

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ
ДЛЯ ОНЛАЙН ПРОДАЖУ АНІМЕ АКСЕСУАРІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF A WEB
APPLICATION FOR ONLINE SALE OF ANIME ACCESSORIES

Виконав: студент(ка) 2 курсу, групи КНТ-222м

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

ЦИММЕРМАН О.Г.

(ПРИЗВИЩЕ та ініціали)

Керівник КАПЛІЄНКО Т.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

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

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЦИММЕРМАНА Олексія Геннадійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація вебзастосунку для онлайн продажу аніме аксесуарів. Research and Software Implementation of a Web Application for Online Sale of Anime Accessories

керівник проєкту (роботи) к.т.н., доцент, КАПЛІЄНКО Тетяна Ігорівна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “14” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

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

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Розробка архітектури програми. 4. Розробка програми. 5. Експлуатація, тестування та експериментальне дослідження програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	КАПЛІЄНКО Т.І., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання « 08 » вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Огляд матеріалів та методів вирішення задачі.	2-3 тижні	Розділ 2
4	Вибір мови програмування та інших технологій розробки.	4 тиждень	Розділ 3
5	Розробка архітектури програми.	5-6 тижні	Розділ 4
6	Розробка програми.	7-8 тижні	Розділ 4, 5
7	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 5
8	Оформлення пояснювальної записки та документів до неї.	10 тиждень	Додатки
9	Нормоконтроль та рецензування.	11 тиждень	
10	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Олексій ЦИММЕРМАН
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна КАПЛІЄНКО
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
111 с., 7 табл., 53 рис., 3 дод., 15 джерел.

ВЕБЗАСТОСУНОК, MYSQL, БАЗА ДАНИХ, HTML, BOOTSTRAP, CSS, PHP, ПРОДАЖ ТОВАРІВ.

Об'єкт дослідження – продаж товарів онлайн.

Предмет дослідження – вебзастосунок для онлайн продажу аніме аксесуарів.

Мета роботи – розробка вебзастосунку для магазину аніме аксесуарів, задля спрощення функціонування магазину та покращення кількості продажів.

Матеріали, методи та технічні засоби: мова розмітки документів HTML, мова стилю сторінок CSS, мови програмування PHP та JavaScript, система керування базами даних MySQL.

Результати. Виконано огляд і аналіз основних принципів зручності користування вебзастосунків. За допомогою мови розмітки документів HTML та мови стилю сторінок CCS, фреймворку Bootstrap й мов програмування PHP та JavaScript створено вебзастосунок, який дозволяє спростити власнику магазину продаж товарів, підвищити ефективність роботи і збільшити прибутки.

Висновки. Розроблено вебзастосунок для магазину аніме аксесуарів, що дозволяє покупцю купувати необхідні йому товари онлайн, а власникам магазину додавати та видаляти пропозиції щодо продажу, змінювати вже наявні пропозиції, отримувати зворотній зв'язок від покупців, а також здійснювати продаж товарів по всіх куточках України і світу.

Галузь використання. Організація продажу аніме аксесуарів, створення платформи для взаємодії покупців та власників магазину.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 111 pages, 7 tables, 53 figures, 3 appendixes, 15 sources.

WEB APPLICATION, MYSQL, DATABASE, HTML, BOOTSTRAP, CSS, PHP, SALE OF PRODUCTS.

Object of study is online sale of products.

Subject of study is a web application for online sale of anime accessories.

The purpose of the work is Web Application Development for an anime accessories store, in order to simplify the operation of the store and improve the number of sales.

Materials, methods, and tools: document markup language HTML, language style pages CSS, programming languages PHP and JavaScript, Database Management System MySQL.

Results. Done a review and analysis of the main principles of usability of web applications. With the help of the HTML document markup language and the CCS page style language, the Bootstrap framework, and the PHP and JavaScript programming languages, a web application has been created that allows the store owner to simplify the sale of products, increase work efficiency and increase profits.

Conclusions. A web application for an anime accessories store, which allows the buyer to buy the products he needs online, and the store owners to add and delete offers for sale, change already existing offers, receive feedback from buyers, as well as sell goods in all corners of Ukraine and the world.

The field of use is a organization of the sale of anime accessories, creating a platform for interaction between buyers and store owners.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	10
1.1 Можливі форми продажу аніме аксесуарів	10
1.2 Програмні засоби продажу аніме аксесуарів	13
1.3 Висновки за першим розділом	15
2 Матеріали і методи	17
2.1 Дослідження та порівняння вебсайтів та вебзастосунків	17
2.2 Дослідження головних принципів зручності користування вебзастосунків	22
2.3 Висновки за другим розділом	31
3 Розробка архітектури програми	33
3.1 Функціональні вимоги до програмного забезпечення	33
3.2 Вибір засобів розробки	35
3.3 Проектування бази даних	39
3.4 Висновки за третім розділом	43
4 Розробка програми	44
4.1 Алгоритм функціонування вебзастосунку	44
4.2 Структура програмного забезпечення	46
4.3 Опис рішень, прийнятих під час реалізації програмного продукту	50
4.4 Висновки за четвертим розділом	54
5 Експлуатація, тестування та експериментальне дослідження програми	55
5.1 Призначення програми	55
5.2 Умови виконання програми	55
5.3 Виконання програми	55
5.4 Тестування програмного засобу	67
5.5 Реалізація принципів зручності користування	71
Висновки	74

Перелік джерел посилання	76
Додаток А Технічне завдання.....	78
Додаток Б Текст програми.....	82
Додаток В Слайди презентації	101

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- CSS – Cascading Style Sheets;
- HTML – HyperText Markup Language;
- JS – JavaScript;
- SQL – Structured Query Language;
- БД – База даних;
- IT – Інформаційні технології;
- СКБД – Система керування базами даних.

ВСТУП

На початку 21 століття аніме було популярне лише на своїй батьківщині, у Японії. Проте у 2010-х, і особливо зараз, у 2020-х роках XXI століття відбувається справжній бум популярності японської мультиплікації. Наприклад у Сполучених Штатах аніме поступово набуває визнання як один з найбільших видів розваг. За даними MarketSplash, США є найбільшим споживачем аніме серед всіх країн світу, за виключенням Японії: близько 56 мільйонів американців регулярно дивляться аніме. А світова вартість ринку аніме у 2022 році становила 25,62 мільярда доларів [1].

В Україні також стрімко зростає кількість шанувальників японської мультиплікації. За даними Google Trends, за 2023 рік пошукові запити зі словом аніме на території України за кількістю лише трохи відстали від запитів зі словом футбол, який є найбільш популярним видом спорту у нашій країні. Це свідчить про те, що аніме стає все більш популярним в Україні. Цей тренд, ймовірно, продовжуватиметься у 2024 році та в наступні роки [2].

Внаслідок стрімкого зростання популярності японської анімації у світі, сфера продажу аніме аксесуарів стала дуже прибутковою. Це сталося з декількох причин. По-перше, аксесуари – це відносно недорогий засіб для того, аби показати свою симпатію до аніме. Завдяки своїй відносно невеликій вартості, аніме аксесуари доступні для дуже багатьох людей. До того ж, аксесуари можуть бути дуже стильними атрибутами одягу, завдяки чому їх можуть купувати навіть люди, які не мають жодного відношення до аніме.

Проте, незважаючи на досить високу популярність аніме, спеціалізованих магазинів для продажу аксесуарів у нашій державі не так багато, а невелика кількість наявних мають певні недоліки.

Тому в цій кваліфікаційній описана проблема має бути вирішена шляхом розроблення спеціалізованого програмного забезпечення для онлайн продажу аніме аксесуарів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Можливі форми продажу аніме аксесуарів

Аніме – стиль анімації, який зародився в Японії. Його головними характеристиками є дуже барвиста графіка, яскравий дизайн персонажів та дуже насичені діями сюжети, що часто мають фантастичну або футуристичну направленість. Також аніме відрізняється від стилів анімації інших країн ще деякими рисами, наприклад:

- персонажі мають великі очі;
- персонажі часто жестикулюють та мають виразну міміку;
- дуже яскрава зовнішність персонажів [3].

Аніме має довгу і багату історію, яка бере свій початок ще у ХІХ столітті. Воно стало популярним у Японії в 1960-х роках минулого століття та з тої пори набуло популярності по всьому світу. Аніме орієнтується на всі типи аудиторій – на дитячу, на підліткову та на дорослу. Воно має величезну кількість шанувальників у Японії й в інших країнах світу, зокрема і в Україні.

На початку ХХІ століття аніме було популярне лише на своїй батьківщині, у Японії. Проте у 2010-х, і особливо зараз, у 2020-х роках нашого століття відбувається справжній бум популярності японської мультиплікації.

Це стало можливим завдяки:

- стрімкому розвитку технологій: дякуючи стрімкому розвитку сучасних технологій, наприклад, соціальних мереж та різноманітних стрімінгових сервісів, переглядати аніме стало набагато легше;
- великому успіху деяких аніме: деякі аніме-фільми та аніме-серіали такі як "Твоє ім'я", "Атака титанів" чи "Клинок, який знищує демонів", стали неймовірно популярними як у Японії, так і в усьому світі, через що інтерес до аніме індустрії збільшився.

Про дуже велику популярність аніме у світі свідчать такі факти і цифри:

- з 2020 по 2022 рік світовий попит на аніме зріс на 118%;

- частка аніме в загальному жанровому попиті зросла з 4,2% у січні 2020 року до 7,1% у 2022 році;
- укладання угод в індустрії аніме зросло, про що свідчить придбання Sony компанією Crunchyroll за 1,2 мільярда доларів;
- у 2021 році більше половини всіх глобальних передплатників Netflix користувалися аніме-контентом [1].

В Україні також стрімко зростає кількість шанувальників японської мультиплікації. В 2022 році, за даними декількох опитувань, аніме дивилося приблизно 40% українців віком від 18 до 35 років. Кількість спеціальних аніме-заходів в Україні також зростає з кожним роком. За даними Google Trends, у 2023 році пошукові запити зі словом ‘аніме’ на території України зросли на 25% порівняно з 2022 роком [2].

Навіть офіційна сторінка Міністерства Оборони України у мережі Twitter в одному з дописів використала цитату головного героя аніме серіалу “Ван Піс” Манкі Д. Люффі, а саме “Сила визначається не вашим розміром, а розміром вашого серця та мріями!” [4].

Внаслідок стрімкого зростання популярності японської анімації у світі, сфера продажу аніме аксесуарів стала дуже прибутковою. Це сталося з декількох причин. По-перше, аксесуари – це відносно недорогий засіб для того, аби показати свою симпатію до аніме. Завдяки своїй відносно невеликій вартості, аніме аксесуари доступні для дуже багатьох людей. До того ж, аксесуари можуть бути дуже стильними атрибутами одягу, завдяки чому їх можуть купувати навіть люди, які не мають жодного відношення до аніме.

Існують різні види аніме аксесуарів, наприклад:

- фігурки персонажів;
- чашки, ручки та інші невеликі сувеніри з персонажами чи логотипами аніме;
- одяг (кепки, футболки тощо).

Існує дві можливості для реалізації аніме атрибутики – у офлайн магазинах та онлайн, у всесвітній мережі Інтернет. Після пандемії COVID-19

та за умов повномасштабної війни в Україні більшою популярністю користуються саме онлайн продаж та купівля речей. Також у онлайн продажу є ще декілька переваг, зокрема:

- доступність та широкий асортимент товарів – покупці можуть купувати будь-які товари. Офлайн можна придбати лише ті товари, які є в наявності в певному магазині;
- зручність – покупці можуть купувати товари перебуваючи в будь-якому місці в будь-який час доби. Завдяки цьому можна дуже заощадити час, адже до офлайн магазинів, які до того ж працюють не цілодобово, потрібно йти чи їхати;
- ціна – оскільки продаючи товари в Інтернеті немає необхідності у сплаті аренди за приміщення, ціни у онлайн магазинах зазвичай є меншими, ніж в офлайн магазинах.

Офлайн магазини також мають певні переваги, як, наприклад, самостійний огляд товару у приміщенні магазину перед покупкою, проте переваг у онлайн продажу набагато більше.

Порівняльну таблицю онлайн та офлайн продажу товарів та подано у табл. 1.1.

Таблиця 1.1 – Порівняння онлайн і офлайн продажу товарів

	Онлайн	Офлайн
Доступність	Краще	Гірше
Зручність	Краще	Гірше
Асортимент	Більший	Менший
Ціни	Менші	Більші
Попередній огляд товару	Майже неможливо	Цілком можливо
Небезпека шахрайства	Більша	Менша

1.2 Програмні засоби продажу аніме аксесуарів

Одним з прикладів програмного засобу для продажу аніме аксесуарів є магазин ITEM SHOP [5], інтерфейс якого наведено на рис. 1.1.

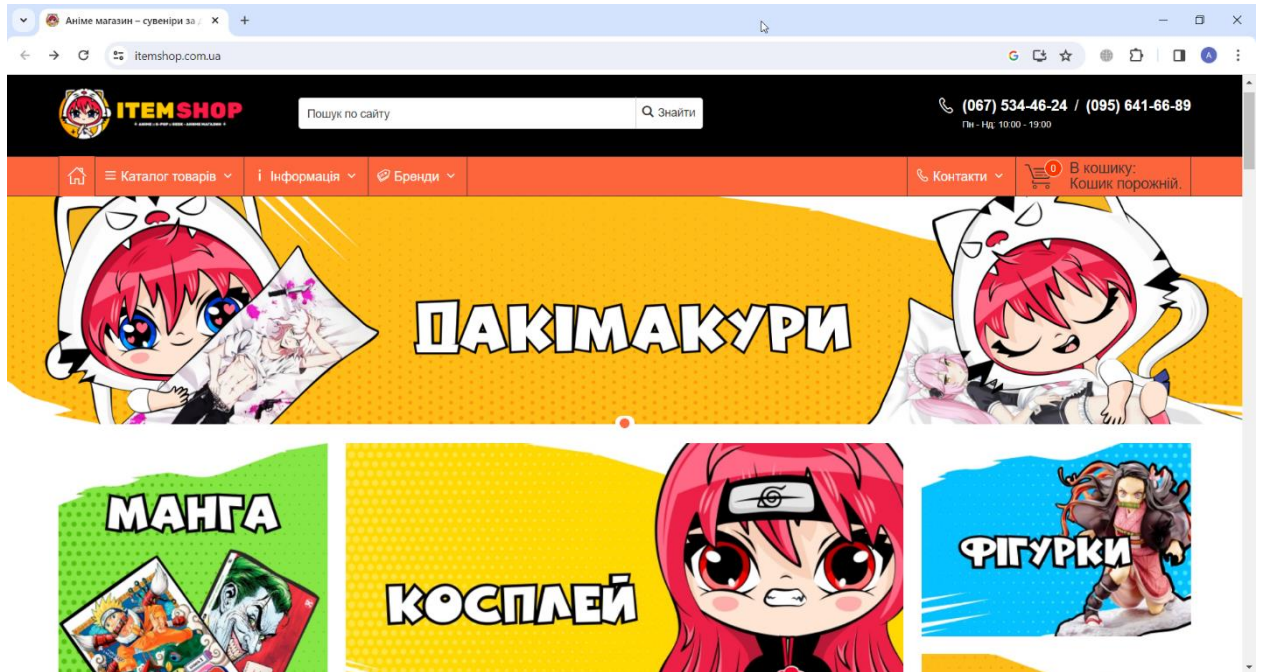


Рисунок 1.1 – Магазин ITEM SHOP [5]

ITEM SHOP дозволяє:

- дізнатися детальну інформацію про магазин;
- переглянути та замовити необхідні товари;
- переглянути наявні скидки та акції;
- ставити оцінки та залишати відгуки про певний товар;
- виконувати пошук необхідних товарів за найменуванням.

Проте у користувачів даного продукту відсутня можливість оцінювати коментарі інших людей, що є доволі важливою функцією, яка дозволяє користувачеві отримати уявлення про корисність або доцільність того чи іншого відгуку.

Іншим прикладом онлайн продажу аніме аксесуарів є магазин LollyBuy [6], інтерфейс якого наведено на рис. 1.2.

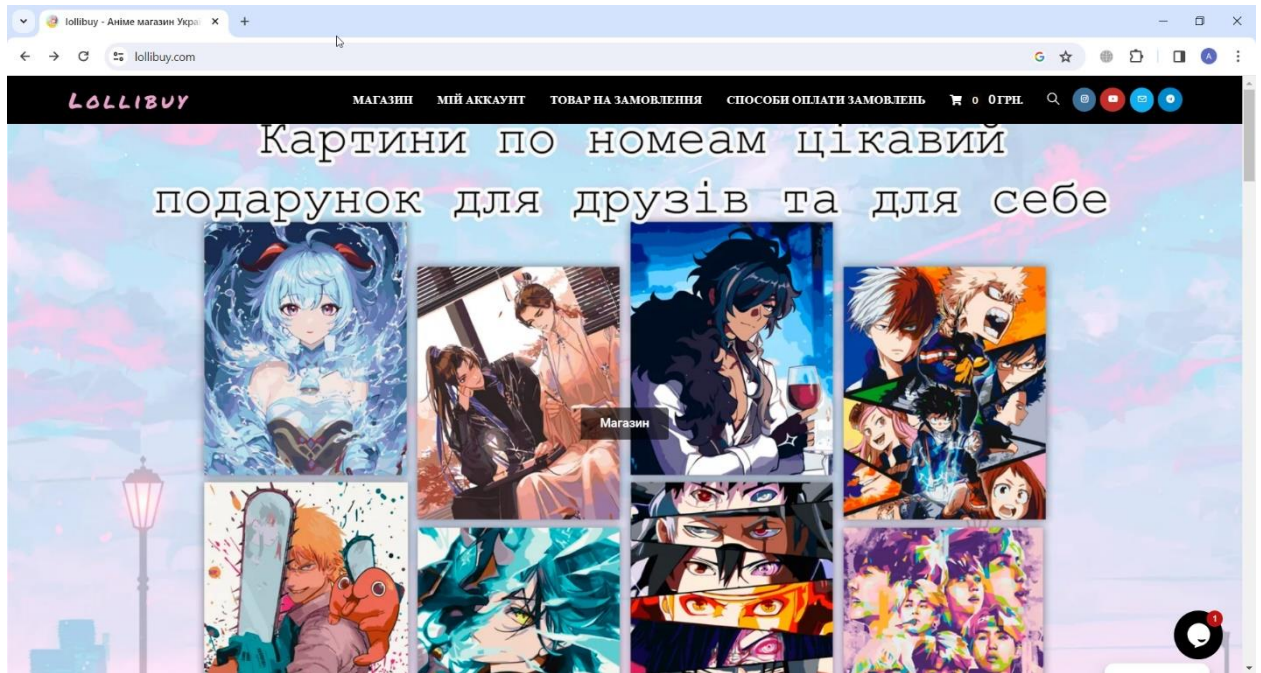


Рисунок 1.2 – Магазин LollyBuy [6]

LollyBuy дозволяє користувачу:

- дізнатися детальну інформацію про магазин;
- переглянути та замовити необхідні товари;
- переглянути наявні скидки та акції;
- ставити оцінки та залишати відгуки про певний товар;
- виконувати пошук необхідних товарів за найменуванням.

Але, як і попередній ITEM SHOP, LollyBuy не надає можливості оцінити коментарі від інших людей. Крім цього, інтерфейс зроблений не повністю українською мовою, що є певним недоліком.

Огляд існуючих програмних засобів для продажу аніме аксесуарів показав, що вони мають певні спільні переваги, але обидва аналоги не надають можливості оцінити відгуки інших користувачів і подивитися кількість оцінок того чи іншого відгука.

Порівняльну таблицю описаних вище програмних засобів та розробленого у роботі продукту подано у табл. 1.2.

Таблиця 1.2 – Порівняння існуючих рішень з розробленим

	ITEM SHOP	LollyBuy	Розроблений програмний засіб
Зручний для користувача інтерфейс	+	+	+
Можливість онлайн купівлі товарів	+	+	+
Пошук на сайті	+	+	+
Можливість залишати коментарі та відгуки про товар	+	+	+
Можливість оцінити коментар іншого користувача	-	-	+
Інтерфейс повністю українською мовою	+	-	+

1.3 Висновки за першим розділом

У даному розділі було виконано аналіз та огляд предметної області, а саме продаж аніме аксесуарів та було виявлено, що найбільш ефективно і зручно продаж і купівлю можна виконувати користуючись мережею Інтернет. Також було проведено аналіз аналогічних програмних засобів для продажу аніме аксесуарів, їх переваг та недоліків. В ході аналізу було виявлено, що спільним недоліком оглянутих засобів є відсутність можливості оцінити коментарі інших людей, що є доволі важливою функцією, яка дозволяє користувачеві отримати уявлення про корисність або доцільність того чи

іншого відгуку. Випускна кваліфікаційна робота магістра спрямована на розробку програмного засобу, у якому дана проблема буде відсутня.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Дослідження та порівняння вебсайтів та вебзастосунків

Задачу онлайн продажу та купівлі аніме аксесуарів найкраще вирішувати за допомогою засобів сфери інформаційних технологій та комп'ютерних мереж.

Інформаційна технологія (ІТ) – це комплекс методів, засобів і процесів, які використовуються для збору, зберігання, обробки та передачі інформації. Інформаційні технології ґрунтуються на використанні сучасних технічних засобів [7].

Комп'ютерна мережа – це система, що дозволяє комп'ютерам взаємодіяти один з одним. Вона може бути локальною, об'єднуючи комп'ютери які знаходяться в одному місці, або глобальною, тобто об'єднуючи комп'ютери з різних частин світу [8].

Одними з найкращих та найпопулярніших засобів для продажу товарів за допомогою ІТ та комп'ютерних мереж є вебсайти та вебзастосунки.

Вебсайт – це сукупність вебсторінок, які об'єднані одним доменним іменем. Вебсайти знаходяться у мережі Інтернет і є доступними завдяки ній. Вебсайти можуть належати окремим особам, компаніям, чи організаціям.

Для того, аби вебсайт працював, у нього мають бути такі компоненти:

- доменне ім'я – це унікальний ідентифікатор вебсайту. Воно складається з, як мінімум, двох рівнів: перший рівень вказує на тип вебсайту, він може бути однаковим у багатьох вебсайтів, а другий рівень є унікальним для конкретного сайту;

- вебсервер (або просто сервер) – комп'ютер, на якому зберігаються всі файли вебсайту;

- веббраузер (або просто браузер) – спеціальна програма, для перегляду вебсайтів на комп'ютерах або смартфонах користувачів вебсайтів. Веббраузер, взаємодіючи з вебсервером, відображає файли вебсайту на екранах користувачів. Прикладами веббраузерів є Chrome, Firefox і Safari.

Алгоритм роботи вебсайту зазвичай виглядає так:

- користувач пише адресу вебсайту в адресний рядок свого веббраузера;
- веббраузер відправляє запит на сервер за адресою, яку ввів у адресний рядок користувач. У ньому міститься інформація про тип запиту (GET або POST) та інші дані, що потрібні для отримання файлів вебсайту;
- сервер отримує запит, після чого відправляє браузеру файли сайту.

Зазвичай ці файли мають розширення .html, .css чи .js;

- веббраузер отримує файли вебсайту і після цього відображає їх на екрані користувача. Файли з розширенням .html визначають структуру вебсторінок, файли з розширенням .css визначають зовнішній вигляд вебсторінок, файли з розширенням .css додають інтерактивність до вебсторінок.

Схему алгоритму роботи вебсайту наведено на рис. 2.1.

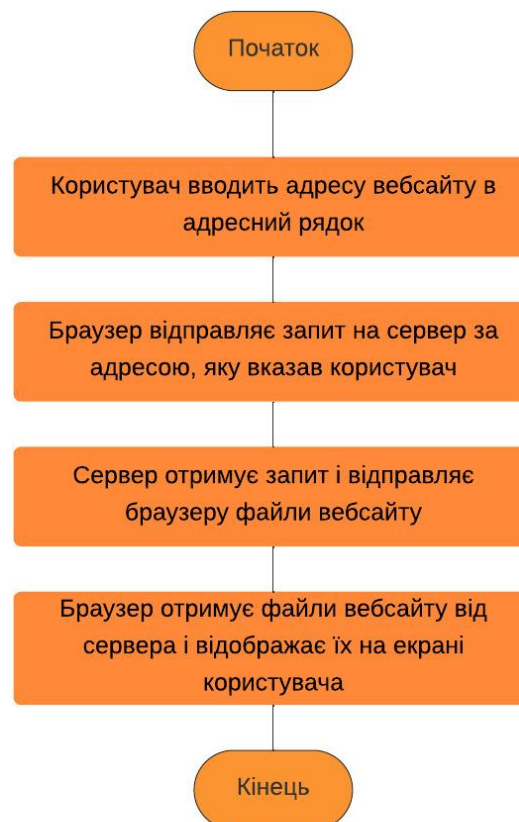


Рисунок 2.1 – Алгоритм роботи вебсайту

Вебзастосунок – це програмне забезпечення, що запускається за допомогою веббраузера і дозволяє користувачу взаємодіяти з вебсервером. Вебзастосунками є, наприклад, електронна пошта або соціальні мережі.

Алгоритм роботи вебзастосунку зазвичай виглядає наступним чином:

- користувач прописує адресу вебзастосунку в адресному рядку свого браузера;
- браузер відправляє запит на сервер за адресою, яку ввів у адресний рядок користувач. У ньому міститься інформація про тип запиту (GET або POST) та інші дані, що потрібні для отримання файлів вебсайту;
- сервер отримує запит і, у відповідь на нього, надсилає браузеру файли вебзастосунку;
- браузер відображає файли вебзастосунку, які були надіслані сервером, на моніторі користувача;
- користувач виконує певні дії у вебзастосунку. Реагуючи на ці дії, браузер знову надсилає запити на сервер;
- сервер отримує нові запити від веббраузера і у відповідь на них генерує нову інформацію, змінює вже існуючі дані чи видаляє певну інформацію з бази даних. Після цього сервер знову надсилає браузеру файли вебзастосунку;
- браузер отримує файли вебзастосунку, які були надіслані у якості відповіді сервера на дії користувача, та знову відображає їх на моніторі комп'ютера користувача;

Схему алгоритму роботи вебзастосунку наведено на рис. 2.2.

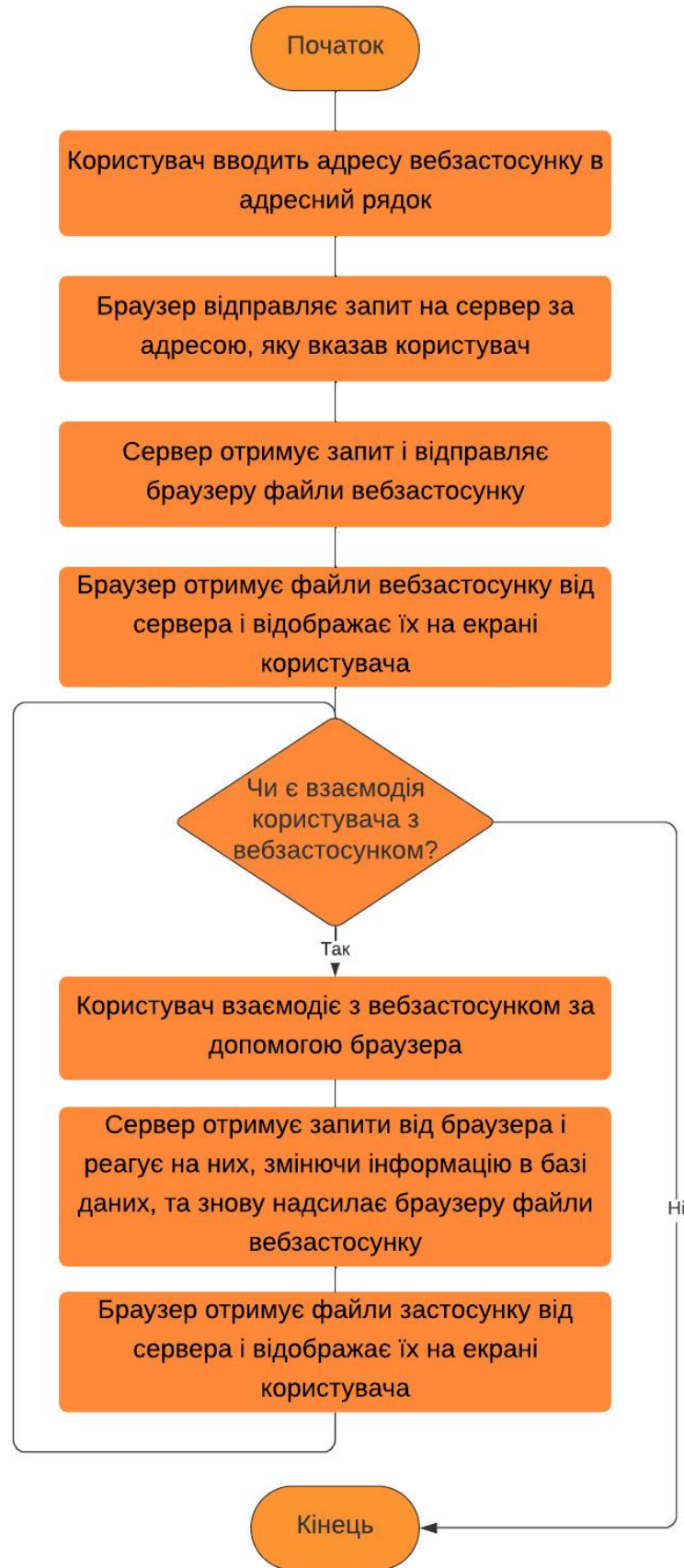


Рисунок 2.2 – Алгоритм роботи вебзастосунку

Для онлайн продажу товарів більш доцільним є використання вебзастосунків, а не вебсайтів. Цей висновок можна зробити з декількох причин:

- вебзастосунки забезпечують набагато більшу, ніж вебсайти, взаємодію з користувачем;
- вебзастосунки можуть обробляти більшу кількість запитів від користувачів, ніж вебсайти, що робить їх ідеальними для великих онлайн-магазинів.

Звичайно, вебзастосунки не є ідеальним рішенням для всіх онлайн-магазинів. Вебсайти можуть бути гарним варіантом для невеликих магазинів або магазинів, які продають обмежений асортимент товарів. Також вебсайти підходять для невеликих магазинів через те, що їх розробка та підтримка коштує набагато менше, ніж розробка вебзастосунків. Однак для великих магазинів, які хочуть надати своїм клієнтам найкращий можливий досвід і хочуть працювати з більшою кількістю покупців, вебзастосунки є пріоритетним вибором.

Порівняння вебсайтів та вебзастосунків наведено у таблиці 2.1.

Таблиця 2.1 – Порівняння вебсайтів та вебзастосунків

	Вебсайт	Вебзастосунок
Простота розробки	Краща	Гірше
Стійкість до мережових атак	Краща	Гірше
Вартість розробки	Менша	Більша
Вартість обслуговування	Менша	Більша
Зручність для користувача	Менша	Більша
Взаємодія з користувачами	Гірше	Краще

Продовження таблиці 2.1

1	2	3
Оновлення контенту	Гірше	Краще

2.2 Дослідження головних принципів зручності користування (usability) вебзастосунків

Зручність користування (usability) вебзастосунку – це комплексний термін, що вказує на зручність взаємодії із ресурсом. До цього поняття входить багато аспектів, таких як: зручність інтерфейсу, якість контенту, швидкість завантаження застосунку, зручна навігація, професійний дизайн, функціональна стабільність застосунку тощо.

Згідно з [9]-[10] принципи зручності користування вебзастосунком можна розділити на три головні категорії:

- навчальність або здатність до швидкого вивчення (learnability) – це легкість, з якою нові користувачі можуть оволодіти функціональністю вебзастосунку та використовувати його без тривалого навчання чи додаткової інформації;
- гнучкість (flexibility) – здатність вебзастосунку адаптуватися до різних потреб користувачів і різних умов використання;
- надійність (robustness) – рівень підтримки, який надається користувачеві для визначення цілей використання вебзастосунку та їх успішного досягнення.

В свою чергу, описані вище три головні категорії можна розділити на більш конкретні принципи. Детально розглянемо категорію навчальність. Її можна розділити на 5 принципів, а саме:

- передбачуваність (predictability);
- синтезованість (synthesizability);
- звичність (familiarity);
- узагальненість (generalizability);

– послідовність (consistency).

Детально розглянемо кожен з вищеназваних аспектів навчальності.

Передбачуваність – здатність користувача легко передбачити результати своїх взаємодій з вебзастосунком (чи може користувач «сказати», що станеться на основі того, що бачив та робив при взаємодії з іншими застосунками у минулому?). Передбачуваність стосується здатності користувача визначати вплив певних операцій на систему. Інша форма передбачуваності пов'язана зі здатністю користувача знати, які саме операції можна виконати. Обидва типи передбачуваності важливі для створення зручних інтерактивних систем [10].

Синтезованість – це здатність користувача оцінювати вплив минулих дій на поточний стан інтерактивної системи (чи може користувач сказати, чому він тут, на основі того, що користувач зробив у минулому). Також синтезованість стосується того, наскільки ефективно користувач може створювати образ та модель системи на основі свого досвіду взаємодії з нею. При цьому важливо, щоб будь-які зміни внутрішнього стану системи були помічені користувачем. Інформативний звіт про такі зміни може бути надано негайно, в цьому випадку зміни стають видимими без додаткових дій від користувача, або з можливістю легко виявити їх у майбутньому [10].

Звичність. Користувачі системи зазвичай мають багатий досвід взаємодії з вебзастосунками. Цей досвід набувається в реальному світі і взаємодії з іншими комп'ютерними системами. Звичність – це ступінь схожості розроблюваної системи з наявними знаннями та досвідом користувача. Це важливо для того, щоб зробити вебзастосунок легким для вивчення та використання. Використання певних аналогій або метафор, які вже знайомі користувачам може допомогти в реалізації принципу звичності. Наприклад, кнопка з зображенням кошика може вказувати на можливість додати товар до кошика для покупок [10].

Узагальненість – це здатність користувачів застосовувати свої знання та досвід взаємодії з однією частиною вебзастосунку на інші частини

застосунку чи на функціональні елементи, які вони раніше не зустрічали. Це важлива частина навчальності, оскільки впливає на те, наскільки легко та ефективно користувачі зможуть адаптуватися до нових частин вебзастосунку чи до його функціональних можливостей [10].

Послідовність є важливим принципом навчальності. Це стабільність та однаковість у представленні інформації та функціональності на різних частинах застосунку. Цей принцип покликаний забезпечити однаковий стиль, формат та взаємодію для подібних завдань чи елементів вебзастосунку. Існує багато способів досягнення послідовності, наприклад використання узгодженої термінології, шаблонів проєктування та механізмів зворотного зв'язку. Іноді звичність та узагальненість також можна розглядати в рамках послідовності [10].

Короткий виклад принципів навчальності зображено у таблиці 2.2.

Таблиця 2.2 – Основні принципи навчальності

Принцип	Визначення принципу
1	2
Звичність	Здатність функціоналу та інтерфейсу вебзастосунку бути звичними та легкими для розуміння для користувачів завдяки їх минулому досвіду
Передбачуваність	Здатність вебзастосунку допомагати користувачам легко передбачити результати своїх запитів до вебзастосунку
Синтезованість	Здатність вебзастосунку допомагати користувачам оцінювати вплив своїх минулих дій та операцій на поточний стан вебзастосунку

Продовження таблиці 2.2

1	2
Послідовність	Здатність вебзастосунку зберігати єдину концепцію та сталість дизайну
Узагальненість	Здатність вебзастосунку допомагати користувачам застосовувати свій досвід роботи з однією частиною чи функцією вебзастосунку під час взаємодії з іншими його частинами, тими, яких вони раніше не зустрічали

Тепер детально розглянемо категорію гнучкість. Можна виділити 5 принципів гнучкості, а саме:

- ініціатива діалогу (dialog initiative);
- багатопотоковість (multi-threading);
- передавання завдань (task migratability);
- взаємозамінність (substitutivity);
- налаштованість (customizability).

Ініціатива діалогу – вказує на те, яка сторона, користувач чи система, бере на себе ініціативу під час взаємодії. Існує два типу діалогів: діалог, ініційований системою і діалог, ініційований користувачем. Під час діалогу, що ініційований системою, система ініціює всі етапи взаємодії, і користувач реагує на запити чи дії, визначені системою. Наприклад, модальне вікно може забороняти користувачу взаємодіяти з іншими частинами системи, поки воно відкрите. У діалозі, ініційованому користувачем, він має повний контроль і може ініціювати будь-які дії взаємодії з системою у будь-який час. Система обмежується реагуванням на користувацькі запити. Це може забезпечити більшу гнучкість, але може призвести до втрати структурованості, якщо користувач втрачає контроль над завданнями. Загалом, краще мати діалоги, ініційовані користувачем, оскільки це дає користувачеві більше гнучкості.

Проте в деяких ситуаціях краще використати діалог, ініційований системою, наприклад, коли користувач потенційно може пошкодити систему [10].

Багатопотоковість – це спосіб, за допомогою якого система виконує кілька завдань одночасно. Це можна зробити двома способами: паралельна багатопотокова обробка та багатопотокова обробка з чергуванням. Паралельна багатопотоковість дозволяє системі обробляти кілька завдань повністю одночасно. Багатопотокова обробка з чергуванням дозволяє системі обробляти кілька завдань по одному, але це робиться досить швидко, що здається одночасним [10].

Передавання завдань – можливість передачі контролю за виконанням завдань між системою та користувачем. Це важливо у випадках, коли користувач хоче взяти на себе завдання, яке вже виконує система, або навпаки. Наприклад, при заповненні форми користувач може перевіряти правопис самостійно, але при виявленні проблем, таких як невірно написане слово чи неправильна пунктуація в реченні, завдання перевірки тексту може передаватися комп'ютеру [10].

Взаємозамінність – можливість заміни еквівалентних значень чи елементів інтерфейсу одне на одне без втрати функціональності та придатності до використання. Наприклад, можливість вводити дані в різних одиницях вимірювання. Це розширює гнучкість та зручність взаємодії користувача з вебзастосунком. Взаємозамінність може також застосовуватися до представлення інформації, де виведення може мати кілька форм (наприклад, текст, графік, аудіо), і користувач може обирати той спосіб, який йому більше підходить для конкретного завдання [10].

Налаштованість – здатність користувачів чи системи змінювати інтерфейс вебзастосунку відповідно до своїх власних потреб і вподобань. Наприклад, користувач має мати можливість переміщення, зміни розміру чи приховання певних елементів. Або система може враховувати попередні дії користувача та пропонувати персоналізовані рекомендації чи налаштування.

Налаштованість важлива, оскільки вона дозволяє вебзастосункам стати більш гнучкими та адаптованими до деяких унікальних потреб користувачів [10].

У таблиці 2.3 зображено короткий виклад принципів гнучкості.

Таблиця 2.3 – Основні принципи гнучкості

Принцип	Визначення принципу
Налаштованість	Здатність вебзастосунку дозволяти користувачам налаштовувати та змінювати під себе деякі елементи інтерфейсу
Взаємозамінність	Здатність вебзастосунку замінювати одні опції на інші, еквівалентні ним, без втрати функціональності
Передавання завдань	Здатність передавати контроль над виконанням завдання від користувача до системи або від системи до користувача в потрібний час
Багатопотоковість	Здатність вебзастосунку одночасно оброблювати та виконувати декілька завдань користувача
Ініціатива діалогу	Здатність вебзастосунку до взаємодії з користувачами за допомогою двох типів діалогів – ініційованих користувачем та ініційованих системою

Детально розглянемо категорію зручності користування вебзастосунком під назвою надійність. Її можна розділити на 4 принципи, а саме:

- наглядність (observability);
- відновлюваність (recoverability);
- швидкість реагування (responsiveness);

- відповідність завданням (task conformance).

Наглядність – це можливість користувача оцінювати внутрішній стан системи через його сприйнятливий представлення. Цей принцип дозволяє користувачеві легко здійснювати контроль та моніторинг за допомогою відображення інформації на екрані. Основні аспекти наглядності включають можливість перегляду, значення за замовчуванням, доступність і стійкість [10].

Розглянемо кожен з цих аспектів:

- можливість перегляду надає користувачеві можливість вивчати різні частини вебзастосунку чи його функціонал через інтерфейс. Наприклад, структуровані меню чи секції можуть полегшити перехід між сторінками [10];
- визначення значень за замовчуванням для деяких параметрів чи опцій може полегшити користувачеві процес взаємодії. Наприклад, автоматичне встановлення значень, які найчастіше використовуються, може спростити введення даних [10];
- доступність – забезпечує можливість навігації між різними сторінками, функціоналом чи розділами вебсайту. Це дозволяє користувачам швидко переходити від одного розділу до іншого та знаходити потрібну інформацію [10];
- стійкість – зберігання деякої інформації чи стану протягом тривалого часу. Наприклад, візуальні позначення (наприклад, позначки або індикатори) можуть залишатися на екрані, щоб нагадувати користувачеві про певні події чи стани [10].

Відновлюваність – можливість користувача виправити помилки або повернутися до бажаного стану після неправильної взаємодії чи дії. Це важливий аспект, оскільки користувачі, як правило, роблять помилки під час взаємодії з вебзастосунком, і їм потрібно мати можливість легко виправляти наслідки помилок. Існує два типи відновлення: пряме та зворотне. Пряме відновлення дозволяє вирішити проблему та продовжити роботу, не втрачаючи попередні результати. Наприклад, якщо під час оформлення

замовлення виникає помилка в полі адреси, користувач може відредагувати адресу та продовжити процес. Зворотне відновлення передбачає можливість відмінити або відновлювати попередні дії. Наприклад, в інтерфейсі редактора тексту кнопка "Скасувати" може дозволити скасування останньої зміни. Важливим аспектом відновлюваності є принцип пропорційних зусиль. Цей принцип стверджує, що складність відміни або виправлення дій повинна відповідати складності їхнього виконання. Це сприяє уникненню ненавмисних помилок та непередбачених наслідків. Наприклад, видалити файл має бути важче, ніж перейменувати його, оскільки видалення файлу є більш руйнівною дією [10].

Швидкість реагування – швидкість реакції системи на дії користувача. Бажаний час реакції зазвичай є якомога коротшим, в ідеалі – миттєвим. Навіть якщо обчислювальні процеси не дозволяють миттєву відповідь, важливо надати користувачеві чітке відзначення того, що його запит отримано, і система працює над обробкою. Також важливо забезпечити стабільність часу швидкості реакції. Швидкість реакції має бути однаковою, незалежно від навантаження або інших факторів. Надто великі коливання в швидкості реакції можуть стати причиною негативного впливу на враження користувача від взаємодії з вебзастосунком [10].

Відповідність завдання – здатність системи підтримувати та спрощувати виконання завдань користувачів, пов'язаних з їхніми цілями та потребами. Вебзастосунок повинен надавати усі необхідні функції та інструменти, щоб користувачі могли легко та ефективно виконувати свої завдання. Це може включати в себе швидкий доступ до інформації, зручний пошук, простоту оформлення замовлень тощо. Також інтерфейс вебзастосунку повинен відображати завдання користувача в зрозумілий та природний спосіб. Наприклад, якщо вебзастосунок використовується для продажу товарів, інтерфейс повинен логічно організовувати вибір товарів, додавання їх до кошика та оформлення покупки [10].

У таблиці 2.4 зображено короткий виклад принципів надійності.

Таблиця 2.4 – Основні принципи надійності

Принцип	Визначення принципу
Відповідність завданням	Ступінь того, наскільки добре можливості вебзастосунку відповідні завданням, які користувачі прагнуть виконати з його допомогою і ступінь того, наскільки можливості вебзастосунку є зрозумілими для користувачів
Швидкість реагування	Здатність вебзастосунку швидко реагувати та відповідати на запити користувачів
Відновлюваність	Здатність вебзастосунку дозволяти користувачам виправляти зроблені ними помилки при роботі з вебзастосунком
Наглядність	Здатність користувача оцінити внутрішній стан системи за допомогою його зрозумілого відображення на екрані

Завдяки слідуванню головним принципам зручності користування вебзастосунку можна досягти багатьох речей, а саме:

- великої кількості відвідувань – слідування принципам зручності користування дозволяє зробити вебзастосунок зручним та зрозумілим для користувачів з різними рівнями навичок та для користувачів різного віку. А ті користувачі, що легко розуміють принципи роботи з вебзастосунком та його функціонал, будуть користуватися вебзастосунком частіше, внаслідок чого,

якщо вебзастосунок призначений для продажу товарів, кількість продажів буде значною;

- більшої ефективності використання вебзастосунку – слідування вищеназаним принципам дуже полегшить виконання користувачами своїх завдань та збільшить ефективність використання вебзастосунку. Це, в свою чергу, призведе до того, що клієнти вебзастосунку під час користування будуть отримувати більше задоволення, внаслідок чого будуть частішати повторні замовлення від клієнтів;

- зменшення помилок – гарний дизайн і зрозуміла структура вебзастосунку можуть дуже сильно зменшити кількість помилок, які можуть виникнути під час користування вебзастосунком. Чим менша кількість помилок та непорозумінь – тим вища ступінь задоволення від користування вебзастосунком та тим більша довіра до власників програмного продукту;

- конкурентоспроможності – клієнти надають перевагу тим вебзастосункам, які під час використання викликають найприємніші емоції та пропонують найзручніший досвід;

- SEO оптимізації – майже всі пошукові системи зазвичай враховують в якості одного з ранжувальних факторів зручність взаємодії з вебзастосунком. Якщо користувача під час роботи з вебзастосунком все влаштовує, то він навряд чи повернеться до сторінки видачі пошукових систем, а навпаки, виконає якусь конверсійну дію.

2.3 Висновки за другим розділом

У результаті виконання другого розділу магістерської роботи було зроблено аналіз засобів сфери інформаційних технологій та комп'ютерних мереж для продажу та купівлі товарів онлайн. У якості основних засобів було виділено два, а саме вебсайти та вебзастосунки. Було виконано детальне дослідження вебсайтів та вебзастосунків, зокрема виділено сутність цих понять, виділено основні переваги та недоліки кожного з них, представлено та

проаналізовано алгоритми їх роботи. У підсумку дослідження в якості основної моделі рішення задачі продажу та купівлі товарів, зокрема аніме аксесуарів, онлайн було обрано саме вебзастосунок. Також у цьому розділі було виконано детальний огляд зручності використання вебзастосунку. Було виділено та досліджено основні пункти та підпункти зручності використання вебзастосунку, описано головні переваги їх використання під час розробки програмного продукту.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

3.1 Функціональні вимоги до програмного забезпечення

Програмний продукт, який було розроблено під час роботи, ділить користувачів на три категорії: неавторизовані користувачі, авторизовані користувачі та адміністратори.

Функціональні вимоги неавторизованого користувача:

- перегляд списку наявних товарів та кожного окремого товару;
- перегляд списку коментарів до певного товару;
- пошук товару за введеним словом;
- реєстрація та авторизація.

Діаграма прецедентів неавторизованого користувача наведена на рис.

3.1.

Функціональні вимоги авторизованого користувача:

- перегляд списку наявних товарів та кожного окремого товару;
- замовлення товару або групи товарів одним замовленням;
- додавання коментарів до певного товару;
- перегляд списку коментарів до певного товару;
- оцінювання коментарів інших користувачів;
- перегляд списку своїх замовлень та їх статусу;
- пошук товару за введеним словом.

Діаграма прецедентів авторизованого користувача наведена на рис. 3.2.

Функціональні вимоги адміністратора:

- авторизація;
- додавання категорій товарів та власне товарів;
- редагування інформації про вже наявні категорії та товари;
- видалення або архівація категорій товарів чи товарів;
- зміна статусу товару;
- перегляд коментарів користувачів та їх видалення.

Діаграма прецедентів адміністратора наведена на рис. 3.3.

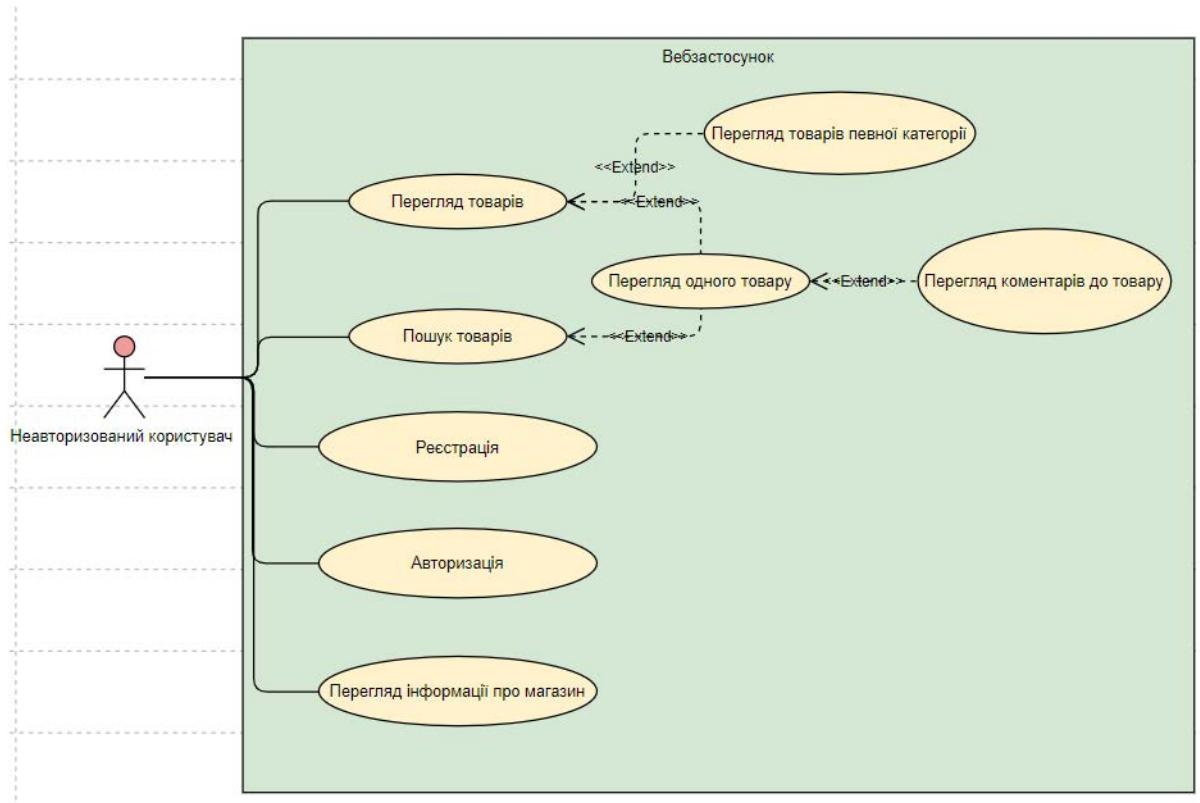


Рисунок 3.1 – Діаграма прецедентів неавторизованого користувача

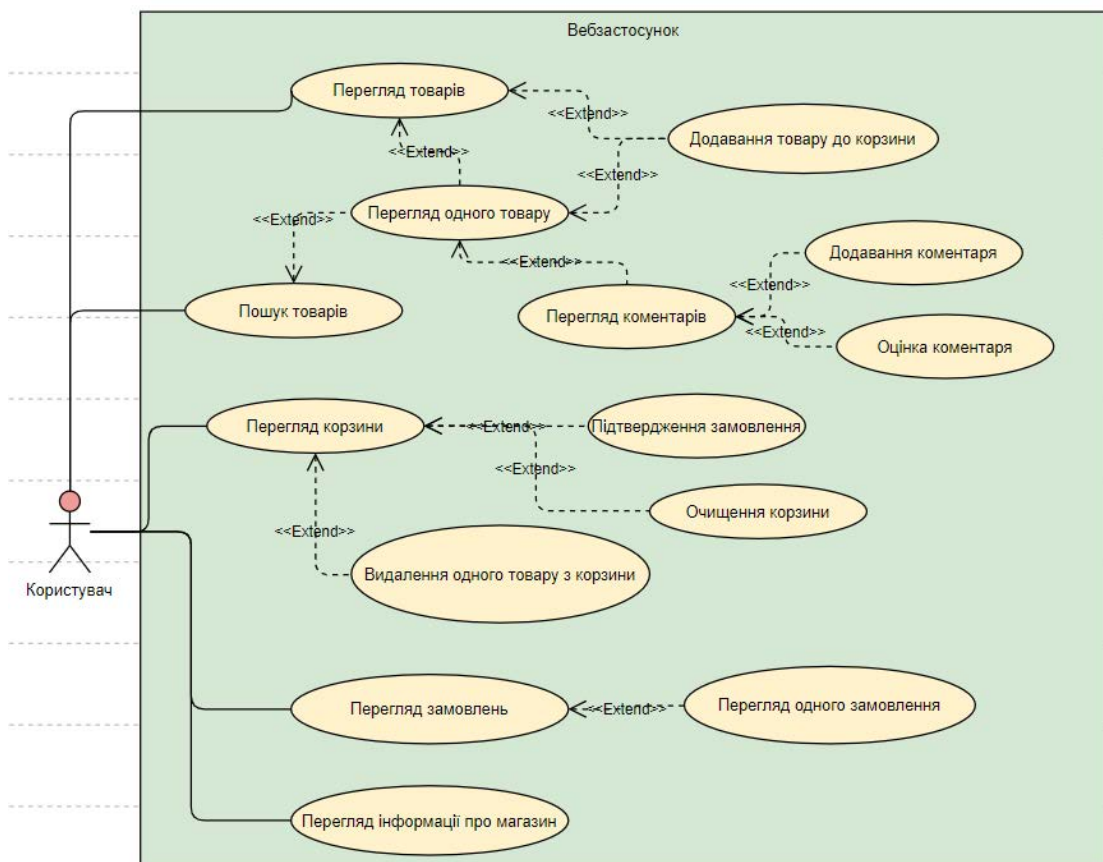


Рисунок 3.2 – Діаграма прецедентів авторизованого користувача

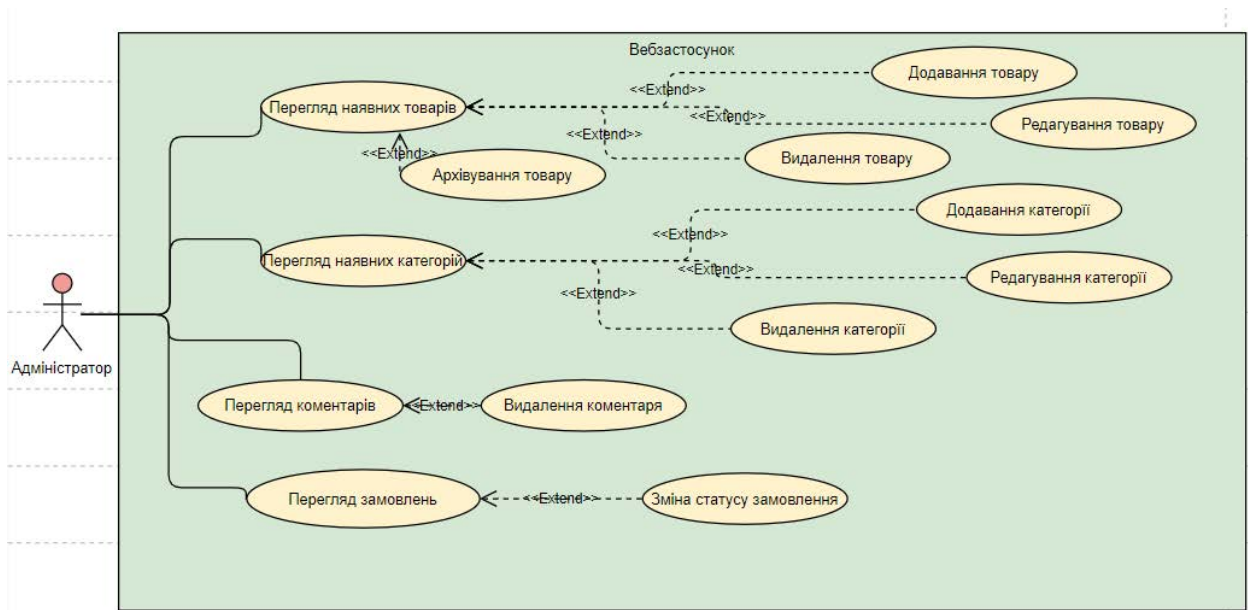


Рисунок 3.3 – Діаграма прецедентів адміністратора

3.2 Вибір засобів розробки

Для розробки клієнтської частини вебзастосунку було обрано три основні технології: мову розмітки документів HTML, мову стилю сторінок CSS й мову програмування JavaScript.

JavaScript – це об'єктно-орієнтована мова програмування, що була розроблена Бренданом Айхом у 1995 році. Вона стала стандартом ECMA у 1997 році. JavaScript використовується для реалізації інтерактивності вебсторінок. Вона дозволяє розробникам створювати вебсторінки, які можуть реагувати на дії користувачів, наприклад, на натискання кнопок або інших елементів сторінки, переміщення миші чи прокручування сторінки. JS може використовуватися для створення різних типів вебзастосунків. Сьогодні JS є однією з найпопулярніших мов програмування у всьому світі [11].

CSS – це мова опису зовнішнього виду елементів вебсторінки, яка використовується для більш красивого оформлення дизайну вебсторінок. Вона дозволяє розробникам робити індивідуальний вигляд і стиль вебсторінки. CSS може використовуватися для зміни шрифту, кольору, розміру, розташування елементів вебсторінки [12].

HTML – це мова, яка використовується для розмітки вебсторінок. Вона визначає структуру контенту вебсторінки. HTML складається з елементів, які називаються тегами. Саме теги повідомляють браузеру, як потрібно відображати контент вебсторінки [13].

Також під час розробки клієнтської частини вебзастосунку активно використовувався клієнтський фреймворк Bootstrap.

Bootstrap – це набір інструментів для веброзробки, який дозволяє створювати гарні функціональні вебсторінки швидше та простіше. Це безкоштовний інтерфейсний фреймворк з відкритим вихідним кодом, розміщеним на GitHub. Bootstrap включає в себе набори готових елементів (кнопки, форми, таблиці, навігаційні меню та інші компоненти), які можна використовувати для створення різних типів вебсторінок. Також Bootstrap надає набори вже готових стилів [14].

Переваги Bootstrap:

- набори готових елементів, наявність яких заощаджує розробникам багато часу;
- за допомогою цього фреймворку легко створювати адаптивні дизайни;
- велика і активна спільнота розробників;
- Bootstrap є дуже простим та зрозумілим у використанні, що полегшує вивчення і використання фреймворка як для початківців, так і для досвідчених розробників;
- набори плагінів JS, які можна використовувати для додавання додаткових функцій до вебсторінок.

Недоліки Bootstrap:

- через те, що фреймворк для розробки надає набори вже готових елементів, вебсторінки, що створені за допомогою Bootstrap, можуть бути схожими між собою;
- через те, що Bootstrap є доволі великим фреймворком, може відбутися збільшення розміру коду вебзастосунку. Це, в свою чергу, може

негативно вплинути на швидкість завантаження та швидкість роботи вебзастосунку і на погане розуміння коду іншими розробниками, що не знайомі з Bootstrap.

У якості мови для розробки серверної частини вебзастосунку було обрано мову програмування PHP.

PHP – це мова програмування з відкритим кодом, яка використовується для створення вебсторінок, що можуть відповідати на дії користувачів та отримувати доступ до баз даних. За допомогою цієї мови програмування можна створювати складні та інноваційні вебсайти та вебзастосунки. PHP легко інтегрується з усіма основними серверами та операційними системами, що робить її дуже популярною мовою серед розробників [15].

Можливості PHP:

- взаємодія з файлами на сервері;
- взаємодія з файлами cookie;
- динамічна генерація вмісту вебсторінок;
- отримання даних з форм, заповнених користувачами;
- ряд функцій безпеки, які допомагають захистити вебзастосунки від злому(шифрування і фільтрація даних, аутентифікація користувачів тощо)
- доступ до інформації, що зберігається в базах даних [15].

Порівняння PHP та Python у якості мов для серверної частини вебзастосунку наведено у таблиці 3.1.

Таблиця 3.1 – Порівняння PHP та Python

	Мова програмування	
	PHP	Python
1	2	3
Спільнота	Велика та активна	Також є великою та активною, але трохи меншою

Продовження таблиці 3.1

1	2	3
Легкість встановлення	Чудово встановлюється у Windows, Linux, Mac	Інколи можуть виникнути певні проблеми
Фреймворки	Наявні та зручні, наприклад Laravel або Symfony	Також є в наявності зручні фремворки, наприклад Django
Розширення та бібліотеки	Велика кількість розширень і бібліотек	Широкий вибір бібліотек, але менше розширень
Швидкість та продуктивність	Минулі версії дещо повільні, але PHP 7 має надзвичайну швидкість	Теж дуже висока швидкість, але трохи менша останню версію PHP
Безпека	PHP має проблеми з безпекою	Вважається безпечнішим, особливо з використанням спеціальних фреймворків, наприклад, Django.
Взаємодія з базами даних	Чудова	Гірша, бо для деяких баз даних потрібен спеціальний драйвер
Синтаксис	Дещо складний	Чіткий і зрозумілий
Популярність	Дуже велика	Дещо менша
Сфера використання	Вузька сфера застосування, переважно веброзробка	Дуже велика сфера застосування

У якості системи керування базами даних (СКБД) було обрано MySQL. Ця СКБД дуже часто використовується разом з мовою PHP.

MySQL – це дуже популярна реляційна СКБД, яка використовується для зберігання даних вебсайтів та вебзастосунків. Поширюється та підтримується компанією Oracle Corporation. MySQL можна використовувати безкоштовно і вона сумісна з усіма основними операційними системами.

Переваги MySQL:

- надійність;
- швидкість;
- безкоштовність використання;
- використовується і для малих і для дуже великих програм;
- простота використання;
- підтримка двох і більше одночасних запитів;
- функціональність;
- можливість масштабування.

3.3 Проєктування бази даних

У якості основних сутностей бази даних було виділено наступні: User, Anime, Category, Stuff, Comment, Rating, Order, Order List.

Сутність User визначає адміністратора або користувача, який взаємодіє з програмним застосунком.

Сутність Anime визначає певне аніме.

Сутність Category визначає певну категорію товару.

Сутність Stuff визначає товар, який продає магазин.

Сутність Comment визначає коментар, який може залишити користувач про певний товар.

Сутність Rating визначає рейтинг певного коментаря.

Сутність Order визначає замовлення товару або групи товарів користувачем вебзастосунку.

Сутність Order List визначає склад певного замовлення певного користувача.

Таблиця Users містить такі поля:

- id – номер користувача;
- check_admin – відомості чи є користувач адміністратором;
- date – дата реєстрації користувача;
- email – пошта користувача;
- login – логін користувача;
- password – пароль користувача.

Таблиця Anime містить такі поля:

- id – номер аніме;
- name – назва аніме;
- description – опис аніме.

Таблиця Categories містить такі поля:

- id – номер категорії;
- name – назва категорії;
- description – опис категорії.

Таблиця Stuff містить такі поля:

- id – номер товару;
- name – назва товару;
- image – шлях до теки, у якій знаходиться зображення товару;
- date – дата додавання товару;
- price – ціна товару;
- description – опис товару;
- anime_name – назва аніме, до якого відноситься даний товар;
- id_category – номер категорії, до якої відноситься даний товар;
- archive – інформація про те, архівовано товар чи ні.

Таблиця Comments містить такі поля:

- id – номер коментаря;
- date – дата додавання коментаря;

- text – текст коментаря;
- id_stuff – номер товару, до якого відноситься даний коментар;
- id_user – номер користувача, який залишив даний коментар.

Таблиця Rating містить такі поля:

- id – номер оцінки коментаря;
- rating – значення оцінки коментаря;
- id_comment – номер коментаря, до якого відноситься дана оцінка;
- id_user – номер користувача, який дав дану оцінку коментаря.

Таблиця Orders містить такі поля:

- id – номер замовлення;
 - date – дата замовлення;
 - total_cost – загальна вартість замовлення;
 - status – статус замовлення;
 - check_temp – інформація про те, чи підтвердив користувач замовлення;
- id_user – номер користувача, що зробив певне замовлення.

Таблиця Order_lists містить такі поля:

- id – номер товару з певного замовлення;
- id_stuff – номер товару, який є у певному замовленні;
- price – вартість товару, який є у певному замовленні;
- id_order – номер замовлення, до якого відноситься товар з замовлення.

Схему бази даних (БД) наведено на рис 3.4.

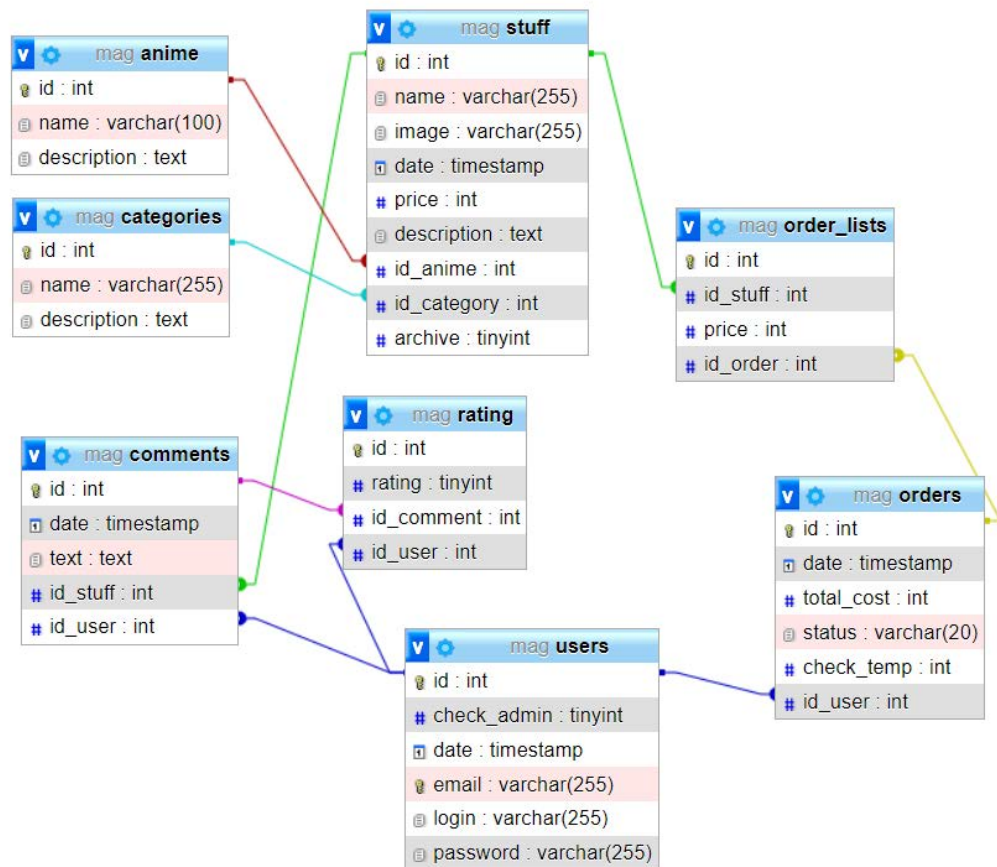


Рисунок 3.4 – Схема БД

Таблиці Anime та Stuff поєднані зв'язком один-до-багатьох, бо до одного аніме може відноситися багато товарів, але певний товар може бути пов'язаний лише з одним аніме.

Таблиці Categories та Stuff поєднані зв'язком один-до-багатьох, бо до однієї категорії може відноситися багато товарів, але певний товар може бути лише однієї категорії.

Таблиця Stuff та Users поєднані зв'язком багато-до-багатьох, цей зв'язок уособлює таблиця Comments.

Таблиці Stuff та Comments поєднані зв'язком один-до-багатьох, бо до одного товару можна залишити багато коментарів, але певний коментар може бути пов'язаний лише з одним товаром.

Таблиці Comments та Rating поєднані зв'язком один-до-багатьох, бо один коментар можуть оцінити багато разів, але певна оцінка буде стосуватися лише одного коментаря.

Таблиці Users та Comments поєднані зв'язком один-до-багатьох, бо один користувач може залишити багато коментарів, але певний коментар може бути залишеним лише одним користувачем.

Таблиці Users та Rating поєднані зв'язком один-до-багатьох, бо один користувач може залишити багато оцінок коментарів, але певна оцінка може бути залишена лише одним користувачем.

Таблиці Users та Orders поєднані зв'язком один-до-багатьох, бо один користувач може зробити багато замовлень, але певне замовлення належати лише одному користувачеві.

Таблиця Orders та Stuff поєднані зв'язком багато-до-багатьох, цей зв'язок уособлює таблиця Order_Lists.

Таблиці Orders та Order_Lists поєднані зв'язком один-до-багатьох, бо в одному складі замовлення може бути багато товарів, але певний товар з певного замовлення може бути лише в складі одного замовлення.

Таблиці Stuff та Order_Lists поєднані зв'язком один-до-багатьох, бо один товар може бути у складі багатьох замовлень.

3.4 Висновки за третім розділом

Під час виконання третього розділу кваліфікаційної роботи магістра було виділено основні функціональні вимоги до програмного засобу і виконано моделювання роботи з програмою користувача та адміністратора за допомогою діаграм прецедентів (Use Case Diagram). Також було зроблено огляд засобів для розробки вебзастосунку та виділено їх основні переваги та недоліки. Для розробки клієнтської частини вебзастосунку було обрано мову HTML, мову CSS, мову JS та фреймворк Bootstrap. Для серверної частини – мову програмування PHP і СКБД MySQL. Також у цьому розділі було виконано проєктування бази даних та розроблено її схему.

4 РОЗРОБКА ПРОГРАМИ

4.1 Алгоритм функціонування вебзастосунку

Алгоритм функціонування вебзастосунку має два режими – режим для звичайного користувача і адміністратора.

Після введення адреси вебзастосунку в пошуковий рядок відкривається головна сторінка застосунку на якій представлені останні пропозиції магазину. Користувач може залишитися на цій сторінці та продовжити перегляд товарів або обрати інші режими роботи. Користувач може виконати:

- перегляд одного товару;
- перегляд коментарів до товару;
- пошук необхідних товарів;
- перегляд інформацію про магазин;
- авторизацію або реєстрацію.

Після обрання режиму авторизації користувач повинен ввести свій логін та пароль. Якщо авторизація не була успішною, користувач має змогу зареєструватися. Без успішної авторизації користувачеві все ще доступні вище названі режими, але до інших можливостей вебзастосунку користувач доступу не отримає.

Після успішної авторизації чи реєстрації користувачеві стають доступними наступні активності:

- додати товар до кошика;
- додати коментар до певного товару;
- оцінити коментар до певного товару;
- переглянути кошик;
- підтвердити замовлення;
- очистити кошик;
- переглянути свої замовлення та їх статус.

У тому випадку, якщо введені користувачем під час авторизації логін та пароль належать адміністратору вебзастосунку, то користувачеві надається

доступ до режимів роботи, що доступні лише адміністраторам програмного продукту, а саме:

- додавання, редагування, видалення аніме;
- додавання, редагування, видалення категорії товарів;
- додавання, редагування, видалення, архівування власне товарів, які продаються в магазині;
- перегляд і видалення коментарів;
- зміна статусу замовлень користувачів.

Схему функціонування вебзастосунку для авторизованих користувачів відображено на рис. 4.1.

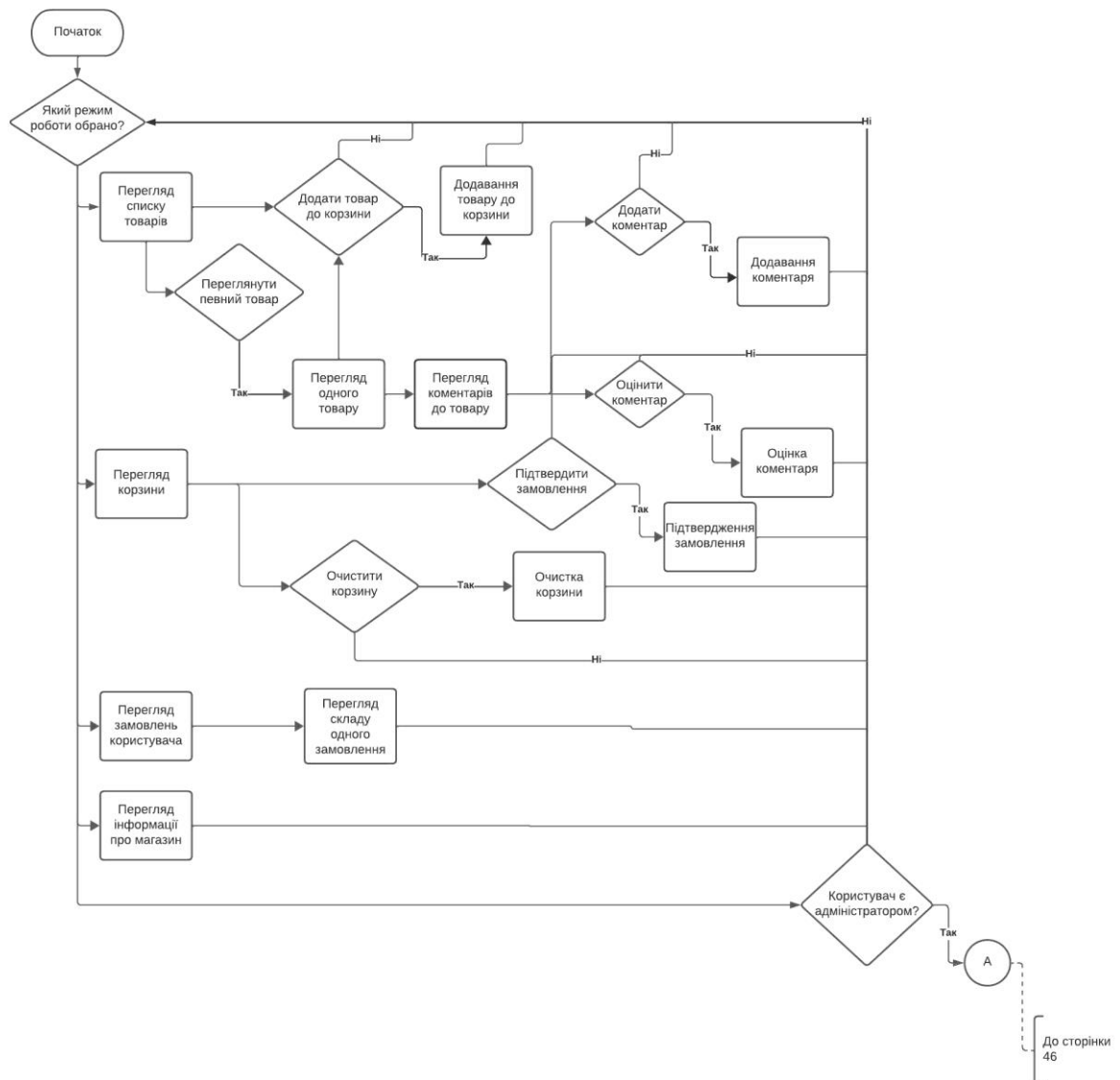


Рисунок 4.1 – Схема функціонування програми, аркуш 1

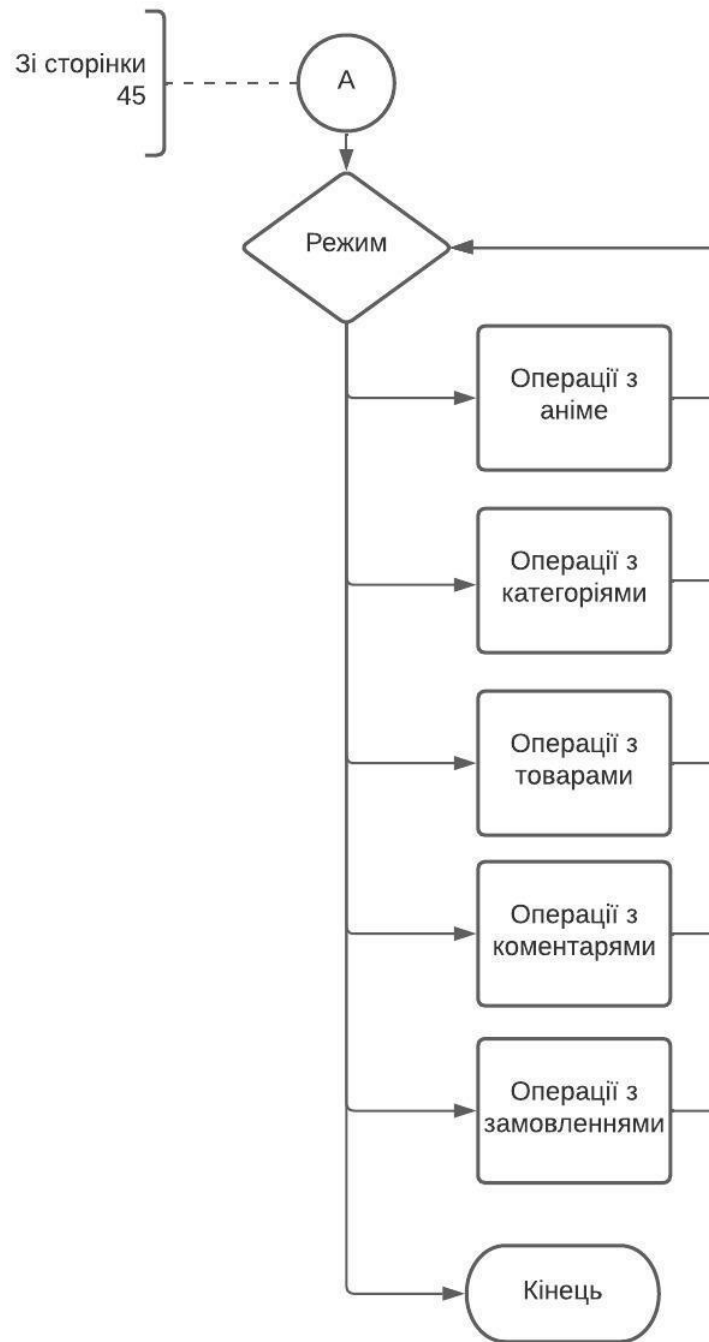


Рисунок 4.1, аркуш 2

4.2 Структура програмного забезпечення

Тека вебзастосунку на сервері розбита на ще декілька тек, у яких знаходяться вебсторінки, призначені для взаємодії з користувачами та з адміністраторами, та спеціальні теки css, db ,images, js, handlers та refactoring,

які призначені для зберігання файлів, що необхідні для нормального функціонування вебзастосунку.

Схема структури вебзастосунку на сервері зображена на рис. 4.2.

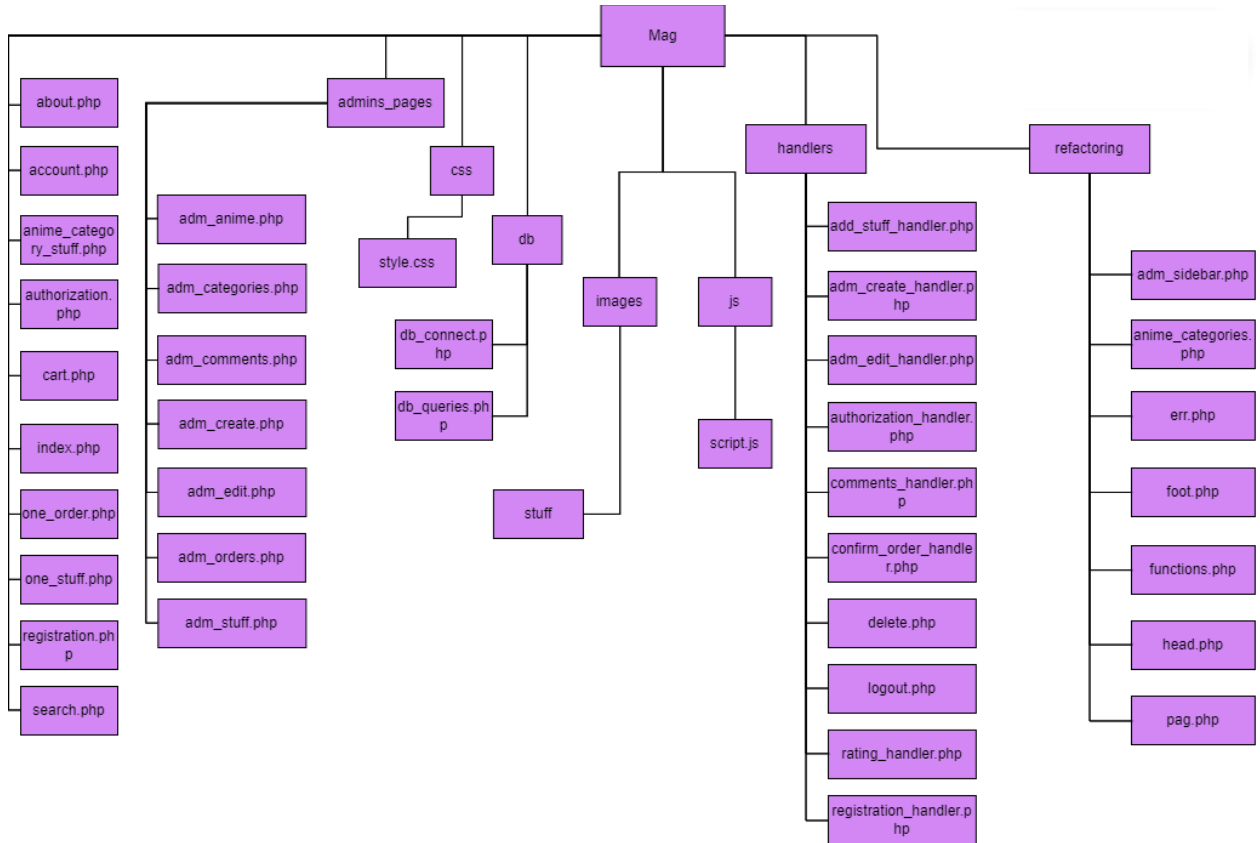


Рисунок 4.2 – Структурна схема теки вебзастосунку на сервері

До вебсторінок, призначених для взаємодії з користувачами належать:

- about.php – вебсторінка з інформацією про магазин;
- account.php – кабінет користувача;
- anime_category_stuff.php – сторінка для відображення товарів, згрупованих за певним аніме чи певною категорією;
- authorization.php – сторінка авторизації користувача;
- cart.php – корзина користувача;
- index.php – головна сторінка вебзастосунку, на якій відображені всі представлені у магазині товари;
- one_order.php – сторінка для перегляду одного замовлення;
- one_stuff.php – сторінка для перегляду одного товару;

- registration.php – сторінка реєстрації нового користувача;
- search.php – сторінка з результатами пошуку певних товару.

Сторінки, призначені для взаємодії з адміністраторами знаходяться у теці `admins_pages`:

- adm_anime.php – сторінка, що відображає всі додані до БД аніме у вигляді зручної таблиці;
- adm_categories.php – сторінка, що відображає всі категорії товарів у вигляді зручної таблиці;
- adm_comments.php – сторінка, що відображає всі коментарі користувачів до товарів у вигляді зручної таблиці;
- adm_create.php – сторінка для додавання категорій товарів чи товарів;
- adm_edit.php – сторінка для редагування категорії товару чи товару;
- adm_orders.php – сторінка, що відображає всі замовлення у вигляді зручної таблиці;
- adm_stuff.php – сторінка, що відображає всі товари у вигляді зручної таблиці.

Тека `handlers` необхідна для зберігання файлів, що містять функції, які викликаються застосунком в результаті певних подій. Файли, що містяться у цій теці:

- add_stuff_handler.php – обробник додавання товарів до кошика;
- adm_create_handler.php – обробник додавання нових категорій товарів та товарів до магазину;
- adm_edit_handler.php – обробник редагування вже наявних категорій товарів та товарів;
- authorization_handler.php – обробник авторизації користувача;
- comments_handler.php – обробник додавання коментарів користувачів до товарів;

- `confirm_order_handler.php` – обробник підтвердження покупки товарів, які містяться у корзині;
- `delete.php` – обробник видалення категорій товарів, товарів, очищення корзини;
- `logout.php` – обробник виходу з облікового запису;
- `rating_handler.php` – обробник оцінки коментаря;
- `registration_handler.php` – обробник реєстрації нового користувача.

Файл `script.js`, який знаходиться в теці `js`, відповідає за поведінку вебсторінок у відповідь на певні дії користувачів вебзастосунку, такі як натискання кнопок, переміщення миші та введення тексту.

Тека `images` містить у собі теку `stuff`. У ній містяться фото товарів, представлених до продажу. Ще у цій теці є файл `image_default.jpg`, який необхідний для того, щоб відображати фото на сторінці товару, якщо адміністратор при додаванні товару не додав його фотографію.

У теці `db` знаходяться два файли – `db_connect.php` та `db_queries.php`. Ці файли необхідні функції для взаємодії вебзастосунку з базою даних.

Тека `css` містить файл `style.css`. Це файл з налаштуваннями, які керують зовнішнім виглядом деяких елементів HTML на веб-сторінці.

У теці `refactoring` зберігаються файли, потрібні для рефакторингу коду:

- `adm_sidebar.php` – сайдбар, що відображається на сторінках, призначених для взаємодії з адміністраторами;
- `anime_categories.php` – список аніме та категорій товарів, який відображається на кількох сторінках вебзастосунку;
- `err.php` – виведення помилок при заповненні форм;
- `foot.php` – футер, який відображається на всіх сторінках вебзастосунку;
- `functions.php` – функції, що часто використовуються під час виконання коду;
- `head.php` – хедер, який відображається на всіх сторінках вебзастосунку;

- pag.php – пагінація сторінок.

4.3 Опис рішень, прийнятих під час реалізації програмного продукту

Для спрощення роботи над розробки вебзастосунку було прийняте рішення винести деякі частини html-коду у окремі файли. Завдяки цьому було досягнуто двох речей. По-перше, вдалося трохи спростити html-коди сторінок через те, що кількість сторінок коду помітно зменшилася, тому іншим розробникам буде легше читати та розуміти ці коди. По-друге, завдяки тому, що ці шматки html-коду дуже часто використовувалися на різних сторінках, процес їх редагування значно спростився. Достатньо зміни певного елемента лише у одному файлі, аби цей елемент змінився на всіх сторінках, де використовується частина коду, у якій міститься даний елемент.

Ось перелік файлів, у яких містяться частини коду, що часто повторюється:

- adm_sidebar.php;
- anime_categories.php;
- err.php;
- foot.php;
- functions.php;
- head.php;
- pag.php.

Як приклад розглянемо файл anime_categories.php, код з якого використовується у деяких сторінках програмного продукту. В цьому файлі міститься html-код, що відображає наявні у базі даних назви аніме та категорії товарів:

```
<div class="categories">
    <h2>Категорії</h2>
    <ul>
```

```

        <?php $all_categories = select_from_table_all('categories',
$clauses = [], $order_by = 'name ASC'); foreach ($all_categories as
$one_category):?>
            <li><a
href="anime_category_stuff.php?table=categories&id=<?=$one_category['id']?>">
<?=$one_category['name']?></a></li>
        <?php endforeach;?>
    </ul>
    <h2>Аніме</h2>
    <ul>
        <?php $all_anime = select_from_table_all('anime', $clauses = [],
$order_by = 'name ASC'); foreach ($all_anime as $one_anime):?>
            <li><a
href="anime_category_stuff.php?table=anime&id=<?=$one_anime['id']?>"><?=$one_
anime['name']?></a></li>
        <?php endforeach;?>
    </ul>
</div>

```

До інших сторінок цей файл можна підключати за допомогою функцій PHP – `include` та `require`, які включають та виконують вказаний файл. У коді сторінки `index.php` підключення файлу `anime_categories.php` відбувається так:

```
<?php require 'refactoring/head.php'?>
```

Так само відбувається підключення інших файлів, що містяться в теці `refactoring` у потрібні частини інших сторінок вебзастосунку.

Також у файлі `db.queries.php` було розроблено функції, які дозволяють оптимізувати запити до бази даних, не прописуючи однакові запити для різних таблиць. Детально розглянемо функцію `select_from_table_all($t_name, $clauses = [], $order_by = '')`, яка повертає всі рядки вказаної таблиці. У якості параметрів приймає ім'я таблиці, з якої необхідно отримати значення, масив, у якому містяться умови про те, які саме рядки з таблиці мають бути отримані та строкову змінну, яка містить інформацію про те, як саме мають бути відсортовані записи з таблиці. Масив з умовами може бути порожнім, в такому випадку функція поверне абсолютно всі рядки, що містяться в таблиці. Змінна,

з інформацією про сортування також може бути порожньою, в такому випадку значення з таблиці будуть відсортовані за зменшенням по значенню поля `id`.

Спочатку у функції оголошується глобальна змінна `$pdo`, яка є екземпляром класу `PDO` і містить у собі параметри для підключення до бази даних. Ця змінна була створена у файлі `db_connection.php`. Після цього створюється змінна `$sql_query`, яка містить в собі запит мовою `SQL` до бази даних на вибірку всіх значень з певної таблиці:

```
global $pdo;
$sql_query = "SELECT * FROM $t_name";
```

Після цього у тому випадку, якщо масив `$clauses` не порожній створюється змінна `$count`, яка є лічильником кількості ітерацій майбутнього циклу:

```
if (!empty($clauses)) {
    $count = 0;
```

Далі за допомогою конструкції `foreach` відбувається перебір масиву `$clauses`. Масив `$clauses` на кожній ітерації розбивається на пари “ключ-значення”, де ключ елемента буде присвоєно змінній `$key`, а значення – змінній `$value`. Якщо значення у змінній `$value` не є числовим, це значення береться у одинарні лапки:

```
foreach ($clauses as $key => $value) {
    if (!is_numeric($value)) {
        $value = "'" . $value . "'";
    }
}
```

У випадку, якщо поточна ітерація є першою, до змінної `$sql_query` за допомогою конкатенації додається ключ елемента масиву та значення елемента масиву, перед якими записується ключове слово “`WHERE`”, якщо ж

поточна ітерація не є першою, до змінної `$sql_query` додається ключ та значення елементу масиву, перед якими записується ключове слово “AND”:

```
if ($count === 0) {
    $sql_query = $sql_query . " WHERE $key = $value";
} else {
    $sql_query = $sql_query . " AND $key = $value";
}
```

Після цього відбувається постфіксний інкремент лічильника ітерацій та закриваються умова не порожності масиву `$clauses` та конструкція `foreach`:

```
    $count++;
}
}
```

Далі за допомогою конкатенації до змінної `$sql_query` додається сортування або за полем `id` потрібної таблиці, або так, як вказано у змінній `$order_by`:

```
if(empty($order_by))      $sql_query = $sql_query . " ORDER BY id
DESC";
else $sql_query = $sql_query . " ORDER BY " . $order_by;
```

Після цього за допомогою методу `prepare()` об'єкту `$pdo` відбувається підготовка запиту до виконання. Далі, використовуючи метод `execute()`, підготований запит запускається на виконання. І у якості результату за допомогою методу `fetchAll()` функція повертає масив, що містить в собі всі рядки результуючого набору:

```
$main_query = $pdo->prepare($sql_query);
$main_query->execute();
return $main_query->fetchAll();
```

Схожим чином працюють і інші функції з файлу `db_queries.php`, наприклад, функції `select_from_table_one($t_name, $clauses = [])`, `insert_into_table($t_name, $clauses)` або `delete_from_table_row($t_name, $id)` які розроблені для відповідно отримання одного рядка з вказаної таблиці, вставки у вказану таблицю рядку з певними значеннями та видалення з вказаної таблиці певного рядка. Розроблені функції дозволяють не прописувати окремо кожний запит на вибірку, додавання, редагування чи видалення даних певної таблиці, а використати в коді вже готову функцію, змінивши лише її вхідні дані.

4.4 Висновки за четвертим розділом

Під час виконання четвертого розділу було розроблено алгоритм функціонування вебзастосунку а також розроблено, описано та показано структурну схему теки вебзастосунку, яка зберігається на сервері. Також зроблено опис рішень, прийнятих під час реалізації програмного продукту, а саме винесення деяких частин `html`-коду в окремі файли, підключення цих файлів до інших сторінок та розробку зручних функцій запитів до бази даних.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Призначення програми

Вебзастосунок призначений для продажу товарів, зокрема товарів аніме атрибутики. Вебзастосунок спрощує функціонування магазину та покращення кількості продажів. Розроблений програмний продукт дозволяє зручно переглядати товари, структуровані за категоріями, виконувати пошук необхідних товарів, замовляти товари, переглядати, додавати та оцінювати коментарі, переглядати список замовлень. Також вебзастосунок дозволяє адміністраторам виконувати перегляд, додавання, редагування, видалення і архівацію аніме, категорій товарів та товарів, зміну статусу замовлень, видалення коментарів користувачів.

5.2 Умови виконання програми

Для стабільної роботи вебзастосунку сервер має бути обладнаний процесором, що має не менше чотирьох ядер з тактовою частотою не менше 3 ГГц, жорсткий диск повинен мати об'єм не менше одного гігабайту, та оперативна пам'ять повинна мати розмір 3 Гб або вище.

Клієнтом є веббраузер користувача вебзастосунку. Для нормального функціонування програмного продукту користувач має мати стабільний доступ до мережі Інтернет та сучасний браузер, який підтримує сучасні стандарти якості.

5.3 Виконання програми

Після введення адреси вебзастосунку в пошуковий рядок відкривається головна сторінка застосунку (рис. 5.1), на якій представлені останні пропозиції магазину, відсортовані по даті додавання.

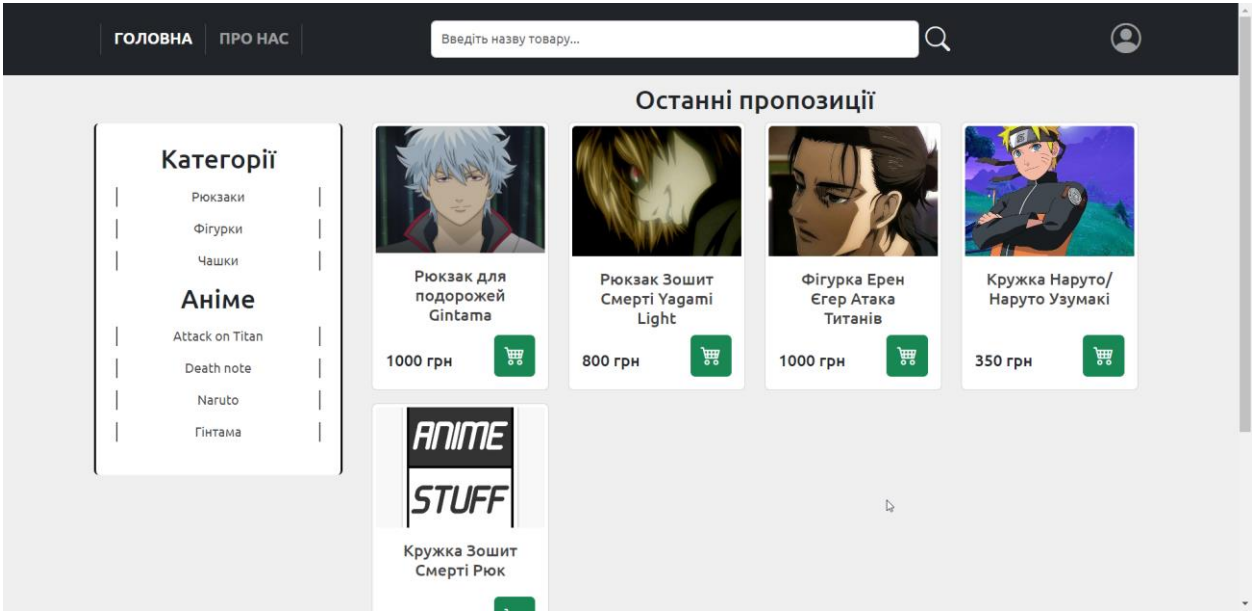


Рисунок 5.1 – Головна сторінка вебзастосунку

На лівій стороні екрану відображаються назви певних аніме та категорії товарів, після кліку по назві аніме чи категорії користувача буде направлено на сторінку(рис. 5.2), де будуть показані товари, що відносяться до даного аніме чи категорії товару.

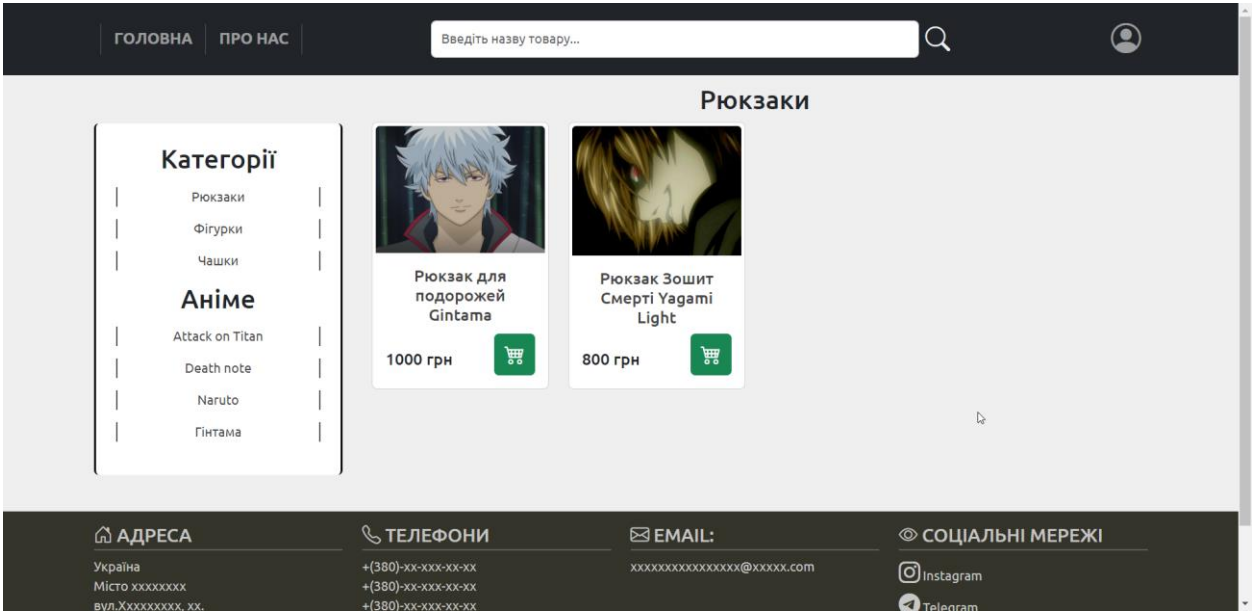


Рисунок 5.2 – Сторінка з товарами певної категорії

Перебуваючи на сторінках з розміщенням товарів користувач може одразу додати потрібний йому товар до корзини. Для цього йому потрібно натиснути на кнопку “Додати до корзини” на відповідній картці відповідного товару (рис. 5.3).

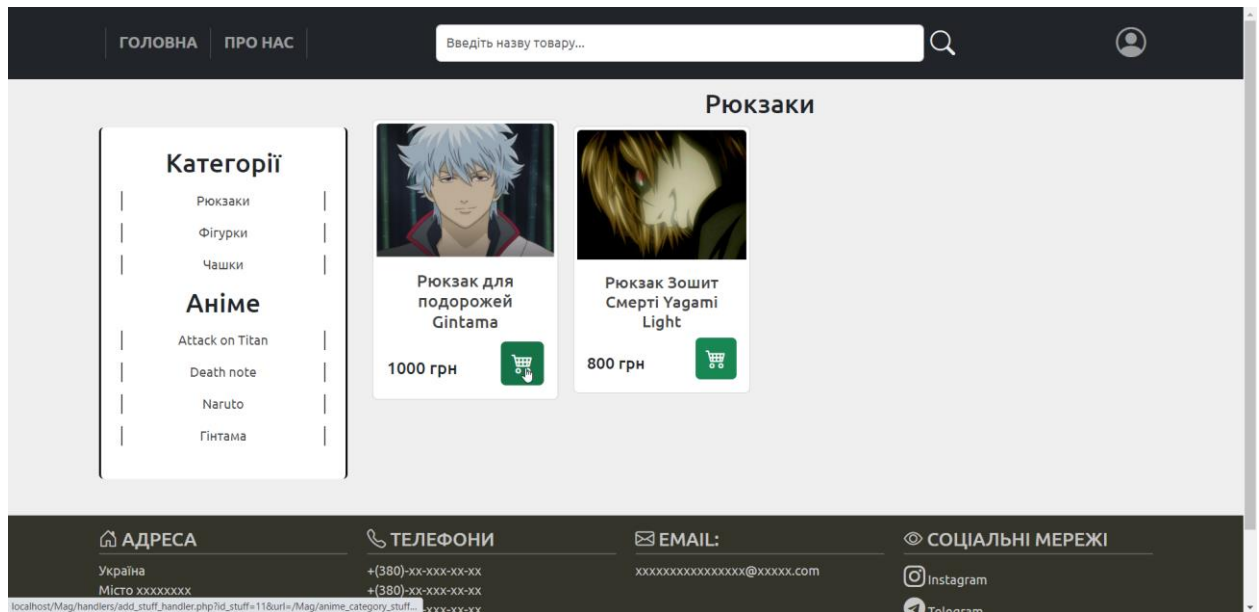


Рисунок 5.3 – Клік на кнопку “Додати до корзини” на картці товару

У випадку, якщо користувач не пройшов авторизацію, при натисканні на кнопку “Додати до корзини” користувача буде направлено на сторінку авторизації (рис. 5.4). На цій сторінці йому буде надано можливість ввести логіну та пароль свого облікового запису. У випадку, якщо введеного користувачем логіну немає у базі даних, або у випадку введення неправильного паролю користувач отримає повідомлення про неправильність введених даних.

Рисунок 5.4 – Сторінка авторизації

Якщо користувач не має облікового запису та хоче зареєструватися, натиснувши на іконку у правому верхньому куті екрану він може перейти на сторінку реєстрації(рис. 5.5), де ввівши свою електронну адресу, логін та пароль він зможе створити новий обліковий запис. Після успішної авторизації або реєстрації у правому верхньому куті екрану буде відображено логін користувача.

Рисунок 5.5 – Сторінка реєстрації

Після авторизації, повернувшись на головну сторінку і натиснувши на назву товару, що зацікавив потенційного покупця, буде відкрито сторінку певного товару (рис. 5.6), на якій користувач зможе прочитати детальний опис товару. Також користувач може додати товар до корзини, натиснувши кнопку “Додати до корзини”. Після цього у правому верхньому куті з’явиться нагадування про те, що корзина вже не порожня та яке буде містити кількість товарів у корзині.

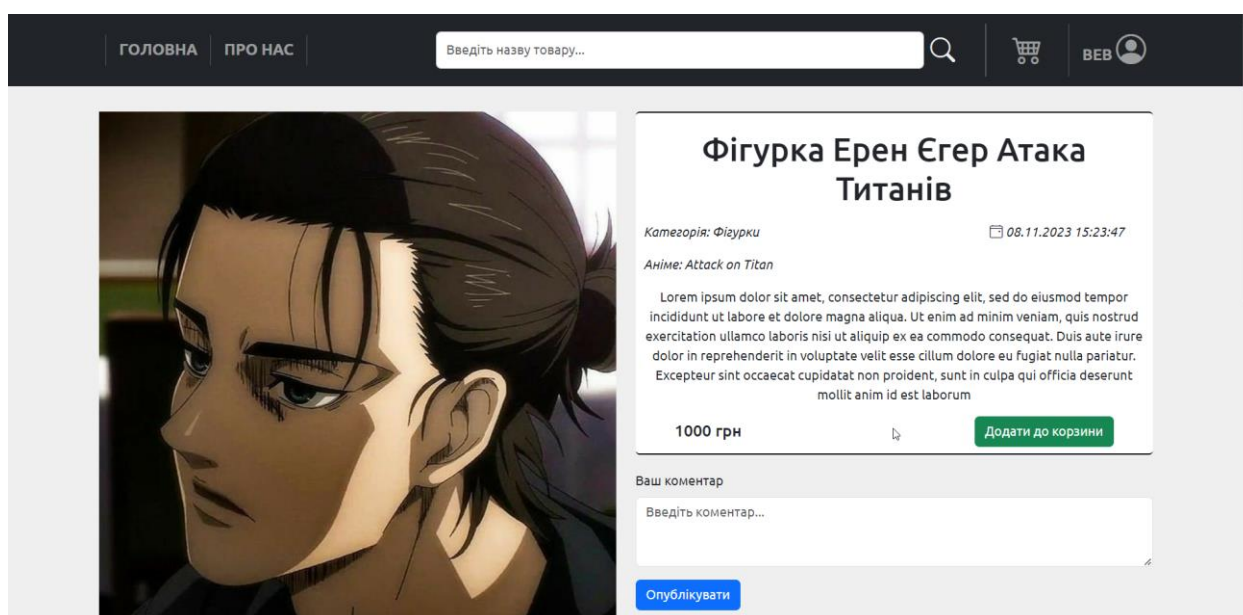


Рисунок 5.6 – Сторінка певного товару

Також перебуваючи на сторінці певного товару, користувач може залишити коментар до даного товару. Якщо користувач хоче це зробити, він має авторизуватися, написати текст коментаря у відповідному полі та натиснути кнопку “Додати коментар” (рис. 5.7). Після цього коментар користувача буде відображено у списку коментарів.

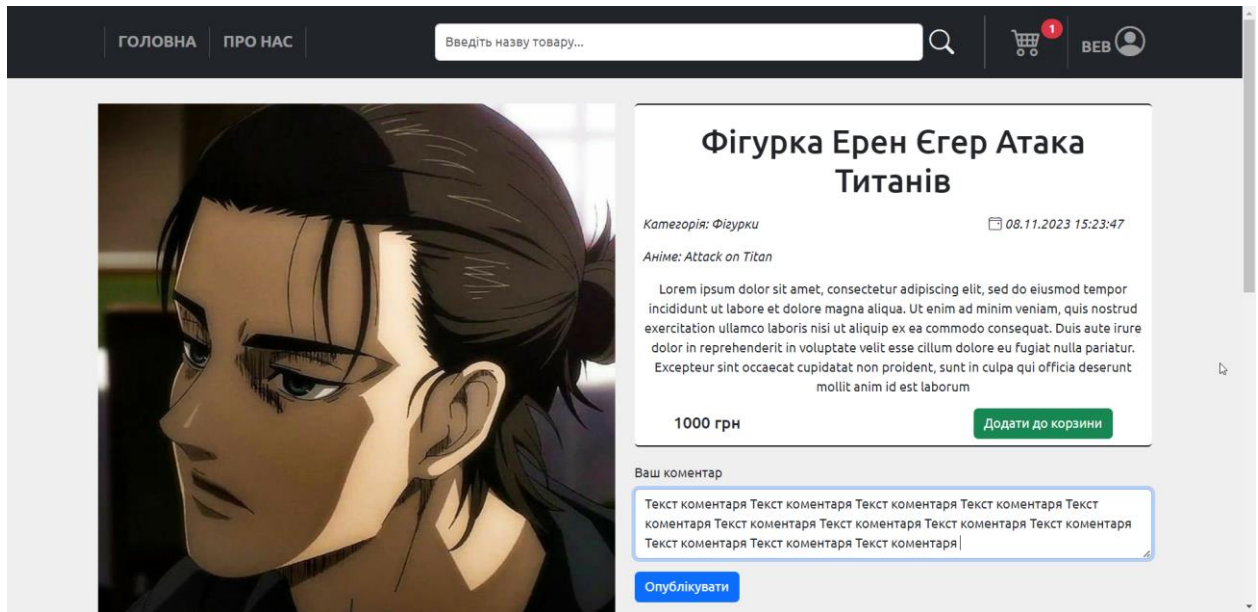


Рисунок 5.7 – Додавання коментаря до товару

Також на сторінці певного товару користувач отримує змогу продивитися відсортовані за датою додавання коментарі інших користувачів про даний товар, та переглянути рейтинги даних коментарів, що може допомогти користувачеві з вибором (рис. 5.8). У випадку, якщо користувачу сподобався або не сподобався певний коментар, він має змогу оцінити цей коментар позитивно або негативно, натиснувши на відповідні кнопки. При спробі оцінити свій же коментар з'явиться відповідне повідомлення про те, що цього робити не можна і оцінка не зарахується.

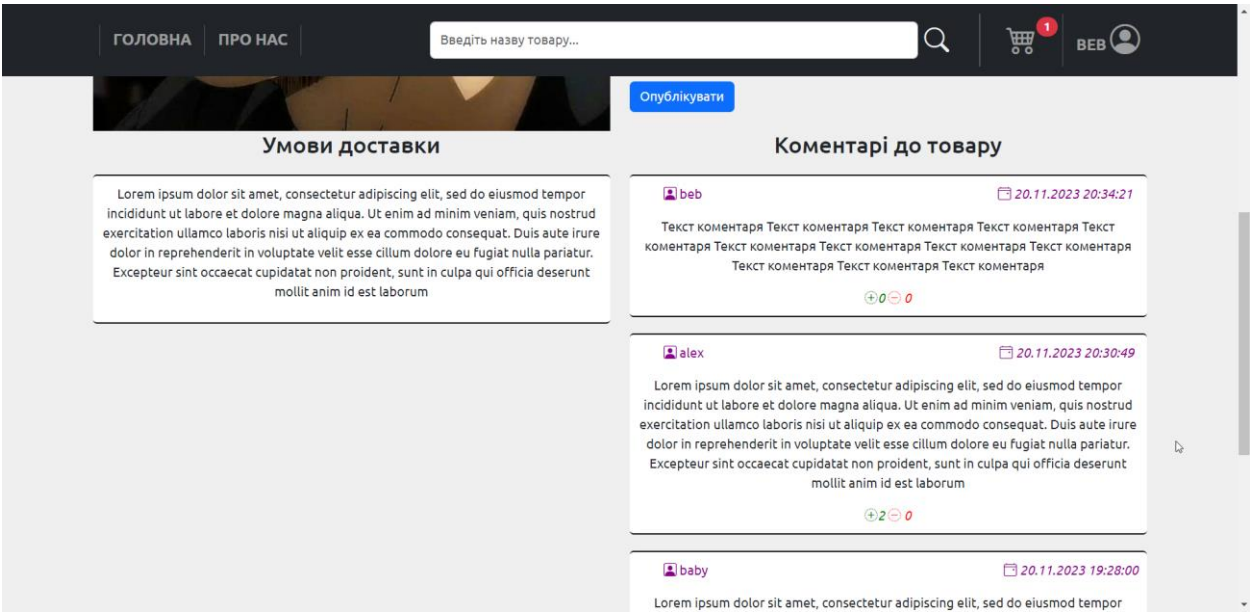


Рисунок 5.8 – Перегляд коментарів до товару

Після того, як користувач закінчить додавання необхідних йому товарів до корзини, він має натиснути на відповідне покликання у верхньому правому куті екрану та перейти на сторінку корзини (рис. 5.9). Тут він зможе переглянути склад поточного замовлення, ціну замовлення, яка буде автоматично підрахована шляхом сумування цін товарів у корзині, видалити товар з корзини чи повністю очистити корзину або підтвердити замовлення.

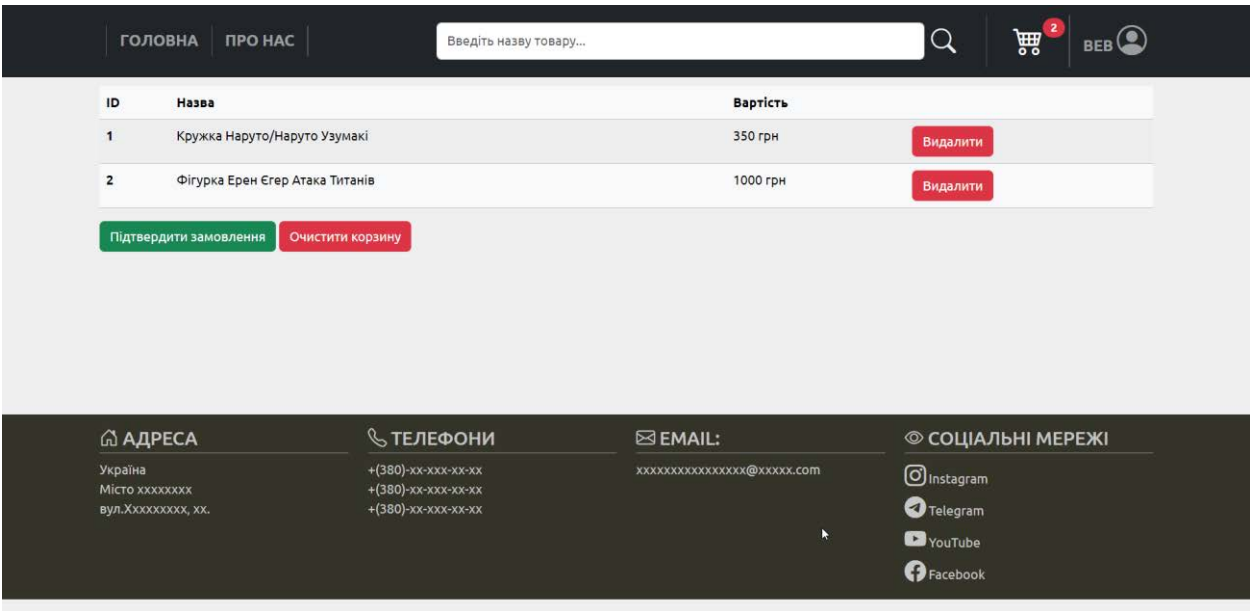


Рисунок 5.9 – Перегляд корзини

Після підтвердження замовлення користувач має змогу перейти до особистого кабінету (рис. 5.10), у якому він може відслідковувати статус замовлення.

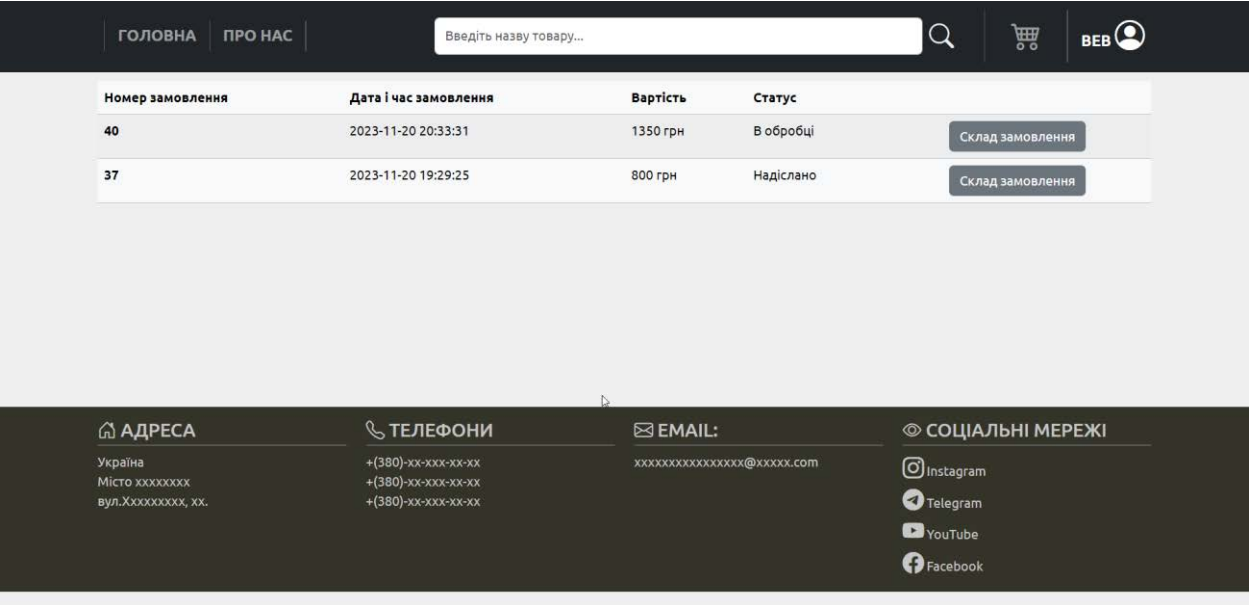


Рисунок 5.10 – Перегляд особистого кабінету

Натиснувши на кнопку “Склад замовлення”, користувач може побачити склад певного замовлення (рис. 5.11).

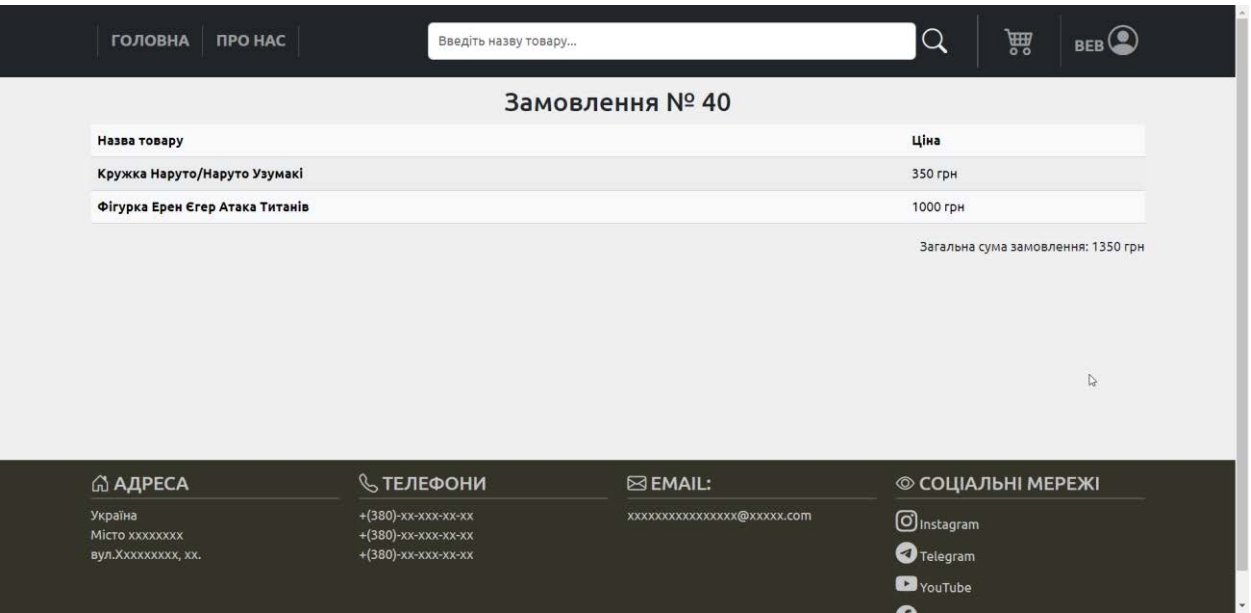


Рисунок 5.11 – Перегляд складу замовлення

Після завершення роботи з вебзастосунком користувач має змогу вийти зі свого акаунту, натиснувши кнопку Вихід у правому верхньому куті (рис. 5.12). Після цього буде здійснено вихід з облікового запису і користувача буде перенаправлено на головну сторінку вебзастосунку.

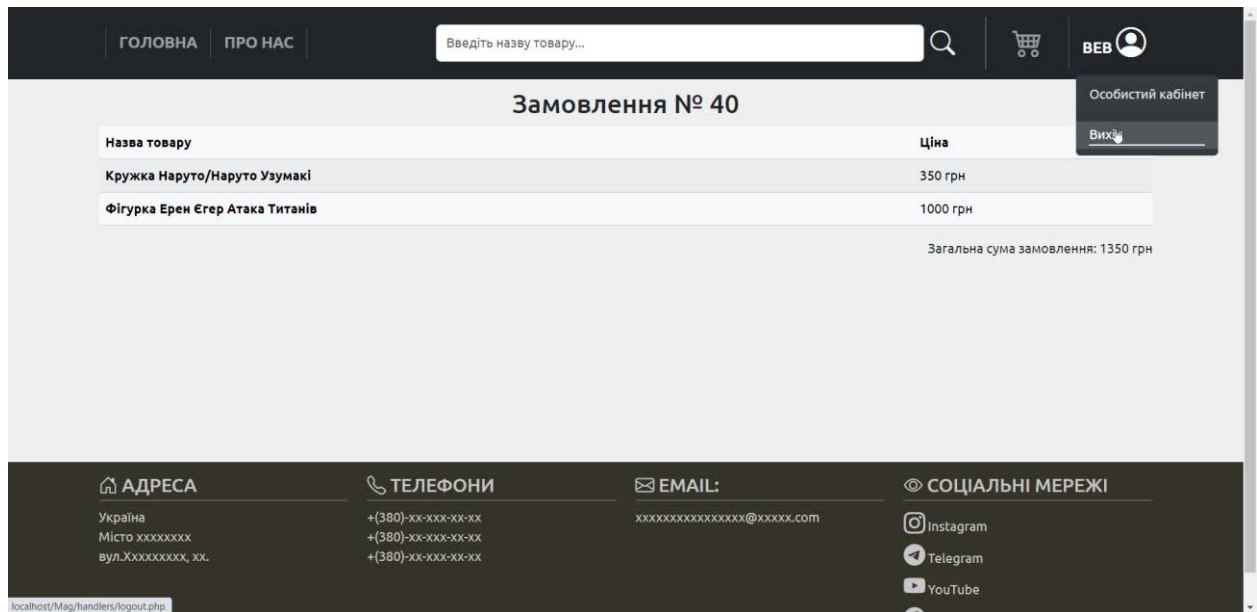


Рисунок 5.12 – Вихід з акаунту

У тому випадку, якщо під час авторизації користувач ввів логін та пароль адміністратора, після кліку на кнопку авторизації його буде перенаправлено на групу сторінок, призначених для адміністратора. Спочатку він опиниться на сторінці таблиці товарів (рис. 5.13), на якій він може переглянути інформацію про товари, додані до бази даних. З цієї сторінки він може перейти на сторінку додавання товару чи редагування товару або видалити чи архівувати певний товар, натиснувши на відповідний чекбокс та кнопку.

ID	Дата	Назва	Вартість	В архів	Керування
11	08.11.2023 21:11:55	Рюкзак для подорожей Gintama	1000 грн	☐	Редагувати Видалити
9	08.11.2023 15:24:14	Рюкзак Зошит Смерті Yagami Light	800 грн	☐	Редагувати Видалити
8	08.11.2023 15:23:47	Фігурка Ерен Єгер Атака Титанів	1000 грн	☐	Редагувати Видалити
7	08.11.2023 15:23:21	Кружка Наруто/Наруто Узумакі	350 грн	☐	Редагувати Видалити
6	08.11.2023 15:22:39	Кружка Зошит Смерті Рюк	2223 грн	☐	Редагувати Видалити

Рисунок 5.13 – Таблиця товарів

На сторінці додавання товару (рис. 5.14) адміністратор може додати товар шляхом заповнення форми з відповідними полями. Після заповнення форми адміністратор має натиснути на кнопку “Додати”. Якщо певні поля форми були залишені порожніми або заповнені не належним чином, адміністратор отримає повідомлення про це, а товар додано не буде.

Заповніть поля нижче

Назва
Введіть назву...

Ціна
Введіть ціну...

Назва аніме
Гінтама

Категорія
Фігурки

Опис
Введіть опис...

Зображення
Вибрати файл Файл не вибрано

[Додати](#)

Рисунок 5.14 – Сторінка додавання товару

На сторінці редагування товару (рис.5.15) адміністратор може редагувати інформацію про товар шляхом зміни даних автоматично заповненої старими даними про товар форми. Після заповнення форми адміністратор має натиснути на кнопку “Редагувати”. Якщо певні поля форми були залишені порожніми або заповнені не належним чином, адміністратор отримає повідомлення про це, а інформацію про товар змінено не буде.

Рисунок 5.15 – Сторінка редагування товару

Також адміністратор може перейти на сторінку аніме чи категорій на яких він може переглянути інформацію про наявні у БД аніме чи категорії товарів. З цієї сторінки він може перейти на сторінку додавання чи редагування аніме або категорії чи видалити аніме або категорію, натиснувши на відповідну кнопку.

Сторінки додавання і редагування аніме чи категорій товарів працюють так само, як і сторінки додавання чи редагування товарів.

При переході на сторінку коментарів (рис. 5.16) адміністратор зможе побачити список коментарів користувачів, та у разі необхідності видалити певний коментар.

ГОЛОВНА

ПРО НАС

Введіть назву товару...

SONNYADMIN

Аніме

Категорії товарів

Товари

Коментарі

Замовлення

Список коментарів

ID	Дата	Текст	Автор	Товар	Керування
23	20.11.2023 20:34:21	Текст коментаря Текст коментаря Текст коментаря Текст коментаря Текст коментаря Текст коментаря	beb	Фігурка Ерен Єгер Атака Титанів	Видалити
22	20.11.2023 20:30:49	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	alex	Фігурка Ерен Єгер Атака Титанів	Видалити
21	20.11.2023 19:28:53	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	alex	Кружка Наруто/Наруто Узумаки	Видалити
20	20.11.2023 19:28:25	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	alex	Рюкзак для подорожей Gintama	Видалити
19	20.11.2023 19:28:00	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	baby	Фігурка Ерен Єгер Атака Титанів	Видалити
18	20.11.2023 19:27:54	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	baby	Рюкзак для подорожей Gintama	Видалити
16	20.11.2023 19:27:00	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	luck	Кружка Наруто/Наруто Узумаки	Видалити
15	20.11.2023 19:26:45	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore ...	luck	Рюкзак для подорожей Gintama	Видалити

Рисунок 5.16 – Сторінка таблиці коментарів

При переході на сторінку замовлень (рис. 5.17) користувач має змогу змінити статус замовлення покупця, що відобразиться у особистому кабінеті певного покупця. Також адміністратор для підвищення зручності роботи має змогу обрати, замовлення з яким статусом будуть відображені у таблиці шляхом натиснення відповідної кнопки. Після завершення роботи з вебзастосунком адміністратор, як і користувач, має змогу вийти зі свого акаунту та закрити вкладку з застосунком у своєму браузері.

ГОЛОВНА	ПРО НАС	Введіть назву товару...	SONNYADMIN
---------	---------	-------------------------	------------

Аніме	Список замовлень				
Категорії товарів	Всі замовлення В обробці Надіслано Прибуло				
Товари	Номер замовлення	Дата	Покупець	Ціна	Статус
Коментарі	40	20.11.2023 20:33:31	beb	1350 грн	В обробці
Замовлення	37	20.11.2023 19:29:25	beb	800 грн	Надіслано
	32	10.11.2023 11:41:57	alex	1600 грн	В обробці
	31	10.11.2023 11:25:14	alex	2000 грн	В обробці
	28	08.11.2023 16:50:14	alex	1500 грн	Надіслано
	27	08.11.2023 16:49:51	test	1000 грн	Надіслано
	26	08.11.2023 16:49:41	test	722 грн	Надіслано
	25	08.11.2023 15:26:25	alex	1222 грн	Прибуло

Рисунок 5.17 – Таблиця замовлень користувачів

5.4 Тестування програмного засобу

Після завершення роботи над розробкою вебзастосунку було проведено його тестування. Під час тестування було зареєстровано користувача з поштою test@gmail.com, логіном та паролем test(рис. 5.18).

Рисунок 5.18 – Реєстрація нового користувача

Після успішної реєстрації нового користувача до його корзини було додано три товари (рис. 5.19). Товари успішно відображаються у корзині користувача. Після цього з корзини було видалено один товар. Товар успішно видалився з корзини та більше не відображається в ній (рис. 5.20)

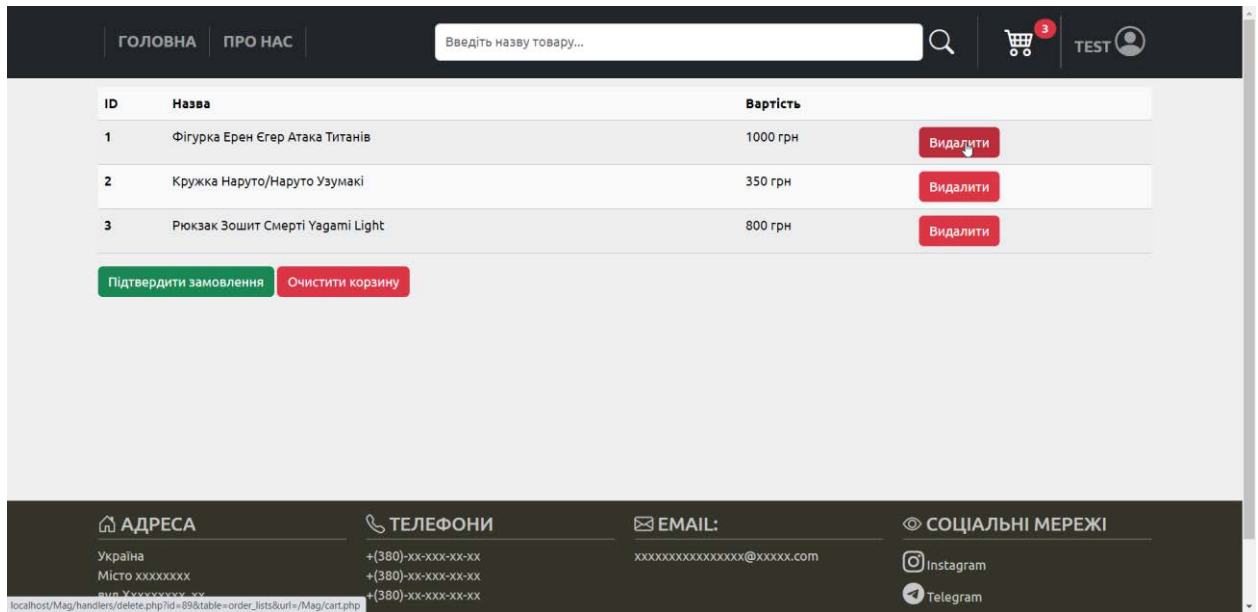


Рисунок 5.19 – Корзина користувача test

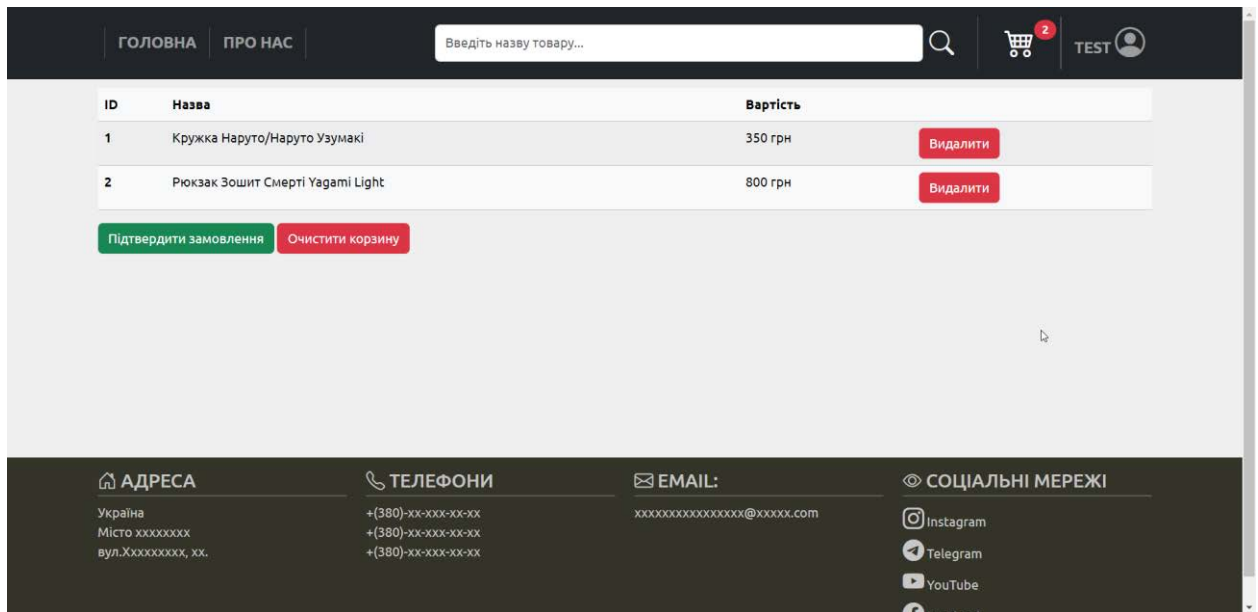


Рисунок 5.20 – Корзина користувача test після видалення одного товару

Після цього дане замовлення було підтверджено, після чого воно коректно відображається в особистому кабінеті користувача (рис. 5.21)

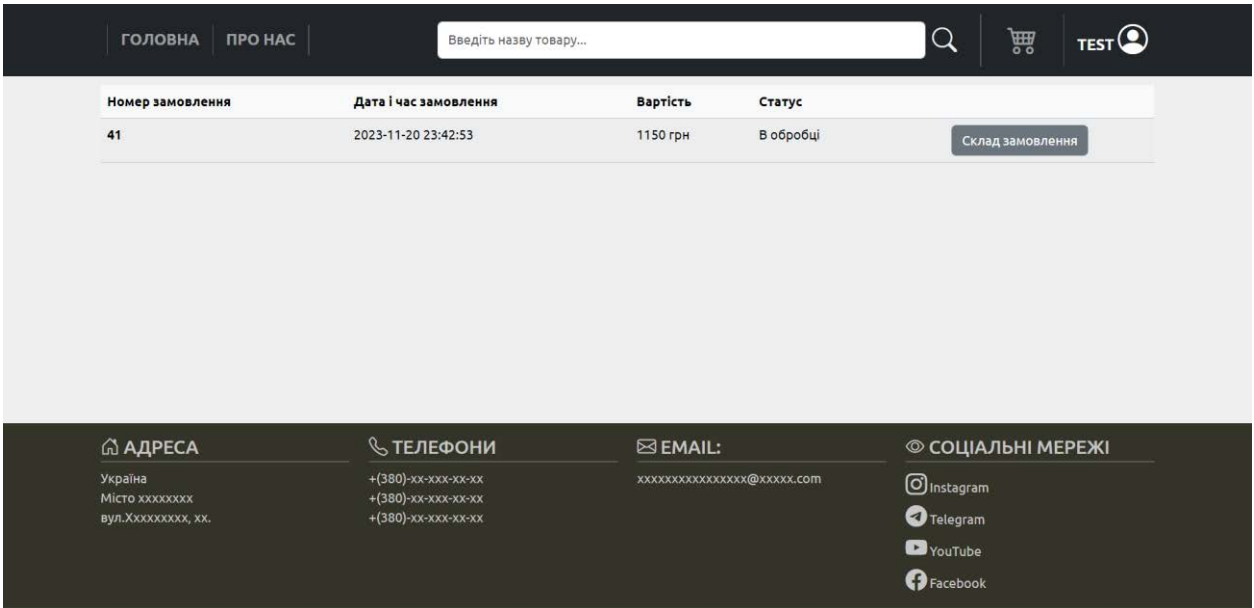


Рисунок 5.21 – Особистий кабінет користувача test після підтвердження замовлення

Після цього був здійснений вихід з акаунту користувача test та здійснена авторизація на вебзастосунку як адміністратор. У таблиці всіх замовлень замовлення користувача test під номером 41 зі статусом «В обробці» відображається цілком коректно (рис. 5.22).

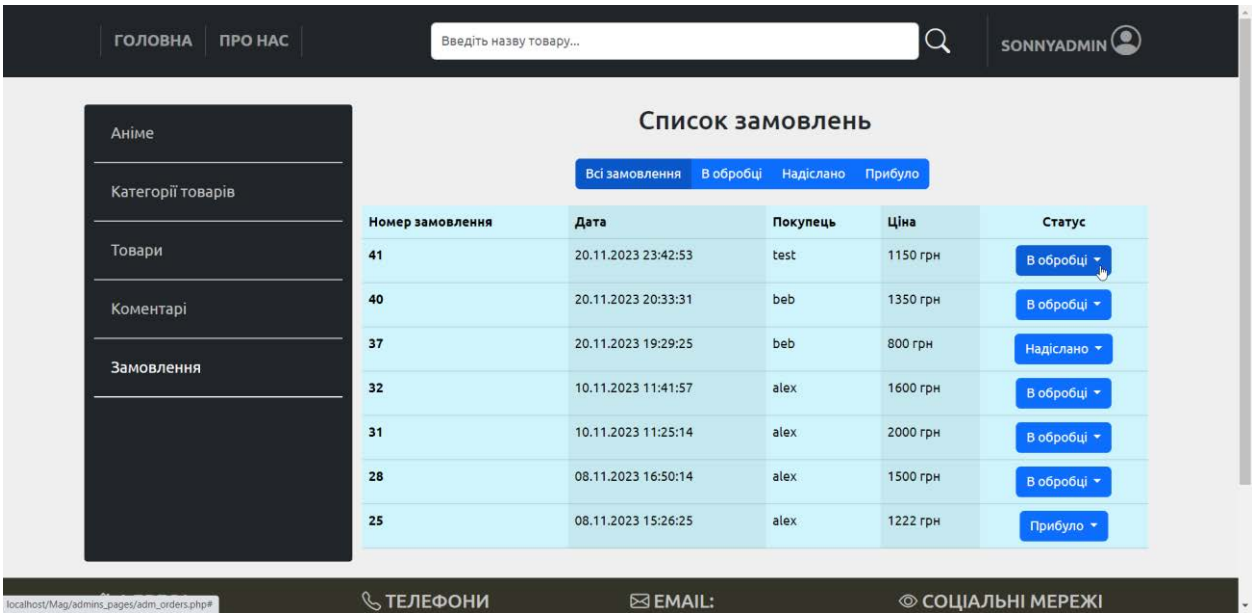


Рисунок 5.22 – Замовлення користувача у таблиці всіх замовлень

Адміністратор в будь-який момент може змінити статус цього замовлення з «В обробці» на «Надіслано» або «Прибуло». Цей статус також буде відображено в особистому кабінеті користувача.

Після цього адміністратор додав до бази даних новий товар з назвою test (рис. 5.23).

[illegible]

Рисунок 5.23 – Додавання замовлення test

Цей товар цілком коректно відображається у таблиці товарі, де адміністратор може виконати з цим товаром необхідні йому операції та на головній сторінці вебзастосунку (рис. 5.24).

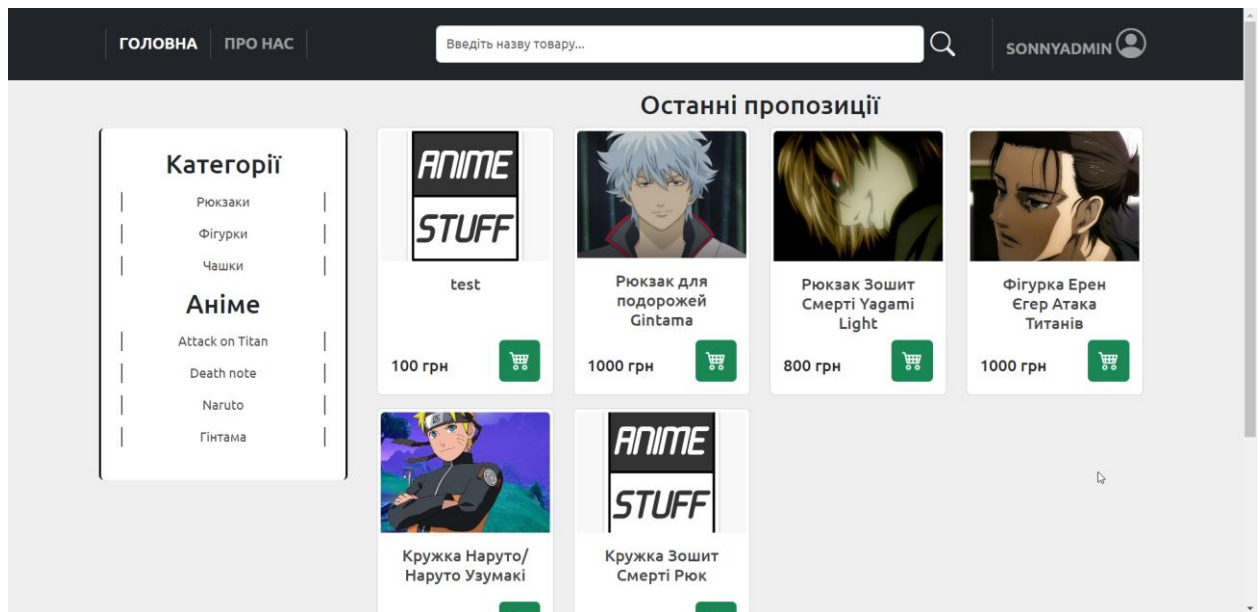


Рисунок 5.24 – Головна сторінка вебзастосунку після додавання замовлення test

Тестування вебсайту пройшло успішно, помилок, які можуть критично впливати на його роботу, виявлено не було.

5.5 Реалізація принципів зручності користування

Принципи Передбачуваності та Звичності реалізуються завдяки схожості деяких елементів вебзастосунку з іншими інтернет-магазинами. Наприклад, у багатьох вебзастосунках для продажу товарів в лівій частині екрану відображаються наявні у базі даних види групування товарів, натиснувши на які користувач отримає змогу продивитися товари, які належать певному групуванню. Передбачуваним для користувача є те, що при кліку на кнопку з зображенням кошика на картці певного товару цей товар буде додано до кошика. Також звичним для користувача є наявність у лівому верхньому куті елементів керування акаунтом (вихід з облікового запису, історія замовлень тощо) чи рядок для введення шуканого товару – на основі свого минулого досвіду взаємодії з вебзастосунками користувач розуміє, що якщо ввести слово у цей рядок та натиснути відповідну кнопку, то програмний

продукт згенерує та покаже користувачеві перелік товарів, назва яких містить в собі шукане слово.

Прикладом реалізації принципу Синтезованості є те, що при кліку на певну категорію товарів, користувач разом з товарами, що відносяться до даної категорії побачить на екрані назву категорії, яку він тільки що обрав. Тобто користувач чудово розуміє, що він потрапив на цю сторінку, тому що натиснув на назву певної категорії товарів. Також прикладом реалізації цього принципу є те, що при додаванні певного товару до порожньої корзини, користувач отримує повідомлення про те, що товар успішно додано, шляхом появи поряд з іконкою корзини у лівому верхньому куті екрану кількості товарів, що містяться у корзини, або збільшення числа, яке відображається поряд з корзиною у тому випадку, якщо корзина не порожня. Тобто інтерфейс показує користувачеві вплив його минулих дій на поточний стан системи.

Принципи Узагальненості та Послідовності реалізуються завдяки тому, що дизайн вебзастосунку зберігає сталість та єдність, немає якихось компонентів, які критично відрізняються від інших частин вебзастосунку та можуть здатися чужорідними елементами. Прикладом реалізації цих принципів є те, що картки кожного товару виглядають однаково, компоненти карток виконані в однаковому стилі та мають однакові розміри, немає якихось елементів, які би виділяли якийсь окремий товар від інших.

Також було реалізовано принцип Ініціативи діалогу. Розроблений програмний продукт передбачає використання здебільшого діалоги, що ініційовані користувачем. Користувач у будь-який момент має змогу ініціювати певні дії у системі, наприклад авторизуватися, зареєструватися, додати товар до корзини тощо.

Прикладом реалізації принципу Передавання завдань є те, що під час реєстрації користувача у тому випадку, якщо користувач ввів некоректні дані в певні поля, наприклад, не вказавши символ «@» у своїй електронній адресі, та не помітив це при самостійній перевірці введених даних, система також

перевірить коректність введеної електронної пошти користувача та повідомить користувача про помилку.

Також під час розробки вебзастосунку було реалізовано принцип Наглядності. Користувач має змогу вивчати різні частини вебзастосунку чи його функціонал через інтерфейс, наприклад для адміністраторів було розроблене спеціальне структуроване меню, що полегшує перехід між сторінками. Також вебзастосунок самостійно визначає значення за замовчуванням для деяких параметрів, наприклад кожному новому замовленню автоматично надається статус «В обробці», або якщо при додаванні товару адміністратор не додав фото товару, то система автоматично прив'яже до товару фото за замовченням.

Принцип Відновлюваності також було реалізовано. Користувач має змогу виправити помилки або повернутися до бажаного стану після неправильної взаємодії чи дії. Наприклад, якщо користувач додав до корзини не той товар або передумав підтверджувати замовлення, то натиснувши на відповідні кнопки він зможе видалити певний товар з корзини чи повністю очистити корзину.

Принцип Відповідності завданням реалізується завдяки тому, що вебзастосунок надає всі необхідні функції та інструменти для того, щоб користувачі могли просто та ефективно виконувати свої завдання, наприклад, зручний пошук, дуже проста організація перегляду товарів певної категорії, зручне додавання товарів до кошика, простий та зрозумілий процес оформлення замовлень тощо.

ВИСНОВКИ

Під час роботи над дипломною кваліфікаційною роботою магістра було зроблено наступне:

- виконано аналіз та огляд предметної області, а саме продаж аніме аксесуарів та було виявлено, що найбільш ефективно і зручно продаж і купівлю можна виконувати користуючись мережею Інтернет. Також було проведено аналіз аналогічних програмних засобів для продажу аніме аксесуарів, їх переваг та недоліків. В ході аналізу було виявлено, що спільним недоліком оглянутих засобів є відсутність можливості оцінити коментарі інших людей, що є доволі важливою функцією, яка дозволяє користувачеві отримати уявлення про корисність або доцільність того чи іншого відгуку;
- зроблено аналіз засобів сфери інформаційних технологій та комп'ютерних мереж для продажу та купівлі товарів онлайн. У якості основних засобів було виділено два, а саме вебсайти та вебзастосунки. Було виконано детальне дослідження вебсайтів та вебзастосунків, зокрема виділено сутність цих понять, виділено основні переваги та недоліки кожного з них, представлено та проаналізовано алгоритми їх роботи. У підсумку дослідження в якості основної моделі рішення задачі продажу та купівлі товарів, зокрема аніме аксесуарів, онлайн було обрано саме вебзастосунок. Також було виконано детальний огляд зручності використання вебзастосунку. Було виділено та досліджено основні пункти та підпункти зручності використання вебзастосунку, описано головні переваги їх використання під час розробки програмного продукту;
- виділено функціональні вимоги до вебзастосунку за допомогою діаграм прецедентів (Use Case Diagram). Також було зроблено огляд засобів для розробки вебзастосунку та виділено їх основні переваги та недоліки та виконано проєктування бази даних та розроблено її схему;
- розроблено алгоритм функціонування вебзастосунку а також розроблено, описано та показано структурну схему теки вебзастосунку, яка

зберігається на сервері. Також було зроблено опис рішень, прийнятих під час реалізації програмного продукту, а саме винесення деяких частин html-коду в окремі файли, підключення цих файлів до інших сторінок та розробку зручних функцій запитів до бази даних;

- розроблено вебзастосунок для магазину аніме аксесуарів, з метою спрощення функціонування магазину та покращення кількості продажів.

В результаті виконання дипломної кваліфікаційної роботи магістра поставлене завдання було виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. MarketSplash: Entrepreneurship, Digital Marketing, Design & Ecommerce Publication [Electronic resource]. – Access mode: <https://marketsplash.com/anime-statistics/>
2. Google Trends [Електрон. ресурс]. – Режим доступу: <https://trends.google.com.ua/trends/>
3. Merriam-Webster: America's Most Trusted Dictionary [Electronic resource]. – Access mode: <https://www.merriam-webster.com/dictionary/anime>
4. Офіційна сторінка Міністерства Оборони України у Twitter [Електрон. ресурс]. – Режим доступу : <https://twitter.com/DefenceU/status/1575738476858773506>
5. Сайт магазину ITEM SHOP [Електрон. ресурс]. – Режим доступу: <https://itemshop.com.ua/>
6. Сайт магазину LOLLYBUY [Електрон. ресурс]. – Режим доступу: <https://lollybuy.com/>
7. Швачич, Г.Г. Сучасні інформаційно-комунікаційні технології: навчальний посібник / [Г.Г. Швачич, В.В.Толстой, Л.М.Петречук та ін.]. – Дніпро : НМетАУ, 2017. – 230 с.
8. Комп'ютерні мережі: навчальний посібник / [А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник]. – Львів : «Магнолія 2006», 2013. – 256 с.
9. Anjum, Nirjhor Evaluation of Web Usability Requirement Model and Web Application Interface Components [Electronic resource] / Nirjhor Anjum. – IJARCSSE, 2019. – Access mode : <https://ssrn.com/abstract=3665742>
10. Dix, Alan Human Computer Interaction 3rd Edition : навчальний посібник / [Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale]. – Pearson Education Limited, 2004. – 834с.
11. W3Schools – a web developer information website [Electronic resource]. – Access mode : https://www.w3schools.com/js/js_intro.asp

12. W3Schools – a web developer information website [Electronic resource]. – Access mode : https://www.w3schools.com/css/css_intro.asp
13. W3Schools – a web developer information website [Electronic resource]. – Access mode : https://www.w3schools.com/html/html_intro.asp
14. W3Schools – a web developer information website [Electronic resource]. – Access mode :
https://www.w3schools.com/bootstrap5/bootstrap_get_started.php
15. W3Schools – a web developer information website [Electronic resource]. – Access mode : https://www.w3schools.com/php/php_intro.asp

ДОДАТОК А
Технічне завдання

Вступ

На початку 21 століття аніме було популярне лише на своїй батьківщині, у Японії. Проте у 2010-х, і особливо зараз, у 2020-х роках 21 століття відбувається справжній бум популярності японської мультиплікації. В Україні також стрімко зростає кількість шанувальників японської мультиплікації. Цей тренд, ймовірно, продовжуватиметься у 2024 році та в наступні роки.

Внаслідок стрімкого зростання популярності японської анімації дуже прибутковою та потрібною багатьом людям стала сфера продажу аніме аксесуарів, бо аксесуари - це відносно недорогий спосіб показати свою любов до аніме, вони є доступними для людей з різним бюджетом. До того ж, аксесуари — це зручний спосіб висловити свою індивідуальність. Вони дозволяють людям показати, що вони шанувальники аніме, але також можуть бути просто стильним доповненням до їхнього одягу.

Проте, незважаючи на досить високу популярність аніме, спеціалізованих магазинів для продажу аксесуарів у нашій державі не так багато, а невелика кількість наявних мають певні недоліки.

А.1 Підстави для розробки

Підставою для розробки вебзастосунку є завдання на випускню кваліфікаційну роботу магістра за темою “Дослідження та програмна реалізація вебзастосунку для онлайн продажу аніме аксесуарів” на основі наказу № 437 від 14 листопада 2023 р. за Національним університетом “Запорізька політехніка”.

A.2 Призначення розробки

Програмний засіб призначений для продажу товарів онлайн та внаслідок цього спрощення функціонування та покращення кількості продажів магазину аніме аксесуарів

A.3 Основні вимоги до програми

A. 3.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик програмного засобу наступні:

- перегляд списку наявних товарів та кожного окремого товару;
- перегляд списку коментарів до певного товару;
- пошук товару за введеним словом;
- авторизація та реєстрація;
- перегляд списку наявних товарів та кожного окремого товару;
- замовлення товару або групи товарів одним замовленням;
- додавання коментарів до певного товару;
- перегляд списку коментарів до певного товару;
- оцінювання коментарів інших користувачів;
- перегляд списку своїх замовлень та їх статусу;
- пошук товару за введеним словом;
- авторизація;
- додавання категорій товарів та власне товарів;
- редагування інформації про вже наявні категорії та товари;
- видалення або архівація категорій товарів чи товарів;
- зміна статусу товару;
- перегляд коментарів користувачів та їх видалення.

А. 3.2 Вимоги до надійності

Вносити інформацію до бази даних мають лише адміністратори та користувачі які успішно авторизувалися і використовуючи для цього лише розроблені спеціально для цього функції вебзастосунку. Неавторизовані користувачі не можуть вносити нових даних.

А.3.3 Умови експлуатації

Для нормального функціонування вебзастосунку потрібен адміністратор, який буде вносити дані до баз даних. Адміністраторів може бути більше, ніж один.

А.3.4 Вимоги до складу та параметрів технічних засобів

Для стабільної роботи вебзастосунку сервер має бути обладнаний процесором, що має не менше чотирьох ядер з тактовою частотою не менше 3 ГГц, жорсткий диск повинен мати об'єм не менше одного гігабайту, та оперативна пам'ять повинна мати розмір 3 Гб або вище.

Клієнтом є веббраузер користувача вебзастосунку. Для нормального функціонування програмного продукту користувач має мати стабільний доступ до мережі Інтернет та сучасний браузер, який підтримує сучасні стандарти якості.

А.4 Порядок контролю та приймання

Контроль за виконанням випускної кваліфікаційної роботи магістра покладається на керівника. Остаточне рішення приймається екзаменаційною комісією після захисту роботи.

ДОДАТОК Б
Текст програми

Б.1 Текст файлу head.php

```

<header class="fixed-top">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container">
            <div
                class="collapse
                navbar-collapse"
id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item py-2 py-lg-1 col-12 col-lg-
auto">
                        <div class="vr d-none d-lg-flex h-100 mx-lg-2
text-white"></div>
                        <hr class="d-lg-none my-2 text-white-50">
                    </li>
                    <li class="nav-item">
                        <a <?php if(current_url() === '/Mag/index.php'
|| current_url() === '/Mag/')?:> class="nav-link active"
                        <?php else?:> class="nav-link" <?php
endif;?:>href="/Mag/index.php">Головна</a>
                    </li>
                    <li class="nav-item py-2 py-lg-1 col-12 col-lg-
auto">
                        <div class="vr d-none d-lg-flex h-100 mx-lg-2
text-white"></div>
                        <hr class="d-lg-none my-2 text-white-50">
                    </li>
                    <li class="nav-item">
                        <a <?php if(current_url() ===
'/Mag/about.php')?:> class="nav-link active"
                        <?php else?:> class="nav-link" <?php
endif;?:>href="/Mag/about.php">Про нас</a>
                    </li>
                    <li class="nav-item py-2 py-lg-1 col-12 col-lg-
auto">
                        <div class="vr d-none d-lg-flex h-100 mx-lg-2
text-white"></div>
                        <hr class="d-lg-none my-2 text-white-50">
                    </li>
                    </li>
                    <form class="d-flex" role="search">
                        <input class="form-control me-2" type="search"
placeholder="Введіть назву товару..." aria-label="Search">

```

```

        <i class="bi bi-search" style="color: #EFEFEF;
font-size: 30px"></i>
        <!--                                <button class="btn btn-success"
type="submit">Шықары</button>-->
        </form>
    </ul>

    <ul class="navbar-nav ms-auto mb-2 mb-lg-0 top-menu">
        <?php if(isset($_SESSION['id'])):?>
            <?php if(!$_SESSION['check_admin']):?>
                <li class="nav-item py-2 py-lg-1 col-12 col-
lg-auto">
                    <div class="vr d-none d-lg-flex h-100
mx-lg-2 text-white"></div>
                    <hr class="d-lg-none my-2 text-white-
50">
                </li>

                <li class="nav-item mx-3">
                    <a href="cart.php" <?php
if(current_url() === '/Mag/cart.php'):?> class="nav-link active"
                    <?php else:?> class="nav-link" <?php
endif;?><i class="bi bi-cart4" style="font-size: 37px"></i>
                    <?php
if(isset($_SESSION['id_cart'])):?>
                        <span class="position-absolute
badge rounded-pill bg-danger"> <?php echo countRow('order_lists', ['id_order'
=> $_SESSION['id_cart']])?> </span>
                    <?php endif;?>
                    </a>
                </li>

                <?php endif;?>
                <li class="nav-item py-2 py-lg-1 col-12 col-lg-
auto">
                    <div class="vr d-none d-lg-flex h-100 mx-
lg-2 text-white"></div>
                    <hr class="d-lg-none my-2 text-white-50">
                </li>

                <li class="nav-item dropdown">

```

```

                                <a      <?php      if(current_url()      ===
'/Mag/account.php' || current_url() === '/Mag/one_order.php'):?>class="nav-
link dropdown-toggle active position-relative"
                                <?php      else:?>      class="nav-link
dropdown-toggle position-relative" <?php      endif;?>      href="#"      data-bs-
toggle="dropdown"><?php echo $_SESSION['login']?>
                                <i      class="bi      bi-person-circle"
style="font-size: 37px;"></i>
                                </a>
                                <ul class="dropdown-menu dropdown-menu-dark
" aria-labelledby="navbarDropdown">
                                <?php if(!$SESSION['check_admin']):?>
                                    <li><a      class="dropdown-item"
href="account.php">Особистий кабінет</a></li>
                                <?php else:?>
                                    <li><a      class="dropdown-item"
href="/Mag/admins_pages/adm_stuff.php">Адмін Панель</a></li>
                                <?php endif;?>
                                    <li><hr class="dropdown-divider"></li>
                                    <li><a      class="dropdown-item"
href="/Mag/handlers/logout.php">Вихід</a></li>
                                </ul>
                                </li>

                                <?php else: ?>
                                    <li class="nav-item dropdown">
                                        <a      <?php      if(current_url()      ===
'/Mag/authorization.php'      ||      current_url()      ===
'/Mag/registration.php'):?>class="nav-link dropdown-toggle active"
                                        <?php      else:?>class="nav-link dropdown-
toggle" <?php      endif;?>      href="#"      data-bs-toggle="dropdown">
                                        <i      class="bi      bi-person-circle"
style="font-size: 37px;"></i></a>
                                        <ul class="dropdown-menu dropdown-menu-
dark" aria-labelledby="navbarDropdown">
                                            <li><a      class="dropdown-item"
href="/Mag/authorization.php">Авторизація</a></li>
                                            <li><hr class="dropdown-divider"></li>
                                            <li><a      class="dropdown-item"
href="/Mag/registration.php">Реєстрація</a></li>
                                        </ul>
                                    </li>
                                <?php endif;?>

```

```

        </ul>
    </div>
</div>
</nav>
</header>

```

Б.2 Текст файла index.php

```

<?php
require 'db/db_queries.php';
require 'refactoring/func.php';
$parameters = pagination(10,'stuff');
$select_stuff = select_from_table_all('stuff', ['archive' => 0]);
?>

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
" rel="stylesheet"
        integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSRlGAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="css/style.css">
    <link
                                                rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-
icons.css">
    <title>Маразин</title>
</head>
<body>
<?php require 'refactoring/head.php' ?>
<div class="container after-header">
    <div class="content row">
        <div class="col-md-3 col-12">
            <?php require 'refactoring/categories.php' ?>
        </div>

```

```

<div class="main-content col-md-9 col-12">
    <h2 class="text-center">Останні пропозиції</h2>
    <div class="container">
        <div class="row gy-3">
            <?php foreach ($all_stuff as $one_stuff): ?>
                <div class="col-3">
                    <div class="card p-1 h-100">
                        
                            <div class="card-body">
                                <h5 class="card-title text-center"><a
href="one_stuff.php?id=<?=$one_stuff['id']?>"> <?=$one_stuff['name']
?></a></h5>
                                <h5 class="position-absolute bottom-0
start-0 mb-4 ms-3"> <?=$one_stuff['price'] ?> грн</h5>
                                <p class="position-absolute bottom-0
end-0 me-3"><a
href="handlers/add_stuff_handler.php?id_stuff=<?=$one_stuff['id']?>&url=<?=$current_url()?>" class="btn btn-success"><i class="bi bi-cart4"></i></a></p>
                                </div>
                            </div>
                        </div>
                    <?php endforeach; ?>
                </div>
            </div>
            <?php require 'refactoring/pag.php' ?>
        </div>
    </div>
    <?php require 'refactoring/foot.php' ?>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.mi
n.js"
        integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
    </body>
</html>

```



```



<div class="card-body">
    <h5 class="card-title text-center"><a
href="one_stuff.php?id=<?=$one_stuff['id']?>"> <?=$one_stuff['name']
?></a></h5>

    <h5 class="position-absolute bottom-0
start-0 mb-4 ms-3"> <?=$one_stuff['price'] ?> грн</h5>

    <p class="position-absolute bottom-0
end-0 me-3"><a
href="handlers/add_stuff_handler.php?id_stuff=<?=$one_stuff['id']?>&url=<?=$current_url()?>" class="btn btn-success"><i class="bi bi-cart4"></i></a></p>
</div>
</div>
</div>
<?php endforeach; ?>
</div>
</div>
<?php require 'refactoring/pag.php' ?>
</div>
</div>
<?php require 'refactoring/foot.php' ?>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.mi
n.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>
</html>

```

Б.4 Текст файлу cart.php

```

<?php
require 'refactoring/func.php';
require 'db/db_queries.php';
$total_cost = 0;
?>

```

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
" rel="stylesheet"
        integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="css/style.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-
icons.css">
    <script src="js/script.js"></script>
    <title>Магазин</title>
</head>
<body>

<?php require 'refactoring/head.php' ?>
<div class="container after-header">
    <table class="table table-light table-striped">
        <thead>
            <tr>
                <th scope="col">ID</th>
                <th scope="col">Назва</th>
                <th scope="col">Вартість</th>
                <th scope="col"></th>
            </tr>
        </thead>

        <tbody>
            <?php if(!empty($_SESSION['id_cart'])):?>
                <?php $order_list =
select_order_list_cart($_SESSION['id_cart']); foreach ($order_list as $key =>
$one_item): ?>
                    <tr>
                        <th scope="row"><?=$key + 1 ?></th>
                        <td><?=$one_item['name']?></td>

```

```

        <td><?=$one_item['price']; ?> грн</td>
        <td><a
href="handlers/delete.php?id=<?=$one_item['id']?>&table=order_lists&url=<?=cu
rrent_url()?>" class="btn btn-danger">Видалити</a></td>
    </tr>
    <?php $total_cost = $total_cost + $one_item['price'];
endforeach; ?>
    <?php endif;?>
</tbody>
</table>
<a
href="handlers/confirm_order_handler.php?total_cost=<?=$total_cost?>"
class="btn btn-success">Підтвердити замовлення</a>
<a
href="handlers/delete.php?id=<?=$_SESSION['id_cart']?>&table=orders&url=<?=cu
rrent_url()?>" class="btn btn-danger">Очистити корзину</a>
</div>

<?php require 'refactoring/foot.php' ?>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.mi
n.js"
    integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>
</html>

```

Б.5 Текст файлу one_stuff.php

```

<?php
require 'handlers/comments_handler.php';
require 'refactoring/func.php';
$stuff = select_from_table_one('stuff', ['id' => $_GET['id']]);
$category = select_from_table_one('categories', ['id' =>
$stuff['id_category']]);
$anime = select_from_table_one('anime', ['id' => $stuff['id_anime']]);
?>

<!doctype html>
<html lang="en">

```

```

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
" rel="stylesheet"
        integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="css/style.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-
icons.css">
    <script src="js/script.js"></script>
    <title>Магазин</title>
</head>
<body>
<?php require 'refactoring/head.php' ?>
<div class="container after-header">
    <div class="row">
        <div class="img col-md-6 col-12">
            
        </div>
        <div class="main_content col-md-6 col-12">
            <div class="one_stuff position-relative">
                <h2><?=$stuff['name']; ?></h2>
                <p class="fst-italic">Категорія: <?=$category['name']?>
<i class="bi bi-calendar4-event" style="margin-left: 280px"> <?=$
date_transform($stuff['date']); ?></i></p>
                <p class="fst-italic" style="">Аніме:
<?=$anime['name']?></p>
                <p class="description text-center">
                    <?=$stuff['description']; ?>
                </p>
                <h5 class="position-absolute bottom-0 start-0 ms-5 mb-
3"> <?=$stuff['price'] ?> грн </h5>
                <div class="position-absolute bottom-0 end-0 me-5 mb-
2"><a href="handlers/add_stuff_handler.php?id_stuff=<?=$stuff['id']?>"
class="btn btn-success ">Додати до коззини</a>

```

```

        </div>
    </div>
    <div class="cpl-md-12 col-12">
        <?php require 'refactoring/err.php'; ?>
        <form action="one_stuff.php?id=<?=$id?>" method="post">
            <input type="hidden" name="id" value="<?=$id?>">
            <div class="mb-3">
                <label for="CommentTextarea" class="form-
label">Ваш коментар</label>
                <textarea name="comment_input" class="form-
control" id="CommentTextarea" rows="3" placeholder="Введіть
коментар..."></textarea>
            </div>
            <div class="col-12">
                <button type="submit" name="button_create_comm"
class="btn btn-primary">Опублікувати</button>
            </div>
        </form>
    </div>
</div>
<div class="container">
    <div class="row main_content">
        <div class="col-6">
            <h3 class="text-center">Умови доставки</h3>
            <div class="one_stuff">
                <p class="text-center">
                </p>
            </div>
        </div>
    </div>
    <div class="col-6">
        <h3 class="text-center">Коментарі до товару</h3>
        <?php foreach ($all_comments as $one_comment): ?>
            <?php
                $all_ratings
                =
select_from_table_all('rating',['id_comment' => $one_comment['id']]);
                $plus = 0;
                $minus = 0;
                foreach ($all_ratings as $one_rating) {
                    if($one_rating['rating'] === '1'){
                        $plus += 1;
                    }else{
                        $minus += 1;
                    }
                }
            </?php>
        </?php>
    </div>
</div>

```

```

    }
  }?>
  <div class="one_stuff">
    <p style="color: #8B008B"><i class="bi bi-person-square" style="margin-left: 32px;"></i> <?=$one_comment['login']?> <i class="bi bi-calendar4-event" style="margin-left: 360px"> <?=$one_comment['date']; ?></i></p>
    <p class="text-center"><?=$one_comment['text']?></p>
    <div class="text-center">
      <a href="handlers/rating_handler.php?id_com=<?=$one_comment['id']?>&rating=plus&id_stuff=<?=$id?>" class="plus"><i class="bi bi-plus-circle-dotted"><?=$plus?></i></a>
      <a href="handlers/rating_handler.php?id_com=<?=$one_comment['id']?>&rating=minus&id_stuff=<?=$id?>" class="minus"><i class="bi bi-dash-circle-dotted"><?=$minus?></i></a>
    </div>
  </div>
</div>
<?php endforeach; ?>
</div>
</div>
<?php require 'refactoring/foot.php' ?>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>
</html>

```

Б.6 Текст файла add_stuff_handler.php

```

<?php
require '../db/db_queries.php';
require '../refactoring/func.php';

check_authorization();

```

```

        if(empty($_SESSION['id_cart'])) {
            $_SESSION['id_cart'] = insert_into_table('orders', ['id_user' =>
$_SESSION['id']]);
        }

        $stuff = select_from_table_one('stuff', ['id' => $_GET['id_stuff']]);

        $id_stuff = insert_into_table('order_lists', ['id_stuff' =>
$_GET['id_stuff'], 'id_order' => $_SESSION['id_cart'], 'price' =>
$stuff['price']]);

        if(isset($_GET['url']))
            header('location:
http://localhost/Mag/index.php');
        else
            header('location: http://localhost/Mag/one_stuff.php?id=' .
$_GET['id_stuff']);

```

Б.7 Текст файлу comments_handler.php

```

<?php
require 'db/db_queries.php';

$id = $_GET['id'];

$error_messages = [];
$text = '';

if (($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['button_create_comm'])) {

    $text = trim($_POST['comment_input']);

    if ($text === '') {
        array_push($error_messages, 'Ви заповнили не всі поля!');
    } else {
        if(empty($_SESSION)) {
            array_push($error_messages, 'Ви не авторизувалися');
        } else {
            $parameters = [
                'text' => $text,

```

```

        'id_stuff' => $_POST['id'],
        'id_user' => $_SESSION['id']
    ];

    $id_last_comment      =      insert_into_table('comments',
$parameters);
    }
}
    $all_comments = select_with_login(['id', 'date', 'text', 'plus',
'minus'], 'comments', ['id_stuff' => $_POST['id']]);
} else {
    $text = '';
    $all_comments = select_with_login(['id', 'date', 'text', 'plus',
'minus'], 'comments', ['id_stuff' => $id]);
}

```

Б.8 Текст файлу confirm_order_handler.php

```

<?php
require 'db/db_queries.php';

$id = $_GET['id'];

$error_messages = [];
$text = '';

if      ($_SERVER['REQUEST_METHOD']      ===      'POST'      &&
isset($_POST['button_create_comm'])) {

    $text = trim($_POST['comment_input']);

    if ($text === '') {
        array_push($error_messages, 'Ви заповнили не всі поля!');
    } else {
        if(empty($_SESSION)){
            array_push($error_messages, 'Ви не авторизувалися');
        }else{
            $parameters = [
                'text' => $text,
                'id_stuff' => $_POST['id'],
                'id_user' => $_SESSION['id']
            ];

```

```

];

$id_last_comment      =      insert_into_table('comments',
$parameters);
    }
}
    $all_comments = select_with_login(['id', 'date', 'text', 'plus',
'minus'], 'comments', ['id_stuff' => $_POST['id']]);
} else {
    $text = '';
    $all_comments = select_with_login(['id', 'date', 'text', 'plus',
'minus'], 'comments', ['id_stuff' => $id]);
}

```

Б.9 Текст файла rating_handler.php

```

<?php
require '../db/db_queries.php';
require '../refactoring/func.php';

if ($_SERVER['REQUEST_METHOD'] === 'GET' && isset($_GET['rating'])) {
    if(empty($_SESSION)){
        header('location: /Mag/authorization.php');
    } else{
        $rating = select_from_table_one('rating', ['id_comment' =>
$_GET['id_com'], 'id_user' => $_SESSION['id']]);
        if($_GET['rating'] === 'plus'){
            if(!empty($rating) && $rating['rating'] === '1')
delete_from_table_row('rating', $rating['id']);
            elseif(!empty($rating) && $rating['rating'] === '-1')
update_data('rating', $rating['id'], ['rating' => 1]);
            else insert_into_table('rating', ['rating' => 1,
'id_comment' => $_GET['id_com'], 'id_user' => $_SESSION['id']]);
        }else {
            if(!empty($rating) && $rating['rating'] === '-1')
delete_from_table_row('rating', $rating['id']);
            elseif(!empty($rating) && $rating['rating'] === '1')
update_data('rating', $rating['id'], ['rating' => -1]);
            else insert_into_table('rating', ['rating' => -1,
'id_comment' => $_GET['id_com'], 'id_user' => $_SESSION['id']]);
        }
    }
}

```

```

        header('location: http://localhost/Mag/one_stuff.php?id=' .
$_GET['id_stuff']);
    }
}

```

Б.10 Текст файлу style.css

```

/* Сторінки з переліком товарів*/

.card{
    transition: all 0.2s;
}

.card:hover{
    transform: scale(1.05);
}

.card img{
    max-height: 50%;
    object-fit: cover;
}

.card i{
    font-size: 25px;
}

/* Сторінки з переліком товарів*/

/*Сайдбар категорій*/

.links {
    background-color: white;
    padding: 25px;
    border-radius: 6px;
    margin-bottom: 25px;
    text-align: center;

    margin-top: 48px;
    border-left: 3px solid black;
    border-right: 3px solid black;
}

```

```
.links li {
    margin: 15px 0;
    border-left: 2px solid #464646;
    border-right: 2px solid #464646;
}
```

```
/*Сайдбар категорій*/
```

```
/*Сторінка одного товару*/
```

```
.img img{
    margin-top: 18px;
}

.info .one_stuff {
    margin-bottom: 19px;
    border-bottom: 3px solid black;
    border-top: 3px solid black;
    background-color: white;
    padding: 10px;
    border-radius: 5px;
    margin-top: 20px;
}

.info h2 {
    padding: 9px 0;
    font-size: 38px;
    text-align: center;
}

.description {
    margin-top: 10px;
    margin-bottom: 50px;
}

/*Сторінка одного товару*/
.bi-plus-circle-dotted{
    color: #008000;
}

.bi-dash-circle-dotted{
    color: #FF0000;
}

.aut_reg button {
    text-align: center;
    width: 32.4%;
    margin-top: 15px;
}

.aut_reg h2 {
```

```

        text-align: center;
        margin: 19px 1px 19px 1px;
    }
    .sidebar-admin {
        border-radius: 4px;
        min-height: 550px;
        margin-top: 22px;
    }
    .sidebar-admin a {
        font-size: 19px;
        color: #C1C1C1;
        display: block;
        padding: 21px;
        border-bottom: 2px solid #C1C1C1;
        transition: all 0.4s;
    }
    .sidebar-admin a:hover {
        transition: all 0.2s;
        border-bottom: 2px solid white;
        color: white;
        padding-left: 30px;
    }
    .admin-content a:hover::after {
        transform: scale(0);
    }
    .active_admin a {
        color: white;
        border-bottom: 2px solid white;
    }
    .admin {
        margin-top: 22px;
    }
    .buttons a {
        margin-bottom: 19px;
        width: 29%;
    }
    .buttons button {
        margin-bottom: 9px;
    }
    .status .btn-group{
        margin: 20px 0px 20px 250px;
    }

```

ДОДАТОК В
Слайди презентації

Диплома робота
Дослідження та програмна реалізація
вебзастосунку для онлайн продажу аніме
аксесуарів

Diploma work
Research and Software Implementation of a
Web Application for Online Sale
of Anime Accessories

Виконав

Студент групи КНТ-222м Олексій ЦИММЕРМАН

Керівник

К.т.н., доцент Тетяна КАПЛІЄНКО

Рисунок В.1 – Слайд № 1

Вступ

Об'єкт дослідження - продаж товарів онлайн.

Предмет дослідження - вебзастосунок для онлайн продажу аніме аксесуарів.

Мета роботи - розробка вебзастосунку для магазину аніме аксесуарів, задля спрощення функціонування магазину та покращення кількості продажів.

Завдання дипломної кваліфікаційної роботи магістра:

- ▶ аналіз предметної області;
- ▶ огляд матеріалів та методів, що допомагають у вирішенні задачі.
- ▶ розробка архітектури програми;
- ▶ розробка програми;
- ▶ експлуатація, тестування та експериментальне дослідження програми.

2

Рисунок В.2 – Слайд № 2

Порівняння онлайн та офлайн продажу

	Онлайн	Офлайн
Доступність	Краще	Гірше
Зручність	Краще	Гірше
Асортимент	Більший	Менший
Ціни	Менші	Більші
Попередній огляд товару	Майже неможливо	Цілком можливо
Небезпека шахрайства	Більша	Менша

3

Рисунок В.3 – Слайд № 3

Порівняння вебзастосунку, що розроблюється, з аналогами

	ITEM SHOP	LollyBuy	Розроблений програмний засіб
Зручний для користувача інтерфейс	+	+	+
Можливість онлайн купівлі товарів	+	+	+
Пошук на сайті	+	+	+
Можливість залишати коментарі та відгуки про товар	+	+	+
Можливість оцінити коментар іншого користувача	-	-	+
Інтерфейс повністю українською мовою	+	-	+

4

Рисунок В.4 – Слайд № 4

Алгоритми роботи вебсайту та вебзастосунку



Рисунок В.5 – Слайд № 5

Порівняння вебсайту та вебзастосунку

	Вебсайт	Вебзастосунок
Простота розробки	Краща	Гірше
Стійкість до мережевих атак	Краща	Гірше
Вартість розробки	Менша	Більша
Вартість обслуговування	Менша	Більша
Зручність для користувача	Менша	Більша
Взаємодія з користувачами	Гірше	Краще
Оновлення контенту	Гірше	Краще

Рисунок В.6 – Слайд № 6

Головні принципи Навчальності

Принцип	Визначення принципу
Звичність	Здатність функціоналу та інтерфейсу вебзастосунку бути звичними та легкими для розуміння для користувачів завдяки їх минулому досвіду
Передбачуваність	Здатність вебзастосунку допомагати користувачам легко передбачити результати своїх запитів до вебзастосунку
Синтезованість	Здатність вебзастосунку допомагати користувачам оцінювати вплив своїх минулих дій та операцій на поточний стан вебзастосунку
Послідовність	Здатність вебзастосунку зберігати єдину концепцію та сталість дизайну
Узагальненість	Здатність вебзастосунку допомагати користувачам застосовувати свій досвід роботи з однією частиною чи функцією вебзастосунку під час взаємодії з іншими його частинами, тими, яких вони раніше не зустрічали

7

Рисунок В.7 – Слайд № 7

Головні принципи Гнучкості

Принцип	Визначення принципу
Налаштованість	Здатність вебзастосунку дозволяти користувачам налаштовувати та змінювати під себе деякі елементи інтерфейсу
Взаємозамінність	Здатність вебзастосунку замінювати одні опції на інші, еквівалентні ним, без втрати функціональності
Передавання завдань	Здатність передавати контроль над виконанням завдання від користувача до системи або від системи до користувача в потрібний час
Багатопотоковість	Здатність вебзастосунку одночасно оброблювати та виконувати декілька завдань користувача
Ініціатива діалогу	Здатність вебзастосунку до взаємодії з користувачами за допомогою двох типів діалогів – ініційованих користувачем та ініційованих системою

8

Рисунок В.8 – Слайд № 8

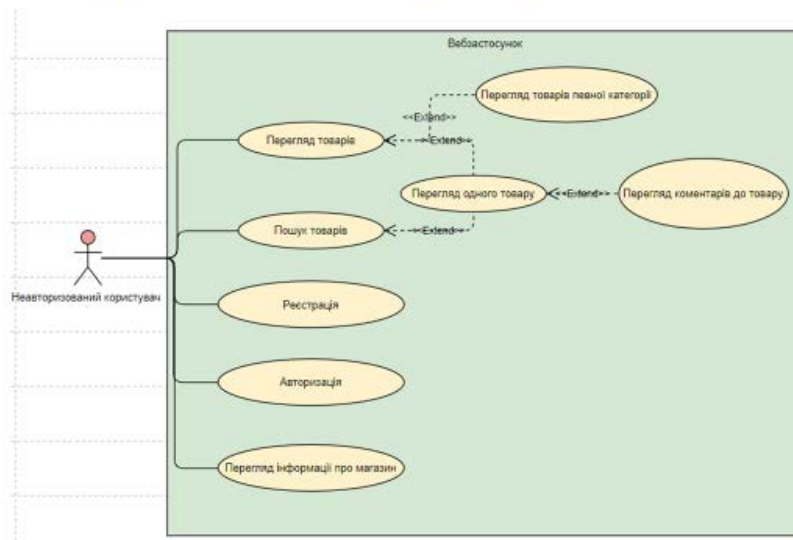
Головні принципи Надійності

Принцип	Визначення принципу
Відповідність завданням	Ступінь того, наскільки добре можливості вебзастосунку відповідні завданням, які користувачі прагнуть виконати з його допомогою і ступінь того, наскільки можливості вебзастосунку є зрозумілими для користувачів
Швидкість реагування	Здатність вебзастосунку швидко реагувати та відповідати на запити користувачів
Відновлюваність	Здатність вебзастосунку дозволяти користувачам виправляти зроблені ними помилки при роботі з вебзастосунком
Наглядність	Здатність користувача оцінити внутрішній стан системи за допомогою його зрозумілого відображення на екрані

9

Рисунок В.9 – Слайд № 9

Діаграма прецедентів неавторизованого користувача



10

Рисунок В.10 – Слайд № 10

Діаграма прецедентів авторизованого користувача

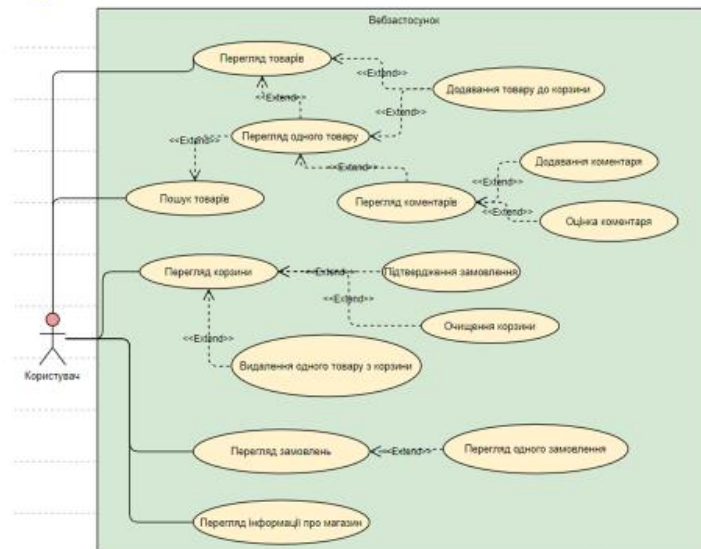


Рисунок В.11 – Слайд № 11

Діаграма прецедентів адміністратора

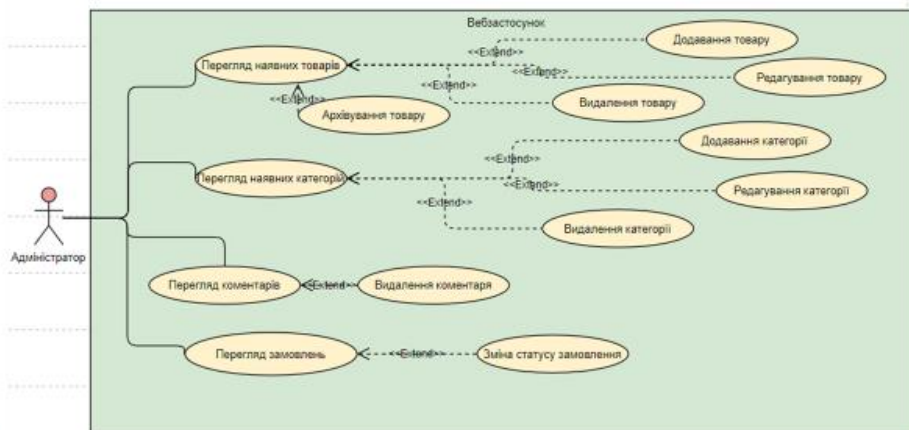


Рисунок В.12 – Слайд № 12

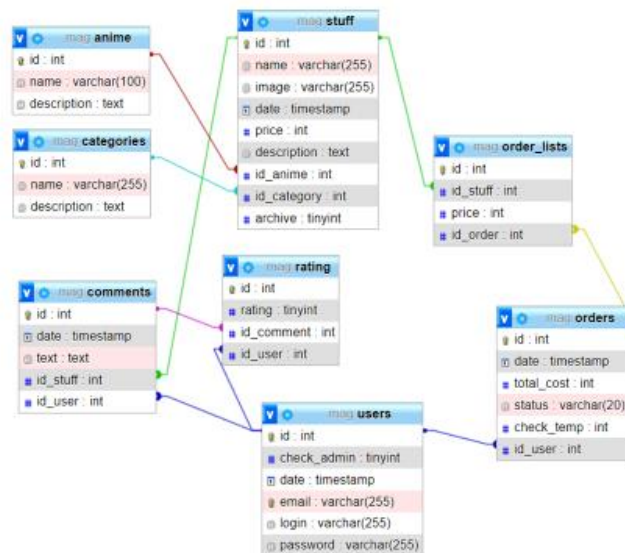
Порівняння PHP та Python

	Мова програмування	
	PHP	ASP.NET
Спільнота	Велика та активна	Також є великою але трохи меншою
Легкість встановлення	Чудово встановлюється у Windows, Linux, Mac	Іноді можуть виникнути певні проблеми
Фреймворки	Наявні та зручні, наприклад Laravel або Symfony	Також є в наявності зручні фреймворки, наприклад Django
Розширення та бібліотеки	Велика кількість розширень і бібліотек	Широкий вибір бібліотек, але менше розширень
Швидкість та продуктивність	PHP 7 має надзвичайну швидкість	Трохи менша між в останньої версії PHP
Безпека	PHP має проблеми з безпекою	Вважається безпечнішим, особливо з використанням Django.
Взаємодія з базами даних	Чудова	Гірша, бо для деяких баз даних потрібен спеціальний драйвер
Синтаксис	Дещо складний	Чіткий і зрозумілий
Популярність	Дуже велика	Дещо менша
Сфера використання	Здебільшого веброзробка	Дуже велика сфера застосування, від веброзробки до штучного інтелекту

13

Рисунок В.13 – Слайд № 13

Схема бази даних



14

Рисунок В.14 – Слайд № 14

Схема функціонування програми

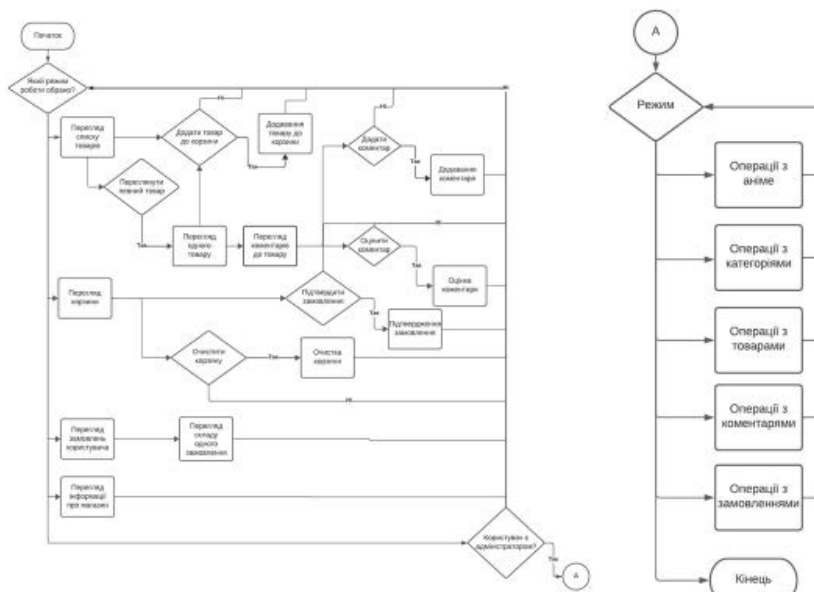


Рисунок В.15 – Слайд № 15

Структурна схема теки вебзастосунку на вебсервері

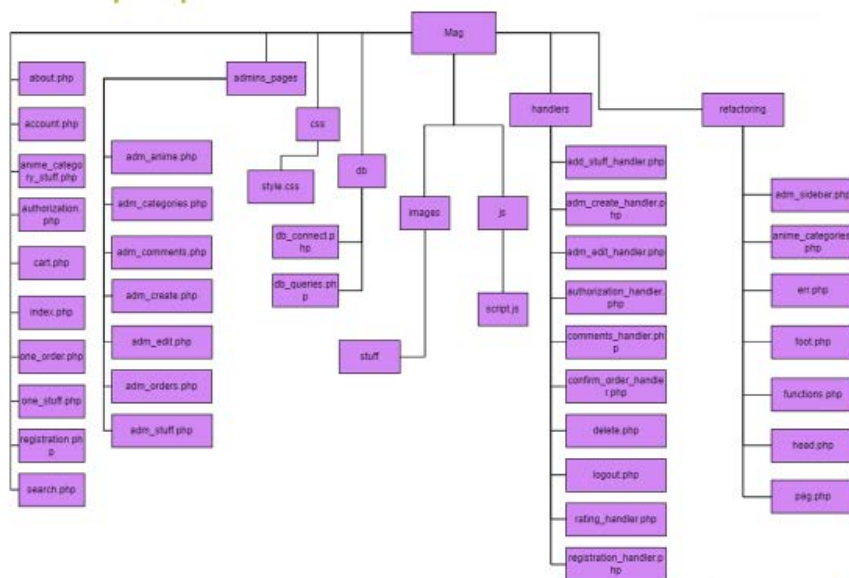


Рисунок В.16 – Слайд № 16

Інтерфейси програми

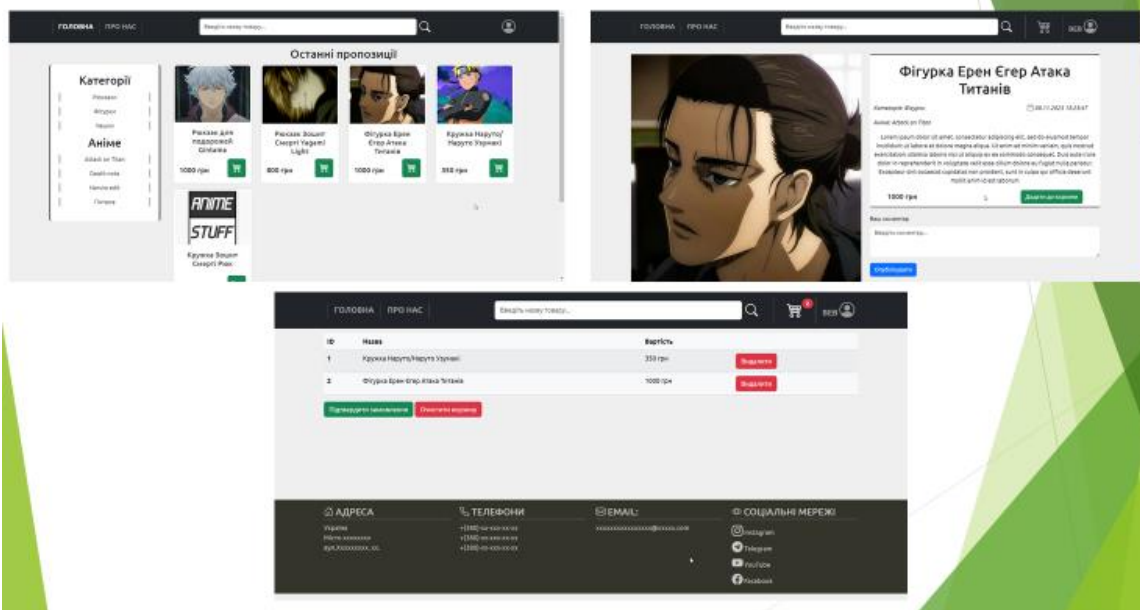


Рисунок В.17 – Слайд № 17

Інтерфейси програми

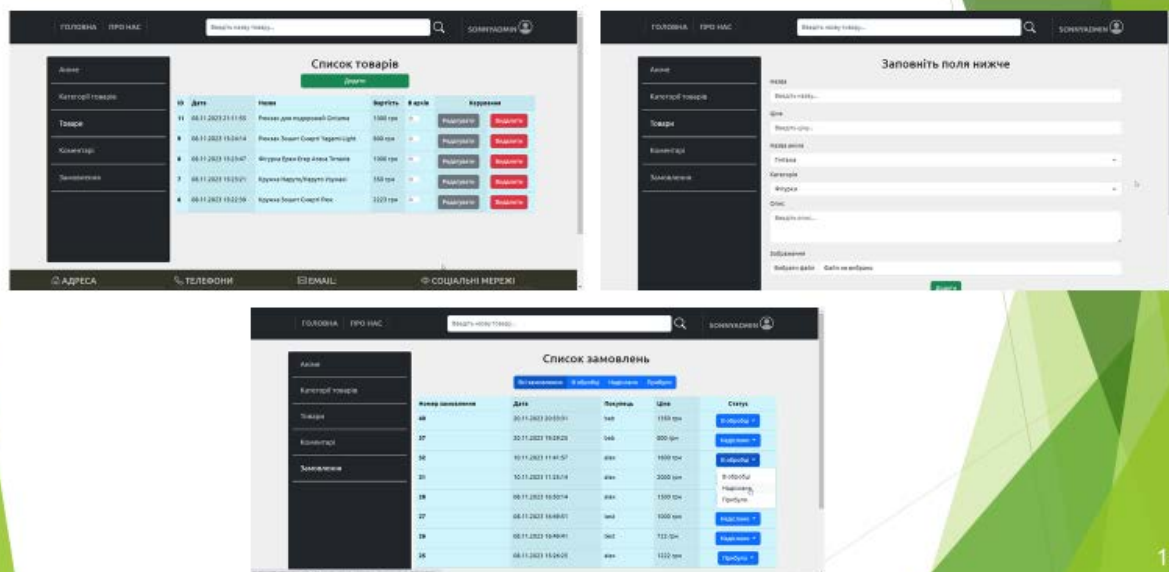


Рисунок В.18 – Слайд № 18

Висновки

Під час роботи над дипломною кваліфікаційною роботою магістра було зроблено наступне:

- ▶ виконано аналіз та огляд предметної області, та було виявлено, що найбільш ефективно і зручно продаж і купівлю можна виконувати користуючись мережею Інтернет. Також було проведено аналіз аналогічних програмних засобів для продажу аніме аксесуарів, їх переваг та недоліків;
- ▶ зроблено аналіз та дослідження засобів сфери інформаційних технологій та комп'ютерних мереж для продажу та купівлі товарів онлайн. Також було виконано детальний огляд зручності використання вебзастосунку. Було виділено та досліджено основні пункти та підпункти зручності використання вебзастосунку, описано головні переваги їх використання під час розробки програмного продукту;
- ▶ виділено функціональні вимоги до вебзастосунку за допомогою діаграм прецедентів (Use Case Diagram). Також було зроблено огляд засобів для розробки вебзастосунку та виділено їх основні переваги та недоліки та виконано проєктування бази даних та розроблено її схему;
- ▶ розроблено алгоритм функціонування вебзастосунку і описано та показано структурну схему теки вебзастосунку, яка зберігається на сервері. Також було зроблено опис рішень, прийнятих під час реалізації програмного продукту, а саме винесення деяких частин html-коду в окремі файли, підключення цих файлів до інших сторінок та розробку зручних функцій запитів до бази даних;
- ▶ розроблено вебзастосунок для магазину аніме аксесуарів, з метою спрощення функціонування магазину та покращення кількості продажів.

Рисунок В.19 – Слайд № 19