ДИПЛОМНА РОБОТА
БАКАЛАВРА

ДРБ. ІП - 21.00.00.000 ПЗ

Група ІП-20-2

Підгаєцький Святослав

2024

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

УДК 004.4

ДИПЛОМНА РОБОТА БАКАЛАВРА

Тема: “ Розробка сервісу електронної комерції для реалізації ювелірних виробів ”

(назва згідно з наказом ректора)

спеціальність – 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДРБ.ІП - 21.00.00.000 ПЗ

(позначення)

Консультанти:

з питань нормоконтролю

асист. /М.М. Піх/
(посада) (підпис) (дата)(розшифровка підпису)

Студент

ІП-20-2 /С.І.Підгаєцький/
(шифр групи) (підпис) (дата) (розшифровка підпису)

Керівник роботи

д.т.н., проф. /В.М.Юрчишин/
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту
Завідувач кафедри

к.т.н., доц. /В.В.Бандура/
(посада) (підпис) (дата) (розшифровка підпису)

2024
(рік)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ ____ ” _____2024 р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ БАКАЛАВРА СТУДЕНТОВІ

Підгаєцькому Святославу Ігоровичу

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) “ Розробка сервісу електронної комерції для реалізації ювелірних виробів”

керівник проекту (роботи) Юрчишин Володимир Миколайович, професор

затверджені наказом закладу вищої освіти від **“** ____ ” травня 2024 р. № _____

2. Строк подання студентом проекту (роботи) 17 червня 2024 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

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

1. Аналіз предметної області розробки ресурсів електронної комерції та постановка задачі
2. Проектування сервісу електронної комерції для реалізації ювелірних виробів
3. Діаграми та їх реалізація в структурному підході
4. Програмна розробка сервісу електронної комерції для реалізації ювелірних виробів
5. Розробка основних методів та функцій

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Основні бізнес моделі електронної комерції (рис. 1.1)
2. Основні критерії вибору інтернет-магазину (рис. 1.2)
3. Статистика використання розширень доменних імен (рис. 1.3)
4. Основні елементи пошукового маркетингу (рис. 1.4)
5. Бізнес-модель типового Інтернет магазину (рис. 2.1)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання **22 травня 2024 р.**

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області розробки ресурсів електронної комерції та постановка задачі	26.05.2024	виконано
2	Проектування сервісу електронної комерції для реалізації ювелірних виробів	29.05.2024	виконано
3	Діаграми та їх реалізація в структурному підході	30.05.2024	виконано
4	Програмна розробка сервісу електронної комерції для реалізації ювелірних виробів	04.06.2024	виконано
5	Розробка основних методів та функцій	10.06.2024	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	17.06.2024	виконано

Студент – дипломник _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Дана робота присвячена розробці ресурсу електронної комерції для продажу ювелірних виробів. Метою проекту є створення веб-платформи, яка надасть користувачам можливість придбати ювелірні вироби онлайн зручним та безпечним способом. Результатом роботи є розроблений та протестований прототип веб-платформи для електронної комерції з великим вибором ювелірних виробів. Платформа дозволяє користувачам зручно та безпечно придбати бажані товари онлайн, що робить процес покупок ще більш доступним та ефективним.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ-МАГАЗИН, ПРОДАЖ ТОВАРІВ, ЕЛЕКТРОННА КОМЕРЦІЯ, ОПТИМІЗАЦІЯ ВЕБ-РЕСУРСУ, РЕАЛІЗАЦІЯ ЮВЕЛІРНИХ ВИРОБІВ, ДІАГРАМА ПРЕЦЕДЕНТІВ.

ANNOTATION

This work is devoted to the development of an e-commerce resource for the sale of jewelry. The goal of the project is to create a web platform that will provide users with the opportunity to purchase jewelry online in a convenient and safe way. The result of the work is a developed and tested prototype of a web platform for e-commerce with a large selection of jewelry products. The platform allows users to conveniently and safely purchase the desired products online, which makes the shopping process even more accessible and efficient.

KEY WORDS: ONLINE STORE, SALE OF GOODS, ELECTRONIC COMMERCE, WEB RESOURCE OPTIMIZATION, JEWELRY MARKETING, CHART OF PRECEDENTS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 8
ВСТУП 10

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ РЕСУРСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ 11
1.1. Значення електронної комерції у розвитку власної справи 11
1.2. Принципи побудови E-Commerce магазинів 13
1.3. Аналіз основних аспектів створення сервісів E-Commerce 18
1.4. Забезпечення інформаційною підтримкою існуючого бізнесу 21
1.5. Організація двохсторонньої взаємодії із клієнтом 22
1.6. Безпека та конфіденційність даних 23
1.7. Аудит та Search Engine Optimization 27

2. АНАЛІЗ ТА ПРОЕКТУВАННЯ СЕРВІСУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ДЛЯ РЕАЛІЗАЦІЇ ЮВЕЛІРНИХ ВИРОБІВ 34
2.1. Аналіз принципів роботи ресурсу електронної комерції 34
2.2. Вимоги до технології розробки сервісу електронної комерції 36
2.3. Процедура здійснення користувацьких покупок 37
2.3.1. Оформлення товару 37
2.3.2. Проведення оплати замовлення 38
2.3.3. Доставка замовлення 39
2.4. Управління життєвим циклом клієнта 40
2.5. Оцінка ризиків при розробці і проектуванні ресурсу електронної комерції 43
2.6 Діаграми та їх реалізація в структурному підході 49

2.7. Модель життєвого циклу програмного забезпечення 53
2.7.1 Waterfall SDLC 54
2.7.2 Iterative SDLC 56
2.7.3 Модель Agile SDLC 57

3. ПРОГРАМНА РОЗРОБКА СЕРВІСУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ДЛЯ РЕАЛІЗАЦІЇ ЮВЕЛІРНИХ ВИРОБІВ 60
3.1 Вибір середовища програмування 60
3.2. Аналіз вимог до розроблюваного програмного продукту 60
3.3. Реалізація дизайну та колірної схема оформлення сервісу електронної коммерції 66
3.4. Реалізація способу зберігання даних 68
3.5. Розробка основних методів та функцій 70
3.5.1. Компонент Lazy-Resources.js 71
3.5.2. Клас ShopHome 71
3.5.3. Клас ShopCategoryData 72
3.5.4. Клас ShopListItem 72
3.5.5. Клас ShopCartData 73
3.5.6. Клас ShopCheckout 74
ВИСНОВКИ 75
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА 77
ДОДАТКИ
БІБЛІОГРАФІЧНА ДОВІДКА

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Еквайр - банк або інша фінансова установа, що надає послуги еквайрингу, тобто, здійснює розрахунки з підприємствами, які приймають оплату від держателів платіжних карток за товари чи послуги або видають їм готівку.

Емітент - банк, що діє за дорученням платника про відкриття акредитива і відповідно до його вказівки, зобов'язується зробити платежі одержувачеві коштів.

Доменне ім’я - це адреса (ім’я) сайту в мережі інтернеті, яка складається з літер і цифр.

FeedBack – зворотній зв’язок клієнта з установою.

GUI - Графічний інтерфейс користувача.

DDoS Attack - розподілена атака на відмову в обслуговуванні системи.

DarkNet - оверлейна мережа, доступ до якої можливий лише через певне програмне забезпечення, налаштування чи авторизацію, часто з використанням нестандартних комунікаційних протоколів та портів.

SSL - криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером.

HTTPS – протокол передачі даних з підтримкою шифрування.

CMS - система керування вмістом.

CSP - це стандарт комп'ютерної безпеки, який вводиться для запобігання міжсайтових сценаріїв, джек-джек і інших атак на введення коду.

VPS - віртуальний виділений сервер.

IP-адреса - це ідентифікатор мережевого рівня, який використовується для адресації комп'ютерів чи пристроїв у мережах.

URL - стандартизована адреса певного ресурсу.

CLM - управління життєвим циклом клієнта.

SDLC – життєвий цикл програмного забезпечення.

Agile – гнучка методологія розробки програмного забезпечення.

Scrum - підхід управління проєктами для гнучкої розробки програмного забезпечення.

Забінджeння (bind) – присвоєння клавішам ПК виконання певних функцій.

Ком’юніті – спільнота, що формується навколо конкретного продукту.

Material Design - принципи дизайну сайтів, програмного забезпечення і застосунків.

IDE – середовище розробки програмного забезпечення.

DPI – кількість пікселів на один квадратний дюйм.

FAQ - підбірка частих питань на певну тему та відповідей на них.

ВСТУП

Сьогодні, в реаліях сучасного суспільства, кожен вид малого та великого бізнесу, кожна компанія, успішна фірма, що розвивається - повинна йти в ногу з часом і повинна освоювати нові тенденції, щоб триматися на висоті. Велика частина успіху кожної справи лежить у правильній подачі контенту та рекламі своєї продукції, запропонованих послуг, тощо. За допомогою своєї власної інтернет сторінки фірми вирішують такі завдання, як представлення компанії в мережі Інтернет, розширюють власну потенційну аудиторію інвесторів та споживачів, підтримують власний бренд, підвищують популярність продукції, інформують клієнтів і так далі.

Розробка комерційних сайтів є актуальною і затребуваною сферою діяльності, тому що сайт компанії в мережі Інтернет дозволяє відносно дешево і масово розміщувати рекламу, дає можливість потенційним та існуючим клієнтам легко отримувати актуальну інформацію про товари і послуги, ділові інтереси, що допоможе залучити нових замовників і партнерів по бізнесу - отже сприяє збільшенню обсягу продажу і рентабельності бізнесу.

Об'єктом дослідження даного дипломного проекту є процес електронної торгівлі. Мета даного проекту: розробка ресурсу E-Commerce для реалізації ювелірних виробів.

Основні завдання до проекту:

1) вивчення існуючого ринку послуг ювелірної майстерні, ознайомлення з провідними конкуруючими компаніями і виявлення її недоліків;
2) обґрунтування необхідності створення Інтернет-магазину;
3) проведення аналізу і вибір технологій для розробки Інтернет-магазину;
4) розробка динамічних WEB-сторінок;
5) оцінка ефективності проведення виконаної розробки і реалізації кінцевого програмного продукту.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ РЕСУРСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Значення електронної комерції у розвитку власної справи

E-Commerce або електронна комерція стосується здійснення ділових операцій онлайн. Комерційна діяльність здійснюється в електронному вигляді через глобальну мережу Інтернет. Як і в звичайному бізнесі, електронна комерція має всі аспекти бізнес-операцій, таких як купівля, продаж та платежі. Основна відмінність полягає в тому, що ця бізнес-модель базується на електронних транзакціях. У сфері електронної комерції компанії розміщують магазини в Інтернеті та надають користувальницькі інтерфейси, що дозволяють купувати та продавати товари. Фізичного контакту між продавцем та покупцем немає, оскільки покупки здійснюються в Інтернеті. Електронна комерція використовує технологію для задоволення потреб клієнтів та врегулювання транзакцій. У цій бізнес-моделі підприємцю не потрібно мати фізичну передумову, лише місце для зберігання товарів.

Можливі такі варіації електронної комерції:

- *Бізнес для бізнесу* (B2B): Цей тип стосується підприємств, що торгують один з одним - наприклад, виробники, що продають своїм дистриб'юторам, або оптові продавці, які здійснюють роздрібну торгівлю. Цей вид послуг має вирішальний вплив на життя будь-якого бізнесу. Бізнес залежить не тільки від компанії, яка надає потрібні їм продукти, але й покладаються на цю послугу, щоб підтримувати свій бізнес і працювати. Це вимагає швидких термінів доставки та простоти при покупці в Інтернеті.
- *Бізнес-споживач* (B2C): це картина, яку уявляють усі, думаючи про електронну комерцію. Модель системи працює з широкою публікою за допомогою електронних кошиків для покупок, без будь-якої взаємодії з людьми. На сайтах є дуже популярний речі, які багато людей

використовують регулярно. Саме з цієї платформи електронна комерція розпочала своє життя.

- *Споживач-Споживач* (C2C): За допомогою цього типу електронної комерції споживачі використовують онлайн-оголошення на різних форумах та спеціальних сайтах-площадках для купівлі та продажу своїх особистих товарів. Старомодний спосіб продажу вживаних автомобілів, холодильників, велосипедів та іншого нового і вживаного майна - став електронним за допомогою цього конкретного способу електронної торгівлі. Незалежно від того, чи є у вас один предмет для продажу, або склад, наповнений ними, ви можете використовувати електронну комерцію, щоб повідомити, здійснити транзакцію і навіть організувати доставку, не бачачи своїх клієнтів віч-на-віч. Приклад: сайти аукціонів, сайти роботи, сайти соціальних мереж тощо.
- *Споживач для бізнесу* (C2B): Споживач може публікувати особистий проект в Інтернеті, а компанії, які зацікавлені в ньому, можуть подати заявку на роботу. Це чудова платформа для багатьох позаштатних працівників, які не пов'язані ні з яким конкретним бізнесом. Художники, письменники, музиканти та інші форми творчого мистецтва процвітали на цьому виді послуг, що робить його дуже популярним у сучасному діловому світі.

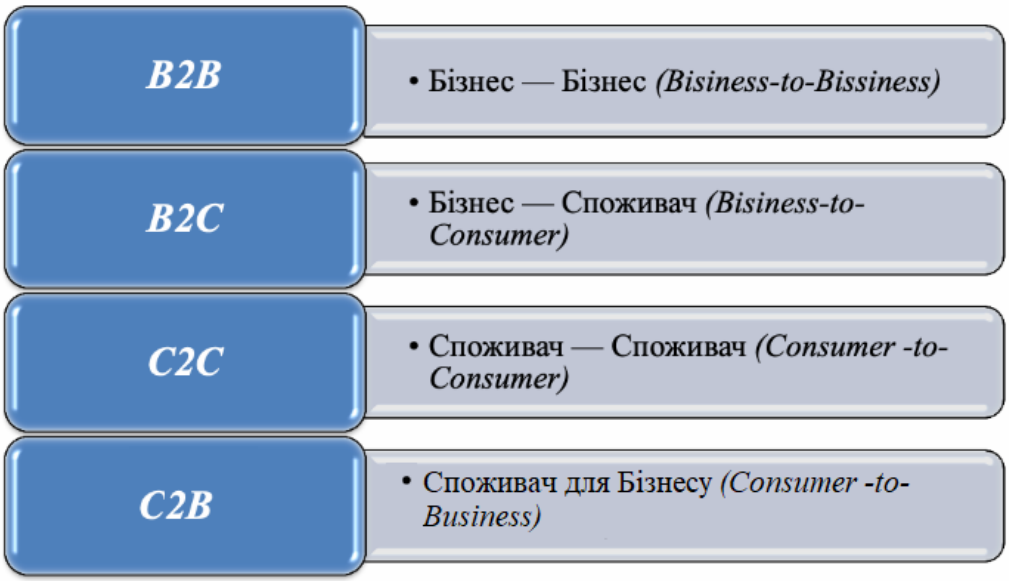


Рисунок 1.1 – Основні бізнес моделі електронної комерції

Залежно від того, яким типом електронної комерції займається людина – потрібно будувати унікальну стратегію. Будь-який бізнес, який шукає своє застосування в Інтернеті, повинен мати власний план розвитку. Від встановлення перших цілей, відкриття рахунків, закупівлі речей, проведення перших транзакцій, до кінцевої доставки продукту клієнту. Кожен етап повинен бути продуманий і змодельований дуже чітко і зрозуміло.

Багато традиційних способів придбання речей стають застарілими через зручність та швидкість онлайн-транзакцій. Навіть утверджені роздрібні продавці, які реалізовують свою продукцію однаково протягом десятиліть, з часом змушені пропонувати свою продукцію в Інтернеті, аби конкурувати з новими роздрібними сайтами. Інтернет швидко перетворюється на головну магістраль бізнесу у світі і надалі буде зростати та впливати на кожного все більше і більше в майбутньому.

1.2 Принципи побудови E-Commerce магазинів

Прогнози відомих бізнес-аналітиків демонструють стрімкий ріст електронної комерції у всьому світі на теперішній рік. Вибираючи бізнес-модель, завжди важливо враховувати світові тенденції та рухатися на ринок зростання, у разі, якщо це можливо. Статистика підтверджує те, що модель електронної комерції - хороший вибір для підприємців, які прагнуть створити нову компанію, або тих, хто зацікавлений у диверсифікації та розширенні власного штабу.

Для підприємців головне значення має фізичне розташування компанії. Часто цей фактор може слугувати розвитком величезної різниці в плані перспектив та реальних продажів. В електронній комерції ж цілком можливо налаштувати WEB-версію магазину під власним доменним іменем і реалізовувати власну продукцію у будь-якому закутку планети. Не потрібно кілька фізичних вітрин, щоб їх можна було побачити у всьому світі, адже кожна

людина має вільний доступ до Інтернету, і ніщо не заважає відкрити веб-сайт і переглянути доступний асортимент. Навіть дедалі частіше стає поширеним явище, коли підприємці-початківці повністю відмовляються від всім звичного фізичного магазину, і переводять всіх працівників на віддалену роботу. Це слугує значною економією коштів як за оренду складських приміщень, так і дозволяє суттєво скоротити кількість потрібних працівників. Тобто виручений бюджет можливо знову і знову вкладати у розвиток власної справи.

Це все здається досить очевидним, але наслідки є більш далекосяжними, ніж можливо уявити. Тенденція E-Commerce магазинів постійно зростає. Власники можуть інформувати про свій бізнес у парках, на автобусних зупинках і станціях метро, або просто на куточках вулиць із потоком людей та високою видимістю - без порушення правил, законів держави. Ну і звичайно, як можна не згадати популярний вислів «заробляти під час сну», що є результатом побудови продуктивної системи, правильної реклами та гнучкої логістики в компанії. З часом інструменти електронної комерції стають дедалі кращими та інноваційними. Від продажів платформ до автоматизації маркетингу та інструментів SEO.

Ключовим фактором для власника бізнесу в галузі електронної комерції є встановлення довіри до продукції у теперішніх та потенційних клієнтів. Від розробки привабливої і легкої навігації веб-сайту до створення унікального та корисного контенту, що допоможе клієнтам. Існує багато різних аспектів для заручення довірою користувачів яким потрібно постійне вдосконалення і вчасно побудована лояльність. Також важливо зазначити те, більшість користувачів довіряють системам, що надають важливу роль питанням безпеки та конфіденційності їхніх даних.

Розглянемо основні переваги електронної комерції над традиційним веденням бізнесу:

1. **Підвищує зручність**: Клієнти можуть здійснювати замовлення товарів за власним бажанням та зі своїх домівок без необхідності подорожувати до

приміщення магазину. Замовлення також доставляються їм у найбільш зручні місцях. Це найкращий варіант покупок для людей, які постійно зайняті. Користувачам стає простіше користуватися будь-де та в будь-який вільний час за допомогою своїх мобільних гаджетів.

2. **Дозволяє порівнювати характеристики товарів та ціни**: знову ж таки, здійснюючи покупку, клієнт очікує отримати найкращу пропозицію. Ця бізнес-модель дозволяє порівняти товари та ціни користувачами, щоб найкращі товари купувались за найвигіднішими цінами. Вони також можуть отримати додаткові переваги та заохочення, такі як знижки, промо-акції, купони на купівлю товарів, а також отримувати найкращі пропозиції в числі перших клієнтів.
3. **Легкий збір коштів для підприємств-новачків:** у багатьох людей виникає бажання зайнятися власним бізнесом, але не вистачає достатніх коштів для реалізації. Оренда фізичного магазину є досить дорогим задоволенням. Електронна комерція полегшує життя новому бізнесу та дає змогу рости і розвиватися.
4. **Ефективність**: електронна комерція має перевагу в ефективності. Всі доступні ресурси задіються максимально ефективно, оскільки більшість бізнес-послуг автоматизовані. Власники звичайного бізнесу нерідко витрачають дуже багато часу і ресурсів на задоволення потреб бізнесу - це значно зменшує фінальний прибуток. Електронна комерція ж процвітає в ефективності.
5. **Доступ до клієнтів:** для багатьох клієнтів простіше знайти Інтернет. Використовуючи таргетовану рекламу в популярних соціальних мережах та хорошій стратегії оптимізації пошукових систем та SEO аудиту - інтернет-бізнес може підвищити пізнаваність бренду та збільшити його клієнтську базу. E-Commerce також має перевагу в тому, що може об’єднувати покупців і продавців з будь-якого куточка земної кулі.

6. **Оперативні безготівкові платежі:** у разі, якщо інтернет-магазини використовують електронні або мобільні способи транзакцій – платежі проходять надзвичайно швидко і безпечно для обох сторін. Система Інтернет транзакцій сприяє значному збільшенню кількості продажів та як результат - збільшення реального доходу для власників бізнесу.
7. **Різноманітність товарів для продажу:** Гнучкість ведення бізнесу в Інтернеті дає можливість підприємцям представляти та реалізовувати будь-які типу товарів, а також задовольняти потреби широких демографічних груп.
8. **Низький запас:** Інтернет-продавці не повинні підтримувати високі запаси. Як тільки вони отримають замовлення, можуть придбати товар у виробника або певного посередника для задоволення запиту.
9. **Мінімізація загальної вартості:** Інтернет-бізнес усуває безліч фізичних процесів, необхідних для ведення традиційного бізнесу. Краще планування ресурсів та потреба у меншій кількості персоналу, тощо - допомагають знизити загальну вартість.

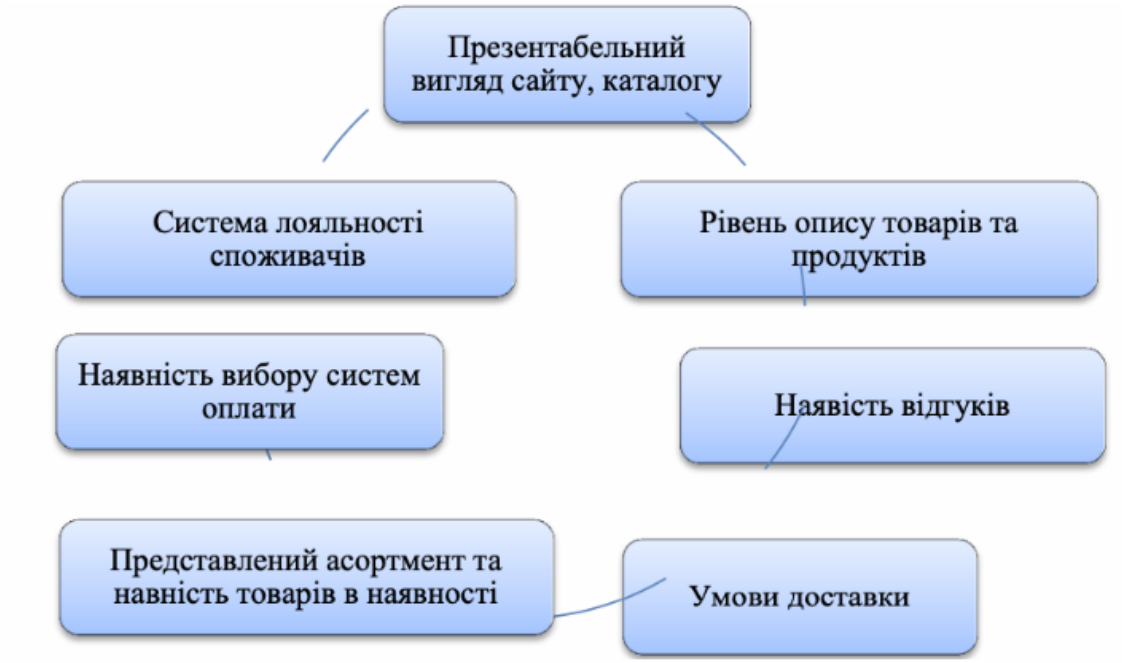


Рисунок 1.2 – Основні критерії вибору інтернет-магазину

У разі, якби модель електронної комерції була настільки ідеальною як її представляють – традиційний бізнес вже давно б зник. Проте це не так, а отже існують певні аспекти, що стримують глобальну популяризацію, і справедливо буде про них також розповісти. Серед основних недоліків можна виділити:

1. Погана якість продукції: Покупець фізично не бачить та не перевіряє те, за що платить, перш ніж отримає замовлений товар поштою, або ж кур'єром. Таким чином, клієнти ризикують стати жертвою поганого маркетингу та придбати неякісну продукцію у віртуального магазину. Звичайно, як правило кошти можна повернути при отриманні, якщо товар не задовольняє очікування клієнта, проте найцінніше в XXI столітті – це згаяний час.

2. Імпульсивні та необдумані покупки: Інтернет-магазини демонструють велику різноманітність товарів, і завдяки простоті покупок клієнти можуть прийняти погані фінансові рішення для себе через надмірну імпульсивність або нестачу вільного часу.

3. Інтернет-шахраї: Інтернет – це дуже корисна річ, проте деякі недобросовісні люди вирішують рухатися не в правильний бік. Шахраї зробили такий тип бізнес-моделі непривабливим для деяких споживачів. Як правило від цього страждають тільки нові бізнеси, які ще не мають власної репутації на Інтернет-ринку.

4. Відсутність підтримки після продажу: Через брак фізичних приміщень клієнтам важко отримати технічну підтримку після купівлі. Перш ніж будь-яка допомога буде надана клієнту, який її потребує, може пройти період від декількох днів до місяців.

5. Швидка зміна бізнес-середовище: Технології розвивається надзвичайно швидко. Деяким підприємцям дуже важко не відставати від тенденцій і вони змушені втрачати дуже багато власних фінансів. Це може призвести до неспроможності зростання бізнесу.

6. Втрата соціального зв’язку: Звичайний бізнес ґрунтується на взаєминах персоналу з клієнтами. Бізнес-модель електронної комерції розмиває цей зв’язок.

7. Доставка товару: Для доставки замовлених товарів потрібен час. Іноді доставка затримується, що не до вподоби більшості клієнтам. Це відрізняє E-Commerce від фізичних приміщень для бізнесу, де клієнти виходять вже із придбаними речами. Крім того, багато критиків електронної комерції стверджують, що такий спосіб купівлі та продажу загрожує життєвим ресурсам традиційних продавців на ринку та власників магазинів, які вважають кращим надавати послуги віч-на-віч.

Електронна комерція передбачає нові електронні технології, такі як:

- Інтернет-банкінг;
- мобільний банкінг;
- інтернет-магазини;
- автоматизована обробка транзакцій;
- Інтернет-реклама та SEO;
- онлайн-системи управління базами даних;
- планування ресурсів підприємства через Інтернет.

1.3. Аналіз основних аспектів створення сервісів E-Commerce

За недавній період, Інтернет став одним з найбільших і найважливіших «мастодонтів» серед засобів маркетингу, що має значні переваги порівняно із традиційними каналами маркетингу. Проте можливості, які були надані глобальною мережею, настільки нові й чужорідні, що більшість фірм не мають уявлення, як вірно використовувати весь її надзвичайно потужний функціонал. Одним з головних аспектів, що потребує маркетингового підходу – являє собою проектування і розробка WEB-сайту.

Щоб розглянути основні маркетингові принципи для реалізації WEB-сайту, потрібно розглянути план їхнього створення:

I. Визначитися з метою і цілями створення WEB-сайту;
II. Визначити головний цільовий сегмент клієнтів;
III. Створення контенту і медіа-наповнення для WEB-сайту;
IV. Створення унікального дизайну для WEB-сайту;
V. Оцінити ефективність роботи створеного WEB-сайту;

Надзвичайно важливу ролі у реалізації кожного веб-сайту відіграє хостингова компанія, що надає послугу розміщення. Зазвичай, якщо власник хоче зекономити кошти і заключає контракт із дешевим хостингом – в майбутньому це додає дуже багато проблем. Низька пропускна здатність з'єднання, повільні серверні накопичувачі, не достача місця на диску, часті технічні збої, погана серверна підтримка тощо – все це поставляється в комплекті із більшістю дешевими веб-хостингами. Кожен цей негативний фактор буде слугувати відторгненням великої долі користувачів. Тому більша інвестиція в хороший і надійний хостинг обов'язково окупиться, і вбереже від більшості проблем.

Також дуже важливо приділити час вірно підібраному доменному імені. *Доменне ім'я* – це домашня адреса веб-сайту в Інтернеті. І хоча це не єдине , що впливає на успіх сайту, є кілька причин, чому важливо ретельно його вибирати:

- Унікальне ім'я, що легко запам'ятовується, швидше за все, залишиться в пам'яті відвідувачів і заохотить їх повернутися знову, а також підвищить пізнаваність бренду.
- Доменне ім'я може одразу дати новим відвідувачам (або користувачам пошукових систем) уявлення про те, про контент на сайті.
- Доменні імена впливають на оптимізацію пошукових систем (SEO).
- Зміна власного доменного імені після створення веб-сайту може призвести до трудомістких ускладнень.

Згідно з дослідженнями *DomainNameStat*, 41% усіх доменів мають розширення "**.com**". Причина цьому банальна - "**.com**" найвідоміший і найлегше запам'ятовуваний. Хоча існує безліч успішних веб-сайтів "**.net**" та "**.org**", буде краще, якщо веб-сайт матиме розширення "**.com**". Це найбезпечніша ставка. Нижче наводжу статистику використання розширень доменних імен (рисунок 1.3).

Загалом, якщо мова йде про довжину доменного імені, то краще використовувати короткі і прості слова. Згідно з дослідженнями сервісу *DataGenetics* - найпоширеніша довжина імені становить приблизно 12 символів. Ці дані показують, що ви повинна зберігатися лаконічність. Потрібно орієнтуватися на 6-14 символів. Швидше за все, більш короткі доменні імена зайняті довгий час тому і продаються за тисячі доларів. Якщо ж не можливо підібрати щось коротке - зробіть його фірмовим. А перш ніж рухатися вперед із певним доменним іменем, потрібно обов'язково перевірити, чи випадково не є це використовується це ім'я на сайтах соціальних мереж, а також чи не існують якісь торгові марки, зареєстровані на це ім'я.

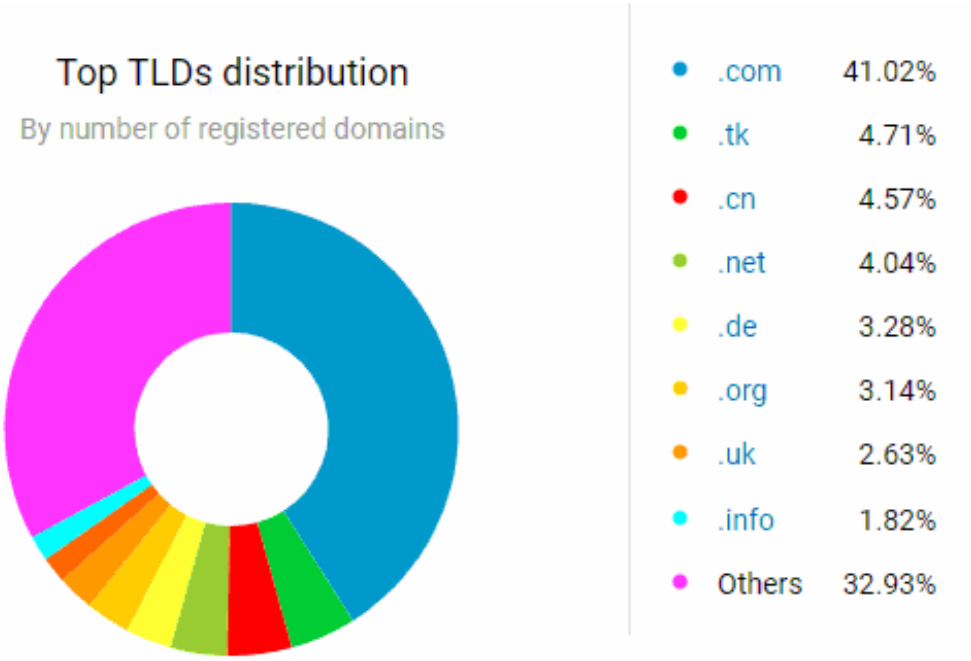


Рисунок 1.3 - Статистика використання розширень доменних імен

Після публікації WEB-сайту, потрібно постійно оновлювати і доповнювати представлену інформацію. Важливе значення безсуперечно відіграє активна реклама сервісу, яку буде корисно розміщувати на так званих «площадках найбільшого скупчення людей». До цього можна віднести персоналізовані оголошення у відомих соціальних мережах, рекламні інтеграції у відео-роликах відомих блогерів на популярному сервісі YouTube, рекламні оголошення у пошукових системах (Google Ads), тощо.

1.4. Забезпечення інформаційною підтримкою існуючого бізнесу

Відсутність сайту у компанії тепер сприймається приблизно так, як відсутність візитки і рекламних матеріалів у менеджера з продажу товарів, що йде на зустріч зі своїм клієнтом. Залежно від масштабів і успішності компанії у них більше розгорнутий і естетично розроблений сайт. Це вже стало звичкою у багатьох компаній і людей.

Завдання інформаційної підтримки бізнесу є розробка діючого рекламно-інформаційного каналу зв’язку, що значно полегшує зв’язок із вже існуючими та потенційними бізнес-партнерами, створення хорошого обліку у своєї компанії чи підприємства, що активно залучує та використовує інформаційні технології. Концептуально, кожен Інтернет-магазину повинен містити актуальну інформацію про компанію, її список товарів, прайс-лист, послуги, сервіс, отримані ліцензії і нагороди та інше.

Наразі, Інтернет-складова кожного бізнесу прямо несе у собі комерційний сенс: розвантаження персоналу компанії, що веде роботу з клієнтами (скорочення витрат на телефонний диспетчерську службу, відправку факсів, телефонні переговори і так далі). За рахунок розміщення актуальних цін на послуги та іншої актуальної інформації на сайті компанії.

1.5. Організація двохсторонньої взаємодії із клієнтом

Інтерактивний WEB-сайт - це Інтернет-сторінка, що задіює різноманітне програмне забезпечення для створення інтерактивного досвіду та змушує людину, що переглядає веб-сторінку, активно взаємодіяти із сервісом. Інтерактивний веб-сайт може слугувати розширенням майже будь-якого звичайного веб-сайту. Головна мета ж цього - дозволити користувачам виходити за рамки простого читання тексту та перегляду зображень. Натомість цей тип веб-сайтів також може дозволити глядачеві змінити спосіб відображення веб-сайту, або дозволити користувачеві грати в ігри та виконувати різні цікаві завдання. Загалом, інтерактивний веб-сайт використовує стандартний графічний інтерфейс користувача (GUI), присутній на всіх веб-сайтах, для створення інтерактивності з клієнтом.

Залучення клієнтів і відвідувачів сайту взаємодії з контентом - зручний інструмент маркетингу. Якщо від багатьох користувачів, що вже купили товар, компанії вдається отримати конструктивні коментарі та реальний фідбек після користування, набагато простіше знаходити власні недоліки та мотивувати в подальшому ще більше нових відвідувачів до покупки. Адже нові клієнти завжди матимуть змогу переконатися в достовірності наданої інформації, завдяки коментарям минулих покупців.

Для організації зв’язку компанії з клієнтами, клієнтів один з одним - на сайті зазвичай створюється форум або книга відгуків (спеціальний розділ, де будь-який відвідувач може залишити свою скаргу або хороший відгук про товар, відповісти на питання інших, брати участь в певному обговоренні). Адміністрація сайту зазвичай активно залучена в дискусії, завжди відповідає на адресовані прямі питання і коментує висловлювання клієнтів. Таким чином, сайт формує у відвідувача певну думку, що компанія дбає про свій імідж та інтереси клієнтів, займає активну і стійку позицію на ринку.

Наявність на сайті системи зворотного зв'язку (feedback) дозволяє відвідувачам відправляти свої повідомлення адміністратору сайту. Для цього їм потрібно всього лише заповнити кілька полів, після чого повідомлення автоматично надсилається на E-mail адміністратора прямо зі сторінки сайту.

Також дуже поширена практика онлайн опитувань. Організувати онлайн опитування можна за допомогою розміщення на сторінках сайту спеціальних форм-анкет з питаннями і варіантами відповідей на них. Завдяки проведенню он-лайн опитувань можна дізнатися у клієнтів їхню думку про нововведення, товарних перевагах і ставлення до компанії.

1.6. Безпека та конфіденційність даних

Досить часто ми чуємо із ЗМІ про те, що доступ до того чи іншого WEB-сайту недоступний, через відмову системи чи обладнання після направлених хакерських атак. Однак це стається відносно рідко, і зловмисники мають зовсім не корисний мотив наживи, а скоріше – показати людям свою силу, чи показати негативне відношення до певних речей (Anonumos). Частіше зустрічаються випадки, коли контент на домашній сторінці був модифікований (як правило негативно). Проте більшість гучних випадків все ж пов’язані із викраденням конфіденційної інформації користувачів – так званий «злив бази». Під час цього процесу бази даних із мільйонами паролів, електронними адресами та реквізитами банківських карт – потрапляють на підпільні ринки в Інтернеті (DarkNet), де їх зможе купити кожен бажаючий за криптовалюту. Це все додає користувачам веб-сайтів як особистих збентежень, так і фінансових ризиків.

Зважаючи на дослідження компанії ESET, що займається розробкою антивірусного програмного забезпечення для багатьох платформ – близько 68% всіх користувачів мають хоча б кілька однакових паролів у різних системах, сайтах та форумах. Це означає те, що у разі отримання незаконного доступу до

бази даних сайту – зловмисник із 68-ми відсотковою вірогідністю також отримає доступ і до інших ваших конфіденційних даних, документів та медіа.

Безпека веб-сайту це дуже важливий аспект. Наявність безпечного веб-сайту так же само важливо, як і розміщення на хостингу. Якщо, наприклад, веб-сайт зламаний і перебуває у чорному списку, він втрачає до 98% свого трафіку. Мати не захищений сайт настільки ж погано, як і взагалі не мати веб-сайту, або навіть гірше. Наприклад, порушення конфіденційності даних клієнтів може призвести до безмежної кількості судових позовів, великих штрафів та повністю зруйнованої репутації у компанії.

Також ще одним популярним способом кіберзлочинності є DDoS-атаки. DDoS Attack - одна з найвідоміших і страхітливих для пересічної людини. Застосовується для виведенню з робочого стану серверу, на якому розміщений сайт. Здійснюється це за рахунок заливання серверів трафіком, який перевищує, те що здатні опрацювати сервери або пропускна здатність. Хакери націлюються на сайти банків, шлюзи для платежів, корпоративні та особисті веб-сайти для майбутньої вигоди, помсти, або шантажу. Часто зустрічається така практика, коли власних замовляє атаку на сервіс конкурента, і в той час коли клієнти не можуть отримати доступ до сайту – йде залучення нової аудиторії до себе. Це поганий, але й надзвичайно дієвий спосіб агресивного маркетингу.

Існує перелік симптомів для DDoS-атак, але він не надто точний, тому що такі самі симптоми можуть з’являтися, якщо у веб-хоста є проблеми з обладнанням або Інтернетом(як правило - у дешевих хостингових компаній):

- Якщо веб-сайт страждає від незвично повільного завантаження сторінок або його функцій;
- Відсутнє з’єднання із сервером;
- незвично велика кількість отриманих електронних листів у техпідтримку;
- незвично велика кількості нових зареєстрованих користувачів у системі;
- велика кількість відгуків, тем на форумі;
- інша помітна спам-активність.

Одразу постає питання, як правильно захищати свій веб-сайт від атак? Способи звичайно існують , і їх не мало. Ось деякі з них:

1) Використання сертифікату SSL та HTTPS протоколу передачі пакетів даних - це найпростіший спосіб захисту, де всі користувацькі дані передаються на сервер у зашифрованому вигляді. І отримавши їх, зловмисник не зможе розшифрувати, або йому треба буде надто великі затрати часу і ресурсів для цього, що не вигідно.
2) Потрібно постійно оновлювати компоненти та програмне забезпечення сайту. Це здається очевидним, проте багато користувачів нехтує цим правилом. Стосується як операційної системи сервера, так і будь-якого програмного забезпечення, яке працює на веб-сайті, наприклад, CMS або форум. Коли в програмному забезпеченні знайдено дірки в безпеці, хакери якнайшвидше намагаються скористатися цим.
3) SQL-Inject вразливість існує, якщо введені зловмисником дані, що передаються до базового оператора SQL, можуть змінити значення виразу. Щоб уникнути подібної атаки, потрібно переконатися, що будь-які дані користувача, передані SQL-запиту, не можуть змінити дані. Один із способів - уникання всіх символів в користувальницьких даних, які мають особливе значення в SQL.
4) У атаці XSS зловмисник вводить на сторінку зловмисний JavaScript код, який потім запускається у браузерах всіх користувачів, і може змінювати вміст сторінки або викрадати (парсити) інформацію для свого розробника. Для захисту, при динамічному генеруванні HTML потрібно використовувати функції, які явно вносять потрібні зміни (наприклад, використовуйте element.setAttribute та element.textContent, який будуть автоматично екрановані браузером, а не встановлювати вручну element.innerHTML), або використовувати функції у шаблонному інструменті, який автоматично виконує відповідне екранування, а не об'єднання рядків чи встановлення неопрацьованого HTML-контенту. Ще один потужний інструмент захисту

від XSS-атаки - це політика безпеки вмісту (Content Security Policy). CSP - це заголовок, який сервер може повернути, що повідомляє браузеру, аби він обмежував запуск і виконання скриптів на сторінці. Наприклад, щоб заборонити запуск будь-яких сценаріїв, які не розміщені на хості – потрібно заборонити вбудований JavaScript або відключити функцію eval(). У Mozilla є відмінне керівництво з деякими прикладами конфігурацій. Це ускладнює роботу сценаріїв зловмисника, навіть якщо вони зможуть запустити їх на вашій сторінці.

5) Потрібно бути обережним з тим, яку інформацію повідомляє сайт в своїх повідомленнях про помилки. Користувачам потрібно знати тільки про мінімальні помилки, аби вони не відображали секретну інформацію, присутню на сервері (на кшталт ключів API або паролів від бази даних). Також не можна надавати повну інформацію про винятки, оскільки це значно спрощує складні атаки, такі як ін’єкція SQL. Зберігати докладні помилки потрібно в журналах сервера і показувати користувачам виключно ту інформацію, яку потрібно знати.
6) Вкрай важливо використовувати надійні паролі для сервера і адміністративних акаунтів веб-сайту. Але не менш важливо наполягати на правильному виборі паролів для ваших користувачів, щоб захистити також безпеку інших їхніх облікових записів. Як би це не подобалося користувачам - дотримання вимог створення паролів, таких як мінімум 8 символів, включаючи великий та малий регістр, цифри - допоможе захистити інформацію в довгостроковій перспективі.
7) Дозвіл користувачам завантажувати власні файли на веб-сайт може представляти серйозну загрозу безпеці, навіть якщо це просто зміна власного аватару. Існує високий ризик того, що будь-який завантажений файл, яким би невинним він не був, може бути склеєним з скриптом зловмисника. Після відвантаження на сервер це повністю відкриє доступ до компонентів сайту.

8) Використовуйте HTTPS протокол. Він може гарантувати те, що окрім користувачів ніхто інший не зможе перехопити або змінити контент, який вони бачать при передачі пакетів.
9) Регулярно тестуйте безпеку сайту. Найбільш ефективний спосіб - задіяння спеціалізованих інструментів(програми для тестуванням на проникнення або ручне тестування). Є безліч комерційних і безкоштовних продуктів, що цьому допоможуть. Вони працюють аналогічно хакерським сценаріям в тому, що перевіряють всі відомі експлойти і намагаються зламати ваш сайт, використовуючи деякі з раніше згаданих методів, таких як SQL-ін'єкція.
10) Для захисту від атак DDoS варто налаштувати захищений VPS хостинг. Багато підприємств вибирають найменші ціни хостингових планів, доступних на ринку. Налаштування захищеного хостингу VPS пропонує захист від DDoS та зменшує вірогідність нападу. Також можна використовувати маршрутизатори і комутатори з брандмауерами. Вони можуть зупинити прості атаки ping і заборонити доступ з недійсні IP адрес.

1.7. Аудит та Search Engine Optimization

Search Engine Optimization (SEO) - абревіатура, що розшифровується як оптимізація пошукових систем. Це процес оптимізації веб-сайту для отримання органічного або безкоштовного трафіку на сторінці результатів пошуку.

Метою оптимізації пошукової системи є те, щоб пошукові системи (Google, Bing, Yahoo, тощо) не тільки знаходили ваш сайт та сторінки, а й спеціально оцінювали релевантність сторінки, щоб вона відображалася у верхній частині результатів пошуку. Процес оптимізації не є разовим процесом, тому вимагає гідного технічного обслуговування, налаштування, постійного тестування та моніторингу.

Коли SEO-оптимізація виконана правильно, ви допомагаєте пошуковим системам легко сканувати, отримувати доступ, індексувати і інтерпретувати

ваш сайт. Ці стратегії не тільки вражають Google, але і виправдовують очікування користувачів.

На основі вивчення наукової літератури можна представити структурну схему пошукового маркетингу (рис. 1.4) та детально розглянути складові елементи зазначеного виду маркетингу віртуального простору.

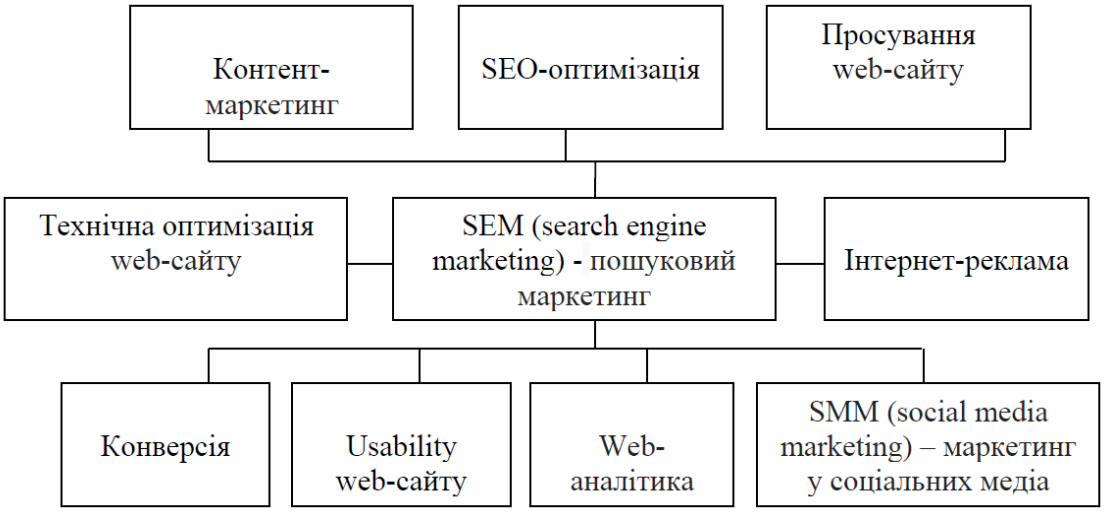


Рисунок 1.4 - Основні елементи пошукового маркетингу

Розглянемо виділені елементи більш детальніше.

Контент-маркетинг – процес створення та розповсюдження інформації, котра є цікавою для цільової аудиторії web-сайту компанії. Контент, що створюється на сайті має бути унікальним та релевантним.

SEO-оптимізація (search engine optimization) – комплекс заходів, котрі впроваджують інтернет-маркетологи для підняття позицій сайту у пошукових системах.

Як правило, це внутрішня робота над сайтом по приведенню контенту у зручний для користувачів вид та на основі вимог пошукових систем до інтернет-інформації, що розміщується на сайті.

Просування web-сайту являє активну фазу покупки посилань. Компанії можуть купувати тимчасові та/або довічні посилання на спеціальних біржах, де

відбираються найбільш релевантні сайти-продавці. В результаті таких дій пошукові системи будуть піднімати позиції певного сайту.

Інтернет-реклама може бути представлена у двох видах: пошукова та банерна (контекстна медійна мережа). Перший вид пов'язаний з розміщенням текстових блоків оголошень над та/або під результатами пошуку систем.

Контекстна реклама представляє процес розміщення та показ банерів на сайтах, де присутня цільова аудиторія.

Маркетинг у соціальних медіа (social media marketing) – комплекс заходів, спрямований на взаємодію з цільовою аудиторією соціальних мереж з метою залучення потенційних клієнтів.

Web-аналітика призначена для проведення діагностики всіх виконаних дій з приводу оптимізації та просування сайту компанії. Основними інструментами цієї складової пошукового маркетингу є Google Analytics. На основі результатів аналітики проводиться планування подальшої роботи з web-сайтом.

Usability web-сайту передбачає зручність використання інтернет-ресурсу. Головними цілями цього процесу є удосконалення сайту з приводу зручності та простоти користування, а також підвищення ступеня довіри цільової аудиторії.

Конверсія – це досягнення на сайті певної дії користувача, яку можна відстежити завдяки технічним засобам. Прикладом кінцевої конверсії може бути завершене оформлення замовлення товару чи послуги на сайті.

Технічна оптимізація web-сайту необхідна, по-перше, для усунення всіх технічних помилок сайту та, по-друге, для покращення взаємодії з роботами пошукових систем.

Перш ніж виконати технічну оптимізацію SEO, потрібно зрозуміти, яку користь потрібно від сайту, і де існують потенційні проблеми. Ось чому кожен без винятку повинен провести технічний аудит SEO. Після того, як він завершиться, можна скласти дієвий список завдань. Існує кілька технічних інструментів для аудиту SEO, такі як Screaming Frog SEO Spider і Google Search Console.

1) Покращення часу відгуку сервера на якому розміщений веб-сайт.

Час відгуку сервера визначається тим, скільки часу займає завантаження веб-сторінки. Google буде сканувати веб-сайт в пошуках неефективних кодів, які уповільнюють швидкість сторінки сайту. Рекомендується намагатися досягти часу відгуку сервера менше 200 мс. Існує кілька безкоштовних інструментів, які можна використовувати для визначення часу завантаження ваших веб-сторінок, таких як GTmetrix, Google PageSpeed Insights і Pingdom Speed Test. Також слід використовувати інструкцію з оптимізації швидкості сторінки, яка виділить кожен крок аудиту швидкості сторінки для заданого сайту і порекомендує правильне рішення.

Сторінки, що повільно завантажуються негативно впливають як на рейтинг сайту, так і на користувацький досвід. Це призводить до більш високих показників відмов, меншій кількості часу, що користувач проводить на сторінці і меншої кількості конверсій.

Нижче наведено кілька способів, як зменшити час відгуку серверу:

- Потрібно скоротити CSS, JavaScript і HTML: видалити зайві або непотрібні дані, такі як коми, пробіли, форматування, коментарі до коду та інші символи. Якщо використовується CMS WordPress - процес спрощує спеціальний плагін WP Rocket.
- Стиснення файлів: варто використовувати додаток, такий як Gzip, щоб зменшити розміри файлів JavaScript, CSS і HTML.
- Задіювання кешування в браузері: файли JavaScript, таблиці стилів CSS і зображення повинні кешуватися, щоб не доводилося перезавантажувати всю сторінку кожен раз, коли на неї потрапляє один і той же користувач.
- Мінімізація перенаправлень сторінок: коли веб-сторінка перенаправляє на іншу сторінку - відвідувач змушений чекати довше, щоб сторінка завантажилася.

- Варто оптимізовувати зображення на сторінці: Зменшення розміру файлів зображень, без помітного користувачеві зменшення якості за допомогою таких інструментів, як JPEG Optimizer, CompressNow і TinyPNG.

2) Створення і забезпечення мобільності сайту.

На даний час компанія Google віддає перевагу смартфонам і мобільним версіям сайтів. Насправді 58% усіх пошукових запитів в Google Search зараз здійснюються з мобільного пристрою. І ця тенденція швидко зростає. За даними Google, на 27,8 мільярдів запитів більше виконуються на мобільних, ніж на десктопах. Якщо ж сайт некоректно відображається на мобільних пристроях, або на ньому повністю відсутня підтримка мобільності - він не буде потрапляти в результати пошуку і в кінцевому підсумку буде дуже низький рейтинг. Варто скористатися Google-Friendly Test, щоб перевірити, чи відповідає сайт критеріям Google. Введіть свій URL в поле пошуку і натисніть «Перевірити URL». Інструмент проаналізує цю сторінку і надає результати тесту. Якщо результати не є оптимальним, потрібно прочитати негайно зайнятися цим питанням, що значно поліпшити свій рейтинг.

3) Оптимізація файлу Robots.txt.

Файл robots.txt знаходиться в кореневому каталозі кожного сайту. Він повідомляє пошуковим системам, які сторінки вашого сайту сканувати і додавати в індексацію. Правильно відформатований Robots.txt буде блокувати тільки ті каталоги й файли, які не потрібно сканувати пошуковим системам. Варто також включити файл robots.txt у власну карту сайту.

4) Використання SEO-friendly URLs.

URL-адреси, зручні для пошукових систем - містять ключові слова, пов'язані з темою сторінки. Потрібно уникати використання підкреслень, дефісів, пробілів та інших спеціальних символів. URL адреси сайту повинні бути логічними і організованими. Слова, які краще читаються повинні переважати над ідентифікаторами, адже вони представляють користувачу те, що він очікує побачити на сторінці. Щоб переконатися, в вірній оптимізації

URL-адреси для SEO варто час-від-часу проводити тестування, що надають спеціалізовані сервіси, такі як SEO Friendly URL Test.

5) Оптимізація внутрішніх посилань.

Внутрішні посилання допомагають пошуковим системам зрозуміти архітектуру сайту. Вони пов'язують доступний контент за допомогою гіперпосилань, що вказують з однієї сторінки на іншу. Google переглядає посилання, щоб визначити пов'язаний контент і його цінність. Використовуючи правильну стратегію внутрішніх посилань, можливо створити ієрархію постів та сторінок, що віддають пріоритет більш важливим.

Правильне проектування сайту із вірною оптимізацією внутрішніх посилань допомагає сайту зайняти більш високе місце в рейтингу пошуку. Нижче наведені деякі рекомендації, за якими варто слідувати:

- Використовуйте фрази із двох або трьох слів;
- Уникайте наповнення ключовими словами;
- Використовуйте комбінації точного, або часткового збігу;
- Використовуйте посилання на важливі сторінки;
- Не використовуйте однаковий якірний текст, більше ніж на одній сторінці.

Одним із способів аудиту та оптимізації внутрішніх посилань є використання Google Search Console. Сторінки з найвищим пріоритетом повинні отримувати кореневі посилання.

6) Виправити неправильні або працюючі посилання та сторінки.

Користувацький досвід грає життєво важливу роль в тому, яке місце займе сайт в рейтингу пошукових систем. Якщо на одній із сторінок відображається помилка 404 – це значить, що власник посилається на неіснуючу сторінку. Google не може сканувати неробочі сторінки, а користувачам не подобається їх відвідувати. Часто буває, що посилання не працює, після видалення або скасування публікації сторінки, на яку посилаються. Тому перед видаленням

сторінки потрібно налаштувати перенаправлення 301. В такому випадку користувачі перейдуть на нову сторінку і не побачать помилку 404.

Якщо використовується CMS WordPress – існує кілька важливих WordPress SEO плагінів, які допомагають значно швидше виконати задачу з аудиту. Або можна використовувати все ту ж Google Search Console. Увійдіть до свого облікового запису та натисніть «Coverage». Якщо на сторінці є помилки - програма зробить акцент на них та виділить червоним кольором. Як виявляться непрацюючі посилання - варто їх одразу виправити. Після цього потрібно надіслати нову карту сайту в Google.

Хороша технічна оптимізація SEO не проходить одразу, стає результатом довгої, нудної і клопіткої роботи. Проте, як тільки виправляються технічні проблеми на сайті – це значно підвищує ефективність відображення у пошукових системах, і, як наслідок, буде відбуватися постійне залучення нових клієнтів.

2. АНАЛІЗ ТА ПРОЕКТУВАННЯ СЕРВІСУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ДЛЯ РЕАЛІЗАЦІЇ ЮВЕЛІРНИХ ВИРОБІВ

2.1. Аналіз принципів роботи ресурсу електронної комерції

E-Commerce або електронна комерція (рисунок 2.1) – це методологія сучасного бізнесу, яка враховує потреби ділових організацій, постачальників та клієнтів у зниженні затрат, підвищенні якості товарів та послуг і одночасному підвищенні швидкості доставки. Електронна торгівля стосується безпаперового обміну ділової інформації, використовуючи наступні засоби:

- Електронний обмін даними (EDI);
- Електронна пошта (E-Mail);
- Електронні дошки оголошень;
- Електронний трансфер коштів(EFT);
- Інші мережеві технології.

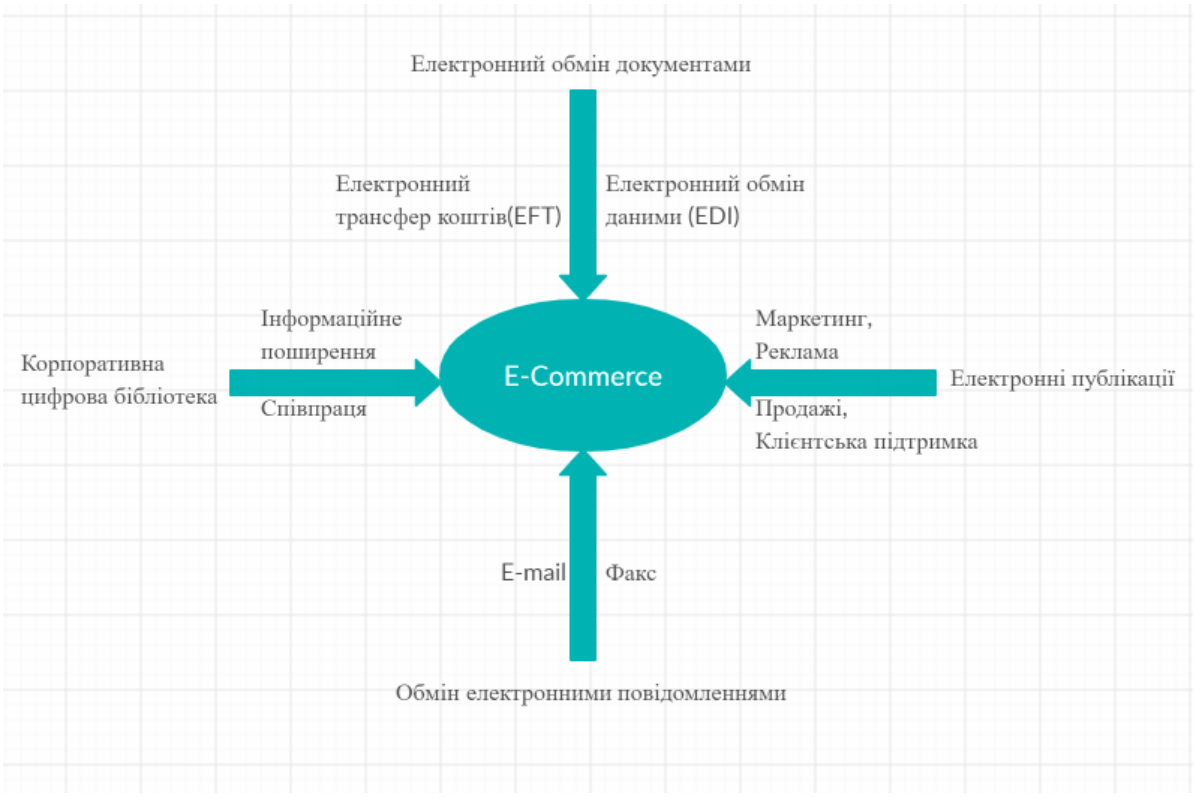


Рисунок 2.1 – Бізнес-модель типового Інтернет магазину

В даному випадку мною розроблялося програмне рішення для бізнес-моделі Business-to-Consumer (від бізнесу для клієнтів). E-Commerce магазин ювелірної майстерні, слідуючи бізнес-моделі B2C, продає свою власну продукцію безпосередньо клієнту. Клієнт може переглядати всі доступні товари, представлені на сайті. У разі, якщо клієнт обрав відповідний товар - одразу може замовити його. Згодом веб-сайт надсилає лист менеджеру ювелірної майстерні електронною поштою, і компанію відправляє продукт товар клієнту у найближчий доступний час (рисунок 2.2).

У моделі B2C бізнес-сайт - це місце, де всі транзакції відбуваються безпосередньо між бізнес-організацією і споживачем. Споживач самостійно відвідує веб-сайт, вибирає каталог з відповідним товаром, замовляє конкретний вибрану річ і відправляє електронного листа бізнес-організації зі своїми контактними даними та реквізитами оплати послуг. Після отримання даних про нове замовлення – воно реалізується і товар відправляється покупцю.

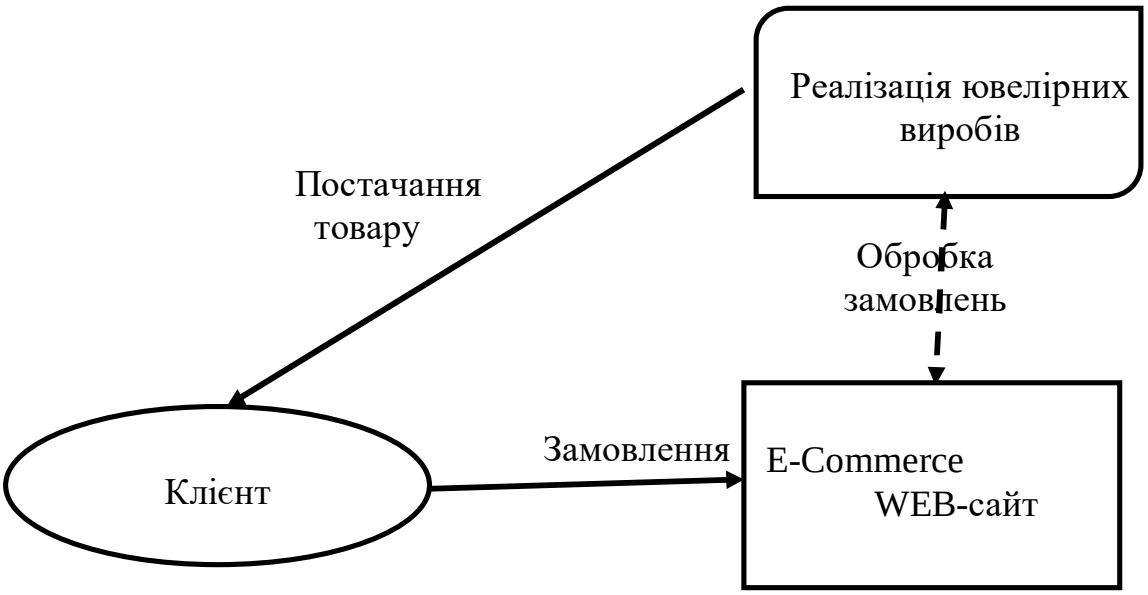


Рисунок 2.2 - Бізнес-модель ресурсу електронної комерції

Нижче наведені ключові особливості даної моделі:

1) Для залучення нових клієнтів потрібна важка реклама;

2) Високі інвестиції з точки зору апаратного і програмного забезпечення системи;
3) Швидка підтримка і відмінне обслуговування клієнтів.

2.2. Вимоги до технології розробки сервісу електронної комерції

Магазин реалізації ювелірних виробів повинен бути створений з додаванням системи управління контентом E-Commerce, а також бути оснащеним обов'язковими та допоміжними (за побажанням власника з майбутньому) системними модулями. Так як зазвичай малі магазини використовують вільне програмне забезпечення - було прийнято рішення розробляти програмний продукт із використанням технології Node.JS та безкоштовної бібліотеки JavaScript з відкритим кодом Polymer 3.0.

Бібліотека Polymer використовується багатьма відомими техно-гігантами IT індустрії (Google, GitHub, Netflix, IBM, Electronic Arts та багатьма іншими) для створення додатків із використанням веб-компонентів. Її складові можливості (багато в чому аналогічні популярним фреймворкам React.JS та Vue.JS) реалізовані не за допомогою мета-платформи і JS-абстракцій поверх звичайного DOM(Document Object Model), а на рівні самого браузера, що відкриває ряд дійсно чудових можливостей і породжує широкий спектр нових підходів в програмуванні. Наприклад, практично зникають кордони між SPA (Single Page Application) та класичною веб-сторінкою, що дозволяє суттєво полегшити роботу із середовищем розробки і значно підвищується універсальність програмних рішень: можливо використовувати власну UI-бібліотеку і реалізувати свою дизайн-систему, задіючи практично будь-який популярний фреймворк або бібліотеку, без необхідності серйозної адаптації.

Сайту інтернет-магазину повинен бути встановлений на віддалений хостинг певного провайдера із помісячною оплатою та хорошим технічним забезпеченням компонентів сервера.

Адміністрування складами товарів, бухгалтерський та податковий облік веб-магазину повинно здійснювати відділом бек-офісу, що не буде доступно звичайному користувачу системи.

2.3. Процедура здійснення користувацьких покупок

Існують конкретно визначені і регламентовані шляхи, що будуть використовуватися в електронній комерції нашого магазину. Завдання клієнта:

- Визначитися, що потрібно від даного сервісу;
- Виконати пошук доступних елементів у веб-магазині, що повністю задовільнятимуть його потреби;
- Порівняти схожі товари за такими критеріями, як ціна, тип доставки та інше;
- Оформлення замовлення;
- Проведення оплати товару;
- Отримання замовлення і перевірка його цілісності;
- Онлайн-консультація з персоналом Ювелірної майстерні, з метою отримання детальної інструкції по обслуговуванню, або з питань повернення продукту, якщо він не відповідає заданим раніше на сайті характеристикам (пошкодження при доставці).

2.3.1 Оформлення товару

Обравши потрібний товар, користувач отримує можливість відразу обрати тип доставки та метод оплати продукту. Всі товари разом, спосіб оплати та тип доставки – це фінальне замовлення, що формується на веб-сайті, до якого також додається певна інформація про клієнта. Дані клієнта будуть додані у клієнтську базу даних ювелірної майстерні, що повинно значно спростити майбутні покупки покупцям. Проте, судячи з даних масового опитування сервісу Bing – обов’язкова реєстрація при покупці часто відбиває бажання

потенційним клієнтам здійснювати свої покупки. В веб-магазині також буде вестися історія купівель та відшкодування клієнтами. І у разі якщо у покупця буде погана статистика – за магазином залишатиметься право обирати, дозволити чи відмовити в подальшому обслуговуванні.

2.3.2 Проведення оплати замовлення

E-Commerce сайт ювелірної майстерні оперує електронними платежами, де електронний платіж відноситься до безготівкових грошових транзакцій. Електронні платежі зробили революцію в бізнес-процесах, скоротивши витрати на оформлення документів, на проведення транзакцій і трудовитрати. Будучи дружнім до користувача і менш трудомістким, ніж фізична обробка - це допомагає бізнес-організації розширювати охоплення ринку. Нижче перераховані деякі розповсюджені типи електронних платежів:

- Транзакція з кредитної картки;
- Картка з електронним чіпом (NFC);
- Електронний переказ коштів зі сторонніх гаманців(EFT);
- Криптовалютні транзакції (Bitcoin, Litecoin, Ethereum, тощо)

Оплата, проведена за рахунок кредитної та дебетової картки - є найпоширенішим способом електронних платежів. Кредитна карта - це невелика пластикова карта з унікальним ідентифікатором, прикріпленим до облікового запису клієнту банку. У неї вбудована магнітна смуга, яка використовується для зчитування інформації, що записана на картку, через кард-рідери. У разі, коли клієнт купує продукт за допомогою кредитної карти, банк-емітент кредитної картки ініціалізує оплату від імені власника картки. Для клієнта буде відведений певний період часу, за який можливо підтвердити оплату, спеціальним способом (PIN, 3D-Secure, SMS, дзвінок з банку, тощо). Якщо процес оплати не вдалося завершити – власнику карти обов’язково повідомляють про невдачу. Це зазвичай цикл оплати кредитною карткою.

Нижче приведені актори в системі транзакцій за кредитними картками:

- Власник карти (card holder);
- Покупець;
- Банк емітента карти;
- банк продавця Карта бренду;
- платіжна система (Visa або MasterCard).

Процес оплати товару на сайті кредитною карткою такий:

1) Банк видає і активує кредитну карту клієнта за його запитом.
2) Покупець вказує інформацію власної кредитної картки на сайті ювелірної майстерні, де бажає придбати товар.
3) Система сайту підтверджує особу покупця, запитуючи дозвіл у компанії-виробника карти (емітента).
4) Кредитна картка авторизується платіжною системою і проходить транзакція кредитних коштів. Покупець отримує фіскальний чек.
5) Магазин отримує кошти на фінансовий рахунок, повідомлення про успішне зарахування коштів та номер замовлення.
6) Банк-еквайр надсилає запит до клієнтського банку з проханням очистити суму кредиту.
7) Платіжна система карти просить очистити суму в банку-емітента та зарахувати на рахунок карти.

2.3.3 Доставка замовлення

Після успішного проведення оплати, наступний етап - відбувається формування та відправка замовлення. Якщо покупець оформив замовлення в робочий період часу – протягом 30 хвилин з ним обов’язково зв’язується представник магазину, уточнює правильність складання замовлення, та пропонує можливі способи доставки. Доставка може здійснюватися такими способами: Нова Пошта, Укрпошта, Ін-тайм, Meest Express, служба поштоматів Приват Банку, кур’єрська служба доставки та самовивіз зі складу в Києві. Якщо замовлення було здійснене в неробочий час – менеджер зв’яжеться з покупцем

в найближчий ранок робочого дня. Електронний трек-номер посилки буде доставлений згодом, у вигляді СМС повідомлення.

2.4. Управління життєвим циклом клієнта

Існує безліч об'єктивів для вивчення даних про поведінку клієнтів в електронній торгівлі, але в основі всього цього лежить управління життєвим циклом клієнтів (CLM). Я вважаю, що CLM - це фундаментальний елемент, який необхідний для того, щоб можна було забезпечували гарну окупність інвестицій в маркетинг або зростання. Якщо існуватиме певне середовище CLM - можна зосередитися на інших аспектах.

Життєвий цикл клієнта - це термін, який використовується для опису послідовності кроків, що проходить клієнт при розгляді, покупці, використанні і збереженні лояльності до товару чи послуги (рисунок 2.3).



Рисунок 2.3 – Життєвий цикл клієнта E-Commerce (базовий)

Управління життєвим циклом клієнта (CLM) - це структура, яка полегшує плавний рух користувачів (не покупок) від придбання до лояльності (постійні активні клієнти) шляхом максимізації цінності, що надається в кожній точці взаємодії з клієнтами, і усунення будь-яких непорозумінь в конверсії.

Нижче наведені аспекти, чому важливо здійснювати керування життєвим циклом клієнтів:

- При обмежених каналах залучення клієнтів витрати на залучення клієнтів будуть продовжувати рости;
- Набагато простіше задовільнити і зберегти існуючого клієнта, ніж придбати нового;
- Щасливий клієнт не тільки купить більше, але і розповість про вас і приверне додаткових клієнтів (органічний трафік SEO).

Жоден підприємець не зможе побудувати стійкий бізнес електронної комерції без постійних клієнтів – тому це питання дуже важливе.

Стандартного способу визначення життєвого циклу клієнта для бізнесу в галузі електронної комерції не визначено, проте багато успішних компаній користуються цим потоком (рисунок 2.4), і він чудово виконує свою роботу.

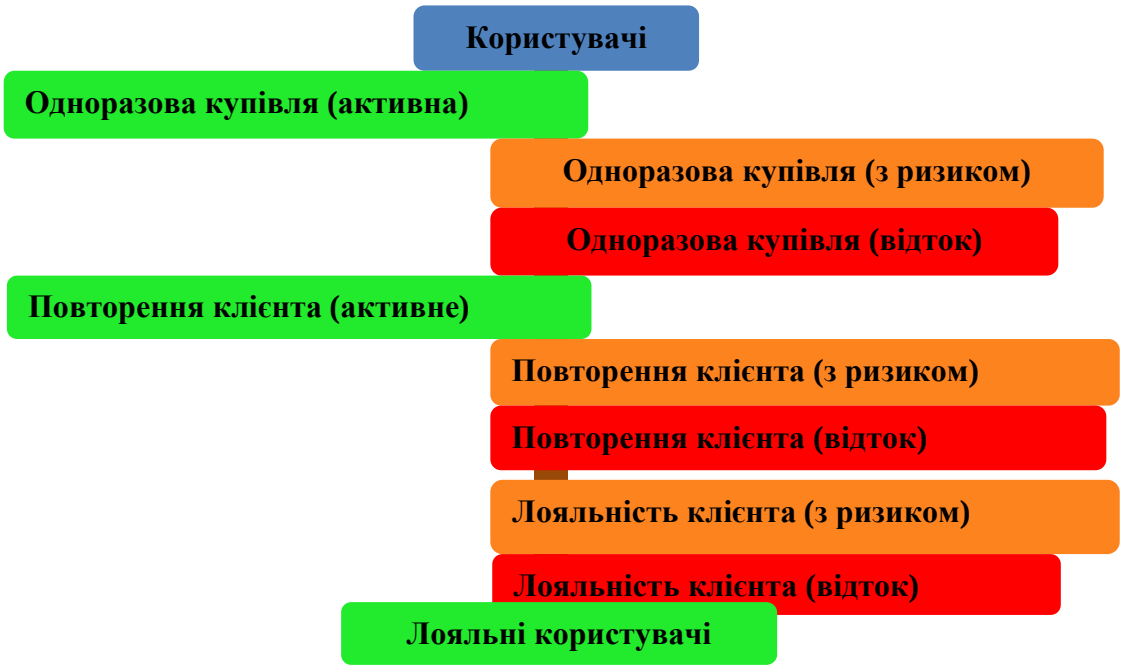


Рисунок 2.4 – Життєвий цикл клієнта (розширений)

Перш ніж перейти до метриків, потрібно швидко зрозуміти, що означає кожен етап життєвого циклу клієнта:

1. Одноразова купівля (активна) – користувачі, які здійснили лише одне замовлення менше X-днів тому назад.
2. Одноразова купівля (з ризиком) – користувачі, які здійснили лише одне замовлення від X до Y днів.
3. Одноразова купівля (відток) – користувачі, які здійснили лише одне більше ніж Y днів тому назад.
4. Повторення клієнта (активне) – користувачі, які здійснили більше ніж одне замовлення, і остання замовлення відбулося менше X-днів тому.
5. Повторення клієнта (з ризиком) – користувачі, які здійснили більше ніж одне замовлення, і остання замовлення відбулося від X до Y днів тому назад.
6. Повторення клієнта (відток) – користувачі, які здійснили більше ніж одне замовлення, і остання замовлення більше Y днів тому назад.
7. Лояльні користувачі (активні) – користувачі, які здійснили більше ніж Z-замовлень, і остання замовлення відбулося менше X-днів тому назад.
8. Лояльні користувачі (з ризиком) – користувачі, які здійснили більше ніж Z-замовлень, , і остання замовлення відбулося від X до Y днів тому назад.
9. Лояльні користувачі (відтік) – користувачі, які здійснили більше ніж Z-замовлень, , і остання замовлення відбулося більше ніж Y днів тому назад.

У цьому змісті пара ключових визначень, які потрібно зрозуміти:

Ризик – кількість днів, протягом яких, якщо клієнт не купує, він наражається на ризик відтоку (X днів).

Відток – кількість днів, протягом яких клієнт зовсім не проявляє ніякої активності (Y днів).

Не існує певного способу визначити постійних і непостійних клієнтів. Для простоти я визначив постійного клієнта як будь-якого, хто розмістив більше одного замовлення. Аналогічно, постійних клієнтів можна визначити декількома способами (замовлення /дохід/кількість і т. Д.), Але я визначив їх на основі кількості замовлень (Z замовлень). Залежно від характеру бізнесу буде можливо вибрати більш конкретні значення для X, Y і Z.

Ключові показники CML:

1) Коефіцієнт активації користувачів = (Всього клієнтів / Всього користувачів) * 100 % ;
2) Рейтинг повторних клієнтів = (Загальна кількість повторних клієнтів / Загальна кількість клієнтів) * 100 % ;
3) Рейтинг постійних клієнтів = (Всього постійних клієнтів / Загальна кількість повторних клієнтів) * 100 % .

Ці ключові метрики CLM електронної комерції - еквівалентні життєво важливим органам людського організму.

2.5. Оцінка ризиків при розробці і проектуванні ресурсу електронної комерції

Найважливіший крок, який ви можливо зробити, щоб оцінити свій план електронної комерції, - це ретельно, реалістично і навіть песимістично поглянути на нього. Проектна група повинна розглянути план і вказати місця, де найбільш ймовірне падіння налагодженої системи, і складе список - в результаті ми отримаємо свій власний список ризиків.

При складанні списку ризиків обов'язково потрібно ознайомитися з переліком переваг, щоб побачити, чи варті нові плани великого ризику. Наприклад, можна виявити, що деякі переваги вимагають використання нових і неперевірених технологій, тому вони самі по собі ризикові.

Ось список ризиків, з яких варто почати оцінку:

- Проект може відвернути на себе ключові ресурси в середині компанії, в результаті чого постраждає інша частина бізнесу.
- Можливість опинитися на тому ж рівні, що основний бізнес-конкурент, який вже веде велику частину свого бізнесу в Інтернеті.
- Клієнти можуть не захотіти змінювати спосіб ведення бізнесу з нами, тому інвестиція у розвиток проекту електронної комерції може не приносити прибуток.
- Збільшення кількості продуктів, пропоновані на веб-сайт, створить проблеми зі складуванням та розповсюдженням.
- Якщо компанія вирішить використовувати альтернативні варіанти складування, вони повинні будуть створити партнерства, які зроблять бізнес вразливим - повинні розраховувати на те, що ці партнери доставлять продукцію вчасно.
- Співробітники можуть відчути загрозу подальшого розвитку проекту і відмовитися від його підтримки або не рекомендувати його клієнтам.

Після ранжування ризиків в стратегії електронної комерції, можна почати дивитися на те, як впоратися з цими ризиками в разі їх виникнення. План не повинен йти далі, доки не буде змодельована чітка схема обходу ризиків, оцінених як «ймовірні». В ідеалі у компанії повинен бути план по зниженню всіх ризиків в її списку. Кожен ризик несе потенційні фінансові втрати, а кожна стратегія пом'якшення пов'язана з фінансовими витратами.

Розглянемо типовий ризик на прикладі нашого E-Commerce магазину ювелірної майстерні:

Сценарій: пожежа у серверному приміщені, на якому розміщений наш сервер.

Ризик 1: втрата прибутку, можлива втрата клієнтів, пошкодження репутації магазину.

Сценарій пом'якшення наслідків №1: у нас є тільки один веб-сервер. Якщо він вийде з ладу, ми можемо втратити бізнес на 4 дні, поки купимо і

налаштуємо новий сервер. Якщо ми отримуємо дохід у розмірі 3 000 доларів в день - наш ризик становить 12 000 доларів, а також вартість нового сервера, збитки, завдані нашій репутації, і втрата клієнтів, які можуть не захотіти повертатися.

Сценарій пом'якшення наслідків №2: ми вирішили зараз придбати другий резервний сервер вартістю 20 000 доларів у Франції.

Актуальне питання: чи виправдовує реальна втрата 20 000 доларів за сервер в даний час в порівнянні з потенційною втратою 12 000 доларів (не кажучи про репутацію, витрати на новий сервер та інші вслід витікаючі проблеми)?

Рішення: Сервер може відімкнутись один раз в місяць, втративши при цьому 12 000 доларів на місяць. Другий сервер може скоротити час очікування нашої компанії для користувачів і стати вигідним вкладенням в надійність і безпеку по мірі зростання компанії. Тому варто обрати другий сценарій.

Також ризики можуть бути пов'язані з шахрайством в Інтернеті, проблемами інформаційної безпеки, способами оплати та навіть із законодавством. Потрапляючи в один з цих ризиків, бізнесу доведеться вирішувати і відновлюватися дорого. Щорічно пом’якшення ризиків для підприємств електронної комерції і прямого маркетингу обходяться мільярдами доларів в рік, тому власникам необхідно розуміти можливі проблеми, пов'язані з веденням бізнесу в Інтернеті. Таким чином, власники бізнесу повинні розробити внутрішню політику для усунення потенційних ризиків і навчити своїх співробітників її впровадження.

Ризик 2: Клієнтські скарги та повернення товару.

Одна з найбільших проблем з онлайн-покупками в індустрії електронної комерції полягає в тому, що клієнти можуть вимагати відшкодування за спірне замовленнями. У більшості випадків суперечка виникає, якщо товар так і не надійшов, а сума була списана з рахунку клієнта. Іноді з клієнта стягується плата двічі, а іноді опис продукту не відповідає реальному продукту.

Рішення :

1) Уникаємо відвантаження неправильного продукту.
2) Не стягуємо плату з клієнта до тих пір, поки ви не впевнені, що усі товари відправлені і доставлені.
3) Використовуємо трек-номер для кожного замовлення, що розміщене і відправлене.

Ризик 3: Порушення безпеки та конфіденційності в Інтернеті.

Інформація, що включає в себе бази даних клієнтів, запити на покупку, процеси оплати – потенційно схильні до можливого взлому зловмисниками. Тому необхідно постійно проводити відповідні міри, щоб уникнути зміни або крадіжки даних. Необхідно налаштувати безпечний і ефективний процес відправлення запитів авторизації через мережу Інтернет, перш ніж почати приймати платежі за банківськими картками.

Рішення:

1) Забезпечуємо додаткові рівні безпеки мережі. Можна почати з правильно налаштованого брандмауера.
2) Після цього додаємо більше шарів на сайт через контактні форми, поля реєстрації та пошукові запити.
3) Постачальник послуг веб-розробки може допомогти нам у налаштуванні безпеки сайту.
4) Переконуємось, що ми ініціюємо та закінчуємо свої транзакції сертифікацією SSL.
5) Обираємо безпечну і надійну платформу електронної комерції, яка використовує парадигми об'єктно-орієнтованого програмування.
6) Регулярно надсилаємо запити у свою хостингову компанію, щоб провести перевірку серверів на наявність вірусів або шкідливого ПО.
7) Видаляємо старі записи з бази даних і зберігаємо мінімальний обсяг даних.

Ризик 4: Порушення інтелектуальної власності.

Порушення інтелектуальної власності також є однією з поширених загроз безпеки бізнесу електронної комерції. Захист інтелектуальної власності дуже важливий в індустрії електронної комерції і включає в себе логотипи сайтів, контент, слогани, продукти та інші зображення і значки. Порушення правил авторського права та інтелектуальної власності може призвести до безлічі судових позовів та величезних втрат.

Рішення:

1) Не копіюємо інформацію з інтернет джерел та в інших компаній.
2) Забезпечуємо дотримання строгих правил авторського права в компанії.
3) Ніколи не розкриваємо конфіденційну інформацію у чатах та месенджерах, тому що вони не мають повної захищеності.
4) Використовуємо строгі письмові протоколи і застосовуємо політику та практику безпеки.

Ризик 5: Низький рейтинг SEO.

Із багатьох потенційних загроз електронної комерції, SEO і цифровий маркетинг нашого бізнесу також може стати серйозним ризиком. Google продовжує змінювати свої алгоритми, і це різко впливає на наш рейтинг. Низький рейтинг SEO означає низький трафік на нашому сайті, що в кінцевому підсумку призводить до меншої кількості продажів. Тому нам потрібно зосередитися на цифровому маркетингу нашого бізнесу електронної комерції.

Рішення:

1) Шукаємо правильні ключові слова для просування наших продуктів.
2) Використовуємо правильні інструменти для пошуку.
3) Розробляємо ефективну стратегію в соціальних мережах.

Ризик 6: Шахрайство з кредитними картами.

Підозрілі транзакції і вкрадені дані кредитних карт є поширеними ризиками в бізнесі електронної комерції. Хакери або будь-хто інший може використовувати вкрадену кредитну карту для власних онлайн-транзакцій.

Наша система безпеки повинна бути достатньо продуктивною і надійною, щоб відловити та заморозити сумнівну транзакцію.

Рішення:

1) Ми повинні мати PCI-сумісність, тому що PCI-захист захищає дані власника кредитної картки і підтримує брандмауер.
2) Використовуємо аутентифікацію Secure Sockets Layer(SSL) для захисту даних.

Ризик 7: Погане обслуговування клієнтів.

Погане обслуговування клієнтів або набутий негативний досвід може стати серйозним фактором для відтоку клієнтів. Зрештою, це може зашкодити нашому бізнесу. Це включає в себе безліч лазівок, починаючи від наших грубих і непрофесійних агентів з обслуговування клієнтів - закінчуючи неефективним управлінням запасами. Неправильні поставки також можуть привести до втрати ефективності бізнесу.

Рішення:

1) Завжди зберігаємо ввічливість, в розмові зі своїми клієнтами.
2) Зосереджуємось на тому, щоб перевершити очікування ваших клієнтів.
3) Завжди витончено відповідаємо на запити наших клієнтів і приносимо вибачення за помилки.
4) Використовуємо ефективне програмне забезпечення для управління запасами, щоб позбавити клієнтів будь-яких незручностей.
5) Стараємося забезпечувати точні і швидкі поставки.

Ризик 8: Слабкі методи аутентифікації.

Якщо у нас слабкі і дуже прості методи аутентифікації - ми схильні до великої кількості кібератак. Якщо ми аутентифікуємо користувача тільки за логіном і паролем, то є велика ймовірність, що ця інформація може бути вкрадена. Нам потрібно вводити більш жорсткі методи аутентифікації в систему, що можуть протистояти атакам.

Рішення:

1) Використовуємо методи двухфакторної аутентифікації, з використанням одноразових СМС кодів, або листів на вказану при реєстрації пошту.

Таким чином, всі перераховані вище методи - лише деякі з безлічі методів обмеження ризиків в E-Commerce. Дотримання цих пунктів, допоможе більш ефективно розвивати наш інтернет-магазин в майбутньому.

2.6. Діаграми та їх реалізація в структурному підході

Для знаходження рішень у завданнях функціонального моделювання на базі структурного аналізу як правило використовують два типи моделей: діаграми IDEF0 та DFD (діаграма потоків). Методологія розробки процесних діаграм застосовується для проведення аналізу підприємства в рамках проектів управлінського консалтингу, а також в задачах з автоматизації масштабних об'єктів при експрес-аналізі.

Нотація діаграм потоків даних відображає як кроки бізнес-процесів, так і управління. На діаграмах можна відобразити засоби автоматизації кроків бізнес-процесів. Зазвичай застосовують для відображення третього і нижче рівня декомпозиції бізнес-процесів (першим рівнем вважається ідентифікований перелік бізнес-процесів, а другим - функції, що виконуються в рамках бізнес-процесів).

Діаграми потоків даних (Data flow diagramming, DFD):

- основний засіб для моделювання функціональних вимог проектованої системи;
- використовують для моделювання існуючого процесу передачі інформації;
- використовують для обробки інформації та опису документообігу;
- застосовують, як доповнення до моделі IDEFO, для більш наочного відображення поточних операцій документообігу (обміну інформацією);

- забезпечують проведення аналізу та визначення основних напрямків реінжинірингу інформаційних систем.

Діаграми DFD доповнюють відображення в моделі IDEF0, оскільки вони описують потоки даних, дозволяють відслідкувати, яким чином відбувається обмін інформацією, як усередині системи між компонентами, так і системи в цілому із зовнішнім інформаційним середовищем.

У випадку наявності в моделюючій системі програмної частини - як правило перевага віддається діаграмі потоків даних з таких міркувань:

1) DFD-діаграми розроблялися як засіб проектування програмних систем, тоді як IDEF0 - як засіб проектування систем в цілому. Саме тому діаграма потоків має масштабніший набір елементів, які вірно відображають специфіку.

2) Міні-специфікації процесів нижнього рівня допомагають виправити незавершеність IDEF0 і будувати функціонально повну специфікацію системи.

3) Доступні алгоритми для автоматичної трансформації ієрархій потоків даних у структурні карти.

За рахунок діаграм потоків даних - вимоги до проектованих інформаційних систем поділяються на процеси і зображуються у вигляді мережі, що пов'язана потоками даних. Головною метою декомпозицій функцій діаграм потоків даних – це наочно відобразити, як кожен окремий процес в системі трансформує вхідні у вихідні дані; виявлення відносин між процесами.

Якщо при моделюванні за допомогою IDEF0 систему розглядають як мережу взаємозв’язаних функцій, то при моделюванні DFD-діаграми систему розглядають, як мережу взаємозв’язаних функцій (сукупність сутностей).

Моделювання процесу роботи інтернет магазину реалізація ювелірних виробів представлено на DFD діаграмі (рисунок 2.5). Тут ми можемо побачити, що процес роботи інтернет-магазину складається з декількох функцій, які виконуються під час взаємодії з клієнтом:

Прийняття запиту від клієнта - функція, що відповідає за отримання запиту від клієнта, коли він зайшов на сайт інтернет-магазину.

Перегляд товарів - функція, що відповідає за відображення товарів на сайті, щоб клієнт міг їх переглянути.

Пошук товарів - функція, що відповідає за пошук товарів на сайті за допомогою пошукової системи.

Додавання товару у кошик - функція, що відповідає за додавання товару, який клієнт обрав, у кошик.

Оформлення замовлення - функція, що відповідає за оформлення замовлення, яке складається з товарів, які клієнт обрав та інформації про нього (адреса, контактні дані тощо).

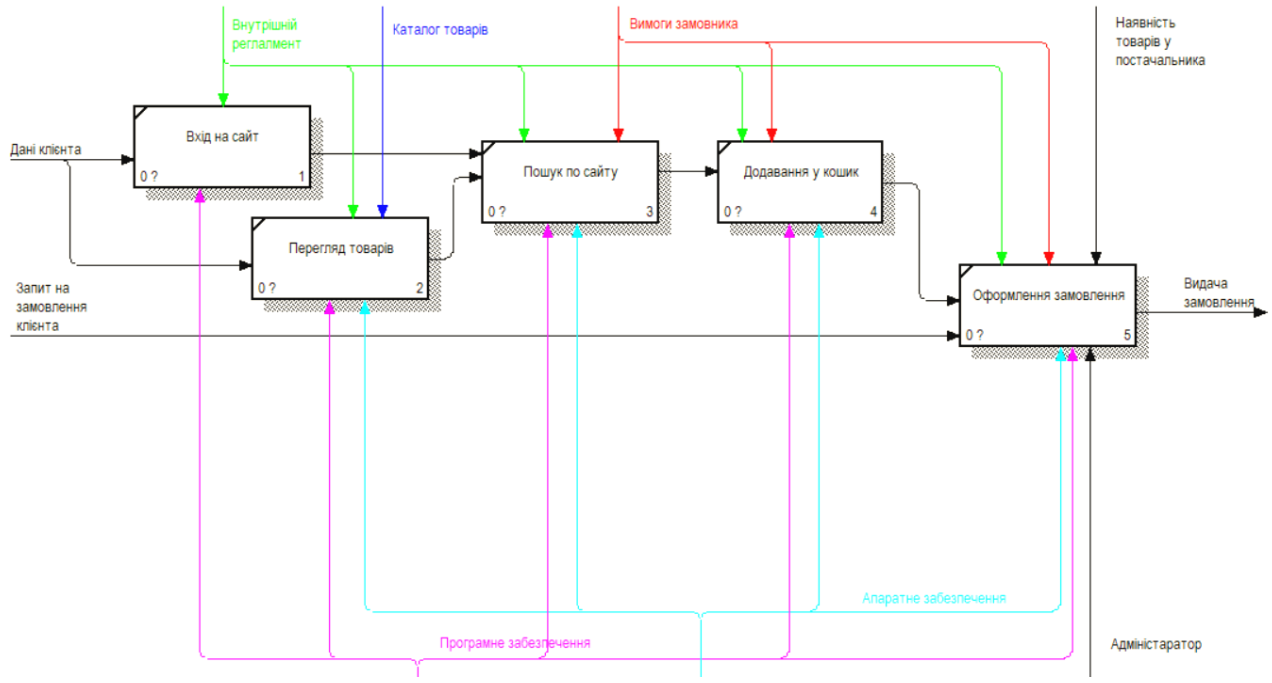


Рисунок 2.5 - Контекстна DFD модель розроблюваного Інтернет-магазину

Крім того, на діаграмі можна побачити ресурси, які використовуються в процесі роботи інтернет-магазину, такі як база даних товарів і кошика, платіжні системи для оплати замовлення, та сервер для забезпечення безперебійної роботи сайту.

Основною метою діаграми варіантів використання є моделювання функціональних вимог до системи та визначення поведінки системи з різних перспектив. Вона допомагає комунікувати та визначати потреби користувачів, а

також слугує основою для подальшого проектування та розробки системи. Нижче наведена діаграма варіантів використання для інтернет-магазину (рисунок 2.6).

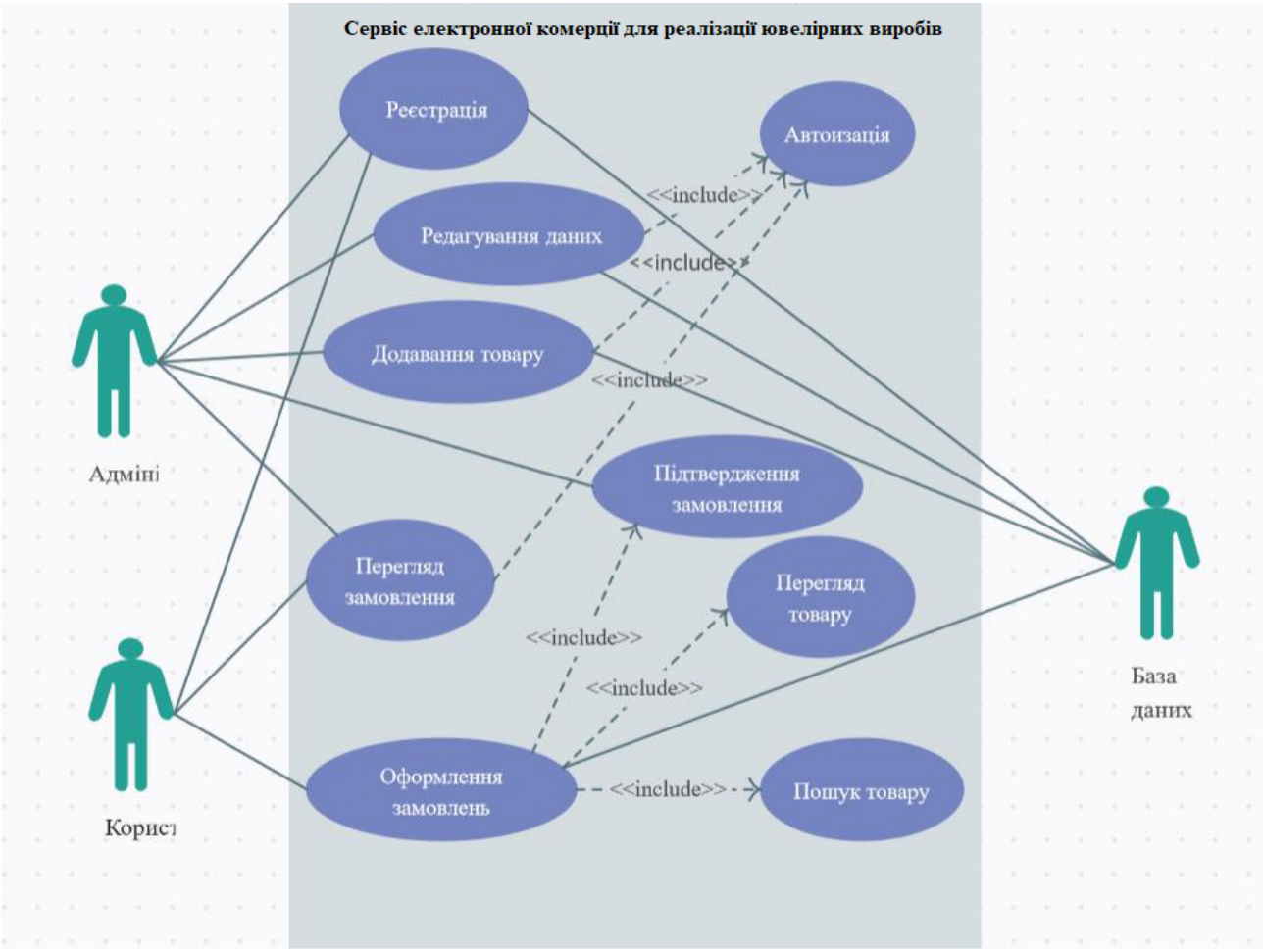


Рисунок 2.6 - Діаграма варіантів використання

На діаграмі можна побачити основні взаємодії користувачів з системою - клієнта та адміністратора:

1. Клієнт може зайти на сайт, переглянути товари, здійснити пошук, додати товар у кошик та оформити замовлення.

2. Адміністратор може додавати/редагувати товари на сайті, керувати замовленнями та користувачами.

Кожен варіант використання має свій опис, який детально описує послідовність дій користувача та системи для досягнення певної мети.

Наприклад, варіант використання "Оформлення замовлення" може мати такий опис:

- Клієнт зайшов на сайт інтернет-магазину.
- Клієнт переглянув товари та здійснив пошук.
- Клієнт додав потрібний товар у кошик.
- Клієнт перейшов до оформлення замовлення.
- Клієнт заповнив необхідну інформацію (адреса, контактні дані тощо).
- Клієнт обрав спосіб оплати та підтвердив замовлення.

Система зберегла замовлення та відправила підтвердження на електронну пошту клієнта.

2.7. Модель життєвого циклу програмного забезпечення

Життєвий цикл розробки програмного забезпечення (SDLC) - це процес, яким користується вся індустрія програмного забезпечення для проектування, розробки і тестування високоякісного програмного забезпечення. SDLC націлений на розробку ПЗ, яке повністю відповідає очікуванням замовників або навіть перевершує їх, в найкоротші терміни завершує роботу і оцінює витрати.

Життєвий цикл програмного забезпечення складається з детального плану, що описує: розробку, підтримку, заміни і заміни, що поліпшують конкретне програмне забезпечення. SDLC визначає методологію вдосконалення програмного продукту та загального процесу розробки ПЗ.

Існують певні спроектовані різновиди моделей SDLC, за якими слідують в процесі розробки програмного забезпечення. Ці моделі також називаються моделями процесів розробки програмного забезпечення. Кожна модель процесу – це серія послідовних кроків, унікальних для кожного типу, щоб забезпечити успіх команди в процесі розробки продукту.

Нижче наведені найважливіші та найпопулярніші моделі життєвого циклу програмного забезпечення, що використовується в галузі IT:

- Waterfall Model;
- Iterative Model;
- Spiral Model;
- V-Model;
- Big Bang Model.

Іншими зв'язаними методолігіями є Agile Model, RAD Model, Rapid Application Development і моделювання прототипів.

2.7.1 Waterfall SDLC

Модель водоспаду (Waterfall) - стала найпершою моделлю процесу, яку представили IT-спільноті. Вона також часто називається лінійно-послідовною моделлю життєвого циклу ПЗ. У цій моделі кожна фаза розробки має бути завершена до того, як почнеться наступна. У цих фазах не може збігів. Модель «водоспаду» ілюструє процес розробки як лінійний послідовний потік, а фази - не перекриваються між собою. Як правило результат кожної попередньої фази слугує входом для наступної.

Нижче наведена графічна ілюстрація WaterFall моделі (рисунок 2.7).

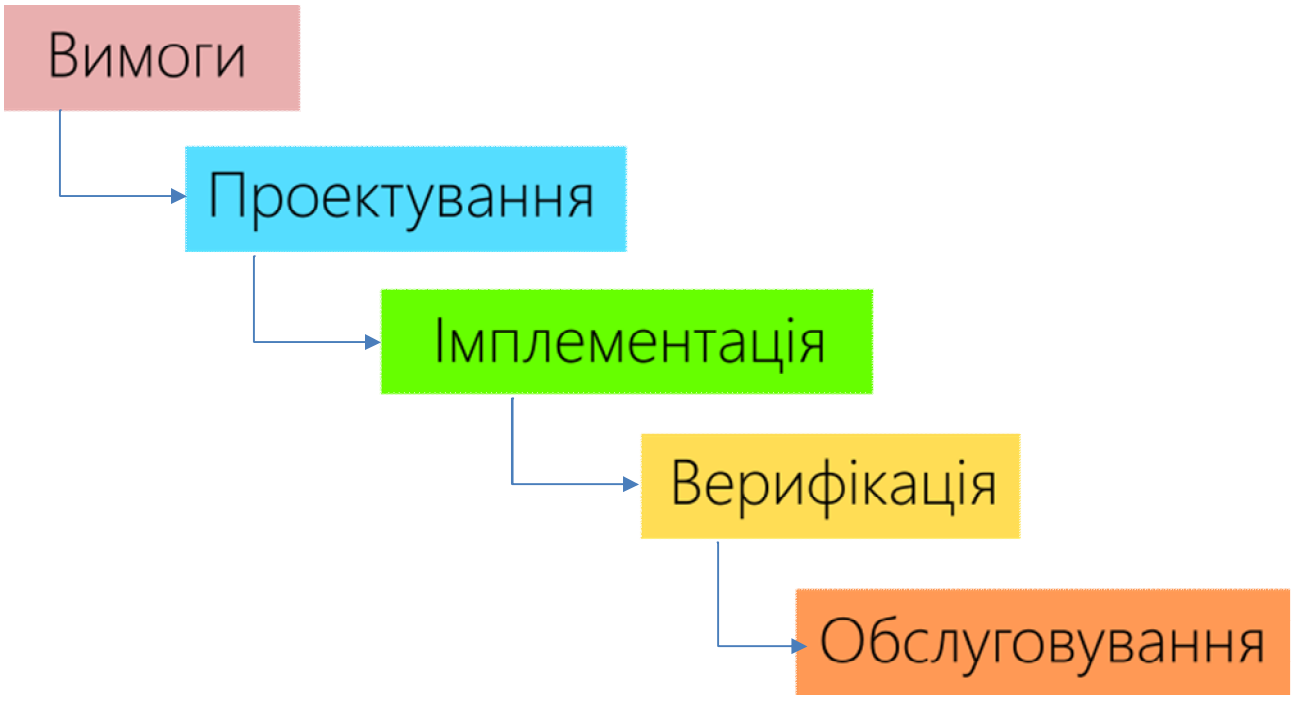


Рисунок 2.7 – WaterFall Model SDLC

Вимоги. На даному етапі виконується збір і аналіз всіх можливих вимог до розроблюваної системи. Все це фіксується і документується в документі специфікації вимог.

Проектування. На цьому етапі вивчаються вимоги і виконується проектування системи. Це допомагає визначити вимоги до обладнання, системи і її загальної архітектури.

Імплементація. Із врахуванням вихідних даних, отриманих під час проектування, система спочатку розробляється у вигляді невеликих програмних модулів, які інтегруються на наступному етапі. Коли вони будуть повністю розроблені – проходить перевірка на функціональність, що також зветься модульним тестуванням.

Верифікація. Коли готові всі модулі, що розроблювались на етапі імплементації - вони інтегруються в систему після тестування кожного модуля. Після верифікації вся система перевіряється на наявність помилок і збоїв.

Обслуговування. Після розгортання продукту часто виникають проблеми в клієнтському середовищі. Етап обслуговування використовується для виправлення цих проблем. Також для поліпшення якості продукту випускають нові ревізії ПЗ (Version 1..). Технічне обслуговування проводиться для впровадження цих змін.

Перевага WaterFall моделі полягає в тому, що вона дозволяє відокремити і контролювати. Графік з конкретно визначеними термінами може бути встановлений для кожного етапу розробки, а продукт буде проходити етапи розробки поступово. Розробка переходить від концепції до проектування, впровадження, тестування, розгортання, виправлення багів, а закінчується експлуатацією та обслуговуванням. Кожен етап розвитку протікає в строгому визначеному порядку.

Недоліком ж можна вважати те, що строго визначена модель не дозволяє багато роздумувати, додавати новий та переглядати існуючий функціонал. Коли додаток знаходиться на стадії тестування, дуже важко повернутися назад і

змінити щось, що не було задокументовано та продумано на стадії проектування ПЗ.

2.7.2 Iterative SDLC

У ітераційній моделі (Iterative Model) процес розробки ПЗ починається із написання невеликого набору вимог до продукту та ітеративного покращення версій, що розвиваються, доки вся система не буде впроваджена і готова до розгортання.

За моделлю ітеративного життєвого циклу – розробники не намагаються почати із повної специфікації вимог до продукту. Замість цього розробка починається з визначення та впровадження тільки частини програмного забезпечення, яка пізніше аналізується для визначення подальших вимог. Потім цей процес повторюється, розробляючи все нову версію продукту в кінці кожної ітерації даної моделі.



Рисунок 2.8 – Ітеративна модель розробки ПЗ

Слідування ітераційній моделі починається із простої реалізації підмножини вимог до програмного забезпечення, що ітеративно розширює версії, які розвиваються, доки не буде реалізована повноцінна і працююча

система (рисунок 2.8). При кожній ітерації вносяться зміни в конструкцію і додаються нові функціональні можливості. Основна ідея цього методу полягає в тому, щоб розробити систему за допомогою повторюваних циклів (ітерацій) і меншими порціями за один раз (поступово).

Успішним використанням ітеративного життєвого циклу розробки програмного забезпечення є валідація вимог, верифікація і тестування кожної наступної версії ПЗ в кожному циклі моделі. Оскільки продукт розвивається послідовними циклами - тести повинні повторюватися і розширюватися для перевірки кожної нової версії ПЗ.

Модель ітеративної розробки найчастіше використовується в наступних сценаріях:

- Вимоги до повної системі чітко визначені і зрозумілі.
- Нова технологія освоюється і використовується командою розробників уже під час роботи над проектом.

Перевага моделі полягає в тому, що на початковій стадії розробки вже існує працююча модель системи, що полегшує пошук функціональних та конструктивних недоліків. Знаходження проблем на ранній стадії розробки дозволяє приймати заходи для корегування в обмеженому бюджеті. Недолік цієї моделі SDLC полягає в тому, що вона може бути застосована тільки до великих і громіздких проектів.

2.7.3 Модель Agile SDLC

В моделі Agile (Гнучка) після кожної ітерації розробки змовник може побачити результат і зрозуміти, чи він його задовольняє (рисунок 2.9). Це одна з переваг моделі життєвого циклу гнучкої розробки програмного забезпечення. Одним з його недоліків є те, що при відсутності певних вимог важко оцінити ресурси і вартість розробки. Екстремальне програмування - одне з практичних застосувань гнучкої моделі. Основа такої моделі складається з коротких щотижневих зустрічей - спринтів, які є частиною підходу Scrum.

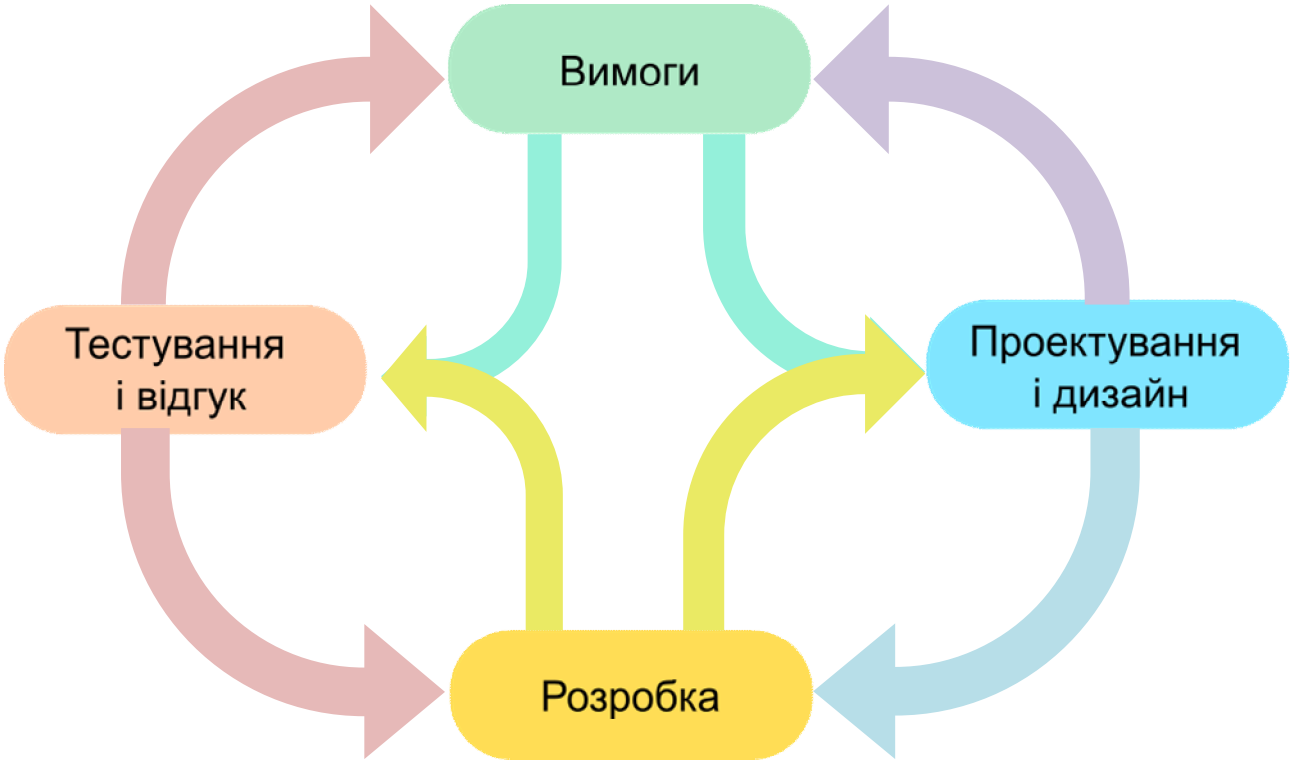


Рисунок 2.9 – Гнучка методологія розробки ПЗ

Нижче наведена таблиця переваг та недоліків гнучкої методології розробки (таблиця 2.1).

Таблиця 2.1 – Недоліки та переваги Agile Model

Переваги	**Недоліки**
Виправлення функціональних вимог впроваджуються в процес розробки для забезпечення конкурентоспроможності	Труднощі з вимірюванням кінцевої вартості через постійні змін
Проект ділиться на короткі і прозорі ітерації	Команда повинна бути високопрофесійною і орієнтованою на клієнта
Ризики зводяться до мінімуму завдяки гнучкому процесу розробки	Нові вимоги можуть конфліктувати з існуючою архітектурою
Швидкий випуск першої версії продукту	З усіма виправленнями і змінами існує ймовірність того, що проект перевищить очікуваний час розробки

Доступні кілька варіантів використання гнучкої моделі розробки:

- Потреби замовника динамічно змінюються;
- Менша ціна за внесення зміни через безліч ітерацій;
- На відміну від моделі Waterfall - для початку проекту потрібно тільки початкове планування.

За роки еволюції SDLC було розроблено безліч моделей з базовою каскадною моделлю для задоволення величезного розмаїття вимог і очікувань користувачів. Не існує єдиної відповідної моделі для всіх проектів, умов запуску і моделі оплати. Навіть на перший погляд, універсальний Agile не може широко використовуватися через непідготовленість деяких клієнтів до масштабування бюджету.

3. ПРОГРАМНА РОЗРОБКА СЕРВІСУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ДЛЯ РЕАЛІЗАЦІЇ ЮВЕЛІРНИХ ВИРОБІВ

3.1. Вибір середовища програмування.

Для виконання поставленої задачі існує безліч спеціалізованих середовищ розробки програмного забезпечення. Головна вимога при виборі – це повноцінна підтримка мови програмування JavaScript, каскадної таблиці стилей, та мови гіпертекстової розмітки. Серед другорядних вимог можна виділити: невисоку вартість пакету програмного забезпечення, або Open-Source розробку.

Для розробки було обрано продукт компанії JetBrains – WebStorm.

WebStorm - це інтегроване середовище розробки (IDE) від компанії JetBrains, призначене для роботи з веб-технологіями, такими як HTML, CSS, JavaScript та інші. Ось декілька ключових особливостей:

- Редактор коду: WebStorm надає потужний та зручний редактор коду з підсвічуванням синтаксису, автодоповненням, відлагодженням коду, рефакторингом та іншими функціями, що спрощують процес розробки.

- Підтримка різних мов та фреймворків: WebStorm підтримує не тільки JavaScript, але й інші мови, такі як TypeScript, CoffeeScript, HTML, CSS та JSON. Він також інтегрується з популярними фреймворками та бібліотеками, такими як Angular, React, Vue.

3.2. Аналіз вимог до розроблюваного програмного продукту

Аналіз вимог до розроблюваного програмного продукту є ключовим етапом у процесі розробки програмного забезпечення. Сутність цього аналізу полягає в тому, щоб зрозуміти і уточнити, які саме функціональні та

нефункціональні вимоги має задовольнити програмний продукт, щоб він відповідав потребам та очікуванням користувачів.

Основні вимоги до розроблюваного сервісу електронної комерції подано на рисунку 3.1.



Рисунок 3.1 – Перелік основних вимог до розроблюваного сервісу електронної комерції

Розглянемо дані вимоги більш детально.

- *Забезпечення адаптивного веб-дизайну*

Мобільні пристрої є важливою частиною нашого повсякденного життя, де споживачі проводять в середньому більше чотирьох-п’яти годин в день. Люди повинні мати можливість переглядати Інтернет сторінки і отримувати доступ до будь-якої потрібної інформації на сайті ювелірної майстерні зі своїх пристроїв, та без будь-яких проблем з відображенням контенту та масштабуванням. Адаптивний веб-дизайн автоматично буде пристосовувати наш E-Commerce магазин до використання на мобільних пристроях з різним розширенням екрану, не спотворюючи вигляд сайту.

- *Реалізація привабливого і лаконічного дизайну*

Привабливий інтернет-магазин супроводжує покупця на кожному етапі процесі купівлі. Спочатку він спокушає покупця відвідати веб-сторінку, потім - надихає на дослідження магазину та інших його товарів. Згідно дослідження компанії *Blue Corona*, у сайту є близько 10 секунд на те, щоб залишити позитивне враження клієнту та розповісти йому, що він отримає від нашого E-Commerce магазину. Якщо сайту не вдається привернути увагу за цей проміжок часу – ймовірно клієнт більше не повернеться.

- *Забезпечення простоти навігації*

Макет інтернет-магазину потрібно зробити інтуїтивно зрозумілим для нових відвідувачів - це хороший спосіб привернути їхню увагу до потрібних речей. Наприклад, організація товарів по логічним категоріям і підкатегоріями, забезпечення високої видимості цих категорій - допомагають клієнтам швидко знаходити потрібні товари, та залишити позитивний досвід користування нашим сайтом. Відображення кількох великих та чітких зображень (DPI > 300 pi) товарів на головній сторінці буде викликати цікавість у користувача. Хороше зображення допоможе клієнтам краще візуалізувати призначення та застосування товару самостійно. Показ схожих товарів також сприятиме переконанням покупцеві зробити покупку. Нижче наведений приклад дотримання хороших і не дуже правил дизайну веб-сторінки (рисунки 3.2 і 3.3).

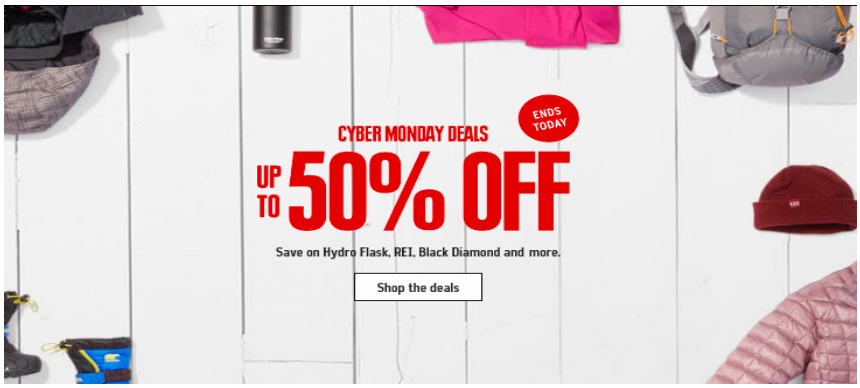


Рисунок 3.2 – Приклад хорошого дизайну головної сторінки ресурсу електронної комерції

А ось приклад дуже поганої реалізації веб-сторінки (рисунок 3.3).



Рисунок 3.3 – Приклад поганого веб-дизайну Інтернет-магазину

- *Вага та доцільність наданої інформації*

Покупки в Інтернеті часто можна порівняти зі «стрибком віри» - покупець не може фізично підібрати і відчути потрібний товар перед самою покупкою, як у звичайному магазині. Фактично, дослідження показало, що 62% покупців обирають звичайні магазини замість їх онлайн-варіантів виключно тому, що хочуть бачити товар на власні очі.

Тому, зважаючи на вище згадане, потрібно включати максимум інформації та медіа на сторінках продукту: такі як фотографії у високому розширенні; детальний опис продуктів; можливості масштабування і повороту; можливість переглядати товар у різних версіях та кольорах; відео-контент з товаром. Надаючи якомога більше інформації про товар, ми повинні зменшити коефіцієнт повернення товарів і збільшимо кількість задоволених клієнтів.

- *Реалізація знижок та акцій*

Розміщення нових віджетів і веб-банерів з рекламними акціями на нашому сайті повинно допомогти клієнтам бачити і поширювати останні спеціальні пропозиції. Інші рекламні функції, такі як таймер з відліком - повинен

сигналізувати, коли починається або закінчується спеціальна пропозиція, що допоможе стимулювати клієнтів до покупки. Можливість створювати необмежені рекламні акції по замовленнях, категоріях, способах доставки товарів тощо - допоможе ефективно зорієнтувати покупців нашого E-Commerce на кожному етапі покупок.

- *Створення інструментів маркетингу*

Під інструментами маркетингу в інтернет магазині розуміються такі інструменту які допомагають здійснити продаж товару. До таких інструментів належать:

- знижки на товар (закреслена стара ціна і нова ціна);
- знижки за оптове замовлення;
- купони на знижку;
- акції (наприклад в честь 10-річчя магазину);
- розпродажі.

- *Реалізація можливості клієнтських відгуків*

Покупці завжди бажають почути від реальних людей інформацію про їхній досвід використання товару, щоб побачити, чи відповідає продукт їх очікуванням. За даними *Kissmetrics*, на рішення 55% покупців впливає інформація з відгуків інших клієнтів. Тому потрібно розмістити сторінку з відгуками на нашому сайті. Хоч позитивні відгуки корисні, не варто боятися публікувати і негативні відгуки - прозорість компанії створює довіру до чесності нашого бренду.

- *Реалізація функціоналу кошика користувача*

Кошик - це не просто місце, де клієнти можуть зберігати товари в очікуванні оформлення замовлення, це ще одна важлива точка контакту, що може надати додаткові можливості для маркетингу. Перехресні продажі при оформленні замовлення допоможуть точніше зорієнтувати покупців про товари, якими вони цікавляться, на основі історії переглядів товарів, що здійснив юзер. В свою чергу, це може збільшити загальну суму покупки. Повна

інформація про продукти, що очікують на оформлення, їхня вартість (включаючи доставку і податок) інформуватиме клієнтів і знижуватиме ймовірність відмови.

- *Процес оформлення покупки (робота з кошиком)*

Процес покупки починається коли покупець переходить в кошик і готовий оформити замовлення. Вимоги до кошика:

- покупець повинен в зручному вигляді побачити весь список покупок;
- повинна бути можливість збільшити / зменшити кількість.

- *Забезпечення простоти оформлення товару*

Потрібно мінімально скоротити складний та тривалий процес оформлення покупки. Недотримання цього - змусить клієнтів відмовитися від свого кошика в останній момент. Проста перевірка дозволить клієнтам швидко заповнити тільки найнеобхіднішу інформацію. Інші поля клієнти можуть заповнити за власним бажанням у формі (наприклад, підписатися на розсилку спеціальних пропозицій по E-Mail, тощо). Попереднє заповнення форм завдяки функціоналу Google Pay, індикатор пройдених кроків/сторінок, можливість зберігати інформацію про кредитну картку - допоможе мінімізувати очікування та як наслідок, зменшить кількість відмов.

- *Реалізація процесу опрацювання замовлення*

Після того як покупець натискає кнопку «Оформити замовлення» запускається цілий комплекс процесів, а інтернет-магазин повинен підтримувати їх:

- Менеджеру приходить лист і SMS про те що було зроблено замовлення;
- Покупцеві має прийти лист електронною пошти з детальною інформацією про покупку;
- Покупцю і менеджеру повинні приходити інформаційні листи в процесі доставки з інформацією про те де зараз знаходиться товар;
- Покупцю і менеджеру повинні приходити інформаційні листи або SMS коли товар буде отриманий;

- Повинні бути механізми для переривання замовлення. Наприклад якщо виявилася нестача товару або якщо покупець відмовився від покупки.

- Реалізація секції FAQ та допомоги

Чим швидше магазин відповідатиме на запитання клієнтів, тим швидше він просунеться вперед у кількостях покупок. Вірне відображення контактної інформації нашого E-Commerce магазину допоможе покупцям дізнатися, куди звертатися у разі виникнення проблем у процесі купівлі.

У звіті KoMarketing йдеться про те, що 44% відвідувачів сайту покинуть сайт магазину, якщо немає контактної інформації або номера телефону. Реалізація чату також є дуже ефективним інструментом, оскільки клієнтам не потрібно перемикатися на інший пристрій/додаток, або іншим чином переривати роботу в Інтернеті. А спілкування з живою людиною може полегшити конверсію продажів. Згідно з декількома дослідженнями, 42% клієнтів вважають за краще користуватися чатом з технічним працівником компанії, а 77% онлайн-покупців не здійснять покупку, якщо немає можливості Live-чату (в реальному часі).

3.3. Реалізація дизайну та колірної схема оформлення сервісу електронної коммерції

Правильно підібрана кольорова палітра це можливість донести індивідуальність власного бренду до потенційних покупців і сформувати позитивний досвід роботи із сайтом E-Commerce. Тому для початку потрібно зрозуміти, які емоції і враження викликають різні кольори:

- Червоний - увага, хвилювання, гнів, симпатія, тепло, затишок, життя, кров, зупинка;
- Жовтий – пригоди, щастя, компетентність, ентузіазм, достаток, витонченість;
- Зелений - баланс, хороший смак, здоров’я, гроші, гармонія;

- Синій - чесність, корпоративність, висока якість, компетентність, холод, лояльність, довіра, надійність;
- Рожевий - любов, вишуканість, щирість, романтика, ніжність;
- Фіолетовий - творчість, авторитет, королівська влада, повага, таємниця;
- Коричневий - дружелюбність, грубість, сум, комфорт, натуральність.
- Чорний - горе, розумність, смерть.
- Білий - простота, порядок, невинність, чистота, порядок, нейтралітет;
- Сірий та срібний - тимчасовість, практичність, нейтральність, сучасність.

При проектуванні веб-магазину ювелірної майстерні моїм вибором стала біла колірна схема (рисунок 3.4). Оскільки компанія займається продажами коштовних ювелірних виробів – білий буде свідчити клієнтам про витонченість, красоту, вишуканість і чистоту.

Також за бажанням власника, при розробці сайту потрібно було утриматися від нагромадження будь-якої інформації. Тобто вигляд повинен бути максимально лаконічний і приємний користувачу.

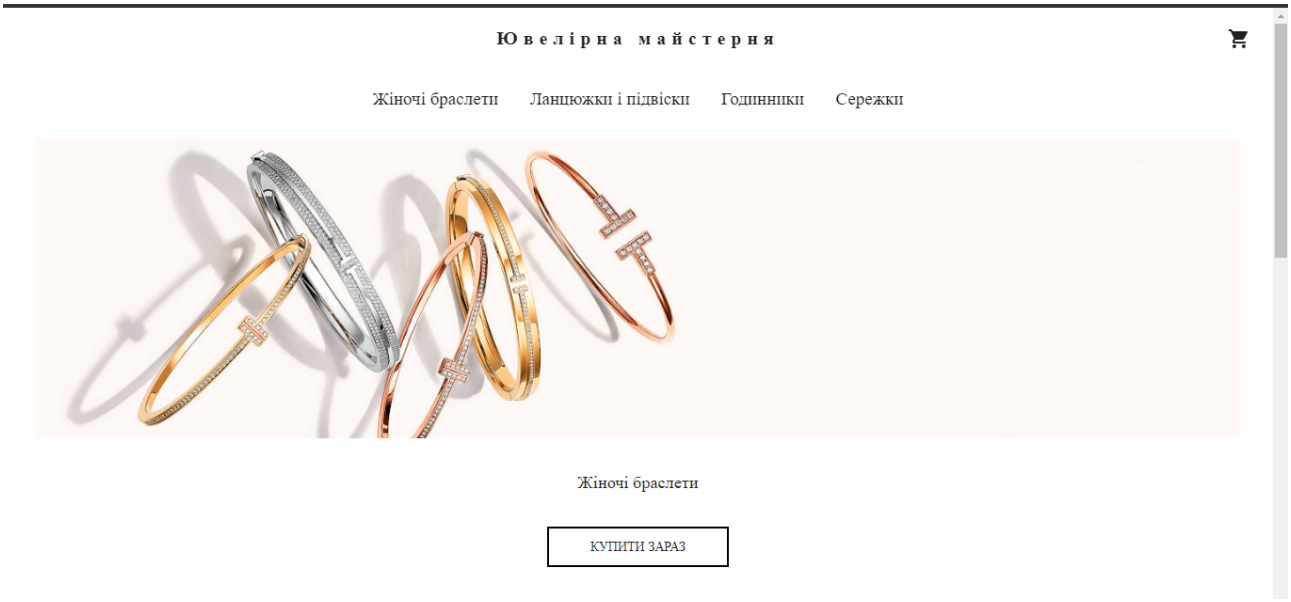


Рисунок 3.4 – Лаконічний дизайн ресурсу електронної комерції реалізації ювелірних виробів

У результаті, текст заголовку, був виконаний у класичному чорно-білому стилі, не відходячи від загальної концепції сайту. А майбутні кнопки вибору – у при наведенні повинні були здійснювати певну анімацію, для привернення уваги користувача. Тому я реалізував затемнення активної кнопки вибору при наведенні курсора миші (рисунок 3.5)

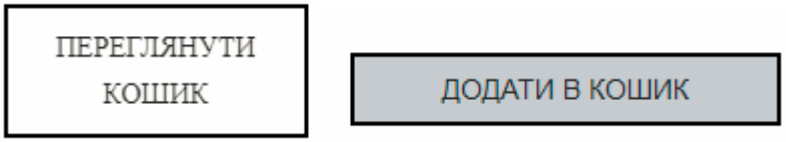


Рисунок 3.5 – Кольорова анімація кнопок при натисненні

По закінченню розробки проекту, пропонована замовником мінімалістична концепція була повністю дотримана, що головне. Проте ці вимоги будуть значно стримувати нас від певних цікавих реалізації UI елементів, ефектів та функціоналу.

3.4. Реалізація способу зберігання даних

Складним питанням при розробці веб-сайту став вибір способу збереження даних магазину. З одного боку інтернет магазин – це складна структура методів та даних, що логічно пов’язані між собою. Тому використання реляційних баз даних – це хороший вибір (звичайному варіанті). Але водночас цьому, складна СУБД по типу MySQL буде споживати надто багато ресурсів. І оскільки власник сайту турбується про максимальну лаконічність, легкість та швидкодію свого сервісу, вибір впав на інше. Так як власник ювелірної майстерні – це початковий підприємець, товарів у нього не настільки багато. Тому вважаю, що вірним рішенням буде представляти дані магазину у вигляді звичайних JSON-файлів.

JSON - це популярний текстовий формат даних, який використовується для обміну даними в сучасних веб та мобільних додатках. JSON також

використовується для зберігання неструктурованих даних в файлах журналів або базах даних NoSQL, таких як Microsoft Azure Cosmos DB. JSON також є основним форматом обміну даними між веб-сторінками та веб-серверами з використанням викликів AJAX.

JSON використовується для передачі даних в основному між додатками, наприклад, зовнішнім і внутрішнім інтерфейсами однієї і тієї ж системи.

Проте пізніше, з одночасним ростом бізнесу власника і кількості доступних товарів – логічніше і правильніше все-таки буде підключити швидкісну базу даних (SQL Lite тощо). І чудово, що майже всі сучасні СУБД без проблем можуть маніпулювати представленими даними у JSON.

Модель сайту передбачає чотири категорії товарів:

- Жіночі браслети – bracelets.json;
- Сережки – earrings.json;
- Підвіски – pendants.json;
- Годинники – watches.json.

Тому для формування списку товарів потрібно було створити чотири JSON файли, в яких дотримана конкретна структура подання файлів:

1) Поле «*Name*» - тут вказаний порядковий номер товару в загальному списку;
2) Поле «*Title*» - вказується повна назва товару, що відображається разом із відповідним фото в загальному списку;
3) Поле «*Category*» - тут вказується категорія до якої потрібно віднести товар (Жіночі браслети, сережки, підвіски та годинники);
4) Поле «*Price*» - тут вказується ціна на конкретний товар. Ціна доступна виключно у іноземній валюті по якій здійснюється закупка матеріалу для виготовлення прикрас (Долари США);
5) Поле «*Description*» - в цьому полі вказується повний опис та детальна характеристика товару. Для кращого візуального ефекту, опис

відображається у вигляді пунктів, що формується засобами гіпертекстової розмітки;

6) Поле «*Image*» - тут вказується повний шлях, за яким доступне фото товару, що відображається у загальному списку. Всі фото для коректного відображення повинні мати розширення 300x300 пікселів.
7) Поле «*LargeImage*» - тут вказується повний шлях, за яким доступне фото, що відображається у при відкритті певного товару. Всі фото для коректного відображення повинні мати розширення 600x600 пікселів.

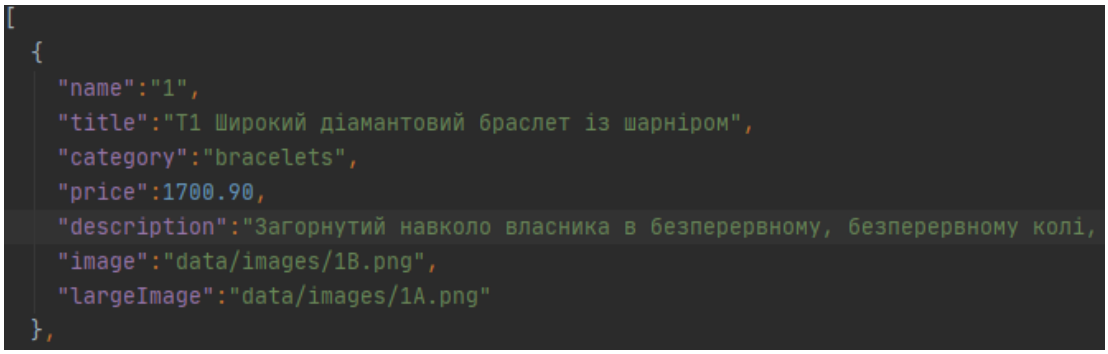


Рисунок 3.6 – Структура JSON файлу

Приклад правильної будови JSON файлу було зображено вище на рисунку 3.6. Даний шаблон однаковий для всіх товарів і категорій.

3.5. Розробка основних методів та функцій

Як вже згадувалося, сайт розробляється з використанням бібліотеки веб-компонентів Polymer. Бібліотека Polymer являє собою набір поліфілії, які допомагають створювати веб-компоненти у всіх основних сучасних браузерах. Вона надає основу для визначення, створення та рендерингу складних, призначених для користувача, елементів у спрощеній формі, аналогічно тегам HTML. Якщо зовсім по-простому – веб-компонент це певний шматок HTML коду, що виділений в окремий блок з власним шаблонуванням, стилями і логікою. Це дозволяє значно підвищити структурованість всього коду,

підвищити читаність і забезпечує можливість повторного використання веб-компонентів.

3.5.1. Компонент Lazy-Resources.js

Ліниве завантаження - це стратегія, що дозволяє ідентифікувати ресурси як неблокуючі (некритичні) та завантажувати їх тільки при необхідності. Це спосіб скоротити довжину критичного шляху рендерингу (послідовність кроків, які проходить браузер для перетворення HTML, CSS і JavaScript в пікселі на екрані), що сприяє скорочення часу завантаження сторінки.

```
import "./shop-icons.js";
import "./shop-analytics.js";
import "./shop-cart-data.js";
import "./shop-cart-modal.js";
import "./shop-snackbar.js";
import "./shop-tabs.js";
import "./shop-tab.js";
import "@polymer/paper-icon-button/paper-icon-button.js";
import "@polymer/app-layout/app-drawer/app-drawer.js";
```

Рисунок 3.7 – Lazy resources load

Ліниве завантаження на сайті ювелірної майстерні відбувається в момент, коли відбуваються деякі взаємодії з сайту з користувачем, таких як прокрутка та навігація. Готова реалізація детально зображена вище на рисунку 3.7.

3.5.2. Клас ShopHome

Клас ShopHome містить в собі методи для генерації відображення домашньої сторінки ювелірної майстерні. В ньому оголошуються методи для швидкого доступу до категорій товарів, розміщення основних фото категорій та формування так званих гарячих кнопок «Купити зараз».

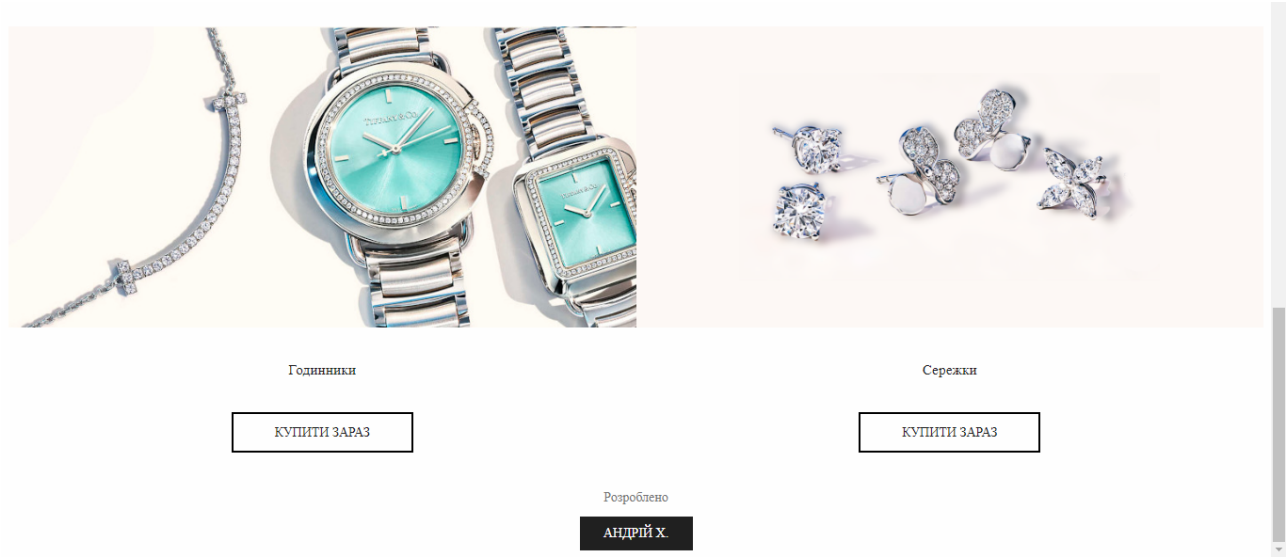


Рисунок 3.8 – Результат повернення функцій класу ShopHome

Приклад відображення реалізованого класу ShopHome наведений вище в рисунку 3.8.

3.5.3. Клас ShopCategoryData

У класі ShopCategoryData проводиться ініціалізація головних пунктів меню E-Commerce. Функція _getCategoryObject () відповідає за формування списку (рисунок 3.9) із чотирьох категорій (Сережки, годинники, підвіски та браслети).



Рисунок 3.9 – Результат функції _getCategoryObject

Метод _getResource () при потребі парсить відповідні дані для заданих пунктів з файлів .JSON.

3.5.4. Клас ShopListItem

Методи класу ShopListItem генерують динамічну веб-сторінку зі всіма елементами однієї з чотирьох категорій товарів.

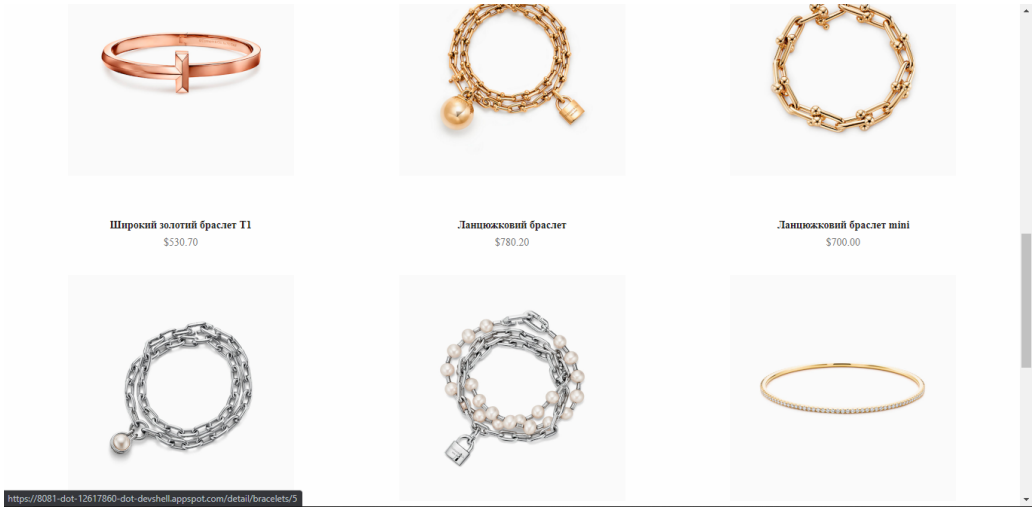


Рисунок 3.10 – Генерація динамічної сторінки з товарами категорії

Всі товари в кінцевому вигляді формують сітку по 3 елементи в рядку та відображають лаконічну інформацію про себе: фото, назву і ціну в доларах США. Приклад даної реалізації зображений вище на рисунку 3.10.

3.5.5. Клас ShopCartData

Даний клас має набір методів для маніпуляції з товарами в кошику (рисунок 3.11).



Рисунок 3.11 – Доступні маніпуляції з кошиком

Детальний опис методів:

- Метод addItem () виконує зміну кількості обраних товарів;
- Метод setItem () виконує перехід на сторінку з детальною інформацією товару;

- Метод clearCart () виконує очищення користувацького кошику;
- _computeNumItems () підраховує загальну кількість товарів;
- _computeTotal () обчислює загальну вартість всіх товарів в кошику.

3.5.6. Клас ShopCheckout

Функції даного класу виконують формування динамічної веб-сторінки і її елементів оформлення замовлення. Також метод _ValidateForm() відслідковує тип та кількість введених символів в поле. Доки не буде введено вірно – відображатиметься активне повідомлення про невірно введені дані (рисунок 3.12).



Рисунок 3.12 – Результат роботи функції Validate

Якщо все ж користувач ввів невідні дані – функція _reset скидає всі введені дані та повертає початковий вигляд сторінки з error-messages, що показано на рисунку 3.13.



Рисунок 3.13 – Результат роботи функції Reset

Після завершення оплати, системою магазину автоматично буде формуватися шаблонний-лист з вказаними користувацькими даними та списком замовлень. Далі він надсилається на E-Mail менеджеру магазину для подальшої реалізації відправки.

ВИСНОВКИ

У ході виконання дипломної роботи було проаналізовано предметну область розробки ресурсів електронної комерції та створено повноцінний сайт реалізації ювелірних виробів, за допомогою якого буде здійснюватися розширення ринку клієнтів та популяризація послуг.

Також було розглянуто важливість розробки B2C онлайн магазинів в наш час, та їхню важливу роль в успішному розвитку бізнесу. Розглянуто основні проблеми, що виникають на стадії розробки та проектування продукту, такі як проблема дизайну, надійності, правильності вибору хостингу, безпеки та конфіденційності даних клієнтів тощо. Досліджена важлива тема оптимізації пошукового трафіку, що при правильному підході - допомагає значно привести органічний трафік на наш сайт. Також підняв проблематику, пов’язану з хакерськими й DDoS атаками, та їхній негативний вплив на життєвий цикл клієнтів.

При створенні проекту була освоєна технологія Polymer, що допомагає значно спростити роботу програміста, розділивши весь код на HTML-подібні блоки, з якими набагато легше працювати, ніж зі стандартною структурою JavaScript. Також бібліотека Polymer має багато стандартних методів, що в подальшому спрощує написання коду. В результаті була згенерована стандартна структура, та реалізовані методи для маніпуляцій та взаємодії клієнта з об’єктами на екрані.

Оскільки власник для якої розроблявся даний Інтернет ресурс наполягав на легкості та лаконічності свого сайту, то було вирішено відмовитися від використання реляційних баз даних, а всі товари та дані зберігати у форматі представлення даних JSON. Оскільки сервер переводить строки JSON в JavaScript об’єкти – досягається максимальна швидкодія сервісу.

Створення програмного продукту здійснюється за ітераційним циклом розробки програмного забезпечення. Тому цілком можливо, що в подальшому

власник даного ресурсу забажає реалізувати й інші функції, від яких спочатку було прийнято рішення відмовитися. Наприклад, при розширенні персоналу і кількості продукції – правильніше буде все ж таки підключить до серверу швидкісну базу даних (наприклад SQLLite). Також можна розглядати доцільність реалізації клієнтських кабінетів (кабінету користувача) на сайті, де буде зберігатися контактна інформація про клієнтів, адреса, спосіб оплати та номер телефону.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Басюк Т.М. Принципи побудови системи аналізу та просування інтернетресурсів / Т.М. Басюк // Вісник Нац. ун-ту “Львівська політехніка” “Комп’ютерні науки та інформаційні технології”. – 2012.– № 784. – С.43–48.

2. Офіційний сайт Google Analytics. – Режим доступу: https://www.google.com/analytics/.

3. Сайт Polymer Library – Режим доступу: https://polymer-library.polymer-project.org/3.0/docs/about_30

4. Синявська О.О. Електронна торгівля в Україні: тенденції та перспективи розвитку. Вісник ХНУ імені В.Н. Каразіна. Серія «Міжнародні відносини. Економіка. Країнознавство. Туризм». 2019. Вип. 9. С. 126–132.

5. Startsev M.V. Electronic commerce as the way of the intensification of business processes. Соціально-економічні явища та процеси. 2011. № 5-6 (027-028). С. 212-215.

6. Zheng Qin. Introduction to E-commerce. Beijing: Tsinghua University Press, 2009. 517 c.

7. eCommerce Usage Distribution in the Top 1 Million Sites – Режим доступу до ресурсу: https://trends.builtwith.com/shop

8. Економіка підприємства: Навч. посібник / [П.В. Круш, В.І. Подвігіна, Б.М.Сердюк]; За заг. ред. П.В. Круша – К. :Ельга–Н, КНТ, 2007. – 780с.

9. Мельник О.В. – Електронна комерція: онлайн підручник: - http://intkonf.org/melnik-ov-elektronna-komertsiya-yak-skladova-chastinaelektronnogo-biznesu/

10. Nanehkaran,Y. An Introduction To Electronic Commerce. International Journal of Scientific & Technology Research, 2 (4), 190B193. 2013

11. Adobe Commerce 2.4 Developer Guide. Architecture Guide. Magento architectural diagrams – Режим доступу : https://devdocs.magento.com/guides/v2.4/architecture/archi_perspectives/arch_diagrams.html

12. Maciej Krawczyk. Building Your Own E-Commerce Keystone.js-Based System — Access Control: https://betterprogramming.pub/building-your-own-e-commerce-keystone-js-based-system-access-control-1a366ed7e064

13. Спірінцев В.В.Web-технології та Web-дизайн: HTML, CSS [Текст]: навч.посіб. / В.В.Спірінцев, В.В.Гнатушенко, О.С.Волковський//.-Дніпро: «Ліра», 2017.-163с.

14. Литовченко І. Л. Методологічні аспекти Інтернет-маркетингу: Монографія. – К. : Наукова думка, 2008. – 196 с.

15. Ілляшенко С. М. Сучасні тенденції застосування Internet-технологій у маркетингу // Маркетинг і менеджмент інновацій. – 2011. – № 4 (2). – С. 64–74.

16. Івашова Н. В., Іванова Т. Є. Застосування Internet-технологій у маркетингу: міжнародний досвід // Механізм регулювання економіки. – 2015. – № 4. – С. 55–68.

17. Сайт Академії інтернет-технологій IMT [Електронний ресурс]: https://imt.academy/.

18. Пасічник О.Г. Основи веб-дизайну /О.Г.Пасічник, О.В.Пасічник, І.В.Стеценко: [Навч.посіб.].-К.:Вид. група BHV.- 2009 .- 336 с.: іл.

19. Трофименко О. Г. Веб-технології та веб-дизайн : навч. посібник / О. Г. Трофименко, О. Б. Козін, О. В. Задерейко, О. Є. Плачінда. – Одеса : Фенікс, 2019. – 284 с

20. Робота з кольором в дизайні. URL: https://cases.media/article/robota-z-kolorom-v-dizaini

21. Навігація інтернет-магазину: розробка структури, типові помилки та вдалі приклади. URL: https://www.insales.com/ua/blogs/blog/navigatsiya-internet-magazina?lang=ua

22. Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів Бородкіна І.Л. – 2021. – 321 с.

ДОДАТКИ

Додаток А

Лістинг програми

Shop-app.js

```
import { PolymerElement, html } from '@polymer/polymer/polymer-element.js';
import '@polymer/app-layout/app-header/app-header.js';
import '@polymer/app-layout/app-scroll-effects/effects/waterfall.js';
import '@polymer/app-layout/app-toolbar/app-toolbar.js';
import { scroll } from '@polymer/app-layout/helpers/helpers.js';
import '@polymer/app-route/app-location.js';
import '@polymer/app-route/app-route.js';
import '@polymer/iron-flex-layout/iron-flex-layout.js';
import '@polymer/iron-media-query/iron-media-query.js';
import '@polymer/iron-pages/iron-pages.js';
import '@polymer/iron-selector/iron-selector.js';
import './shop-category-data.js';
import './shop-home.js';
import { afterNextRender } from '@polymer/polymer/lib/utils/render-status.js';
import { timeOut } from '@polymer/polymer/lib/utils/async.js';
import { Debouncer } from '@polymer/polymer/lib/utils/debounce.js';

window.performance && performance.mark && performance.mark('shop-app - before
register');
class ShopApp extends PolymerElement {
  static get template() {
    return html`
    <style>
      :host {
        display: block;
        position: relative;
        padding-top: 130px;
        padding-bottom: 64px;
        min-height: 100vh;
        --app-primary-color: #202020;
        --app-secondary-color: #757575;
        --app-accent-color: #172C50;
        --paper-button-ink-color: var(--app-accent-color);
        --paper-icon-button-ink-color: var(--app-accent-color);
        --paper-spinner-color: var(--app-accent-color);
        -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
        color: var(--app-primary-color);
      }

app-header {
        @apply --layout-fixed-top;
        z-index: 1;

        background-color: rgba(255, 255, 255, 0.95);
        --app-header-shadow: {
          box-shadow: inset 0px 5px 6px -3px rgba(0, 0, 0, 0.2);
          height: 10px;
          bottom: -10px;
        };
      }
paper-icon-button {
        color: var(--app-primary-color);
      }
      .logo {
```

```
        text-align: center;
      }
      .logo a {
        font-size: 16px;
        font-weight: 600;
        letter-spacing: 0.3em;
        color: var(--app-primary-color);
        text-decoration: none;
        display: inline-block;
        pointer-events: auto;
      }
      .left-bar-item {
        width: 40px;
      }
      .menu-btn {
        display: none;
      }
      .cart-btn-container {
        position: relative;
        width: 40px;
      }
      .announcer {
        position: fixed;
        height: 0;
        overflow: hidden;
      }
:host(:not([page=detail])) .back-btn {
        display: none;
      }
      [hidden] {
        display: none !important;
      }
      #tabContainer {
        position: relative;
        height: 66px;
      }

      shop-tabs, shop-tab {
        --shop-tab-overlay: {
          border-bottom: 2px solid var(--app-accent-color);
        };
      }
      shop-tabs {
        height: 100%;

      }
      shop-tab {
        margin: 0 10px;
      }
      shop-tab a {
        display: inline-block;
        outline: none;
        padding: 9px 5px;
        font-size: 13px;
        font-weight: 50
        text-decoration: none;
        color: var(--app-primary-color);
      }
      .cart-badge {
        position: absolute;
        top: -2px;
        right: 0;
```

```
  width: 20px;
  height: 20px;
  background-color: var(--app-accent-color);
  border-radius: 50%;
  color: white;
  font-size: 12px;
  font-weight: 500;
  pointer-events: none;
  @apply --layout-vertical;
  @apply --layout-center-center;
}
.drawer-list {
  margin: 0 20px;
}
.drawer-list a {
  display: block;
  padding: 0 16px;
  line-height: 40px;
  text-decoration: none;
  color: var(--app-secondary-color);
}
.drawer-list a.iron-selected {
  color: black;
  font-weight: bold;

shop-cart-modal {
  z-index: 2;
}
app-drawer {
  z-index: 3;
}
iron-pages {
  max-width: 1440px;
  margin: 0 auto;
}
footer {
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
  text-align: center;
  margin-top: 20px;
  line-height: 24px;
}
footer > a {
  color: var(--app-secondary-color);
  text-decoration: none;
}
footer > a:hover {
  text-decoration: underline;
}
.demo-label {
  box-sizing: border-box;
  width: 120px;
  padding: 6px;
  margin: 8px auto 0;
  background-color: var(--app-primary-color);
  color: white;
  text-transform: uppercase;
}
/* small screen */
@media (max-width: 767px) {
```

```
        :host {
          padding-top: 64px;
        }
        .menu-btn {
          display: block;
        }
        :host([page=detail]) .menu-btn {
          display: none;
        }
      }
    </style>
    <shop-analytics key="UA-39334307-16"></shop-analytics>
    <app-location route="{{route}}"></app-location>
    <app-route route="{{route}}" pattern="/:page" data="{{routeData}}"
tail="{{subroute}}"></app-route>
    <iron-media-query query="max-width: 767px" query-
matches="{{smallScreen}}"></iron-media-query>
    <shop-category-data categories="{{categories}}"></shop-category-data>
    <shop-cart-data id="cart" cart="{{cart}}" num-items="{{numItems}}"
total="{{total}}"></shop-cart-data>
    <app-header role="navigation" id="header" effects="waterfall" condenses=""
reveals="">
      <app-toolbar>
        <div class="left-bar-item">
          <paper-icon-button class="menu-btn" icon="menu" on-
click="_toggleDrawer" aria-label="Categories">
          </paper-icon-button>
          <a class="back-btn" href="/list/[[categoryName]]" tabindex="-1">
            <paper-icon-button icon="arrow-back" aria-label="Go back"></paper-
icon-button>
          </a>
        </div>
        <div class="logo" main-title=""><a href="/" aria-label="Домашня
сторінка">Ювелірна майстерня</a></div>
        <div class="cart-btn-container">
          <a href="/cart" tabindex="-1">
            <paper-icon-button icon="shopping-cart" aria-label\$="Shopping cart:
[[_computePluralizedQuantity(numItems)]]"></paper-icon-button>
          </a>
          <div class="cart-badge" aria-hidden="true"
hidden\$="[[!numItems]]">[[numItems]]</div>
        </div>
      </app-toolbar>
      <!-- Lazy-create the tabs for larger screen sizes. -->
      <div id="tabContainer" sticky\$="[[_shouldShowTabs]]"
hidden\$="[[!_shouldShowTabs]]">
        <dom-if if="[[_shouldRenderTabs]]">
          <template
            <shop-tabs selected="[[categoryName]]" attr-for-selected="name">
              <dom-repeat items="[[categories]]" as="category" initial-
count="4">
                <template>
                <shop-tab name="[[category.name]]">
                  <a href="/list/[[category.name]]">[[category.title]]</a>
                </shop-tab>
                </template>
              </dom-repeat>
            </shop-tabs>
          </template>
        </dom-if
      </div>
    </app-header>
```

```
    <!-- Lazy-create the drawer for small screen sizes. -->
    <dom-if if="[[_shouldRenderDrawer]]">
      <template>
      <!-- Two-way bind \`drawerOpened\` since app-drawer can update \`opened\`
itself. -->
      <app-drawer opened="{{drawerOpened}}" swipe-open="" tabindex="0">
        <iron-selector role="navigation" class="drawer-list"
selected="[[categoryName]]" attr-for-selected="name">
          <dom-repeat items="[[categories]]" as="category" initial-count="4">
            <template>
              <a name="[[category.name]]"
href="/list/[[category.name]]">[[category.title]]</a>
            </template>
          </dom-repeat>
        </iron-selector>
      </app-drawer>
      </template>
    </dom-if>
    <iron-pages role="main" selected="[[page]]" attr-for-selected="name"
selected-attribute="visible" fallback-selection="404">
      <shop-list name="list" route="[[subroute]]" offline="[[offline]]"></shop-
list>
      <!-- detail view of one item -->
      <shop-detail name="detail" route="[[subroute]]"
offline="[[offline]]"></shop-detail>
      <!-- cart view -->
      <shop-cart name="cart" cart="[[cart]]" total="[[total]]"></shop-cart>
      <!-- checkout view -->
      <shop-checkout name="checkout" cart="[[cart]]" total="[[total]]"
route="{{subroute}}"></shop-checkout>      <shop-404-warning name="404"></shop-
404-warning>
    </iron-pages>
    <footer>
      <a href="https://www.polymer-project.org/3.0/toolbox/">Розроблено</a>
      <div class="demo-label">Андрій Х.</div>
    </footer>
    <!-- a11y announcer -->
    <div class="announcer" aria-live="assertive">[[_a11yLabel]]</div>
`;
  }
  static get is() { return 'shop-app'; }
  static get properties() { return {
    page: {
      type: String,
      reflectToAttribute: true,
      observer: '_pageChanged'
    },
    numItems: {
      type: Number,
      value: 0
    }
    _shouldShowTabs: {
      computed: '_computeShouldShowTabs(page, smallScreen)'
    },

    _shouldRenderTabs: {
      computed: '_computeShouldRenderTabs(_shouldShowTabs, loadComplete)'
    },

    _shouldRenderDrawer: {
      computed: '_computeShouldRenderDrawer(smallScreen, loadComplete)'
```

```
    }
  }}  static get observers() { return [
    '_routePageChanged(routeData.page)'
  ]}
  constructor() {
    super();
    window.performance && performance.mark && performance.mark('shop-
app.created');
  }
  ready() {
    super.ready();
    // Custom elements polyfill safe way to indicate an element has been
upgraded.
    this.removeAttribute('unresolved');
    // listen for custom events
    this.addEventListener('add-cart-item', (e)=>this._onAddCartItem(e));
    this.addEventListener('set-cart-item', (e)=>this._onSetCartItem(e));
    this.addEventListener('clear-cart', (e)=>this._onClearCart(e));
    this.addEventListener('change-section', (e)=>this._onChangeSection(e));
    this.addEventListener('announce', (e)=>this._onAnnounce(e));
    this.addEventListener('dom-change', (e)=>this._domChange(e));
    this.addEventListener('show-invalid-url-warning',
(e)=>this._onFallbackSelectionTriggered(e));
        afterNextRender(this, () => {
      window.addEventListener('online', (e)=>this._notifyNetworkStatus(e));
      window.addEventListener('offline', (e)=>this._notifyNetworkStatus(e));
    });
  }
  _routePageChanged(page) {
    if (this.page === 'list') {
      this._listScrollTop = window.pageYOffset;
    }

    this.page = page || 'home';
    this.drawerOpened = false;
  }
  _pageChanged(page, oldPage) {
    if (page != null) {
      let cb = this._pageLoaded.bind(this, Boolean(oldPage));
      switch (page) {
        case 'list':
          import('./shop-list.js').then(cb);
          break;
        case 'detail':
          import('./shop-detail.js').then(cb);
          break;
        case 'cart':
          import('./shop-cart.js').then(cb);
          break;
        case 'checkout':
          import('./shop-checkout.js').then(cb);
          break;
        default:
          this._pageLoaded(Boolean(oldPage));
      }
    }
  }
  _pageLoaded(shouldResetLayout) {
    this._ensureLazyLoaded();
    if (shouldResetLayout) {
      // The size of the header depends on the page (e.g. on some pages the tabs
      // do not appear), so reset the header's layout only when switching pages.
```

```
      timeOut.run(() => {
        this.$.header.resetLayout();
      }, 1);
    }
  }
  _ensureLazyLoaded() {
    // load lazy resources after render and set `loadComplete` when done.
    if (!this.loadComplete) {
      afterNextRender(this, () => {
        import('./lazy-resources.js').then(() => {
          // Register service worker if supported.
          if ('serviceWorker' in navigator) {
            navigator.serviceWorker.register('service-worker.js', {scope: '/'});
          }
          this._notifyNetworkStatus();
          this.loadComplete = true;
        });
      });
    }
  }
  _notifyNetworkStatus() {
    let oldOffline = this.offline;
    this.offline =  !navigator.onLine;
    if (this.offline || (!this.offline && oldOffline === true)) {
      if (!this._networkSnackbar) {
        this._networkSnackbar = document.createElement('shop-snackbar');
        this.root.appendChild(this._networkSnackbar);
      }
      this._networkSnackbar.innerHTML = this.offline ?
          'Ви не в мерережі' : 'Ви в мережі';
      this._networkSnackbar.open();
    }
  }
  _toggleDrawer() {
    this.drawerOpened = !this.drawerOpened;
  }
  _setMeta(attrName, attrValue, content) {
    let element =
document.head.querySelector(`meta[${attrName}="${attrValue}"]`);
    if (!element) {
      element = document.createElement('meta');
      element.setAttribute(attrName, attrValue);
      document.head.appendChild(element);
    }
    element.setAttribute('content', content || '');
  }
 _onChangeSection(event) {
    let detail = event.detail;
    let scrollTop = 0;
    if (this.page === 'list') {
      if (this.categoryName === detail.category) {
        scrollTop = this._listScrollTop;
      } else {
        this._listScrollTop = 0;
      }
    }
    scroll({ top: scrollTop, behavior: 'silent' });
    this.categoryName = detail.category || '';
    if (detail.title) {
      document.title = detail.title + ' - SHOP';
      this._announce(detail.title + ', loaded');
      this._setMeta('property', 'og:title', detail.title);
```

```
      this._setMeta('property', 'og:description', detail.description ||
document.title);
      this._setMeta('property', 'og:url', document.location.href);
      this._setMeta('property', 'og:image', detail.image || this.baseURI +
'images/shop-icon-128.png');
      // Set twitter card metadata
      this._setMeta('property', 'twitter:title', detail.title);
      this._setMeta('property', 'twitter:description', detail.description ||
document.title);
      this._setMeta('property', 'twitter:url', document.location.href);
      this._setMeta('property', 'twitter:image:src', detail.image ||
this.baseURI + 'images/shop-icon-128.png');
    }
  }  _onAddCartItem(event) {
    if (!this._cartModal) {
      this._cartModal = document.createElement('shop-cart-modal');
      this.root.appendChild(this._cartModal);
    }
    this.$.cart.addItem(event.detail);
    this._cartModal.open();
    this._announce('Товар доданий в кошик');
  }
  _onSetCartItem(event) {
    let detail = event.detail;
    this.$.cart.setItem(detail);
    if (detail.quantity === 0) {
      this._announce('Товар видалено');
    } else {
      this._announce('Кількість змінено на ' + detail.quantity);
    }
  }
  _onClearCart() {
    this.$.cart.clearCart();
    this._announce('Кошик очищено');
  }
  _onAnnounce(e) {
    this._announce(e.detail);
  }
  _announce(message) {
    this._a11yLabel = '';
    this._announceDebouncer = Debouncer.debounce(this._announceDebouncer,
      timeOut.after(100), () => {
        this._a11yLabel = message;
      });
  }
  _domChange(e) {
    if (window.performance && performance.mark && !this.__loggedDomChange) {
      let target = e.composedPath()[0];
      let host = target.getRootNode().host;
      if (host && host.localName.match(this.page)) {
        this.__loggedDomChange = true;
        performance.mark(host.localName + '.domChange');
      }
    }
  }
  _onFallbackSelectionTriggered() {
    this.page = '404';
  }
  _computeShouldShowTabs(page, smallScreen) {
    return (page === 'home' || page === 'list' || page === 'detail') &&
!smallScreen;
  }
```

```
  _computeShouldRenderTabs(_shouldShowTabs, loadComplete) {
    return _shouldShowTabs && loadComplete;
  }
  _computeShouldRenderDrawer(smallScreen, loadComplete) {
    return smallScreen && loadComplete;
  }
  _computePluralizedQuantity(quantity) {
    return quantity + ' ' + (quantity === 1 ? 'item' : 'items');
  }
}
customElements.define(ShopApp.is, ShopApp);
```

Lazy-resources.js

```
import "./shop-icons.js";
import "./shop-analytics.js";
import "./shop-cart-data.js";
import "./shop-cart-modal.js";
import "./shop-snackbar.js";
import "./shop-tabs.js";
import "./shop-tab.js";
import "@polymer/paper-icon-button/paper-icon-button.js";
import "@polymer/app-layout/app-drawer/app-drawer.js";
import "./shop-network-warning.js";
import "./shop-404-warning.js";
import "./shop-cart-item.js";
import "@polymer/iron-form/iron-form.js";
import "@polymer/paper-spinner/paper-spinner-lite.js";
```

Shop-404-warning.js

```
import { PolymerElement, html } from '@polymer/polymer/polymer-element.js';
import '@polymer/iron-icon/iron-icon.js';
import './shop-button.js';
import './shop-icons.js';

class Shop404Warning extends PolymerElement {
  static get template() {
    return html`
    <style include="shop-button">
      :host {
        display: block;
        text-align: center;
        color: var(--app-secondary-color);
      }
      iron-icon {
        display: inline-block;
        width: 60px;
        height: 60px;
      }
      h1 {
        margin: 50px 0 50px 0;
        font-weight: 300;
      }
    </style>
    <div>
      <iron-icon icon="error"></iron-icon>
      <h1>Вибачте, ми не знайшли цю сторінку..</h1>
    </div>
    <shop-button>
```

```
      <a href="/">Перейти на домашню сторінку</a>
</shop-button>
`;
  }
  static get is() { return 'shop-404-warning'; }
}
customElements.define(Shop404Warning.is, Shop404Warning);
```

shop-list-item.js

```
import { PolymerElement, html } from '@polymer/polymer/polymer-element.js';
import '@polymer/iron-flex-layout/iron-flex-layout.js';
class ShopListItem extends PolymerElement {
  static get template() {
    return html`
    <style>
      :host {
        @apply --layout-vertical;
        @apply --layout-center-justified;
        text-align: center;
        margin: 0 48px;
      }
      shop-image {
        margin: 32px 0 16px;
      }
      shop-image::before {
        content: "";
        display: block;
        padding-top: 100%;
      }
      .title {
        color: var(--app-primary-color);
        font-weight: bold;
      }
      .price {
        color: var(--app-secondary-color);
      }
      @media (max-width: 767px) {
        :host {          margin: 0 12px;
        }
      }
    </style>
    <shop-image src="[[item.image]]" alt="[[item.title]]"></shop-image>
    <div class="title">[[item.title]]</div>
    <span class="price">[[_formatPrice(item.price)]]</span>
`;
  }
  static get is() { return 'shop-list-item'; }
  static get properties() { return {
    item: Object
  }}
  _formatPrice(price) {
    return price ? '$' + price.toFixed(2) : '';
  }
}
customElements.define(ShopListItem.is, ShopListItem)
```

Shop-cart-data.js

```
import { PolymerElement, html } from '@polymer/polymer/polymer-element.js';
import '@polymer/app-storage/app-localstorage/app-localstorage-document.js';
```

```
class ShopCartData extends PolymerElement {
  static get template() {
    return html`
    <app-localstorage-document key="shop-cart-data" data="{{cart}}"></app-
localstorage-document>
`;
  }
  static get is() { return 'shop-cart-data'; }
  static get properties() { return {
    cart: {
      type: Array,
      value: () => [],
      notify: true
    },
    numItems: {
      type: Number,
      computed: '_computeNumItems(cart.splices)',
      notify: true
    },
    total: {
      type: Number,
      computed: '_computeTotal(cart.splices)',
      notify: true
    }
  }}
  addItem(detail) {
    let i = this._indexOfEntry(detail.item.name, detail.size);
    if (i !== -1) {
      detail.quantity += this.cart[i].quantity;
    }
    this.setItem(detail);
  }
  setItem(detail) {
    let i = this._indexOfEntry(detail.item.name, detail.size);
    if (detail.quantity === 0) {
      // Remove item from cart when the new quantity is 0.
      if (i !== -1) {
        this.splice('cart', i, 1);
      }
    } else {
      // Use Polymer's array mutation methods (`splice`, `push`) so that
observers
      // on `cart.splices` are triggered.
      if (i !== -1) {
        this.splice('cart', i, 1, detail);
      } else {
        this.push('cart', detail);
      }
    }
  }
  clearCart() {
    this.cart = [];
  }
  _computeNumItems() {
    if (this.cart) {
      return this.cart.reduce((total, entry) => {
        return total + entry.quantity;
       }, 0);
    }
    return 0;
  }
  _computeTotal() {
```

```
    if (this.cart) {
      return this.cart.reduce((total, entry) => {
        return total + entry.quantity * entry.item.price;
      }, 0);
    }
    return 0;
  }
  _indexOfEntry(name, size) {
    if (this.cart) {
      for (let i = 0; i < this.cart.length; ++i) {
        let entry = this.cart[i];
        if (entry.item.name === name && entry.size === size) {
          return i;
        }
      }
    }
    return -1;
  }
}
customElements.define(ShopCartData.is, ShopCartData);
```

БІБЛІОГРАФІЧНА ДОВІДКА

Індивідуальна тема дипломної роботи бакалавра: «Розробка сервісу електронної комерції для реалізації ювелірних виробів**»**

Обсяг пояснювальної записки: 78 аркушів

Кількість таблиць 4

Кількість рисунків 20

Кількість додатків 1