і приймати нові моделі отримання прибутку.

На сьогоднішній день найбільшою локальною мережею, яка забезпечує людей швидким обміном інформацією вважається мережа Інтернет. До його переваг можна віднести: доступність будь-яких послуг, можливість швидкої і надійної комунікації, світовий масштаб діяльності і т. д. Саме домінування плюсів над мінусами стало підставою для розвитку таких видів діяльності, як електронний бізнес та електронна комерція. Однією із найзначніших змін, яка спостерігається у бізнесі, є перехід від традиційного способу торгівлі до електронної комерції. Різняться вони низкою параметрів, а саме: майже повна відсутність затрат, автоматизація самого процесу торгівлі і Інтернет сам створює безпосередню обробку даних. З урахуванням стрімкого розвитку Інтернет-технологій, онлайн торгівля нарощує свої обсяги у світовому масштабі і вже вважається окремою ланкою економіки.

Останнім часом все більшої популярності набуває електронна торгівля, оскільки вона є одним з найбільш ефективних механізмів інформування цільової аудиторії та розповсюдження товарів. Враховуючи та аналізуючи всі

аспекти впливу електронної комерції на розвиток економіки, в роботі було виявлено позитивні та негативні сторони поширення даного процесу. Виявлено, що високорозвинені країни показують швидкий темп розвитку електронних механізмів торгівлі і разом з тим суспільство вдосконалюється, пристосовуючись до змін. Особлива увага приділяється Європейському Союзу, який займає провідне місце у розвитку цифрової економіки та електронної комерції, адже для Європейського Союзу дана галузь є допоміжним інструментом у досягненні різносторонніх цілей розвитку. В цілому ЄС може слугувати прикладом для багатьох країн, оскільки електронна торгівля дійсно пришвидшує розвиток, підштовхує до інноваційного прогресу та просуває економіку інтеграційного об'єднання на відносно вищий рівень.

Splice - одна з провідних онлайн-платформ для музикантів і продюсерів, що пропонує широкий спектр інструментів для створення музики [35]. Платформа здобула популярність завдяки своєму інноваційному підходу та зручним функціям (рис 1.1.).

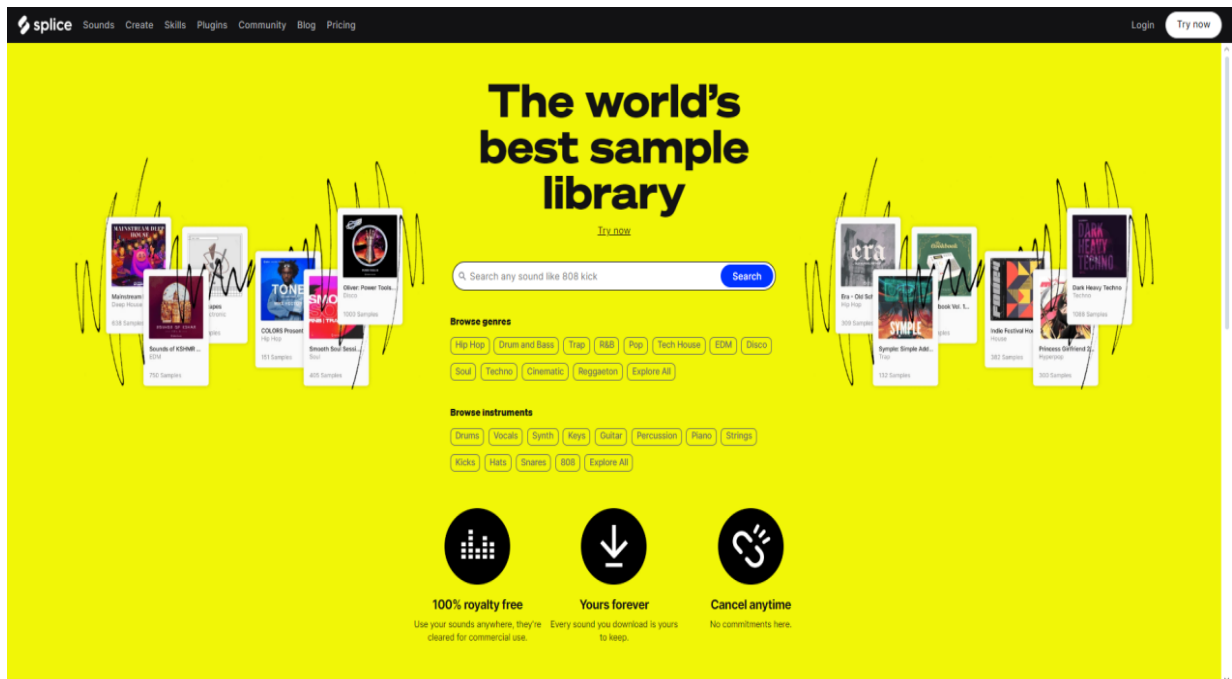


Рисунок 1.1 - Вікно музичної платформи Splice

Основні можливості

1. Велика бібліотека семплів Splice Sounds: величезний каталог семплів, петель та звукових ефектів, що охоплює різноманітні жанри та стилі. Музиканти можуть знайти те, що їм потрібно для своїх треків.

2. Плагіни та віртуальні інструменти Rental to Own: унікальна система оренди плагінів, яка дозволяє користувачам поступово купувати необхідні їм інструменти.

3. Доступний широкий вибір плагінів, включаючи звукові процесори, синтезатори та ефекти.

4. Проекти та співпраця Splice Studio - інструмент для зберігання та синхронізації проектів у хмарі. Музиканти можуть зберігати проекти та ділитися ними з іншими музикантами для спільної роботи. Також є можливість перегляду журналів змін і відновлення попередніх версій проектів.

5. Плагіни та інтеграція: платформа підтримує інтеграцію з популярними DAW (цифровими аудіо робочими станціями), такими як Ableton, Logic Pro і FL Studio. Бібліотеки семплів і корисні інструменти для організації та управління проектами.

6. Спільнота: величезна та активна спільнота музикантів, де користувачі можуть обмінюватися ідеями, отримувати підтримку та надихатися роботами інших.

7. Splice Discover: розділ, де ви можете знаходити та ділитися музикою, створеною іншими користувачами платформи.

8. Навчання та підтримка: відеоуроки, статті та інші навчальні матеріали допомагають користувачам вивчати нові техніки та інструменти; цілодобова підтримка клієнтів доступна для вирішення будь-яких питань;

Splice пропонує інноваційні рішення для музичного продакшну, дозволяючи створювати музику будь-кому, незалежно від рівня навичок. Splice пропонує інноваційні рішення для створення музики незалежно від

рівня підготовки, роблячи процес виробництва музики більш доступним і зручним для кожного.

Sweetwater - це більше, ніж просто магазин музичних інструментів та аудіообладнання, це справжній партнер для музикантів, звукорежисерів та продюсерів [38] (рис 1.2.).

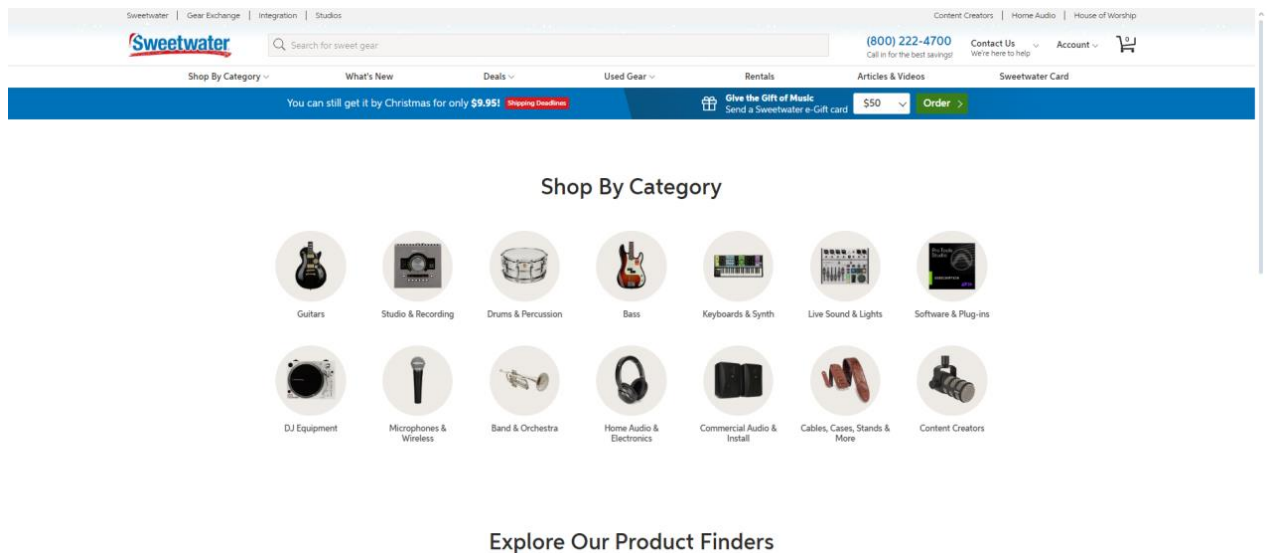


Рисунок 1.2 - Інтерфейс платформи Sweetwater

Заснований у 1979 році, магазин зберіг відданість своїм клієнтам та високий рівень обслуговування, ставши одним з найбільших постачальників музичного обладнання у світі. Однією з найсильніших сторін Sweetwater є асортимент продукції:

Гітари: від класичних акустичних гітар до сучасних електрогітар від відомих брендів, таких як Fender, Gibson і PRS.

Клавійні інструменти: синтезатори, цифрові піаніно та MIDI-контролери від Yamaha, Roland, Korg та інших.

Ударні інструменти: все, що потрібно барабанщику, від акустичних комплектів до електронних ударних установок.

Професійне звукове обладнання Sweetwater: широкий вибір обладнання для запису та живого звуку: мікрофони. Мікрофони: широкий вибір студійних, живих та USB мікрофонів.

Аудіоінтерфейси Від домашньої студії до професійної.

Монітори Студійні та сценічні монітори для чистого, детального звуку.

Однією з відмінних рис Sweetwater є рівень технічної підтримки. Команда кваліфікованих консультантів завжди готова допомогти вам вибрати правильне обладнання та надати кваліфіковану пораду. Крім того, вони пропонують безкоштовне налаштування гітари при покупці, а також цілий ряд навчальних програм і вебінарів.

Інновації Sweetwater постійно впроваджує нові технології та розробки, щоб надавати своїм клієнтам найкращі рішення. Це включає інтеграцію з різними цифровими платформами для музикантів, що робить процес створення музики більш зручним. Як результат, Sweetwater - це не просто магазин, а місце, де музиканти можуть знайти все необхідне для своєї творчості. Високоякісний сервіс, великий вибір та професійна підтримка роблять Sweetwater вибором багатьох музикантів по всьому світу.

Світовий лідер у сфері музичного продакшну та діджеїнгу Native Instruments - один з провідних виробників програмного забезпечення, та аудіоапаратури, для комп'ютерного музичного продакшну, та діджеїнгу [33]. Заснована у 1996 році, компанія має широкий асортимент продукції, для задоволення потреб музикантів, продюсерів та діджеїв (Рис 1.3.).

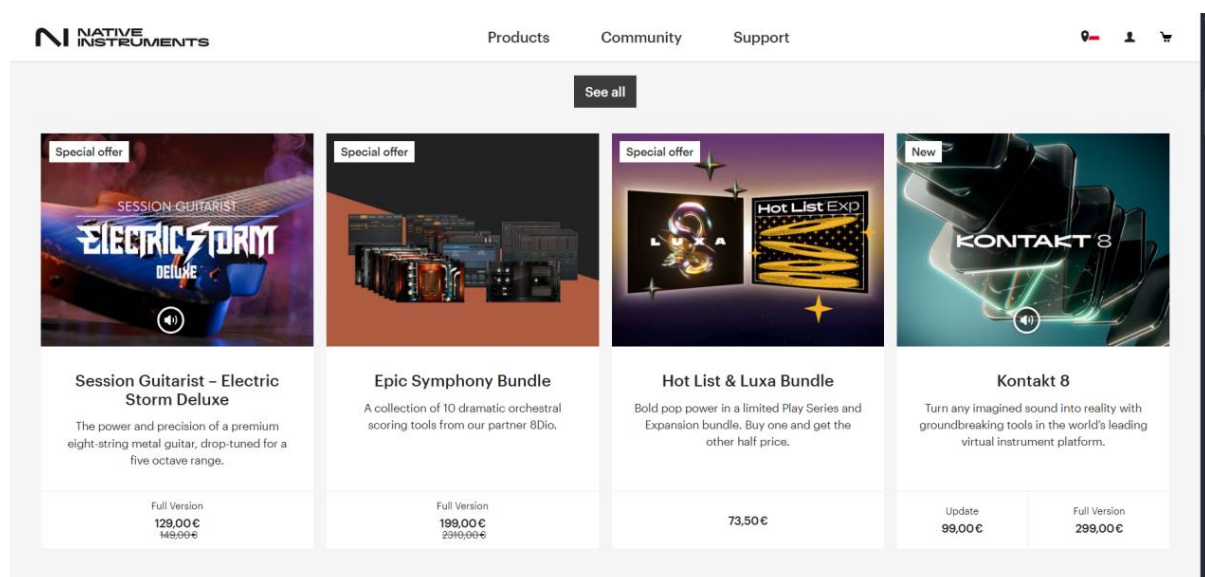


Рисунок 1.3 - Сторінка сервісу Native instruments

Серія Komplete, пропонує все необхідне, для музичного продакшну, включаючи синтезатори, семплери та ефекти; Traktor встановлює стандарти, в діджейській індустрії, завдяки надзвичайній гнучкості та контролю, а Maschine поєднує в собі апаратні, та програмні інструменти для виробництва. Асортимент включає флагманські продукти. Native Instruments також пропонує широкий вибір MIDI-контролерів, аудіоінтерфейсів, та інших інструментів, які допоможуть музикантам і продюсерам, реалізувати свої творчі ідеї. Сайт компанії пропонує безліч ресурсів, для навчання, та підтримки, включаючи безкоштовні оновлення програмного забезпечення, знижки, та інтеграцію з популярними DAW. Компанія також активно підтримує свою спільноту користувачів, за допомогою форумів, вебінарів, та інших заходів, що робить її незамінним партнером, для музикантів по всьому світу. Порівняльна характеристика сервісів, для продажу музичних плагінів наведена у таблиці 1.2.

Таблиця 1.2 - Порівняльна характеристика програмних продуктів

Назва програмного продукту	Характеристика, основні функції, та задачі програмного забезпечення.
iZotope [33]	Пропонуючи широкий спектр продуктів для зведення, мастерингу, ремастерингу та обробки вокалу, такі плагіни, як Ozone, RX, Nectar і Neutron, відомі своєю інтелектуальною обробкою звуку та високою якістю. iZotope також пропонує пакети та бібліотеки для різних аспектів музичного виробництва, які користуються популярністю серед музикантів та продюсерів по всьому світу.

Продовження табл. 1.2

Waves [39]	Waves відома своїми емуляціями класичного студійного обладнання, такого як SSL 4000 і Abbey Road Studios, а плагіни, такі як CLA-2A Compressor, J37 Tape і Retro-Fi, славляться своєю високою якістю і точністю емуляції. Waves також пропонує сучасні плагіни мульти-ефектів, такі як Waves Tune і Waves Nx, які цінуються сучасними продюсерами. Компанія пропонує пакети і підписки, такі як Waves Creative Access, що дає доступ до всіх плагінів за щомісячну плату.
FabFilter [30]	Такі плагіни, як Pro-Q3, Pro-C 2, Pro-L 2, Saturn і Pro-R відомі своєю точністю, інтуїтивно зрозумілими моделями з сенсорним екраном і високою якістю звуку. FabFilter також пропонує студентські пакети і знижки для молодих музикантів і продюсерів. Компанія активно підтримує свою спільноту користувачів за допомогою форумів, вебінарів та інших заходів, що робить її незамінним партнером для музикантів по всьому світу.

Підсумовуючи дослідження, можна зробити висновок, що магазини Splice, Native Instruments, Sweetwater, iZotope, Waves та FabFilter, пропонують комплексні інструменти та послуги для задоволення різноманітних потреб музикантів та продюсерів. Від великих звукових бібліотек, та інноваційних програмних продуктів (Splice, Native Instruments), до високоякісних емуляцій класичного обладнання, та інтелектуальних плагінів, для зведення та мастерингу (iZotope, Waves, FabFilter). Кожен з цих магазинів має свої сильні сторони, при цьому Sweetwater виділяється своїм обслуговуванням клієнтів, і широким вибором інструментів, та професійного

аудіообладнання. Разом ці магазини забезпечують музикантів усім необхідним, для реалізації творчих ідей, та професійного зростання, що робить їх незамінними партнерами у світі музичного продакшну.

1.3. Технічне завдання на розробку веб-додатку з продажу музичних плагінів

Будь-яка програмна розробка (довідково-інформаційна система, програмно-педагогічний засіб, електронний ресурс тощо), на початковому етапі повинна передбачати створення технічного завдання (ТЗ). Цей документ регламентує функціональність програмного засобу та визначає вимоги до процесу впровадження та звітної документації. При цьому виконавець робіт повинен зрозуміти сутність завдання, показати замовнику приблизний інтерфейс майбутнього програмного продукту, а замовнику зрозуміти, підтвердити та розписатися в тому, що саме йому потрібно у цьому програмному засобі.

При написанні ТЗ ми орієнтувалися на ГОСТ 34.602-89. Цей стандарт розповсюджується на автоматизовані системи для автоматизації різних видів діяльності (управління, проектування, дослідження тощо), включаючи їх поєднання, і встановлює склад, зміст, правила оформлення документа «Технічне завдання на створення (розвиток або модернізацію) системи».

Зміст

1. Вступ

1.1 Загальні визначення

1.2 Призначення документу

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

2.2 Характеристика об'єктів автоматизації

- 3. Вимоги до програмного засобу
 - 3.1 Вимоги до оформлення інтерфейсу
 - 3.2 Вимоги до зберігання даних і мов програмування
 - 3.3. Вимоги до розділення доступу
 - 3.4 Вимоги до програмного та технічного забезпечення
- 4. Стадії і етапи розробки програмного засобу
 - 4.1 Склад і зміст робіт із розробки програмного засобу
- 5. Порядок контролю і приймання
 - 5.1 Перевірка працездатності роботи програми
 - 5.2. Вимоги до введення програмного засобу в дію

1. Вступ

1.1 Загальні визначення

Повна назва програмного продукту: Інтернет-магазин з продажу музичних інструментів та плагінів.

Замовник – організація, яке замовляє створення програмного засобу.

Виконавець – програміст, який розробляє сайт, та завантажує його на хостинг замовника, в даному випадку це студент 2 курсу магістратури, Кеда І.О.

СУБД – система управління базами даних.

React JS та Visual Studio Code - є потужними інструментами для ефективної розробки сучасних веб-додатків, оскільки React JS дозволяє створювати динамічні користувацькі інтерфейси з використанням компонентного підходу та віртуального DOM, Visual Studio Code - багатофункціональне середовище розробки, яке підтримує велику кількість мов програмування, інтегрується з Git і може бути розширене, що робить процес написання, налагодження та управління кодом максимально комфортним і продуктивним для розробників. максимально продуктивним для розробників.

Vue JS - це просунутий JavaScript фреймворк, призначений для розробки інтерактивних користувацьких інтерфейсів. Створений для поступової

інтеграції в існуючі проекти, Vue JS поєднує в собі найкращі риси Angular і React, щоб забезпечити простий у використанні, гнучкий і масштабований інструментарій для розробки сучасних веб-додатків. Завдяки компонентній архітектурі, двонаправленому зв'язуванню даних та адаптивності, Vue JS дозволяє розробникам швидко та ефективно створювати продуктивні, зручні для користувача інтерфейси.

1.2 Призначення документу

У цьому документі приводиться повний перелік вимог до програмної реалізації та впровадження програмного засобу. У цьому документі Замовник і Виконавець підтверджують їх згоду з нижченаведеними фактами і умовами:

1. Виконавець підготував і розробив цей документ, що іменується Технічне Завдання, який містить перелік вимог до виконуваних робіт.
2. Замовник згоден зі всіма положеннями цього Технічного Завдання.
3. Замовник не має права вимагати від Виконавця в рамках поточного Договору виконання робіт або надання послуг, прямо не описаних у справжньому Технічному Завданні.
4. Виконавець зобов'язується виконати роботи в об'ємі, вказаному в справжньому Технічному Завданні.
5. Замовник не має права вимагати від Виконавця дотримання яких-небудь форматів і стандартів, якщо це не вказано в справжньому Технічному Завданні.
6. Все неоднозначності, виявлені в справжньому Технічному завданні після його підписання, підлягають двосторонньому узгодженню між Сторонами. В процесі узгодження можуть бути розроблені додаткові вимоги, які оформляються додатковою угодою до Договору і відповідним чином оцінюються.

2. Призначення розробки

2.1 Призначення і мета розробки програмного засобу

Предметом розробки є веб-сайт з продажу музичних інструментів і плагінів.

Метою розробки інтернет-магазину для продажу інструментів та плагінів є створення зручної, ефективної та надійної платформи для автоматизації процесу продажу, надання інтуїтивно зрозумілого користувацького інтерфейсу, підвищення рівня обслуговування, забезпечення безпеки та розширення ринку збуту. Магазин зробить інструменти та плагіни доступними для більшої кількості клієнтів, збільшить продажі та доходи, а також сприятиме розвитку музичної індустрії в цілому.

2.2 Характеристика об'єктів автоматизації

Електронна комерція, яка надає музикантам, продюсерам і меломанам швидкий і легкий доступ до необхідного їм обладнання та програмного забезпечення, відіграє важливу роль у сучасній музичній індустрії, а онлайн-платформи, такі як Sweetwater і Thomann, дозволяють користувачам переглядати різні продукти, порівнювати ціни та характеристики, читати відгуки інших клієнтів і приймати обґрунтовані рішення, не виходячи з дому. Це особливо важливо в сучасному швидкоплинному світі, де час і зручність мають вирішальне значення.

Інтернет-магазини, що спеціалізуються на продажу музичних інструментів і плагінів, пропонують широкий асортимент продукції від різних виробників, збільшуючи вибір покупців. Guitar Centre, наприклад, пропонує безкоштовну доставку та повернення, що підвищує задоволеність клієнтів. Інформація про нові продукти та спеціальні пропозиції доступна онлайн, що дозволяє музикантам легко йти в ногу з останніми тенденціями та технологіями.

Електронна комерція також відіграє важливу роль у розвитку бізнесу в музичній індустрії. Невеликі магазини та продюсери, такі як Reverb та Bandsamp, можуть вийти на глобальні ринки, розширити свою аудиторію та збільшити продажі. Водночас маркетингові стратегії, такі як таргетована реклама та просування в соціальних мережах, допомагають залучити нових клієнтів і підвищити впізнаваність бренду.

Таким чином, електронна комерція забезпечила успішний продаж

інструментів і плагінів, сприяючи зростанню і розвитку музичної індустрії в цілому.

3. Вимоги до програмного засобу

3.1 Вимоги до оформлення інтерфейсу

Графічний інтерфейс програмного засобу повинен бути легким (розрахованим на користувача з початковим рівнем інформаційної культури та навичок роботи за комп'ютером), але в той же час функціональним. Вікно для введення тексту та типу замовлення повинен мати мінімальну кількість елементів управління (кнопок, пунктів меню, полів введення) (рис. 2.1.).

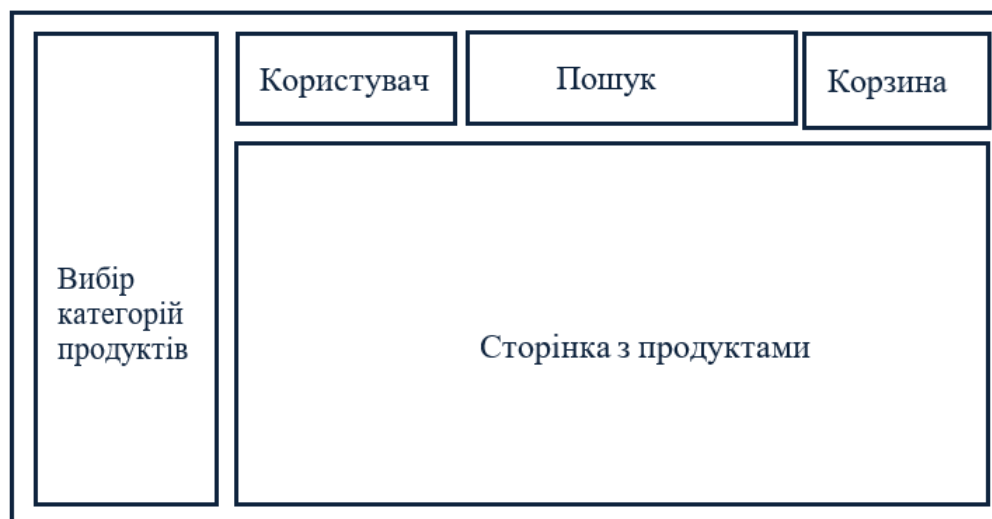


Рисунок 1.4 - Інтерфейс головної сторінки

Сторінка огляду товарів музичного інтернет-магазину повинна бути зрозумілою і містити якісні зображення товарів з різних ракурсів, детальний опис характеристик, функціональності та переваг, відгуки користувачів, функції попереднього та демонстраційного перегляду, інформацію про наявність на складі та ціни, а також зручні кнопки для додавання товарів до кошика або списку бажань. Дизайн повинен бути інтуїтивно зрозумілим. Також важливо, щоб сторінка містила блок рекомендацій щодо схожих товарів, інтегровані соціальні кнопки для швидкого обміну посиланнями та можливість швидкого зв'язку зі службою підтримки для отримання консультації. (рис. 2.2.).



Рисунок 1.5 - Інтерфейс сторінки продукту

3.2 Вимоги до зберігання даних і мов програмування

Дані про користувачів, покупки і транзакції в музичному інтернет-магазині повинні зберігатися в захищеній базі даних MySQL. Щоб використовувати MySQL в Visual Studio Code, спочатку потрібно встановити сервер MySQL і створити базу даних. Далі нам потрібно створити «базу даних MySQL» в VS Code. Потім встановимо розширення, такі як «MySQL» і «SQLTools» у VS Code для підтримки операцій з базами даних. Потім відкремо панель розширень, обираємо розширення, вводимо дані для з'єднання і підключаємось до бази даних.

3.3 Вимоги до розділення доступу

Коли ви реєструєтесь на сайті “Pluginesco”, ви створюєте обліковий запис, який дозволяє вам переглядати ваші покупки і зберігати налаштування вашого профілю. Автентифікація забезпечує безпеку і гарантує, що тільки зареєстровані користувачі можуть здійснювати покупки і залишати відгуки. Це важливо для забезпечення безпеки транзакцій і допомагає сервісу надавати персоналізовані пропозиції, підвищуючи зручність і задоволеність клієнтів.

3.4 Вимоги до програмного та технічного забезпечення

Для функціонування програмного засобу необхідне наступне програмне та мінімальне апаратне забезпечення:

1. Доменне ім'я та хостинг: зареєстроване доменне ім'я і хостинг-платформа, де буде розміщено сайт.

2. Фреймворк: React і Vue, для розробки інтерфейсу, та Node.js, для роботи над серверною частиною.

3. База даних: MySQL для зберігання даних.

4. Мова програмування: основна мова для розробки (HTML, CSS, JavaScript).

5. SSL сертифікат: для забезпечення безпеки даних користувачів.

4. Стадії і етапи розробки програмного засобу

4.1 Склад і зміст робіт із розробки програмного засобу

Розробка програмного засобу для інтернет-магазину музичних інструментів, припускає виконання наступних етапів:

1. Аналіз наочної області, яка буде реалізована в програмному засобі, в даному випадку інтернет-комерція музичних приладів.

2. Вибір бази даних, в якій будуть зберігатися потрібна інформація для роботи програмного засобу.

3. Вибір інструментального середовища, за допомогою якого буде створений програмний продукт.

4. Створення структури бази даних, що використовується для зберігання даних.

5. Створення модулів програмного засобу для інтернет-магазину музичних інструментів.

6. Тестування і перевірка працездатності програмного продукту в реальних умовах.

7. Виправлення помилок роботи програми і проведення завершального показу Замовнику.

5. Порядок контролю і приймання

5.1 Перевірка працездатності роботи програми

Приймання програмного засобу для інтернет-магазину музичних інструментів і плагінів до експлуатації складається з наступної перевірки:

1. Перевірка створення облікового запису, скидання логіну та пароллю.
2. Перевірка зображення товару, опис функцій та відображення відгуків користувачів.
3. Перевірка роботи пошуку та фільтрації товарів за категоріями, цінами та брендами.
4. Перевірка додавання товару до кошика, редагування, кількість та видалення товарів.
5. Перевірка актуальності персональних даних користувача та перегляд історії замовлень.

5.2. Вимоги до введення програмного засобу в дію

Підготовка програмного засобу до експлуатації припускає:

1. Отримання Замовником інтернет-магазину музичних інструментів і плагінів у вигляді теки із здійснимим файлом програми та всіма необхідними для роботи файлами. Додатково Виконавець може створити інсталятор програмного засобу.
2. Відкриваємо браузер, вводимо посилання на сайт і з'являється головна сторінка.
3. Показ відповідальному представнику Замовника (головному менеджеру) всіх програмного засобу.

РОЗДІЛ 2. СПЕЦИФІКАЦІЯ ВИМОГ ДО МУЗИЧНОГО ВЕБ-ДОДАТКУ

2.1 Концептуальна модель використання музичного веб-додатку

Таблиця 2.1 - Варіанти використання музичного веб-додатку.

Термін	Опис терміну
1. Основні поняття та категорії предметної області та проекту	
Користувач	Людина, яка має, мав, або, можливо, буде мати доступ до системи для здійснення операцій.
Веб-додаток	Це програмне забезпечення, яке працює у веб-браузері.
СУБД	СУБД, або система управління базами даних, - це програмне забезпечення, яке дозволяє створювати, керувати і використовувати бази даних.
Бази даних	Це організовані колекції даних, які дозволяють легко зберігати, змінювати та керувати інформацією. Вони використовуються в різних галузях для зберігання великих обсягів інформації та забезпечення її доступності для користувачів.
2. Користувачі системи	
Адміністратор	Це людина, яка має доступ: реєстрації користувача, управління запитами, управління замовленнями, керування користувачами, управління проблемами, управління сторінками,

	очікування замовлення.
Користувач	Це особа, що має можливість: реєструватись, впровадити пошук продукту, робити замовлення.
Фронтенд-розробник	Фронтенд розробник (frontend developer) - це фахівець, який відповідає за створення і підтримку інтерфейсу користувача веб-додатків або веб-сайтів. Його головне завдання - зробити веб-додаток або веб-сайт зручним, привабливим і функціональним для користувачів.
Бекенд-розробник	Це фахівець, який працює над серверною частиною веб-додатків або веб-сайтів, забезпечуючи їх функціональність, продуктивність і безпеку. Його головне завдання - забезпечити безперебійну роботу серверної логіки, баз даних і інтеграцію з іншими системами

Щоб створити потужне програмне забезпечення з використанням певного інструментального середовища, спочатку необхідно спроектувати його структуру та функціональність. Важливо детально описати логіку роботи програми, яка задовольняє системним вимогам і можливим обмеженням [25, с.10]. Крім того, потреби користувачів можуть змінюватися і вдосконалюватися в процесі побудови та експлуатації інформаційної системи, що ускладнює розробку і супровід такої системи. Щоб уникнути або вирішити ці проблеми, рекомендується будувати моделі, які відображають

структурні та функціональні характеристики майбутніх інформаційних систем.

Модель - це об'єкт або опис об'єкта, який замінює систему-оригінал з метою вивчення або відтворення її характеристик. Процес моделювання передбачає розробку комп'ютерних моделей, які підтримують весь життєвий цикл моделі. Системи комп'ютерного моделювання є високоінтегрованими та інтерактивними.

Універсальною мовою об'єктно-орієнтованого моделювання є UML (Unified Modeling Language), яка була розроблена в середині 90-х років на основі декількох методів і нотацій для опису інформаційних систем [25, с.12]. Сьогодні UML є загальноприйнятим стандартом для документування процесу створення інформаційних систем та програмного забезпечення. Поточна версія - UML 2.0. програмне забезпечення, що підтримує UML, включає IBM Rational Rose, Telelogic TAU G2, Microsoft Visio, Enterprise Architect і Modelio. в рамках UML, комплекс Всі аспекти моделі системи представляються у вигляді спеціальних схем, які називаються діаграмами. Діаграми - це графічні представлення елементів моделі у вигляді зв'язних графів, вершинам і ребрам яких присвоєна певна семантика.

В UML 1.5 визначено 12 типів діаграм, розділених на три групи: чотири типи діаграм є статичними в додатку, 4 типи діаграм відображають статичну структуру додатку (діаграми класів, діаграми об'єктів, діаграми компонентів, діаграми розгортання), 5 типів діаграм відображають аспекти поведінки (діаграми прецедентів, діаграми послідовностей, діаграми взаємодії, діаграми станів, діаграми діяльності) і 3 типи діаграм відображають фізичні аспекти функціональності (діаграми реалізації).

Кілька слів про нотації, що використовуються для всіх типів діаграм. Нотація - це набір символів і правил їх використання, що використовуються для представлення понять і зв'язків між ними. Термін «нотація» підкреслює, що UML є графічною мовою для малювання, а не опису моделей (діаграм); в UML використовуються чотири типи елементів нотації: фігури (що

представляють вузли інфраструктури), лінії (що з'єднують вузли), піктограми та мітки.

Об'єктивні труднощі, пов'язані з формальним описом предметної області та створенням повних і функціонально узгоджених інформаційних моделей, сприяли появі програмно-технічних засобів (CASE-засобів), які підвищують ефективність розробки різноманітного програмного забезпечення CASE-технології визначаються як сукупність методів і засобів проектування програмного забезпечення, які можуть бути використані для створення та аналізу предметної області можна моделювати візуально і аналізувати цю модель на всіх етапах розробки та супроводу програмного забезпечення [24, с. 6].

Більшість існуючих CASE-засобів базуються на методологіях структурного та об'єктно-орієнтованого підходів. Варто зазначити, що об'єктно-орієнтований підхід рекомендується для моделювання інформаційних систем, які використовують потужні програмні засоби, особливо бази даних. Основна ідея об'єктно-орієнтованого аналізу та проектування полягає в тому, щоб розглядати цільову предметну область і вирішувати проблеми в термінах об'єктів (концепцій або сутностей). Об'єктно-орієнтований аналіз фокусується на визначенні та описі об'єктів в термінах предметної області. В процесі об'єктно-орієнтованого проектування визначаються логічні програмні об'єкти (атрибути і методи), які реалізуються за допомогою певної об'єктно-орієнтованої мови програмування [6, с. 223].

Так як акторами данної системи є "Користувач", "Адміністратор", веб-додаток, то ми відображаємо таку діаграму варіантів, як на рисунку 2.1.

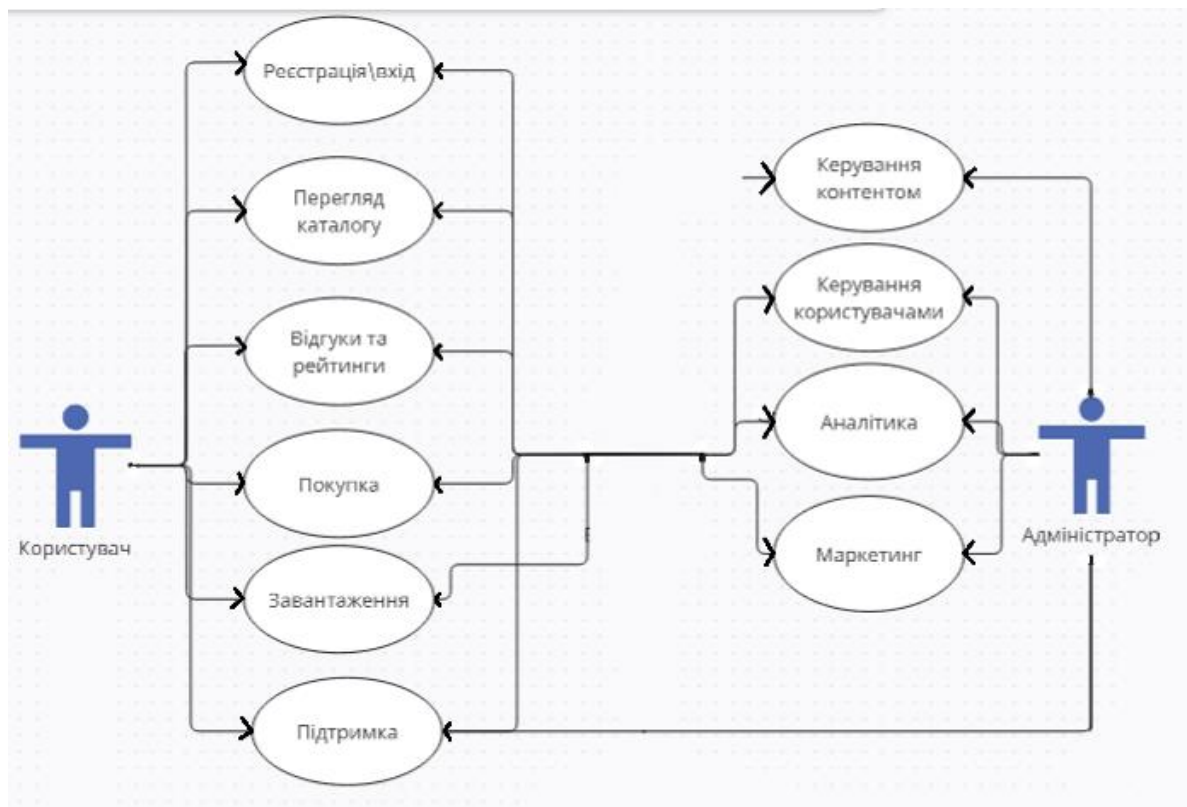


Рисунок 2.1 – Діаграма варіантів використання акторів системи

Для початку, в діаграмі акторів, все починається з того, що користувач баче перед собою першу сторінку, де він може увійти, або зареєструватись. Далі, переглядаючи каталог, він натикається на відгуки о товарах. Інформацію о відгуках, забезпечує актор - “Веб-додаток”. Бо подібна текстова інформація, міститься в базі даних, яка є частиною цього актора. При покупці товару, веб-додаток забезпечує для користувача можливість сплатити, та обробити платіж, і звісно ж, вислати інформацію про платіж до актора Адміністратора, який в свою чергу, надсилає до актора “Веб-додаток”, сповіщення, про вдалу транзакція. Та видимість цієї операції для користувача, вже забезпечує веб-додаток.

Розкадровка – це логічний і концептуальний опис функціональних можливостей системи для певного сценарію, який включає необхідну взаємодію між системою та її користувачами. За результатами розкадрування повинні бути сформульовані вимоги, які підлягають узгодженню з зацікавленими особами.

В даній роботі розкадровка зроблена за допомогою програмного продукту Creately.

Розкадровка варіанту використання «Кошик», на якій містяться повідомлення про відсутність товару наведена на рисунку 2.2.

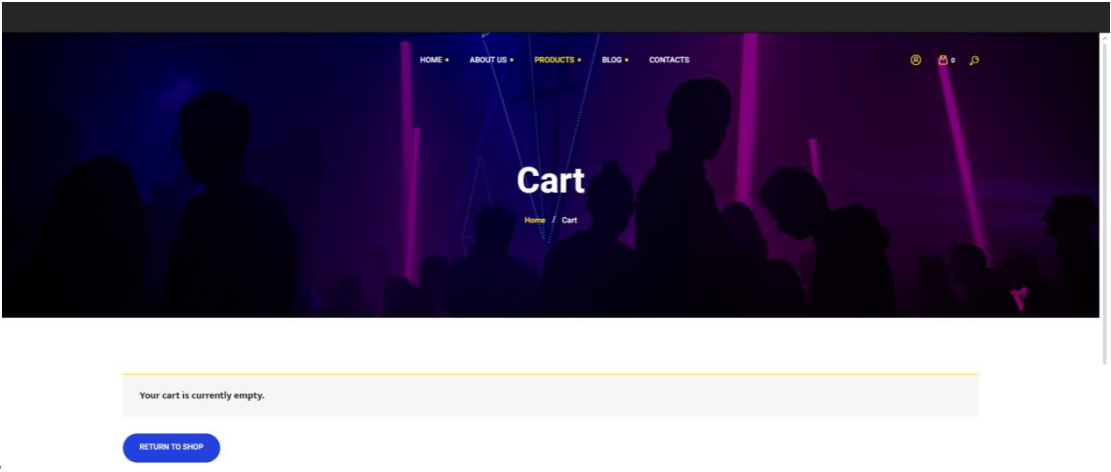


Рисунок 2.2 - Розкадровка варіанту “Кошик”

Розкадровка варіанту використання «Додавання товару», на якій містяться кнопки для взаємодії з обраним товаром, і сторінка товарів, наведена на рисунку 2.3.

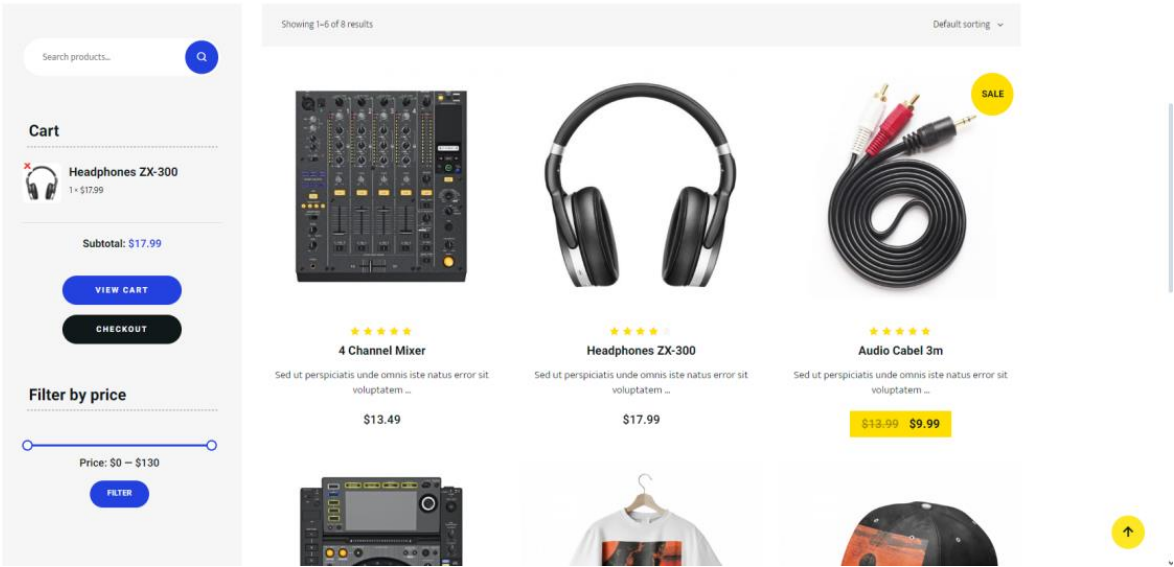


Рисунок 2.3 - Розкадровка варіанту використання “Додавання товару”

Розкадровка варіанту використання «Рекомендовані товари», на якій містяться інформація о товарах, їх ціна, та можливість додати до кошику на рисунку 2.4.

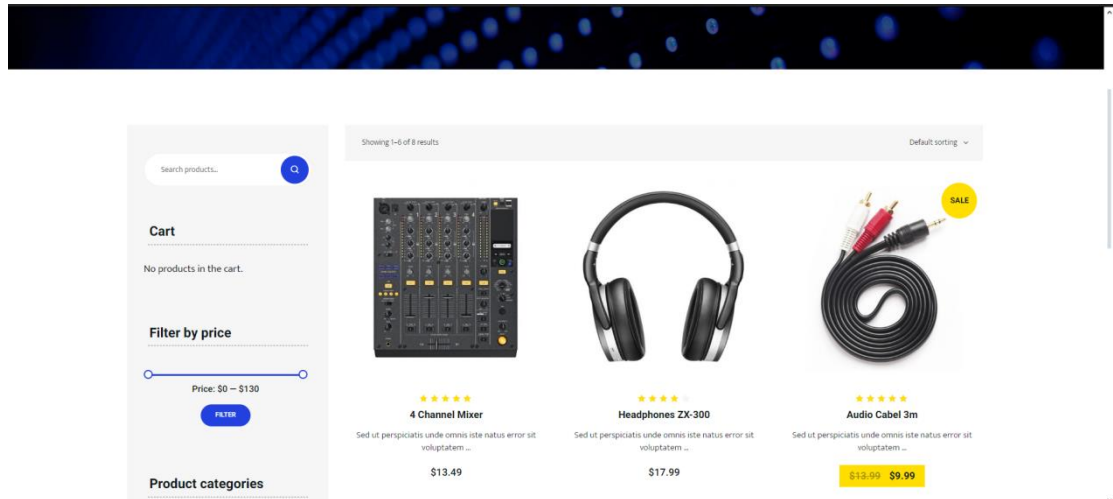


Рисунок 2.4 - Розкадровка варіанту “Рекомендовані товари”

2.2. Функціональна модель веб-додатку

IDEF0 - це методологія функціонального моделювання та графічного опису процесів. Завдяки методології функціонального моделювання систему, що розробляється, можна розглядати як набір взаємопов'язаних функцій (функціональних блоків). Побудова моделі IDEF0 починається з того, що розробник розглядає систему як єдине ціле - один функціональний блок з дугою інтерфейсу, що простягається через межі предметної області. Процес починається з розгляду системи як єдиної сутності. Така діаграма з одним функціональним блоком називається контекстною діаграмою і позначається ідентифікатором «A-0».

Історія IDEF0 дуже цікава і пов'язана з розвитком методів моделювання бізнес-процесів: у 1970-х роках ВПС США зіткнулися з проблемою інтеграції різних систем для управління виробничими процесами. Для вирішення цих завдань була розроблена програма Integrated Computer Aided Manufacturing (ICAM), покликана підвищити ефективність виробничих процесів за

допомогою комп'ютерного управління. В рамках програми ICAM була розроблена стандартизована методологія моделювання та аналізу функцій виробничих процесів. Необхідно було створити стандартизовану методологію моделювання та аналізу функцій виробничого процесу. Так з'явилася IDEF (Integrated Definition), яка включала в себе кілька методологій, зокрема IDEF0, IDEF1 та IDEF2. IDEF0 була спеціально розроблена для моделювання функцій і процесів і надає користувачам інструменти для візуалізації та аналізу систем в ієрархічній структурі. IDEF0 дозволяє представити функції організації у вигляді блоків, пов'язаних між собою входами, виходами, механізмами та елементами управління. Такий підхід допомагає краще зрозуміти зв'язки між різними частинами системи та виявити можливі точки оптимізації. З часом IDEF0 стала широко використовуваним інструментом як у військовому, так і в комерційному середовищі, дозволяючи компаніям оптимізувати свої бізнес-процеси та підвищити продуктивність. Методологія активно використовується і сьогодні, залишаючись одним з ключових інструментів для аналізу та вдосконалення бізнес-процесів.

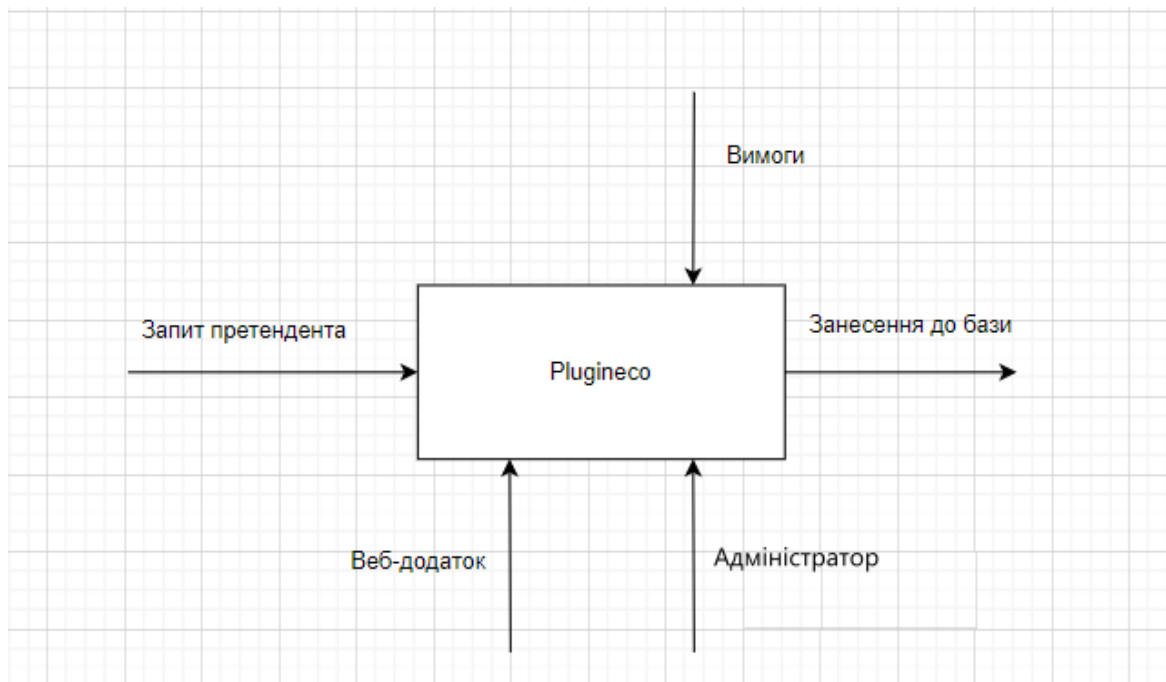


Рисунок 2.5 – Контекстна діаграма IDEF0

Спеціалізуючись на музичних плагінах, таких як VST, AU та інших ефектах і віртуальних інструментах, Plagineso пропонує широкий вибір продуктів від провідних виробників. Веб-додаток пропонує зручний інтерфейс для перегляду та вибору потрібного плагіна, а також докладні описи та відгуки користувачів для кожного продукту. Користувачі можуть легко створити власний обліковий запис на Plagineso, де вони можуть зберігати свої улюблені плагіни, отримувати персоналізовані рекомендації щодо плагінів та брати участь у спеціальних акціях. Щоб створити обліковий запис, необхідно заповнити просту форму з основною інформацією про користувача, яка забезпечує зручний і персоналізований досвід покупок.

Щоб створити обліковий запис у веб-додатку, користувач повинен заповнити форму, що містить кілька основних полів: email для введення дійсної адреси електронної пошти, ім'я користувача для створення унікального логіну, безпека облікового запису для встановлення безпеки облікового запису, пароль для підтвердження правильності пароля та прапорець «Умови використання» для підтвердження згоди користувача з Умовами використання та Політикою конфіденційності веб-додатку.

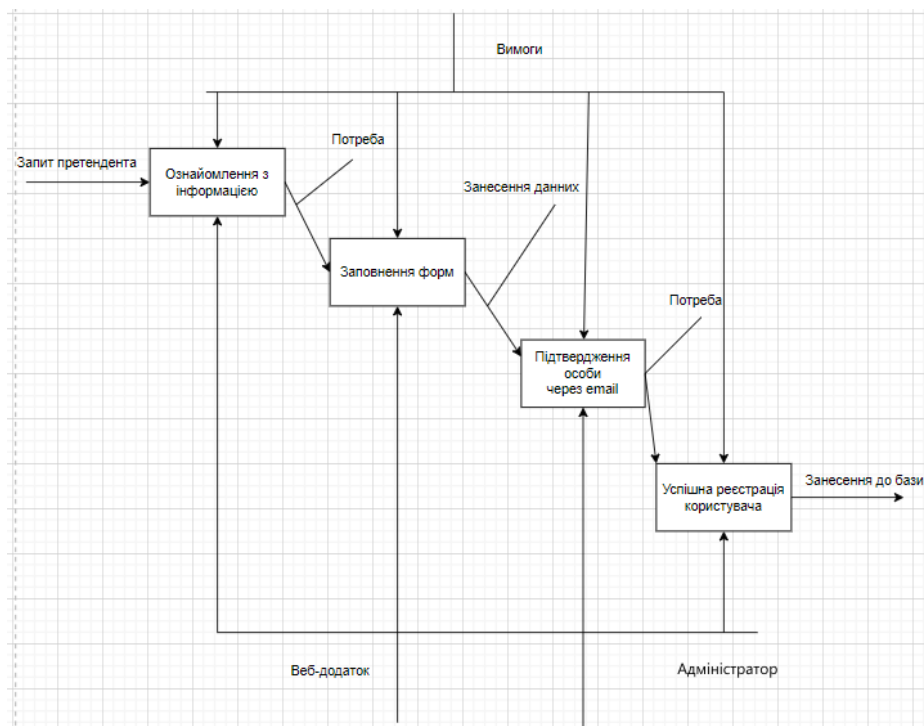


Рисунок 2.6 - Діаграма вхідних параметрів

Користувач вводить особисті дані, такі як ім'я, електронна пошта та пароль, у відповідні форми на сайті. Після цього система надсилає на вказану електронну адресу лист із посиланням для підтвердження. Користувач відкриває лист і переходить за посиланням, щоб підтвердити свою особу. Після завершення підтвердження з'являється повідомлення про те, що обліковий запис успішно зареєстровано.

2.3 Специфікація функціональних та нефункціональних вимог

Функціональні вимоги описують функції, які повинно виконувати програмне забезпечення. Функціональні вимоги є похідними від вимог, встановлених для користувача.

Таблиця 2.2 - Специфікація функціональних вимог

№	Назва вимоги (варіанта використання)	Атрибути вимог		
		пріоритет	складність	Контакт/виконавець
1	Реєстрація	обов'язкове	середнє	користувач
2	Авторизація	обов'язкове	середнє	користувач
3	Пошук	рекомендоване	середнє	користувач
4	Огляди	рекомендоване	середнє	користувач
5	Кошик	обов'язкове	середнє	користувач
6	Оплата	обов'язкове	середнє	користувач
7	Історія покупок	рекомендоване	середнє	користувач
8	Оновлення плагінів	обов'язкове	висока	адміністратор
9	Підтримка користувачів	обов'язкове	висока	адміністратор

10	Управління користувачами	рекомендоване	висока	адміністратор
11	Каталог плагінів	обов'язкове	висока	адміністратор
12	Управління замовленнями	обов'язкове	висока	адміністратор/користувач
13	Фінансова статистика	рекомендоване	висока	адміністратор
14	Безпека	обов'язкове	висока	адміністратор
15	Сповіщення користувачів	обов'язкове	висока	адміністратор

Таблиця 2.3. Нефункціональні вимоги

№	Назва вимоги	Характеристика
1	Застосовність	Веб-додаток призначений для покупки музичних плагінів, та їх продажу лейблами
2	Час для навчання користувачів	Веб-додаток мусить бути зрозумілим користувачу на інтуїтивному рівні
3	Надійність	Система повинна передбачувати паролі для кожного користувача та зберігати їх у форматі md5 у базі даних
4	Мова програмування	Веб-додаток мусить бути розроблений на мовах програмування: JS, CSS, HTML
5	Бізнес-правила	При початком користування веб-додатком користувач системи видає вже створений ним логін і пароль адміністратору
6	Зовнішні інтерфейси	Забезпечити запис в базу даних останнього відвідування клієнта

РОЗДІЛ 3. ВИБІР ТА ОБҐРУНТУВАННЯ ТЕХНОЛОГІЙ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ

3.1 Загальний огляд технологічного стеку

Інтерпретатор у програмуванні - це програма, яка перетворює код рядок за рядком у машинні інструкції та виконує їх [4, с.477]. Це дозволяє розробникам швидко тестувати та змінювати код, що особливо корисно для створення прототипів та експериментів; мови-інтерпретатори, такі як Python, JavaScript та Ruby, дозволяють змінювати типи даних під час виконання, що робить їх гнучкими та корисними для багатьох завдань. Однак інтерпретатори мають і недоліки. Вони можуть працювати повільніше, ніж компілятори, оскільки код перетворюється на машинні інструкції під час виконання. Інтерпретатори також споживають більше ресурсів і можуть потребувати спеціального інтерпретатора для запуску коду на різних платформах. Тим не менш, їх гнучкість і зручність роблять інтерпретовані мови популярним вибором серед розробників.

Visual Studio Code (VS Code) - популярний безкоштовний редактор коду, розроблений компанією Microsoft. Відомий своєю швидкістю, легкістю та розширюваністю, VS Code підтримує багато мов програмування завдяки вбудованим функціям та розширенням, які можна завантажити з ринку вбудованих розширень. Це робить його ідеальним інструментом для розробників, які працюють з різними технологіями. Користувачі можуть змінювати теми, додавати плагіни з підтримкою різних мов програмування та інструментів, а також налаштовувати робочі процеси за допомогою численних параметрів. Інші важливі функції включають вбудований термінал, відладчик, підтримку Git, автозавершення коду, інтеграцію з Docker та багато інших інструментів для полегшення розробки програмного забезпечення.

ReactJS - це популярна бібліотека JavaScript, розроблена Facebook спеціально для створення інтерфейсів користувача для односторінкових

програм. Це дозволяє розробникам створювати складні інтерфейси за допомогою компонентного підходу. Кожен компонент є незалежним блоком, що відповідає за певну частину інтерфейсу. Це робить код зрозумілішим і легшим для керування та повторного використання [27, с.26].

Однією з головних особливостей React є використання віртуальної DOM (Document Object Model), яка значно покращує продуктивність. Замість того, щоб безпосередньо взаємодіяти з реальним DOM, React спочатку оновлює віртуальний DOM, а потім ефективно застосовує зміни до справжнього DOM. Це зменшує кількість операцій над фактичним DOM, що є однією з найдорожчих операцій у браузері. Завдяки цій функції ReactJS став одним із найпопулярніших інструментів для створення сучасних веб-додатків.

HTML5 - це остання версія мови гіпертекстової розмітки, яка використовується для створення та структурування веб-сторінок і веб-додатків. Основною метою HTML5 є покращення підтримки мультимедіа та інтерактивності без необхідності додаткових плагінів, таких як Flash. Ця версія містить багато нових елементів і атрибутів, які допомагають створювати більш семантичні, сумісні та зручні структури веб-сторінок.

HTML5 представляє кілька нових функцій, включаючи вбудовану підтримку відео та аудіо за допомогою елементів `video` і `audio`, елемент для малювання графіки та анімації за допомогою JavaScript, а також нові типи полів введення та перевірки для створення та обробки форм інструменти для: Крім того, HTML5 підтримує різні API, такі як `LocalStorage` для зберігання даних на клієнті та `Geolocation` для визначення місцезнаходження користувача, що робить веб-розробку більш потужною та гнучкою та забезпечує кращий досвід роботи з користувачами. Це дає нові можливості для веб-додатків [15, с. 40].

CSS3 (Cascading Style Sheets Level 3) - остання версія мови таблиць стилів, яка використовується для опису зовнішнього вигляду веб-сторінок, написаних у HTML або XHTML. CSS3 представляє безліч нових

властивостей і модулів, які дозволяють розробникам створювати більш привабливі та інтерактивні дизайни. Ключовою особливістю CSS3 є його модульна структура, яка дозволяє додавати нові функції, не порушуючи старі. Наприклад, модуль може містити селектори, кольори, текст, зображення, фони, шрифти тощо [15, с. 42].

Однією з головних особливостей CSS3 є можливість анімації та трансформації. Це означає, що розробники можуть створювати плавні переходи та анімацію, щоб зробити веб-сторінки більш динамічними та привабливими. Наприклад, властивості трансформації, переходу та анімації дозволяють використовувати CSS для обертання, масштабування, переміщення та трансформації елементів, які раніше потребували JavaScript. Також варто відзначити підтримку нових одиниць вимірювання, таких як Rem і VH/VW. Це дозволяє створювати більш гнучкі та адаптивні конструкції.

CSS3 також має покращену підтримку медіа-запитів, що дозволяє створювати адаптивні дизайни, які адаптуються до різних розмірів екрана та типів пристроїв. Це особливо важливо в епоху мобільних пристроїв. Ваш веб-сайт має чудово виглядати як на великих моніторах, так і на маленьких екранах смартфонів. Медіа-запити дозволяють застосовувати різні стилі залежно від ширини, висоти, орієнтації та роздільної здатності екрана, що забезпечує кращу взаємодію з користувачем на різних пристроях.

CSS3 швидко стає незамінним інструментом для веб-розробників, оскільки пропонує багато нових функцій і більший контроль над виглядом і поведінкою веб-сторінок. Це дозволяє створювати сучасні, креативні та високофункціональні веб-дизайни, які відповідають сучасним вимогам.

Vue.js – це вдосконалений фреймворк JavaScript для створення інтерфейсів користувача та односторінкових програм. Він був розроблений Еваном Ю і вперше випущений у 2014 році. Однією з головних переваг Vue.js є його гнучкість і простота використання. Ви можете легко інтегрувати його в існуючі проекти або створювати нові програми з нуля. Vue.js має

модульну архітектуру, і його основні функції можна використовувати з різними бібліотеками та інструментами, такими як Vue Router для маршрутизації та Vuex для керування станом програми [8, с. 12].

Vue.js використовує декларативний підхід до створення інтерфейсів користувача, що полегшує читання та підтримку коду. Інфраструктура компонентів Vue.js дозволяє створювати невеликі відокремлені компоненти, які можна легко повторно використовувати в різних частинах програми. Ключовою особливістю Vue.js є реактивність. Це дозволяє інтерфейсу автоматично оновлюватися при зміні даних. Це досягається завдяки двосторонньому зв'язку між даними та дисплеєм, що значно спрощує розробку інтерактивних програм.

Крім того, у Vue.js є велика та активна спільнота, яка створює багато розширень, плагінів та інструментів, які полегшують розробку. Офіційна документація Vue.js є однією з найкращих у своїй категорії, що робить фреймворк дуже простим у вивченні та використанні навіть для новачків. Vue.js стає все більш популярним серед розробників і компаній, які створюють сучасні веб-додатки, завдяки своїй продуктивності, гнучкості та простоті використання.

3.2 Обґрунтування вибору мови програмування

Мови програмування по суті діють як міст між людьми та машинами. Це дає змогу перетворювати абстрактні ідеї, креативні рішення та наукові відкриття в конкретні інструкції, яким можуть слідувати машини. У світі, де комп'ютери та технології стали невід'ємною частиною нашого життя, мови програмування надають нам інструменти для керування цими технологіями та створення нових інновацій. Вони відкривають двері до безмежних можливостей у різних сферах – від медицини до дослідження космосу.

Мови програмування потрібні в багатьох різних ситуаціях. Під час створення програмного забезпечення або веб-додатків вам потрібна мова програмування для розробки інтерфейсу користувача, обробки даних і взаємодії з базами даних. У наукових дослідженнях мови програмування допомагають аналізувати великі обсяги даних, моделювати фізичні процеси та прогнозувати майбутні події. Автоматизація дозволяє керувати роботами та пристроями без втручання людини. І це лише верхівка айсберга. Мови програмування є ключем до освоєння світу технологій і створення майбутнього, де технології полегшують наше життя та розширюють горизонти можливостей.

Мови програмування справили великий вплив на музичну індустрію, відкривши нові можливості для створення, запису, обробки та розповсюдження музики. Одним із найяскравіших прикладів є розробка музичного програмного забезпечення. Такі програми, як Ableton Live, Logic Pro і FL Studio, дозволяють композиторам і продюсерам створювати та редагувати музику за допомогою передових алгоритмів і технологій, написаних на таких мовах програмування, як C++, Python і JavaScript. Ці програми пропонують широкий спектр можливостей, включаючи створення різних звукових ефектів, аранжування треків і створення високоякісного музичного контенту.

Ще однією важливою областю є алгоритмічна композиція та генеративна музика, яка передбачає використання мов програмування для створення музики за допомогою алгоритмів. Програмісти можуть створювати алгоритми, які генерують мелодії, гармонії та ритми на основі заданих параметрів. Це дозволяє створювати унікальні музичні композиції, створення яких вручну було б неможливим або займало б дуже багато часу. Наприклад, мова програмування Sonic Pi спеціально розроблена, щоб дозволити вам писати код для створення музики в реальному часі. Крім того, він використовує методи штучного інтелекту та машинного навчання для

аналізу музичних творів і створення нових композицій, що забезпечує абсолютно новий рівень музичних експериментів [11, с. 36].

Мови програмування також стають все більш важливими в сфері потокової передачі та розповсюдження музики. Такі платформи, як Spotify, Apple Music і YouTube Music, використовують складні алгоритми, щоб рекомендувати музику на основі ваших смаків і треків, які ви слухаєте. Ці алгоритми, написані такими мовами, як Python і Java, аналізують великі обсяги даних, щоб надати кожному користувачеві найрелевантніші заголовки, створюючи унікальний досвід користувача. Мови програмування також використовуються для створення музичних візуалізаторів та інтерактивних інструментів, які можуть розширити межі традиційної музики та зробити музику доступною для більш широкої аудиторії.

Тому мови програмування відіграють важливу роль у музичній індустрії, дозволяючи створювати інноваційні інструменти, автоматизувати рутинні процеси та нові способи взаємодії з музикою. Вони відкривають широкі можливості для композиторів, продюсерів, виконавців і слухачів, роблячи процес виробництва музики більш творчим і технічно досконалим.

Мови програмування відіграють життєво важливу роль у створенні веб-додатків, оскільки вони є основою для зовнішньої та внутрішньої розробки. На передній частині ви створюєте інтерфейс користувача за допомогою таких мов, як HTML, CSS і JavaScript. HTML забезпечує структуру веб-сторінки, а CSS визначає стиль і макет. JavaScript також додає інтерактивність, дозволяючи створювати динамічні елементи, такі як анімація, форми з перевіркою тощо. Ці мови дозволяють розробникам створювати привабливі, інтуїтивно зрозумілі та функціональні інтерфейси, які покращують взаємодію з користувачем.

Сервер обробляє запити користувачів, керує базою даних і виконує бізнес-логіку за допомогою мов програмування, таких як Python, Ruby, PHP, Java і Node.js. Сервер забезпечує безпечну взаємодію з базами даних, обробляє автентифікацію користувачів, керування сесіями та багато інших

важливих функцій, які забезпечують стабільну та надійну роботу вашої веб-програми. Поєднуючи зовнішні та внутрішні технології, ви можете створювати складні веб-програми, які відповідають потребам сучасних користувачів і компаній, забезпечуючи високу продуктивність, безпеку та масштабованість.

Мови програмування відіграють важливу роль у створенні та розвитку електронної комерції, оскільки вони забезпечують функціональність, безпеку та масштабованість веб-сайтів і програм онлайн-комерції. На передній частині ви використовуєте HTML, CSS і JavaScript, щоб створити зручний і привабливий інтерфейс, який дозволяє користувачам легко знаходити та купувати ваші продукти. Бібліотеки та фреймворки, такі як React, Angular і Vue.js, дозволяють розробникам створювати динамічні інтерактивні інтерфейси користувача, які покращують взаємодію з користувачем.

Сервер використовує такі мови програмування, як Python, Ruby, PHP, Java і Node.js для керування базами даних, обробки транзакцій і забезпечення безпеки даних користувача. Серверні інфраструктури, такі як Django, Ruby on Rails і Express, дозволяють розробникам створювати потужні та безпечні системи для обробки замовлень, управління запасами та інтеграції платіжної системи. Безпека особливо важлива в секторі електронної комерції, а мови програмування надають інструменти, необхідні для захисту конфіденційної інформації користувачів і запобігання кіберзагрозам.

Мови програмування складають основу всіх аспектів онлайн-комерції, від створення користувацьких інтерфейсів до обробки даних і забезпечення безпеки, що дозволяє підприємствам ефективно продавати свої продукти та послуги в Інтернеті.

ReactJS, HTML, CSS і Vue.js є основними інструментами для створення комерційних веб-додатків, що дозволяє розробникам створювати ефективні, інтерактивні та привабливі інтерфейси користувача.

ReactJS - це бібліотека JavaScript, яка дозволяє створювати складні інтерфейси за допомогою компонентного підходу. Він пропонує високу

продуктивність через віртуальний DOM і підтримує рендеринг на стороні сервера для покращеного SEO та швидшого завантаження сторінки. HTML визначає структуру веб-сторінки та дозволяє створювати семантично правильний і зручний для пошукових систем вміст. Це покращує оптимізацію SEO, а також забезпечує підтримку мультимедійного вмісту без додаткових плагінів [15, с. 40]. CSS визначає зовнішній вигляд ваших веб-додатків, дозволяючи створювати привабливі дизайни, які чудово виглядатимуть на різноманітних пристроях, від смартфонів до великих моніторів. Він також підтримує анімацію, трансформації та медіа-запити для створення сучасних динамічних інтерфейсів [15, с. 42].

Vue.js - це прогресивний фреймворк JavaScript, який забезпечує архітектуру компонентів і реактивність, що дозволяє розробникам швидко створювати та підтримувати комерційні веб-додатки. Для створення інтерактивних інтерфейсів використовуються одностороння і двостороння прив'язка даних. Він також має потужну екосистему, включаючи Vue Router для маршрутизації та Vuex для керування станом програми. Поєднання цих технологій дозволяє створювати комерційні веб-програми з високою продуктивністю, чудовим дизайном і чудовим користувацьким досвідом, що є важливим для досягнення успіху в конкурентному середовищі електронної комерції.

ReactJS ідеально підходить для створення веб-додатків для продажу музичних плагінів завдяки своїй ефективності, гнучкості та здатності забезпечувати багатий досвід користувача. Компонентний підхід React дозволяє розробникам створювати повторно використововані компоненти, прискорюючи розробку та полегшуючи підтримку коду. Наприклад, компоненти для відображення списку продуктів, кошика для покупок і окремих сторінок продукту можна розробити один раз і використовувати в різних частинах вашої програми, що значно підвищує ефективність розробки [27, с. 26].

Крім того, ReactJS досягає високої продуктивності завдяки своїй віртуальній DOM, ефективно оновлюючи інтерфейс користувача без виконання непотрібних операцій над реальною DOM. Це особливо важливо для комерційних додатків, де швидкість завантаження сторінки та зручність роботи користувача мають вирішальне значення. Крім того, React підтримує рендеринг на стороні сервера, що забезпечує кращу оптимізацію SEO та швидший початковий час завантаження сторінки. Це важливо для залучення та утримання клієнтів [27, с.27].

React також має широку екосистему інструментів і бібліотек, зокрема: Наприклад, використання React Router для керування маршрутизацією та Redux для керування станом програми полегшує розробку складних масштабованих веб-додатків. Ці інструменти забезпечують зручну роботу користувача та дозволяють розробникам зосередитися на функціональності та дизайні своїх програм. Ці переваги роблять ReactJS ідеальним вибором для створення веб-додатку для продажу музичних плагінів, оскільки він пропонує гнучкість, продуктивність і чудову взаємодію з користувачем.

Гнучкість, легкість вивчення та потужна екосистема Vue.js роблять його чудовим вибором для створення веб-програми для продажу музичних плагінів. По-перше, Vue.js використовує компонентний підхід, що дозволяє вам створювати модульні багаторазові компоненти, які полегшують розробку та підтримку вашого коду. Це особливо корисно під час створення динамічних інтерфейсів користувача, таких як каталоги продуктів, візки для покупок і сторінки продуктів, де важлива швидка та проста навігація [8, с. 12].

Крім того, Vue.js має вбудовану підтримку реактивності, тому інтерфейс автоматично оновлюється при зміні даних. Це забезпечує безперебійну та безперебійну роботу користувача, що має вирішальне значення для комерційних програм. Двостороннє зв'язування даних полегшує роботу з формами та інтерактивними елементами, дозволяючи користувачам швидко та легко взаємодіяти з вашим веб-додатком.

Vue.js також має потужну екосистему інструментів і бібліотек, таких як Vue Router для керування маршрутизацією та Vuex для керування станом програми, що полегшує створення складних, масштабованих веб-додатків. Vue Router забезпечує плавну навігацію без перезавантаження сторінок, що значно покращує взаємодію з користувачем. Vuex дозволяє ефективно керувати глобальним станом вашої програми та зберігати дані про користувачів, продукти та транзакції.

Vue.js простий у вивченні та використанні, тому навіть початківці можуть швидко освоїти його та розпочати розробку повнофункціональних програм. Він також надає досвідченим розробникам усі інструменти, необхідні для створення масштабованих, продуктивних комерційних веб-додатків. Це робить Vue.js ідеальним вибором для створення веб-програми, що продає музичні плагіни, які можуть забезпечити чудову взаємодію з користувачем, високу продуктивність і надійну функціональність [8, с.13].

У висновку, багато переваг і широка підтримка JavaScript роблять його важливим інструментом для створення веб-додатків. По-перше, JavaScript - це мова програмування на стороні клієнта, яка дозволяє створювати інтерактивні та динамічні інтерфейси користувача. Це усуває потребу в безперервних запитах до сервера, що забезпечує швидшу та плавну взаємодію та кращий досвід користувача. Більшість сучасних веб-фреймворків і бібліотек, таких як React, Angular і Vue.js, засновані на JavaScript і тому є важливою частиною екосистеми веб-розробки.

Крім того, JavaScript - це мова загального призначення, яка працює в усіх сучасних веб-переглядачах без будь-яких додаткових плагінів чи конфігурації. Це забезпечує широку доступність і підтримку між платформами, що важливо для комерційних веб-додатків, яким потрібно охопити широку аудиторію. Крім того, JavaScript надає можливість працювати з API, обробляти події, перевіряти форми, створювати анімацію та багато інших функцій, які роблять веб-додатки більш функціональними та привабливими для користувачів [16, с.01].

3.3 Інструменти для розробки та тестування

Середовище розробки (IDE) необхідне, щоб допомогти розробникам писати, тестувати та налагоджувати код більш ефективно, об'єднуючи різні інструменти в єдиний простий у використанні інтерфейс. Вони забезпечують аналіз, підсвічування коду, автозаповнення, керування проектами, вбудовані засоби налагодження та можливості тестування, що дозволяє вам зосередитися на логіці програми, а не на повсякденних завданнях. Це значно підвищує продуктивність і якість коду, а також скорочує час налагодження. Середовище розробки також надає інструменти для роботи з системами контролю версій, сприяючи груповій розробці та інтеграції коду. Зрештою, це допомагає створити більш організований робочий процес, який дозволяє розробникам швидко адаптуватися до змін і ефективно керувати своїм часом і ресурсами [28, с.39].

Серед багатьох IDE для створення веб-додатків деякі з найпопулярніших включають Visual Studio Code, IntelliJ IDEA, PyCharm і WebStorm. Microsoft Visual Studio Code має багато розширень для роботи з різними мовами програмування та фреймворками, що робить його універсальним інструментом. IntelliJ IDEA від JetBrains підтримує широкий спектр мов програмування та має потужні інструменти для розробки додатків Java. PyCharm, також від JetBrains, є специфічною для Python IDE, яка надає нам усі інструменти, необхідні для створення веб-додатків цією мовою. Інший продукт від JetBrains, WebStorm, ідеально підходить для розробки на JavaScript, TypeScript і Node.js і пропонує глибоку інтеграцію з популярними фреймворками та бібліотеками. Кожне з цих середовищ розробки має свої особливості та переваги, тому вибір має залежати від конкретних потреб проекту та особистих уподобань розробника.

Розробники часто стикаються з необхідністю змінювати типи елементів програми. Це завдання вимагає оновлення не тільки самого типу, але й усіх пов'язаних з ним посилань в коді, що займає багато часу і є досить

трудомістким процесом. Хоча зміни типів відбуваються частіше, ніж перейменування, більшість сучасних інструментів IDE надають лише часткову автоматизовану підтримку для обмеженого набору типів. Нещодавні дослідження запропонували підхід, що базується на аналізі даних, для виведення правил переписування API для шаблонів змін типів у Java на основі історії комітів коду. Враховуючи ці досягнення, було створено IntelliTC - інструмент для рефакторингу змін типів у Java. Цей інструмент реалізовано як плагін для IntelliJ IDEA, популярного IDE для Java, розробленого компанією JetBrains. Пропонуються три різні режими роботи з цим інструментом: класичний режим, запропонований рефакторинг і режим інспекції. Для оцінки зручності використання IntelliTC було опитано 22 досвідчених розробників програмного забезпечення, які високо оцінили корисність цього інструменту [37, с. 115].

WebStorm – це інтегроване середовище розробки (IDE) для веб-розробки, розроблене компанією JetBrains. Воно призначене для роботи з різними технологіями веб-розробки, включаючи HTML, CSS, JavaScript, TypeScript та різні фреймворки, такі як React, Angular, Vue.js та інші. В загальному, використання WebStorm як середовища розробки для проекту на TypeScript/React допомагає підвищити продуктивність, надійність та якість розробки. Він надає потужні інструменти та підтримку для роботи з цими технологіями, спрощуючи процес розробки та підтримки проекту [10, с.125].

Visual Studio Code (VS Code) є одним із найпопулярніших середовищ розробки Microsoft. Він відомий своєю гнучкістю та універсальністю, що робить його ідеальним для розробки веб-додатків. VS Code підтримує багато мов програмування та технологій завдяки багатьом розширенням, доступним у Marketplace. Наприклад, ви можете знайти розширення для HTML, CSS, JavaScript, Python, Java, C++ та багатьох інших мов [13, с. 40].

Варто також зазначити, що робоче середовище PyCharm є одним з найкращих, повнофункціональних, спеціалізованих та універсальних для розробки на Python. Воно має багато функцій, які економлять час,

допомагаючи з рутинними завданнями. У головному вікні створеного нами програмного продукту ми обираємо алгоритм шифрування, мову введення даних та ключ шифрування [32 , с.110].

Однією з головних переваг VS Code є простота використання та швидкість, що важливо для великих проектів. Він також підтримує налагодження, підсвічування синтаксису, автоматичне доповнення коду та інтеграцію з Git та іншими системами контролю версій. Ці функції дозволяють розробникам легко бачити зміни коду, виконувати налагодження та тестування, а також працювати в командах.

Крім того, VS Code - це кросплатформний інструмент, який працює на Windows, macOS і Linux. Це дозволяє розробникам працювати у звичному середовищі незалежно від операційної системи, яку вони використовують. Завдяки відкритому вихідному коду та активній спільноті користувачів VS Code постійно оновлюється та отримує нові функції, що робить його ще більш потужним і практичним інструментом розробки.

Visual Studio Code (VS Code) ідеально підходить для створення комерційних веб-додатків, оскільки він надає багато функцій для роботи з популярними фреймворками JavaScript, такими як ReactJS і Vue.js. Щоб розпочати роботу з ReactJS у VS Code, ви можете встановити такі розширення, як ES7+ React/Redux/React-Native-Snippets, які надають попередньо зібрані фрагменти коду для швидкого створення компонентів. Крім того, інтеграція з TypeScript і IntelliSense забезпечує автозавершення та підказки, які полегшують написання коду. VS Code також підтримує інтеграцію з інструментами налагодження, такими як React DevTools, що полегшує налагодження та оптимізацію веб-додатків.

ReactJS – це бібліотека на основі компонентів, яка використовується для розробки інтерактивних інтерфейсів користувача. На сьогодні це найпопулярніша фронтенд JS бібліотека. Вона об'єднує шар відображення (V) у проекті MVC (Model View Controller). ReactJS підтримується Facebook, Instagram та спільнотою індивідуальних розробників і організацій. ReactJS

дозволяє розробляти великі та складні онлайн-додатки, які можуть змінювати свої дані без повторного завантаження сторінки. Мета використання ReactJS – забезпечити кращий користувацький досвід та розробку швидких і потужних веб-додатків. ReactJS також може інтегруватися з іншими бібліотеками або фреймворками JavaScript в MVC, такими як AngularJS [29, с.1154].

Для роботи з Vue.js VS Code також надає спеціальні розширення, такі як «Vetur», які надають такі корисні функції, як підсвічування синтаксису, автозавершення коду та форматування. Розробники можуть використовувати Vue CLI для створення та керування новими проектами безпосередньо з терміналу VS Code, що спрощує процес налаштування та розробки.

Ці розширення та вбудовані інструменти роблять VS Code потужним інструментом створення комерційних веб-додатків, який робить роботу з ReactJS і Vue.js легкою та ефективною, дозволяючи вам швидко писати, налагоджувати та тестувати свій код в одному місці. середовище.

Visual Studio Code - це середовище розробки надає вам усе, що вам потрібно, від швидкого налаштування до гнучкості та потужності численних розширень. Підтримка фреймворків, таких як ReactJS і Vue.js, дозволяє легко створювати сучасні функціональні програми. Функції автозавершення, інтеграції Git і налагодження роблять ваш процес розробки швидким і зручним, дозволяючи вам зосередитися на творчих аспектах вашого проекту. Кінцевий результат – це не тільки ефективний інструмент розвитку, а й справжнє задоволення від роботи [17, с.26].

Гнучкість Visual Studio Code, потужний набір інструментів і численні функції розширення роблять його ідеальним для створення веб-додатків. Автоматичне завершення коду, інтеграція з системами контролю версій і функції налагодження роблять наш процес розробки швидким і ефективним, дозволяючи нам зосередитися на створенні високоякісного веб-додатку.

Основні характеристики інтегрованих середовищ відображені у таблиці

Таблиця 3.1 Основні характеристики інтегрованих середовищ

Параметр	Visual Studio Code	IntelliJ IDEA	PyCharm	WebStorm
Розробник	Microsoft	JetBrains	JetBrains	JetBrains
Основні мови	Багато (JavaScript, Python)	Багато (Java, Kotlin, Groovy)	Python	JavaScript, TypeScript, Node.js
Розширюваність	Висока (через Marketplace)	Вбудована	Вбудована	Вбудована
Інтеграція з VCS	Git, GitHub, Bitbucket	Git, GitHub, SVN, Mercurial	Git, GitHub, SVN, Mercurial	Git, GitHub, SVN, Mercurial
Налагодження	Вбудоване	Вбудоване	Вбудоване	Вбудоване
Синтаксичне підсвічування	Так	Так	Так	Так
Автодоповнення	Так	Так	Так	Так
Кросплатформеність	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux
Вартість	Безкоштовно	Платна	Платна	Платна
Цільова аудиторія	Універсальне	Професійні розробники	Python-розробники	JavaScript-розробники

Як бачимо з таблиці 3.1, Visual Studio Code є безкоштовною, що має майже аналогічні можливості у відповідності до платних IDE. Тому у якості середовища розробки ми обрали саме Visual Studio Code. Його можливостей цілком достатньо для досягнення мети нашого дослідження.

РОЗДІЛ 4. ДОСЛІДНА ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ

4.1 Особливості програмної реалізації сайту

Ключовим елементом розробки веб-додатків є структура проекту. Структура проекту визначає організацію файлів і папок для зручності використання та підтримки коду. Продумана структура дозволяє розробникам швидко знаходити потрібні файли, розуміти логіку проекту та забезпечувати його масштабованість. Веб-додаток зазвичай містить папки для компонентів інтерфейсу користувача, логіки обробки даних, зображень, стилів та інших елементів. Правильна структура проекту сприяє безперебійній роботі команди розробників і сприяє подальшому розширенню функціональності програми.

У нашому випадку ми створили веб-додаток для продажу музичних плагінів. Структура проекту така:

`components` - ця папка зберігає всі елементи інтерфейсу користувача, які використовуються на сторінках сайту. До них належать такі компоненти, як кнопки, форми та панелі навігації, які повторюються на різних сторінках.

`features` - ця папка відповідає за елементи фрагментів сайту. Фрагмент (або фрагменти) - це модуль або фрагмент коду, який забезпечує певні функції на нашому веб-сайті. Наприклад, такі модулі, як обробка даних користувачів, керування продуктами та обробка платежів.

`images` - тут ми зберігаємо всі зображення, які використовуються на нашому сайті. Зображення мають різні формати (JPEG, PNG, SVG тощо) і використовуються для візуалізації контенту, ілюстрацій товарів, банерів тощо.

`styles` - ця папка зберігає всі стилі, написані в CSS, які визначають вигляд і поведінку елементів вашого сайту. До них відносяться кольорові схеми, шрифти, макет сторінки та інші візуальні аспекти.

utils - утиліти, написані на JavaScript, які виконують допоміжні функції у нашому проєкті. Це функції обробки даних, роботи з датою та часом, обробки масивів та об'єктів тощо.

index.js - це основний файл, який відображає весь сайт і містить відповідний код для проєкту. Це початкова точка для нашої програми, де всі компоненти, функції та стилі об'єднуються для створення готового продукту.

Назва, аккаунт, пошук, продукти що сподобалися, кошик	
Категорії товарів, служба підтримки, умови використання	Хіти продажу, замовити одразу
Трендові товари	

Рисунок 4.1 - Шаблон головної сторінки

Наш веб-додаток, шаблон якої зображений на рисунку 4.1, який використовується для продажу музичних плагінів, базується на сучасних веб-технологіях, таких як JavaScript, HTML і CSS. Ключовим елементом інтерфейсу користувача є заголовок, створений за допомогою JSX. Це дозволяє ефективно інтегрувати різні функціональні елементи, включно з навігаційними компонентами, такими як функція пошуку та кошику для покупок, що забезпечує більш зручну роботу. Категорії в лівій частині сторінки відіграють важливу роль у структурі веб-додатку. Реалізація використовує JSX, що робить вміст гнучким і адаптованим до змін.

Не менш важливим компонентом є маршрутизація, яка реалізована за допомогою ReactJS. Це дає змогу створювати багатосторінкові програми з

плавними переходами між сторінками, покращуючи загальну взаємодію з користувачем. Домашня сторінка веб-сайту містить розділ “Лідери продажів”, який виконує важливу маркетингову функцію, привертаючи увагу користувачів до найпопулярніших товарів, а також надає можливість швидко замовити продукт. Наведені нижче популярні продукти допоможуть користувачам отримати уявлення про поточні пропозиції ринку.

Структура програми відображає останні тенденції веб-розробки та використовує такі інструменти, як ReactJS, HTML, CSS і, трошки, Vue.js, щоб створити потужну, продуктивну та просту у використанні веб-програму, яка відповідає сучасним вимогам створити додаток. Стандарти та вимоги до продуктивності.

Наш веб-додаток використовує систему керування базами даних MySQL (СУБД), яка дозволяє нам ефективно організовувати та керувати даними. MySQL дозволяє створювати та зберігати профілі користувачів, у тому числі зберігати такі важливі дані, як ім'я, адреса електронної пошти та пароль. Крім того, ця СУБД надає потужні інструменти для керування замовленнями продуктів.

MySQL дозволяє зберігати дані про замовлення, включаючи інформацію про вибрані елементи, кількість, вартість і статус замовлення. Це спростить процес обробки та виконання замовлень і підвищить зручність для користувачів додатку. MySQL також надає інструменти звітності та аналізу, які допоможуть вам краще зрозуміти потреби клієнтів і оптимізувати процеси.

Таким чином, використання MySQL як СУБД для нашої веб-програми є найкращим вибором, оскільки воно забезпечує надійну та масштабовану платформу для зберігання та керування даними, необхідними для створення профілів користувачів і обробки замовлень.

Для створення нашої системи управління базами даних (СУБД) ми використовуємо Node.js, який забезпечує швидку та ефективну обробку даних. Node.js дозволяє писати серверний код на JavaScript, що спрощує

взаємодію з MySQL і надає різноманітні інструменти для роботи з даними. Завдяки Node.js ми можемо легко керувати запитами до бази даних і виконувати операції зберігання, оновлення та отримання даних, роблячи ваші веб-додатки більш динамічними та надійними.

```
CREATE DATABASE music_plugins_store;

USE music_plugins_store;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE plugins (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    plugin_id INT NOT NULL,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE
CASCADE,
    FOREIGN KEY (plugin_id) REFERENCES plugins(id) ON DELETE
CASCADE
);
```

Ось короткий опис структури бази даних для веб-додатку з продажу музичних плагінів:

1. Таблиця ``users``:

- зберігає інформацію про користувачів;
- поля: ``id`` (унікальний ідентифікатор), ``username`` (логін), ``password`` (хешований пароль), ``email`` (електронна адреса), ``created_at`` (дата створення).

2. Таблиця ``plugins``:

- зберігає інформацію про музичні плагіни;
- поля: ``id`` (унікальний ідентифікатор), ``name`` (назва плагіна), ``description`` (опис), ``price`` (ціна), ``created_at`` (дата створення).

3. Таблиця ``orders``:

- зберігає інформацію про замовлення користувачів;
- поля: ``id`` (унікальний ідентифікатор), ``user_id`` (ідентифікатор користувача), ``plugin_id`` (ідентифікатор плагіна), ``order_date`` (дата замовлення);
- ``user_id`` у таблиці ``orders`` посилається на ``id`` у таблиці ``users``.
- ``plugin_id`` у таблиці ``orders`` посилається на ``id`` у таблиці ``plugins``.

Ця структура дозволяє ефективно зберігати та управляти даними про користувачів, та замовлення в системі.

Щоб створити дизайн нашого веб-додатку, ми використали комбінацію технологій HTML, CSS і Vue.js для створення інтерактивного та гарного інтерфейсу користувача.

HTML служить основою для структури веб-сторінки та визначає розташування різних елементів, таких як заголовки, абзаци, зображення, кнопки тощо. Використовуючи CSS, ми можемо стилізувати ці елементи, додаючи кольори, шрифти, відступи та інші візуальні функції, щоб створити гармонійний, зручний дизайн.

Vue.js додає динамічності та інтерактивності вашим програмам, спрощуючи керування станом компонента та його оновлення в реальному часі. Vue.js дозволяє створювати реактивні компоненти, які взаємодіють з

користувачем, створюючи більш плавний і приємний досвід користувача. Поєднання цих технологій дозволяє створювати сучасні, функціональні та цікаві веб-додатки, які відповідають найвищим стандартам дизайну та зручності використання.

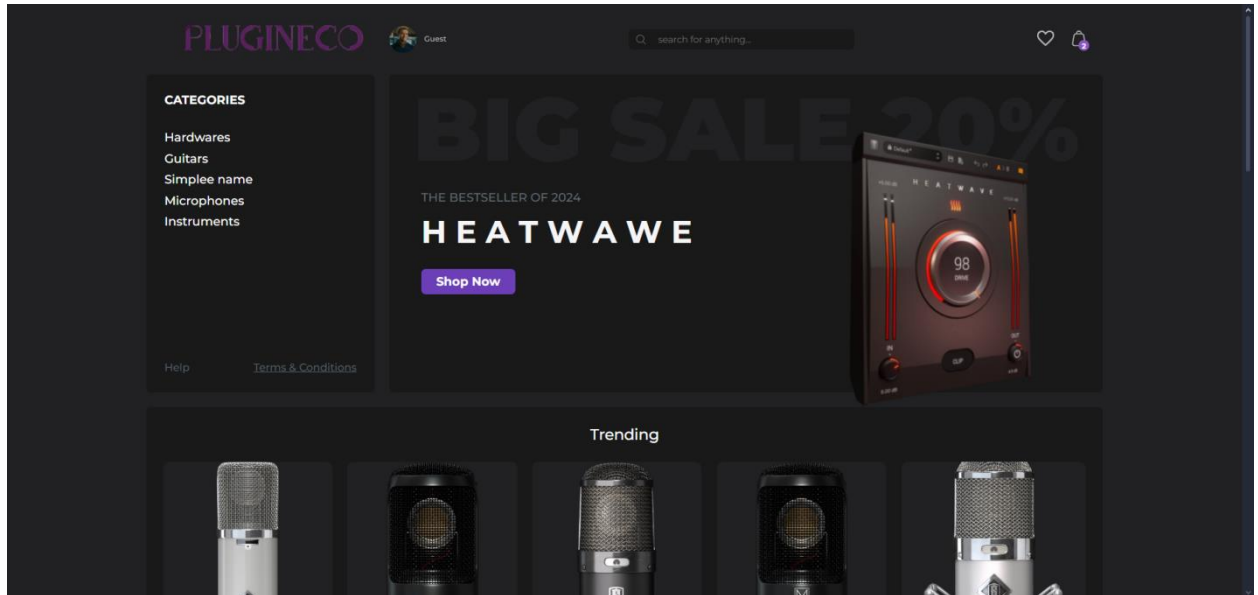


Рисунок 4.2 - Дизайн сторінки

На головній сторінці нашого веб-додатка ми використовуємо фіолетову кнопку з білим написом "Shop Now". Ця кнопка створена за допомогою Vue.js, використовуючи JSX для забезпечення інтерактивності та динамічності. Шаблон кнопки розміщується в секції `<template>`, де визначено HTML-структуру кнопки. Стилзація кнопки, включаючи колір, розміри та ефекти при наведенні курсора, визначається в секції `<style scoped>`.

Для створення кнопки на сторінці з товарами ми використовуємо синю кнопку з білим написом "Add to Cart". Вона також реалізована з використанням Vue.js і JSX. Шаблон кнопки в `<template>` визначає її структуру, а стилі, включаючи фоновий колір, текстовий колір та ефекти при наведенні, визначаються в `<style scoped>`.

Кожна кнопка експортується як окремий компонент у секції `<script>`, що дозволяє легко реюзати її в різних частинах додатка. Цей підхід забезпечує структурованість, масштабованість та полегшує підтримку коду.



Рисунок 4.3 - Кнопки як результат Vue.js

Насамкінець мусимо підкреслити, що наш веб-додаток для продажу музичних плагінів створено з використанням сучасних засобів розробки. Використовуючи такі технології, як HTML, CSS, Vue.js, ReactJS, Node.js, MySQL, ми можемо реалізувати візуально привабливі інтерфейси користувача, а також забезпечити надійну функціональність і високу продуктивність наших програм. Завдяки цим інструментам можна створювати інтерактивні, динамічні та зручні продукти, які оптимізують процеси створення та утримання користувачів, а також виконання замовлень. Ми мусимо прагнути створювати високоякісні та ефективні рішення, які відповідають найновішим стандартам і вимогам ринку, забезпечуючи чудову взаємодію з користувачами та високий рівень задоволеності клієнтів.

4.2. Інструкція користувача веб-додатку

Перед використанням сайту користувач перенаправляється на головну сторінку. У лівій частині сторінки є меню категорій продуктів, що полегшує користувачам навігацію та пошук потрібних музичних плагінів, оскільки вони структуровані відповідно до відповідних категорій.

Центральне вікно на головній сторінці відображає самий ходовий товар. Тут користувач може дізнатися, які музичні плагіни є найпопулярніший і затребуваним серед покупців, допомагаючи їм вибрати найкращий продукт.

Верхня панель містить кілька важливих елементів. У верхньому лівому куті є значок облікового запису користувача, який можна використовувати для входу або реєстрації на сайті. З лівої сторони із значком облікового запису є рядок пошуку, що дозволяє користувачам вводити ключові слова, щоб швидко знайти потрібний плагін. Поруч із рядком пошуку, в правому куті, є значок кошика для покупок, який показує кількість вибраних продуктів і дозволяє переглядати та замовляти їх.

Ці елементи полегшують навігацію на вашій домашній сторінці, а користувачам – легко знайти та придбати потрібний музичний плагін.

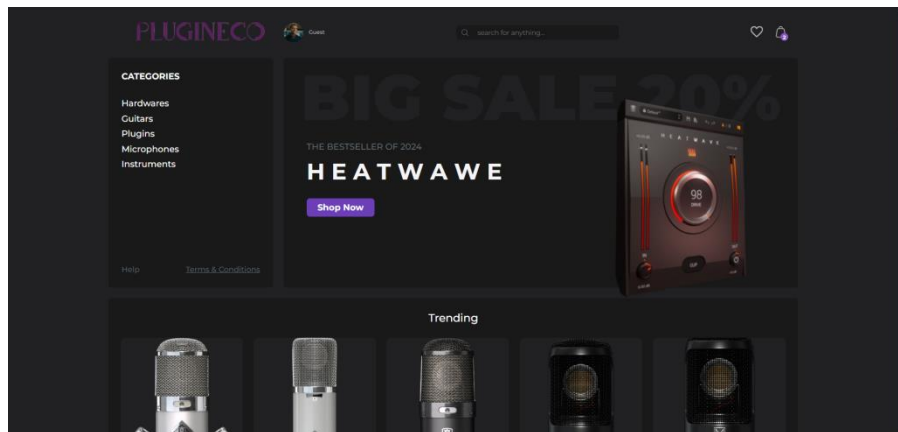


Рисунок 4.4 - Головна сторінка сайту

Коли користувач прокручує сторінку вниз, він бачить, що магазин пропонує безліч популярних товарів. Цей елемент виглядає як рядок, який відображає трендові товари, які можна придбати. Користувач може прокручувати цю хронологію, щоб побачити популярні музичні плагіни, які є актуальними та рекомендованими для придбання. Кожен товар у цьому розділі має фотографію, назву та короткий опис, щоб ви могли швидко зрозуміти, що пропонується.

Ці елементи полегшують навігацію на нашій домашній сторінці, а користувачу – легко знайти та придбати потрібний музичний продукт.

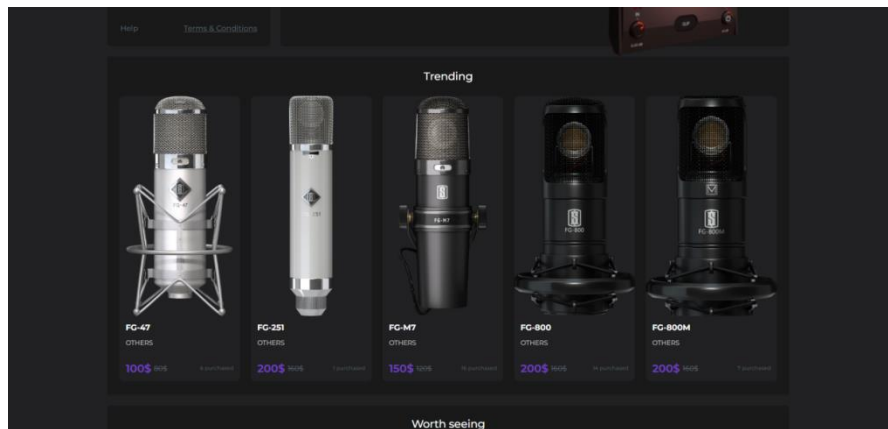


Рисунок 4.5 - Сторінка з трендовими товарами

Користувач може вибрати потрібний продукт і натиснути його, щоб перейти на сторінку з додатковою інформацією про цей продукт. На цій сторінці ми можемо прочитати детальний опис продукту та дізнатися про його технічні характеристики та особливості. Крім того, користувач може побачити кілька фотографій продукту, що допоможе йому отримати краще уявлення про те, як виглядатиме продукт.

На сторінці продукту користувач також може виконувати кілька важливих дій. По-перше, він може замовити товар, заповнивши всі необхідні дані та підтвердивши покупку. Потім користувач може додати продукт у свій кошик, щоб продовжити покупки та оформити замовлення пізніше. Нарешті, у нас є можливість додати продукт до списку бажань, щоб зберегти його для майбутніх покупок або рекомендацій. Ці функції роблять сайт зручним і зручним для користувачів, що дозволяє користувачам легко знаходити та купувати потрібні музичні плагіни.

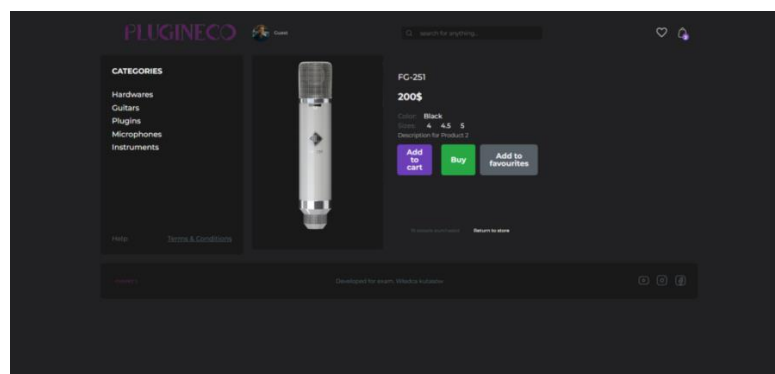
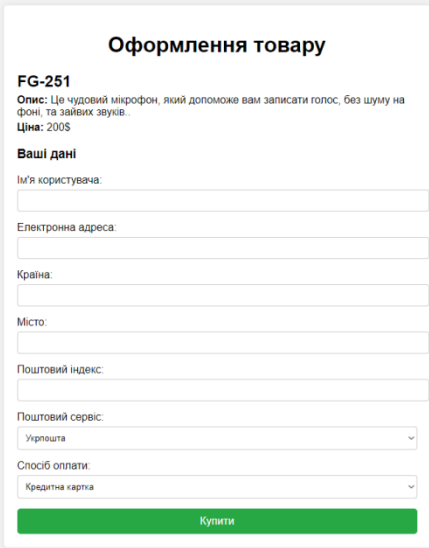


Рисунок 4.6 - Сторінка продукту

Якщо користувач вирішив придбати товар, він натискає кнопку «Buy» і перенаправляється на сторінку оформлення замовлення. На цій сторінці користувачеві пропонується ввести наступні дані:

-ім'я, адреса електронної пошти, країна, місто, поштовий індекс, вибір поштової служби, спосіб оплати. Кожне поле містить докладні підказки, щоб уникнути плутанини та допомогти користувачеві безпомилково ввести всю необхідну інформацію. Після заповнення всіх необхідних полів користувач може продовжити оплату, натиснувши відповідну кнопку. На сторінці оплати користувач може переглянути огляд замовлення, перевірити введені дані, вибрати найбільш зручний спосіб оплати та завершити покупку. Цей процес дозволяє нам легко та зручно завершити покупку музичних продуктів на нашому сайті та забезпечує нашим користувачам позитивні враження від використання нашого магазину.



Оформлення товару

FG-251
Опис: Це чудовий мікрофон, який допоможе вам записати голос, без шуму на фоні, та зайвих звуків.
Ціна: 200\$

Ваші дані

Ім'я користувача:

Електронна адреса:

Країна:

Місто:

Поштовий індекс:

Поштовий сервіс:

Спосіб оплати:

Рисунок 4.7 - Сторінка оформлення замовлення

4.3. Тестування веб-додатку

Тестування під час веб-розробки є дуже важливим кроком для забезпечення якості продукту. По-перше, тестування допомагає виявити та виправити помилки на ранній стадії розробки. Це економить час і ресурси, оскільки виправлення помилок на наступних етапах або після випуску може

бути дуже важким і дорогим процесом. Крім того, будь-які помилки можуть спричинити серйозні збої у нашому додатку, негативно вплинувши на взаємодію з користувачем.

Ще одна важлива причина для тестування - забезпечення безпеки нашого веб-додатку. Сайти часто обробляють конфіденційну інформацію користувача, таку як особисті дані, платіжну інформацію тощо. Без належного тестування можуть існувати вразливості, якими зловмисники можуть скористатися для отримання несанкціонованого доступу до цих даних. Регулярне тестування допомагає виявити й усунути потенційні загрози безпеці, переконатися, що дані користувачів захищено, а також покращити загальну безпеку вашої програми.

Крім того, тестування допомагає покращити взаємодію з користувачем. Тестування передбачає перевірку різних аспектів функціональності програми, таких як навігація, швидкість завантаження сторінок і точність відображення вмісту. Це гарантує, що веб-додаток залишається стабільним та зручним для користувача. Якість взаємодії з користувачами є ключовим фактором утримання користувачів і залучення нових, безпосередньо впливаючи на успіх нашого веб-додатку.

Тому тестування є важливою частиною процесу веб-розробки, гарантуючи якість, безпеку та зручність використання продукту. Це не просто крок у розробці, а стратегічний підхід, який допоможе створити надійну та конкурентоспроможний продукт.

Тестування сайту починається з відкриття середовища розробки, яке відіграє ключову роль у цьому процесі. Середовище розробки надає вам усі інструменти, необхідні для запуску нашої програми, і дозволяє виявляти кожен момент налагодження нашої програми.

Сучасне середовище розробки не тільки полегшує написання коду, але й дозволяє зосередитися на виявленні та виправленні помилок. У нашому випадку ми будемо використовувати Visual Studio Code (VS Code), який

популярний серед розробників завдяки своїй універсальності та зручному інтерфейсу.

Крім того, VS Code надає широкий набір розширень, які додають додаткові можливості тестування та налагодження. Наприклад, різні розширення для статичного аналізу коду можуть допомогти нам виявити потенційні проблеми на ранній стадії розробки. Інтегрується з системами контролю версій, такими як Git, що дозволяє легко відстежувати зміни коду та повертатися до попередніх версій, якщо потрібно.

Дебагінг - це процес виявлення, діагностики та виправлення помилок (або багів) у програмному забезпеченні. Це один із найважливіших етапів розробки програмного забезпечення, оскільки він забезпечує стабільність і правильну роботу програмного забезпечення.

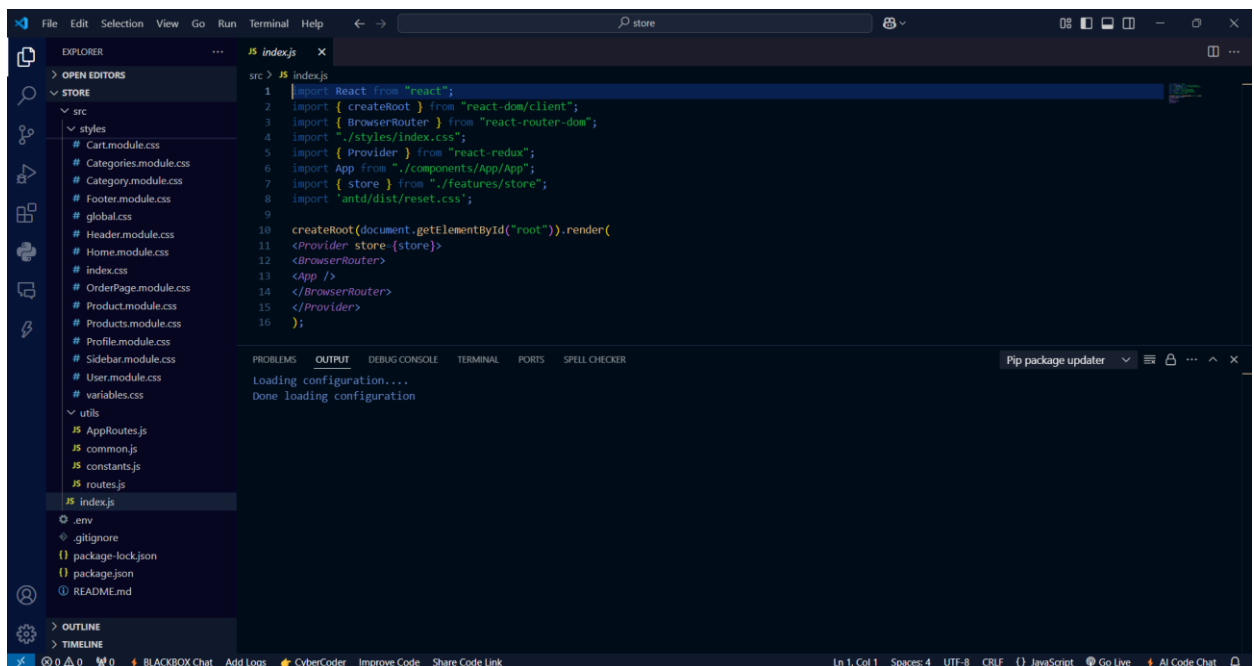


Рисунок 4.8 - Вікно середовища розробки

Для моніторингу процесу налагодження ми маємо важливий інструмент у будь-якому середовищі розробки: термінал. Термінал дозволяє запускати різні процеси в режимі реального часу, включаючи тестування коду. Це дозволяє відстежувати прогрес нашої програми та негайно виявляти будь-які

помилки, які можуть виникнути. У нашому випадку ми почнемо процес тестування за допомогою команди «yarn start».

Yarn - це менеджер пакетів JavaScript, який забезпечує швидке, надійне та безпечне керування залежностями у наших проектах. Це дозволяє встановлювати та оновлювати пакети, які використовуються у нашому проекті, і гарантувати, що всі залежності працюють належним чином. Однією з головних особливостей Yarn є його швидкість. Він використовує кеш для встановлених пакетів і виконує всі операції паралельно, що значно прискорює процес встановлення.

Команда «yarn start» запускає нашу програму в середовищі розробки, дозволяючи нам негайно побачити результати коду та виявити помилки в реальному часі. Це забезпечує ефективний процес налагодження, дозволяючи швидко реагувати на проблеми та вносити необхідні виправлення. Термінал відображає всі повідомлення про помилки, попередження та іншу важливу інформацію, яка допомагає нам краще зрозуміти, що відбувається у нашому коді під час виконання.

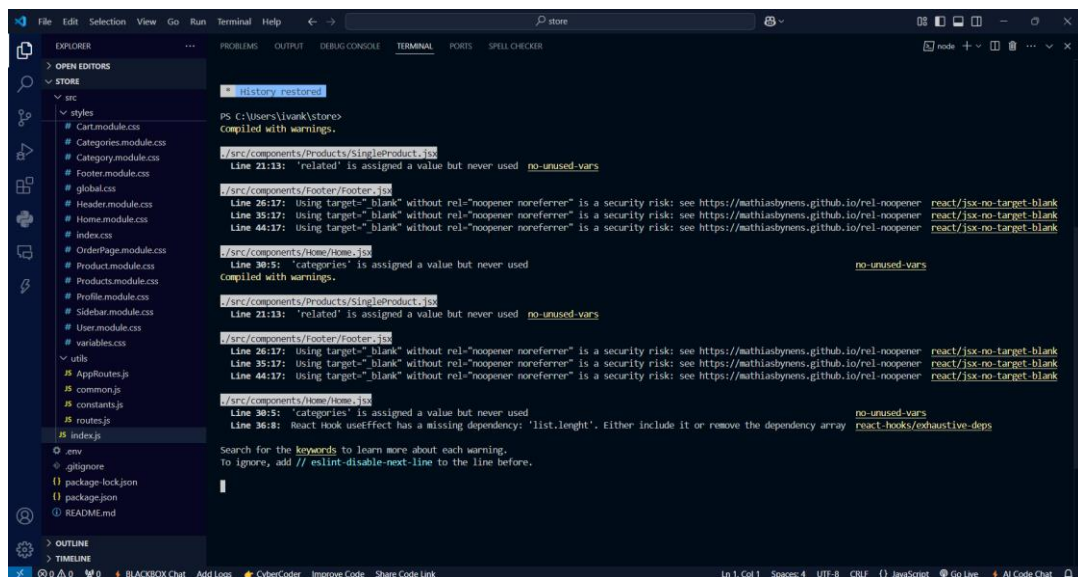


Рисунок 4.9 - Термінал з дебагінгом

Процес дебагінгу є дуже важливою частиною тестування програми, оскільки він дозволяє виявити та виправити помилки, які можуть вплинути на функціональність або зручність використання програми. Під час

тестування програми ми виявили деякі помилки в коді, які створювали незручності під час використання.

По-перше, ми виявили деякі зайві невикористовувані змінні у файлі "SingleProduct.jsx". Ця змінна просто займає місце у нашому коді та робить його менш зрозумілим і організованим. Крім того, у файлі "SingleProduct.jsx" були знайдені змінні без значень, що також викликає проблеми під час запуску програми. Далі ми знайшли невикористану змінну у файлі "Footer.jsx", яка залишилася після рефакторингу коду. Знову ми знайшли невикористані змінні, що залишилися після змін коду у файлі "Footer.jsx".

У Home.jsx знайдено невідому змінну. Його використання не задокументовано або налаштовано неправильно, що спричиняє помилки. Після завершення тестування ми виявили, що дві змінні знову не використовуються в Home.jsx, що вказує на те, що нам потрібно додатково очистити й оптимізувати наш код.

Додаткові змінні можуть відволікати розробника та ускладнювати читання коду, що може призвести до плутанини та ускладнити роботу з кодом. Подібні помилки часто виникають під час створення програмних продуктів. Це особливо часто трапляється під час швидкої розробки або внесення змін до існуючого коду. Це може статися з різних причин, зокрема через недостатнє тестування, прогалини в комунікації всередині команди розробників або просто людський фактор. Такі помилки тестування не обмежуються клієнтською частиною веб-програми. Це також може статися на серверній частині, яка обробляє та обробляє дані. Щоб знайти серверні помилки та вразливості, переходимо до MySQL.

```
[2025-01-21 12:00:00] ERROR: Помилка підключення до бази даних:  
Access denied for user 'wrong_user'@'localhost' (using password: YES)
```

```
[2025-01-21 12:01:00] INFO: Підключено до бази даних як ID 5
```

```
[2025-01-21 12:02:00] ERROR: Помилка виконання запиту: Duplicate  
entry 'testuser' for key 'users.username'
```

[2025-01-21 12:03:00] INFO: Результати запиту: { insertId: 1, affectedRows: 1, changedRows: 0 }

[2025-01-21 12:04:00] ERROR: Помилка виконання запиту: Table 'music_plugins_store.plugins' doesn't exist

[2025-01-21 12:05:00] INFO: Результати запиту: [
 { id: 1, name: 'Super Synth', description: 'Чудовий синтезатор для створення унікальних звуків.', price: 49.99 },
 { id: 2, name: 'Drum Machine', description: 'Драм машина для створення ритмів.', price: 39.99 }
]

[2025-01-21 12:06:00] ERROR: Помилка виконання запиту: Foreign key constraint fails for order with user_id 1 and plugin_id 3

[2025-01-21 12:07:00] INFO: Результати запиту: { insertId: 1, affectedRows: 1, changedRows: 0 }

[2025-01-21 12:08:00] INFO: З'єднання з базою даних закрито.

Під час процесу налагодження веб-додатку MySQL, було виявлено кілька критичних помилок, через які система не функціонувала належним чином. По-перше, у нас виникли проблеми з підключенням до бази даних через неправильні облікові дані, що призвело до помилки «Access denied for user». Після виправлення даних доступу з'єднання буде успішно встановлено і ми зможемо продовжити роботу з базою.

Ми також отримали помилку під час виконання запитів для вставки нових користувачів і замовлень. Наприклад, коли ми намагалися додати нового користувача, ми отримали повторювану помилку, оскільки користувач із таким логіном уже існував. Після вирішення цієї проблеми запит на вставку виконано успішно. Однак, коли ми намагалися вставити нове замовлення, ми отримали повідомлення про помилку порушення зовнішнього ключа, яке вказувало на відсутність відповідного запису в таблиці користувачів або таблиці плагінів.

Нарешті, після виправлення всіх помилок, система успішно виконала запит на отримання плагіна та вставлення замовлень, підтверджуючи, що база даних функціонує правильно. Завдяки успішному результату запиту ми змогли отримати необхідні дані про плагін і зберегти нове замовлення. Закриття з'єднання з базою даних також пройшло без помилок, що свідчить про стабільність системи після завершення всіх операцій.

У процесі тестування нашого веб-додатку ми виявили кілька важливих помилок і недоліків, які необхідно було негайно виправити. Виявлення таких помилок важливо для забезпечення належної якості програмного продукту, тим самим покращуючи взаємодію з користувачем і загальну стабільність програми.

Перед налагодженням MySQL для веб-додатку, нам слід уважно вивчити облікові дані, структуру бази даних і зв'язки між таблицями. Виявлені помилки, такі як неправильні облікові дані, дублікати записів та порушення зовнішніх ключів, підкреслюють необхідність належної валідації даних перед виконанням запитів. Після вирішення цих проблем система успішно виконала запити на введення та отримання даних, підтверджуючи стабільність і готовність до продовження роботи. Цей процес налагодження не тільки покращив функціональність програми, але й надав цінний досвід, який стане в нагоді для подальшої роботи з базами даних.

Таким чином, регулярне тестування та налагодження веб-додатків дозволило нам виявити та виправити критичні помилки, забезпечити стабільність і покращити загальну якість наших програмних продуктів.

ВИСНОВКИ

Цифровізація музичної індустрії охоплює використання онлайн-платформ, віртуальних інструментів, програмного забезпечення для аранжування, а також відеоуроків та майстер-класів. Такі ресурси та інструменти забезпечують доступ до навчальних матеріалів із будь-якої точки світу та відкриває можливості для експериментів із різними стилями музики за допомогою цифрових інструментів, таких як синтезатори, електронні барабани або цифрові піаніно.

Для реалізації електронної комерції в музичній індустрії використовуються різноманітні сайти або онлайн майданчики. Наприклад, веб-сайти музичних проєктів публікувати свій контент, надавати інформацію про історію проєкту, анонсувати тури та концерти. Такі сайти стають віртуальним середовищем для взаємодії з любителями музики, забезпечуючи зручність спілкування через блоги, форуми або інтеграцію із соціальними мережами. Важливою складовою музикальних веб майданчиків є можливість продажу альбомів, брендових товарів, квитків на концерти тощо.

У ході виконання роботи був успішно розроблений інтернет-магазин, що використовує сучасні технології, такі як Payload CMS, Next.js, MongoDB, GraphQL, SCSS та TypeScript. Проєкт має гнучку та масштабовану архітектуру, яка дозволяє керувати контентом, легко розширювати функціональність та забезпечувати зручний досвід для користувачів.

Основна увага була зосереджена на адаптації серверної частини під специфічні функціональні вимоги проєкту, зокрема через налаштування Payload CMS для оброблення та зберігання даних. Було визначено структури даних для основних колекцій, таких як користувачі, продукти, замовлення, категорії та медіаресурси. Для деяких полів додано бізнес-логіку через хуки, що автоматизує процеси та забезпечує валідацію даних, створюючи таким чином гнучку архітектуру, здатну адаптуватися до змін і потреб інтернет-магазину.

Клієнтська частина проєкту була розроблена за допомогою Next.js, що дозволило інтегрувати її з серверною частиною через GraphQL. Це сприяло оптимізації обміну даними між клієнтом і сервером та динамічному оновленню інтерфейсу. Ключові сторінки магазину, такі як авторизація, кошик, замовлення та адміністративна панель, були налаштовані з фокусом на зручність для користувачів та адміністраторів.

Окремо було налаштовано інтеграцію з платіжною системою Stripe для безпечних і зручних платежів, що є важливим аспектом для онлайн-торгівлі. Після налаштування, система успішно проходила тестування, що підтвердило правильність роботи як платіжних операцій, так і загальної функціональності магазину.

Таким чином, мета дипломної роботи, що полягала у розробці веб-додатку з продажу музичних плагінів з дисципліни «Комп'ютерні науки» та експериментальній перевірці його ефективності, була досягнута, а всі поставлені у ході виконання дипломної роботи задачі виконані.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Ануфрієва О. Л. Електронна комерція в системі цифрової економіки. *Науковий вісник УМО. Серія: Економіка та управління*. 2017. С. 1–12.
2. Волощук К.Б. Електронна комерція в Україні та основні інноваційні тренди її розвитку. *Podilian Bulletin: Agriculture, Engineering, Economics*. 2019. № 31. С. 98–109.
3. Грабовський Є. М., Матейко А. В. Дослідження специфіки розробка сайту для просування музичного проекту. Поліграфічні, мультимедійні та web-технології : тези доп. IX Міжнар. наук.-техн. конф., 14-18 травня 2024 р. Т. 1. С. 175–176.
4. Дмитренко Т., Толочин М. Сучасні онлайн компілятори і інтерпретатори. Тези 72-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету, присвяченої 90-річчю Національного університету «Полтавська політехніка імені Юрія Кондратюка» (21 квіт.-15 трав. 2020 р., м. Полтава). Полтава : Національний університет імені Юрія Кондратюка, 2020. Т. 1. С. 477– 478.
5. Єремеева Х. А., Стогул О. І. Розробка методики побудови функціональних діаграм в нотаціях IDEF0. *Менеджмент організацій та адміністрування*. 2015. С. 126–128.
6. Зінов'єва О. Г. Використання CASE-засобів для проектування інформаційних систем. Українські студії в європейському контексті. 2023. № 7. С. 220–227.
7. Івашко Л. М. Роль інформаційно-комунікаційних технологій при навчанні та реалізації основних положень дисципліни Електронна комерція. *Наукові праці Донецького національного технічного університету. Сер.: Економічна*. № 4. 2014. С. 286–293.
8. Кисельова А. Д. Розробка Web додатку з використанням фреймворку Vue.js : кваліфікаційна робота магістра. Запоріжжя, 2020. 49 с.

9. Кишакевич Б. Ю., Кишакевич С. В., Стець Г. В. Сучасні тенденції цифровізації музичної освіти. *Академічні візії*. 2024. № 27. С. 1–10.
10. Козійчук А. О., Романюк О. В. Розробка веб-додатку для цифровізації фільмотек. *Електронні інформаційні ресурси: створення, використання, доступ*. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції (20-21 листопада 2023 р., м. Вінниця). 2023. С. 123–126.
11. Кравчук О. А. Віртуальний музичний синтезатор у форматі VST плагіну: дипломний проект бакалавра. Київ, 2021. 96 с.
12. Кучерков А. О. Напрямки використання криптовалюти. *Українські студії в європейському контексті*. 2023. № 6. С. 97–101.
13. Ларіоненко О. І. Розробка веб-додатку «Інтернет-магазин з продажу книжок» із застосуванням reactjs: випускна робота молодшого спеціаліста. Запоріжжя, 2023. 62 с.
14. Лебеденко М. С., Лученко І. В. Веб-ресурс як ефективний інструмент маркетингових комунікацій. *Вісник Хмельницького національного університету*. 2011. № 2(1). С. 178–182.
15. Луцков М. П. Веб-додаток для аналізу та покращення функціональності веб-сайтів: Кваліфікаційна робота магістра. Вінниця, 2024. 68 с.
16. Майданюк І. Г. Розробка веб додатків за допомогою JavaScript. Матеріали ІІ науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2022) : збірник доповідей. Вінниця : ВНТУ, 2022. С. 14–18.
17. Максимович Д. Розробка інтернет-магазину цифрових гаджетів: кваліфікаційна робота бакалавра. Київ, 2021. 48 с.
18. Паламарчук Д.М., Тимошенко А.С. Електронна комерція як інструмент забезпечення сталого розвитку: досвід ЄС. *Збірник наукових праць Черкаського державного технологічного університету. Серія: Економічні науки*. № 49. 2018. С. 76–83.

19. Поплавський М. М., Трач Ю. В. Цифровізація музичної індустрії: тенденції і перспективи. *Вісник Національної академії керівних кадрів культури і мистецтв*. 2022. № 2. С. 30–39.
20. Федишин І.Б. Електронний бізнес та електронна комерція (опорний конспект лекцій для студентів напрямку «Менеджмент» усіх форм навчання). Тернопіль, ТНТУ імені Івана Пулюя, 2016. 97 с.
21. Філіппова М. В., Демченко М. О., Матвієнко С. М. Моделювання технологічного процесу складання за допомогою IDEF0. *Східно-Європейський журнал передових технологій*. 2013. Т. 2. № 3(62). С. 44–47.
22. Хижняк О.С. Недоліки електронного бізнесу в Україні і шляхи їх усунення. *Науковий вісник Херсонського державного університету. Сер.: Економічні науки*. 2016. № 16(1). С. 90–94.
23. Чуприна О. О. Електронна комерція як основна ланка торгівлі майбутнього. Тези конференції «Економіка і менеджмент – 2021». Секція 8. Моделювання процесів інноваційного розвитку в економіці. URL: http://www.confcontact.com/eim2021/8_Chupryna.pdf
24. Шаров С. В., Скрипка С. О. Використання мови UML для інфологічного моделювання реляційної бази даних. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення*. 2014. № 7. С. 5–9.
25. Шаров С. В., Шамардак О. А. Концептуальне моделювання інформаційної системи за допомогою мови UML. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення*. 2014. № 7. С. 10–14.
26. Шокур О. І. Інтерактивний веб додаток для інтернет галереї «Shuraz_art»: кваліфікаційна робота бакалавра. Суми, 2022. 41 с.
27. Юдін Д. Ф. Розробка фулстек веб-додатку для продажу електроніки на фреймворку React та NodeJS: кваліфікаційна робота бакалавра. Хмельницький, 2024. 103 с.

28. Якубич К. О. Розробка веб-додатку моніторингу закладів освіти місцевого рівня: кваліфікаційна робота бакалавра. Вінниця, 2022. 60 с.
29. Bhalla A., Garg S., Singh P. Present day web-development using reactjs. *International Research Journal of Engineering and Technology*. 2020. Т. 7. № 05. P. 1154–1157.
30. FabFilter Redefining EQ. URL: <https://www.fabfilter.com/products/pro-q-4-equalizer-plugin-in>.
31. iZotope by Native Instruments. URL: <https://www.izotope.com>.
32. Kornilovska N. V., Vyshemyrska S. V., Kolmykov M. Development of python electronic message information protection system using the pycharm working area. *Вісник Херсонського національного технічного університету*. 2021. № 1(76). С. 106–112.
33. Native Instruments overview, All Products. URL: <https://www.native-instruments.com/en/catalog/>.
34. Presley A., Liles D. H. The use of IDEF0 for the design and specification of methodologies. Proceedings of the 4th industrial engineering research conference. 1995. С. 442–448.
35. Royalty-Free Sounds, FX, Presets & More. URL: <https://splice.com/features/sounds>.
36. Sharov S., Rassokha I. Specific features of using Web 2.0 technology in the educational process. *Bulletin of the Cherkasy National University named after Bohdan Khmelnytskyi. Series: Pedagogical sciences*. 2022. № 3. С. 26–33.
37. Smirnov O. et al. Intellitc: Automating type changes in intellij IDEA. *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering*. 2022. С. 115–119.
38. Sweetwater, shop by category. URL: <https://www.sweetwater.com/shop/by-category/>.
39. Wawes Creative Access. URL: <https://www.waves.com/subscriptions>.

Код (лістинг) програми

```
import React, { useEffect } from "react";
import Poster from '../Poster/Poster'; // Проверь путь
import Products from "../Products/Products";
import Categories from "../Categories/Categories";
import Banner from "../Banner/Banner";
import { useDispatch, useSelector } from "react-redux";
import { filterByPrice } from "../../features/products/productsSlice";
const productsData = [
  { id: 1, images: ['https://slatedigital.com/wp-content/uploads/2019/12/slate-digital-FG-47-1.png'], title: 'FG-47', category: { name: 'Others' }, price: 100 },
  { id: 2, images: ['https://slatedigital.com/wp-content/uploads/2019/12/slate-digital-FG-M7-1.png'], title: 'FG-M7', category: { name: 'Others' }, price: 150 },
  { id: 3, images: ['https://slatedigital.com/wp-content/uploads/2019/12/slate-digital-FG-800-1.png'], title: 'FG-800', category: { name: 'Others' }, price: 200 },
];
const categoriesData = [
  { id: 1, name: 'Microphones', image: ['https://slatedigital.com/wp-content/uploads/2023/01/Screen-Shot-2022-11-11-at-2.04.02-PM.png'] },
  { id: 2, name: 'Plugins', image: ['https://slatedigital.com/wp-content/uploads/2021/11/Murda-Melodies-High-Res-GUI-2x.png'] },
  { id: 3, name: 'Instruments', image: ['https://slatedigital.com/wp-content/uploads/2022/11/P1001379-1@2x.png'] },
  { id: 4, name: 'Guitars', image: ['https://www.guitarcenter.com/static/gc/2024/tent-pole/GAT-Home/mobile/GAT-Fall-2024-Mobile-C2-NewArrivals-310x312.jpg'] },
```

```

    {    id:    5,    name:    'Hardwares',    image:    ['https://www.native-
instruments.com/typo3temp/pics/img-packshot-maschine-plus-product-finder-
3a7886479dd4f209f7615e9522a407d0-d.jpg'] },
];

const Home = () => {
  const dispatch = useDispatch();
  const { products: {list, filtered },
categories } = useSelector((state) => state);
  useEffect(() => {
    if(!list.lenght) return;
    dispatch(filterByPrice(100))
  }, [dispatch, list.length]);
  return (
    <>
      <Poster />
      <Products products={productsData} amount={5} title="Trending" />
      <Categories products={categoriesData} amount={5} title="Worth seeing" />
      <Banner/>
      <Products products={filtered} amount={5} title="Trending" />
    </>
  );
};

export default Home;
import React from "react";
import { Link } from "react-router-dom";
import styles from "../styles/Header.module.css";
import { ROUTES } from "../utils/routes";
// Импорт логотипа в формате PNG
import LOGO from "../images/logo.png";
import AVATAR from "../images/avatar.jpg";

```

```

const Header = () => {
  return (
    <div className={styles.header}>
      <div className={styles.logo}>
        <Link to={ROUTES.HOME}>
          {/* Замена формата логотипа на PNG */}
          <img src={LOGO} alt="Stuff" />
        </Link>
      </div>
      <div className={styles.info}>
        <div className={styles.user}>
          <img className={styles.avatar} src={AVATAR} alt="User Avatar" />
          <div className={styles.username}>Guest</div>
        </div>
        <form className={styles.form}>
          <div className={styles.icon}>
            <svg className="icon">
              <use
xlinkHref={`$${process.env.PUBLIC_URL}/sprite.svg#search`} />
            </svg>
          </div>
          <div className={styles.input}>
            <input
              type="search"
              name="search"
              placeholder="search for anything..."
              autoComplete="off"
              onChange={() => {}}
              value=""
            />
          </div>
        </form>
      </div>
    </div>
  )
}

```

```

    </div>
    {false && <div className={styles.box}></div>}
  </form>
  <div className={styles.account}>
    <Link to={ROUTES.HOME} className={styles.favourites}>
      <svg className={styles["icon-fav"]}>
        <use
xlinkHref={`${process.env.PUBLIC_URL}/sprite.svg#heart`} />
      </svg>
    </Link>
    <Link to={ROUTES.CART} className={styles.cart}>
      <svg className={styles["icon-cart"]}>
        <use xlinkHref={`${process.env.PUBLIC_URL}/sprite.svg#bag`}
      />
    </svg>
    <span className={styles.count}>2</span>
  </Link>
</div>
</div>
</div>
);};

```

```

export default Header;
import React from "react";
import { Link } from "react-router-dom";

import styles from "../../styles/Footer.module.css";
import { ROUTES } from "../../utils/routes";
import LOGO from "../../images/logo.png";
const Footer = () => {

```

```

return (
  <section className={styles.footer}>
    <div className={styles.logo}>
      <Link to={ROUTES.HOME}>
        {/* Замена формата логотипа на PNG */}
        <img src={LOGO} alt="Stuff" />
      </Link>
    </div>
    <div className={styles.rights}>
      Developed for exam. Władca kutasów
    </div>
    <div className={styles.socials}>
      <a
        href="https://www.youtube.com/@OfficialDoctorBonov"
        target="_blank"
        rel="noreferrer">
        <svg className="icon">
          <use xlinkHref={`${process.env.PUBLIC_URL}/sprite.svg#youtube`}
        />
        </svg>
      </a>
      <a href="https://instagram.com"
        target="_blank"
        rel="noreferrer">
        <svg className="icon">
          <use
xlinkHref={`${process.env.PUBLIC_URL}/sprite.svg#instagram`} />
        </svg>
      </a>
      <a

```

```

href="https://facebook.com"
target="_blank"
rel="noreferrer">
<svg className="icon">
  <use
xlinkHref={`${process.env.PUBLIC_URL}/sprite.svg#facebook`} />
  </svg>
</a>
</div>
</section>
  });
export default Footer;

```