

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сайту інтернет-магазину з продажу одягу

Виконав(ла): студент(ка) 4 курсу, групи СП-41

спеціальності 121 Програмна інженерія

(шифр і назва спеціальності)

Барна Т.В.

(підпис)

(прізвище та ініціали)

Керівник

Петрик М.Р.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Стоянов Ю.М.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2024

РЕФЕРАТ

Основною метою кваліфікаційної роботи є перевірка функціональності, надійності та відповідності вимогам програмного забезпечення. Шляхи досягнення мети включають аналіз предметної області, формування вимог, опис ПЗ, постановку задачі, аналіз вимог, функціональне тестування, відслідковування багів, навантажувальне тестування та автоматизоване тестування. Робота буде зосереджена на конструктивних, технологічних та техніко-експлуатаційних показниках та характеристиках ПЗ для оптимального функціонування сайту. Галузь застосування включає електронну комерцію і торгівлю.

Ключові слова: WEB-САЙТ, FRAMEWORK REACTJS, HTML5, CSS3, SCSS, MYSQL, FRAMEWORK FUELPHP, ПРОЄКТУВАННЯ, АРХІТЕКТУРА, ТЕСТУВАННЯ.

ABSTRACT

The main purpose of the qualification work is to check the functionality, reliability and compliance of the software. Ways to achieve the goal include domain analysis, requirements generation, software description, problem statement, requirements analysis, functional testing, bug tracking, load testing, and automated testing. The work will be focused on constructive, technological and technical and operational indicators and characteristics of the software for the optimal functioning of the site. The field of application includes e-commerce and trade.

Keywords: WEBSITE, FRAMEWORK REACTJS, HTML5, CSS3, SCSS, MYSQL, FRAMEWORK FUELPHP, DESIGN, ARCHITECTURE, TESTING.

ЗМІСТ

РЕФЕРАТ	4
ABSTRACT	5
ВСТУП	8
1. РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	10
1.1. Аналіз вимог до програмної системи	10
1.1.1. Аналіз предметної області.	10
1.1.2. . Постановка завдання.	12
1.1.3. Класифікація інтернет-магазинів.....	14
1.1.4. Аналіз програмних продуктів – аналогів.	15
1.2 Проектування програмного забезпечення.....	18
1.2.1. Структура побудови інтернет-магазину.....	18
1.2.2. Моделювання основних принципів роботи web-сайту.	22
1.3 Програмна реалізація інтернет-магазину.....	26
1.3.1. Вибір та огляд програмних засобів реалізації проекту.	26
1.3.2. Розробка основних функцій магазину.	27
1.3.3. Інструкція користувача.	31
1.4 Висновок до розділу 1	37
2. ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ.....	38
2.1 Тестуванн інтерфейсу	38
2.2 Відслідковування багів	42
2.3 Навантажувальне тестування.....	45
2.4 Автоматизоване тестування.....	47
2.5 Висновки до розділу 2.....	51
3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	53
3.1 Управління силами та засобами об’єктів господарювання під час надзвичайних ситуацій.	53
3.2 Розрахунок місцевого освітлення для верстату чи обладнання	55

ВИСНОВОК	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А – ДИСК ІЗ КВАЛІФІКАЦІЙНОЮ РОБОТОЮ БАКАЛАВРА	63

ВСТУП

Інтернет-торгівля - це процес продажу фізичних та нефізичних товарів через спеціалізовані електронні платформи можливість віддаленого розміщення замовлення. Інтернет-торгівля в Україні розвивається цілком успішно. На основі результатів дослідження ринку електронної комерції в Україні, обсягів продажу постійно зростає кількість таких магазинів.

Хоча інтернет давно і міцно увійшов у наше життя, багато підприємців і навіть великих компаній не розуміють, як його використовувати, що вони отримують від створення власного веб-додаток, тому що є інші способи самореклами, що добре зарекомендували себе. Форми самореклами: телебачення, радіо, ЗМІ, банери, листівки та інші.

Економічна криза підштовхує маркетологів до розвитку онлайн-торгівлі. Це впливає на високу вартість традиційного безкоштовного вендінгу через високу орендну плату, витрати на персонал, а також зміни у поведінці споживачів: дедалі більше українців перед покупкою шукають дешевші товари через інтернет. Крім того, на ринку спостерігається тенденція до зниження в автономному режимі та значно скоротити їх. Однак більшість роздрібних торговців все ще намагаються створити багатоканальний механізм продажу. Причинами поширення інтернет-магазинів є:

- Комфортність. Здійснювати покупки, перебуваючи вдома, - є мрією любителів довгочасового традиційного шопінгу. За декілька кліків, можна замовити той чи інший товар у інтернет-магазині без відволікань від поточних справ.

- Заощадження часу. Враховуючи вищесказане, користувачам подобаються платформи електронної комерції власне із-за можливості зекономити час. Адже, час витрачений на покупки, можна більш корисно провести.

- Можливість порівняти ваші улюблені товари та ціни на них. Майже всі сучасні інтернет-магазини використовують фільтраційні системи, за допомогою

яких можна відсортувати товар, відкинути непотрібний і порівняти позиції, представлені у каталозі, одночасно.

Комфориабельність шопінгу є пріоритетністю, тому, звертаючи увагу на вищезгадані причини, варто зауважити, що якщо власники хочуть залишатися конкурентоспроможними на сучасному ринку, то їм не обійтися без онлайн-версії свого магазину.

Мета кваліфікаційної роботи – можливість створення інтернет-магазину з продажу одягу та розробити гнучку, стійку до змін архітектуру та часткову реалізацію системи, метою якої є пошук та купівля обраного товару зі зручним інтерфейсом. Додаток повинен врахувати всі переваги конкурентів, вже представлених на ринку, і надавати користувачеві додаткові можливості, яких немає у конкурентів, щоб бути конкурентоспроможним.

1. РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

1.1. Аналіз вимог до програмної системи

1.1.1. Аналіз предметної області.

Для початку зупинюсь на використаних в роботі словникових термінах. Інтернет-магазин – це сайт, на якому можна зробити покупку через Інтернет-мережу, включаючи доставку. У той же час відправка інформації споживачеві, замовлення товару та угода відбуваються там же, усередині мережі.

Інтерфейс користувача – засоби, що дозволяють користувачеві комфортно взаємодіяти з інформаційною системою.

Електронна комерція – це сфера цифрової економіки, яка включає фінансові та комерційні трансакції, які проводяться із використанням комп'ютерних мереж, та відповідні з ними бізнес-процеси.

Електронні закупівлі — це купівля-продаж товарів, робіт, послуг поміж організаціями, підприємством та споживачем чи між підприємствами та державою через Інтернет-мережу, а також інші інформаційні та мережеві системи.

Кошик – список товарів, які Покупець планує придбати. Існують різні версії кошиків, однак, ми розглянемо розробку класичного модуля для інтернет-магазинів.

Замовлення – належним чином оформлений запит Покупця, що включає перелік обраних товарів у каталозі інтернет-магазину, який є підставою щодо прийнятого рішення про здійснення покупки.

Авторизація – контроль рівнів та засобів доступу до певного захищеного ресурсу, як у фізичному сенсі, так і у сфері цифрових технологій та системних ресурсів, на основі ідентифікації користувача та пароля або надання певних повноважень на виконання певних дій у системі оброблення даних.

Покупець – особа, яка користується послугами інтернет-магазину.

Рахунок-фактура – це документ, що спрямовується покупцю, який засвідчує прийняте інтернет-магазином замовлення, визначає його остаточну вартість і є підставою для оплати замовлення.

Внаслідок швидкого зросту технологій, бізнес-організаціями зроблено перехід від встановлених традиційних методів продажу товарів до електронних із використанням Інтернет-мережі, як основної, при проведенні комерційних операцій.

Електронна комерція – це послуга, яка дозволяє людям робити покупки в інтернеті і є поширеним видом діяльності в інтернеті в секторі B2C (business to customer) тобто продаж товарів або послуг приватним особам [1].

Вони не обмежені просторово, і на web-сайтах представлено багато різновидів товарів. Це допоможе покупцям-аналітикам купити обраний товар, за результатами пошуку.

Інтернет-покупки є однією із форм електронної комерції. З її використанням, споживач, через Інтернет мережу, може купити товар чи послугу. Споживач знаходить товар, який його зацікавив, на вебсайті роздрібної торгівлі або шляхом його пошуку посеред інших постачальників використовуючи торгову пошукову систему, де вказано, чи доступний потрібний товар та його ціна у різних точках продажу [2].

Інтернет-магазини надають клієнту розширений перелік товарів чи послуг, на відміну від звичайних магазинів, та надають користувачеві значний об'єм інформації щодо них. А це, відповідно, впливає на вибір та прийняття ним рішення щодо необхідності здійснити покупку. Вище сказане є головною особливістю таких магазинів. За допомогою веб-технологій можна налаштувати інтернет-магазин як спосіб збільшити продажі, тобто мати можливість запропонувати потрібний товар у потрібний час потрібному споживачеві. Також, слід зауважити, що для Інтернет-магазину не потрібно значних витрат на його утримання та організацію роботи, оскільки для його роботи потрібна мінімальна кількість працівників та матеріальна база (приміщення для роботи).

Покупки онлайн мають свої недоліки. Найпоширеніші з них, шахрайські транзакції та комісії платіжних систем, потенційні проблеми зі службами доставки, висока конкуренція на ринку та суперечки з покупцями про те, чи товар відповідає якостю основним параметрам.

1.1.2. . Постановка завдання.

Програмний продукт призначений для створення роздрібного інтернет-магазину одягу, який запропонує покупцям зручний інтерфейс для роботи, можливість пошуку товарів за категоріями та фільтрації за ціною та інші практичні функції. Для продавців буде реалізовано простий спосіб додавання товарів та забезпечено безпечну взаємодію між покупцями та продавцями.

Функціональні вимоги:

Система повинна передбачати наступні можливості:

- можливість реєстрації у системі;
- можливість авторизації у системі;
- можливість створення продавцем нового товару;
- можливість редагування продавцем створеного товару;
- можливість адміністратору блокувати користувачів;
- можливість покупцеві додавати товар до кошика;
- можливість покупцеві видаляти товар з кошика;
- можливість пошуку товару за назвою;
- можливість фільтрувати товари по ціні;
- можливість для продавця позначати товар як акційний;
- можливість для покупця оформити замовлення.

При розробці цей список функцій може бути розширений під час розробки програми.

Нефункціональні вимоги:

1. Надійність:

- Зручність використання. Інтерфейс розробленого програмного продукту має повністю відповідати потребам користувачів, бути зручним та інтуїтивно зрозумілим.

- Допомога в режимі онлайн. У разі виникнення проблем або збоїв у роботі системи, користувач може звернутися до технічних адміністраторів, які допоможуть йому вирішити проблему.

2. Продуктивність:

- підтримка одночасної безперебійної роботи ста і більше користувачів;
- час відгуку навіть при серйозному навантаженні на сервера не повинна перевищувати однієї секунди.

З урахуванням особливостей роботи ПЗ сформулюємо вимоги щодо його захищеності:

- доступ до сайту здійснюється з використанням інтернет-браузерів, які встановлені на комп'ютері чи мобільному телефоні (Android, IOS);
- реалізація доступу з застосуванням захищеного протоколу HTTPS;
- кешування паролів.

Перелік вимог до інтерфейсу ПЗ:

- однакове відображення згенерованих веб-сторінок при використанні різних інтернет браузерів;
- реалізація шаблонів за допомогою таблиць CSS- стилів (інтерпретація SCSS в CSS);
- простота та зрозумілість інтерфейсу;
- адаптивна верстка для планшетів та мобільних телефонів.

Збереження інформації повинно бути у MongoDB.

Формулювання вимог до ПЗ:

- ОС: Windows, версії 7, 8, 10, Linux. Android, IOS;
- браузери: Google Chrome, Mozilla Firefox.

1.1.3. Класифікація інтернет-магазинів.

Інтернет-магазини класифікують за наступними характеристиками [3]:

- Метод дрібного продажу товарів в мережі: власне інтернет-магазин; веб-вітрина, торгівельна система; торгівельний ряд; контентний проєкт (енциклопедія споживача, система інтернет-замовлень та ін.);
- Бізнес-модель: цілком on-line магазин і поєднання off-line-бізнесу з онлайнним (інтернет-магазин базується на основі діючої реальної торгівельної точки);
- Взаємовідносини з постачальниками:: магазини з власним складом (є реальні товарні запаси);
- Робота по договору з постачальниками (великі товарні запаси відсутні);
- Автоматизація торгівельної системи інтернет-магазину (ІМ): веб-вітрина, власне ІМ та торгівельна інтернет-система (ТІС) [3]. Вона є характерною у системі обробки замовлень, це уможливить індивідуальний підхід до кожного зареєстрованого клієнта. Спільним для ІМ та ТІС є проведення повного торгового циклу при мережевому підключенні. Тут ТІС додатково інтегрують у систему внутрішнього документообігу. Система постачання товару є неавтоматизованою. Типову структуру базових форм організації інтернет-магазину представлено на рисунку 1.1 [3].

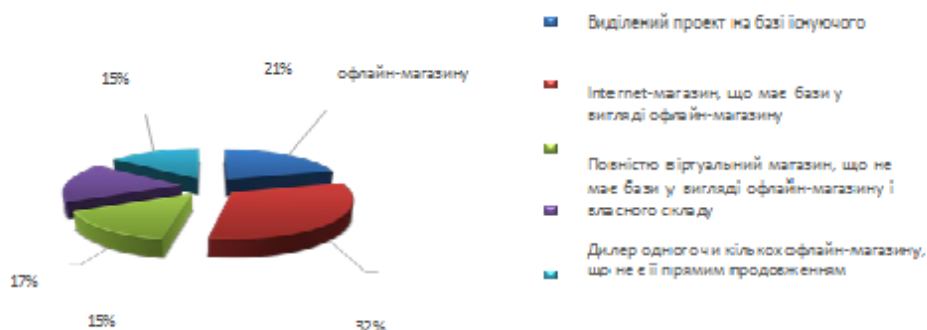


Рисунок 1.1 – Зображення форм організації ІМ в українському мережевому сегменті

Провівши аналіз електронної комерції, було визначено основні причини інтернет-покупок:

- час;
- зручність;
- широкий вибір товару;
- цінова політика;
- проінформованість;
- порівнювання товарів від різних виробників.

Недоліками є: відсутність можливості огляду чи примірки, домовленості щодо ціни, не визначені терміни очікування, безпечність, тощо.

1.1.4. Аналіз програмних продуктів – аналогів.

На даний момент на ринку існує декілька хороших рішень для даної предметної області. Для аналізу було обрано декілька сайтів зі схожою тематикою і призначенням. У кожного з цих рішень є свої плюси та мінуси.

Одним із таких сайтів являється сайт магазину для чоловічого і для жіночого одягу «Staff».

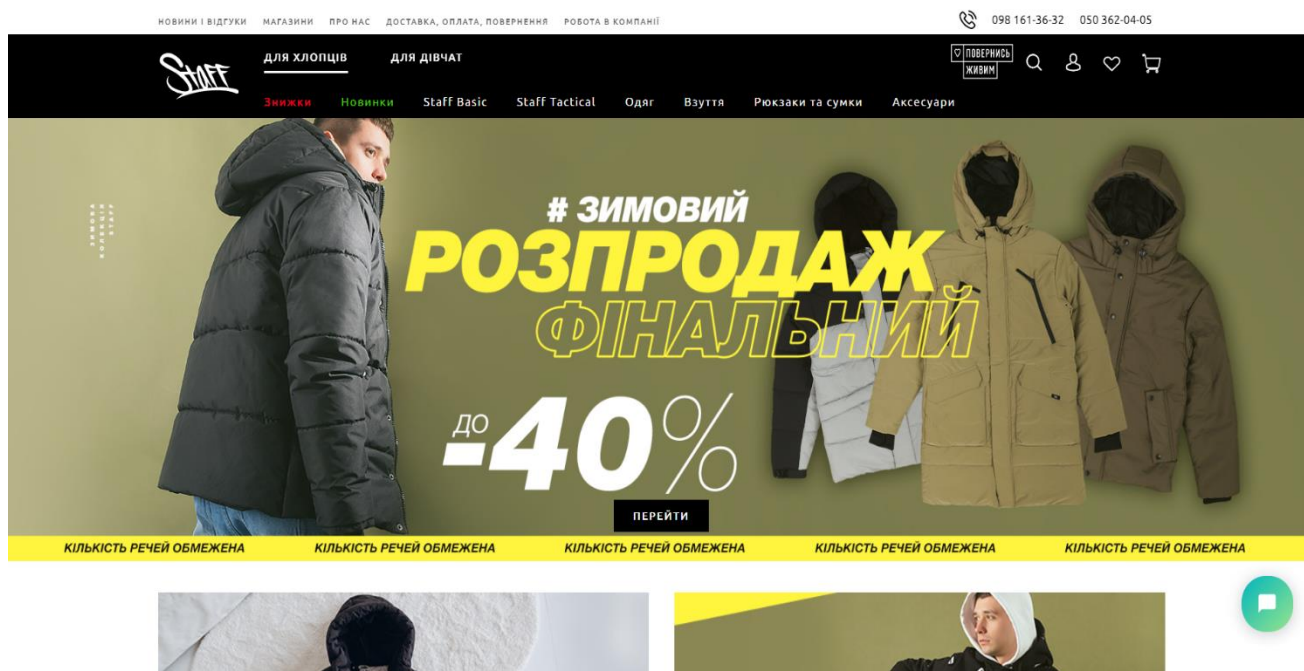


Рисунок 1.2 – Інтернет магазин «Staff»

В нього багато переваг, а саме великий функціонал в плані вибору та пошуку товару, багато фільтрів які дозволяють максимально відсортувати потрібний користувачеві товар, також можливість посортувати товар за його новизною, рейтингом і ціною, що досить часто стає в пригоді. Оскільки це магазин одягу, то потрібно звернути увагу на функціонал «опції товару» за допомогою якого можна вибрати розмір, колір тощо.

Схожу тематику і призначення також має сайт «George» (Рис 1.3).

Перевагою даного сайту є досить зручний інтерфейс, можливість додавати коментарі та можливість збільшити фото товару. Однак цей сайт має і недоліки, а саме відсутність зміни кольору одягу, обмежена кількість доступних розмірів. Крім того, кожен товар не перенасичений описом.

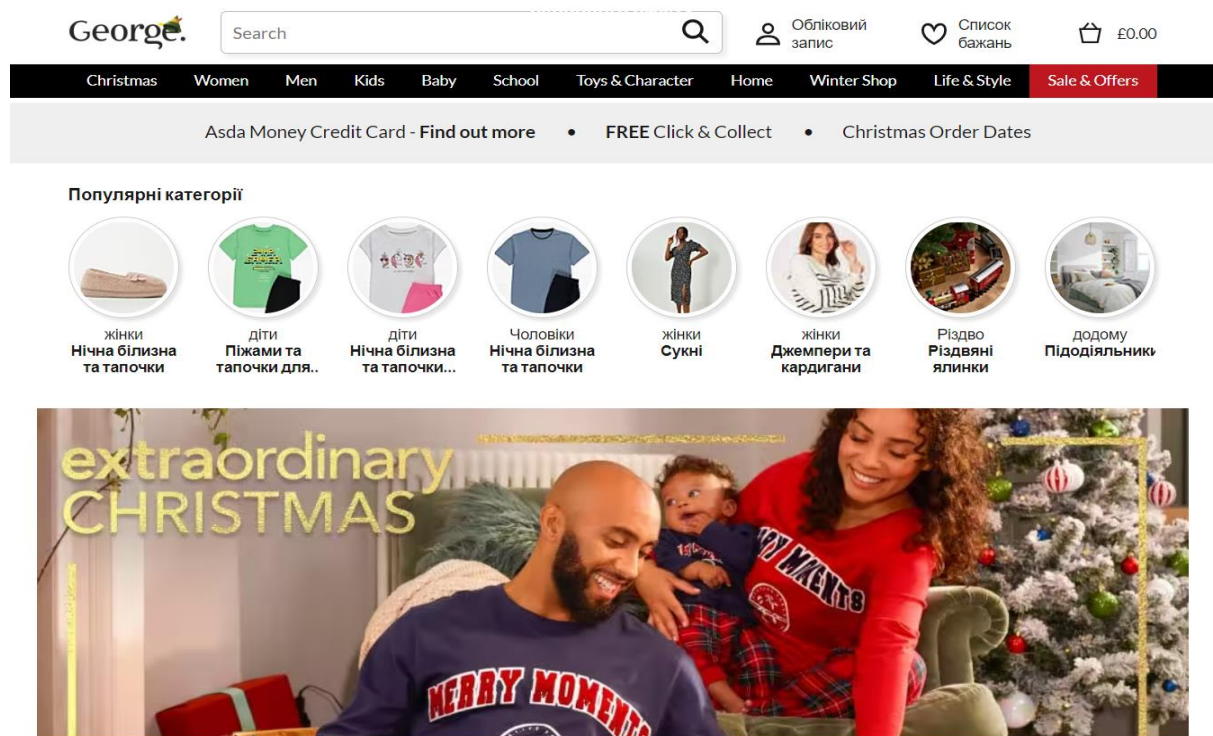


Рисунок 1.3 – Інтернет магазин одягу «George»

Також проаналізовано сайт одягу «SHOPSTYLE» (Рис. 1.4).

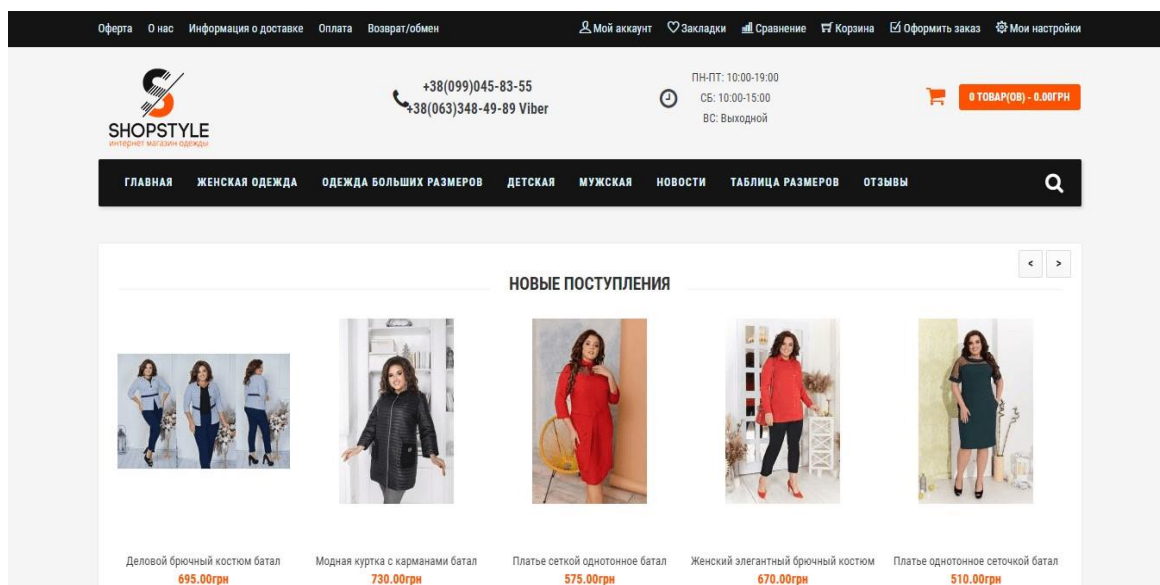


Рисунок 1.4 – Головна сторінка сайту «SHOPSTYLE»

Як і попередні проаналізовані сайти, цей сайт також має недоліки, а саме неможливість вибрати розмір одягу та колір. Крім того, відсутність докладного опису продукту не дозволяє оцінити його в повному обсязі.

На сьогоднішній день розроблено безліч таких додатків, але вони мають

певні недоліки.

Проаналізувавши більшість вже розроблених програм, можна виділити кілька недоліків, а саме

- Неоновлювання інформація. Розвиток компанії триває, щодня зміна асортименту та наявність товару на складі, тривають промоушени, але бракує часу, що поновити дані.
- Простота використання. Поширена помилка розробників: потенційний клієнт не знає, як оформити замовлення, де натиснути кнопку отримання умов доставки і де набрати номер, щоб обраний товар доставили йому додому.
- Застарілий дизайн.
- Повільна обробка замовлень.

1.2 Проектування програмного забезпечення

1.2.1. Структура побудови інтернет-магазину

Зараз існує багато платформ електронної комерції, які пропонують покупцям різноманітні онлайн-ресурси на вибір.

Якщо покупець вибере один з них, йому не доведеться довго чекати відповіді від сервера або адміністратора, а, можливо, закриє вкладку і вибере більш зручний і швидкий онлайн-ресурс для покупки товару.

Чим більше часу покупець заощадить, працюючи з вашим сайтом, тим більше шансів стати клієнтом цього магазину.

Не менш важливо, щоб покупцям було комфортно користуватися магазином, і має бути налагоджена відповідна система.

Це своєрідний «фреймворк», який точно вказує, що потрібно включити на сторінку, щоб нею було легше користуватися.

Наприклад, враховуючи онлайн-ресурс, який містить корисну для користувачів інформацію, якщо текст не створений так, щоб бути приємним для

людського ока, користувач, швидше за все, відмовиться від використання цього порталу.

Те ж саме відбувається і з інтернет-магазинами.

Клієнти відмовляться від використання ресурсів, якщо потрібно затратити багато часу на потрібну інформацію.

Тому, розробка структури вашого інтернет-магазину є одним із найважливіших кроків.

Структура сайту є результатом ретельної роботи, свідомого дизайну та чіткої організації.

Це передбачає побудову блоків сайту у вигляді дерев для надання йому практичної та логічної структури [6].

Розглядаючи магазин більш детально, то проглядається принципова схема розташування блоків категорій, товарних карток, кошиків, особистих кабінетів, розсилок тощо.

Отже, розглядаючи планову структуру сайту, бачимо, що: картинка стає чіткішою, показується доцільність окремих блоків, зрозумілішими можливі зміни.

Коли ви оцінюєте загальний вигляд ваших онлайн-ресурсів продажів, ви можете прийти до висновку, що більшість ресурсів містять однакові функціональні частини, і що сайт має унікальний зовнішній вигляд.

При розробці сайту особлива увага приділяється головній сторінці, яка повинна бути наочною і більше відображати те, що шукають самі покупці.

Щоб покупці відчували себе на вашому сайті комфортно і здійснювали дії, які допоможуть вибрати потрібний товар, ваш сайт, перш за все, повинен мати логічну і продуману структуру.

Якщо ваш сайт структурований неправильно, важливі дані на вашому сайті можуть бути візуально втрачені, тому ви не тільки втратите клієнтів, але й важливі дані на вашому сайті.

Тому головним завданням є проектування гнучкої конструкції.

На веб-сайтах існує типи структур організації інформації: лінійна (послідовна), ієрархічна, мережева та комплексна [20 - 26].

Лінійна структура. Найпоширеніший і найпростіший спосіб подання інформації – це її заповнення.

Ця структура зазвичай зустрічається в книгах, журналах тощо.

Такі структури можуть бути від загального до конкретного, як-от довідники, словники, енциклопедії тощо, або мати логічне хронологічне розташування тем в алфавітному порядку.

Завдяки своїй конфігурації лінійна структура ідеально підходить для навчальних закладів, де користувачі навчаються.

Очікується, що сайт виконуватиме послідовні операції над набором матеріалів і повертатиме відповідь, яка містить лише посилання, що підтримують лінійний шлях навігації.

Веб-сайти з послідовною структурою вимагають від користувачів покрокового перегляду вмісту [7].

Комбінована структура. Цей тип структури є комбінацією двох різних типів структури.

Наприклад, ваш сайт може мати розроблену ієрархічну структуру, але при проектуванні деяких сторінок використовувати цю структуру нелогічно, тому ви можете використовувати іншу структуру, яка є більш відповідною (наприклад, лінійну структуру), щоб замінити такі процеси .

Однак найчастіше веб-сайти з такою структурою виглядають абсолютно неорганізованими.

Такі сайти дозволяють переходити від сторінки до сторінки абсолютно різними способами і виявляють, що вам не потрібно повертатися на потрібну сторінку тим самим способом [7].

Тому, проаналізувавши типи структур сайту, ми прийняли складену структуру, яка може створити просту та легку у використанні структуру.

Структура інтернет-магазину з продажу етнічних костюмів представлена на рисунку 2.1.

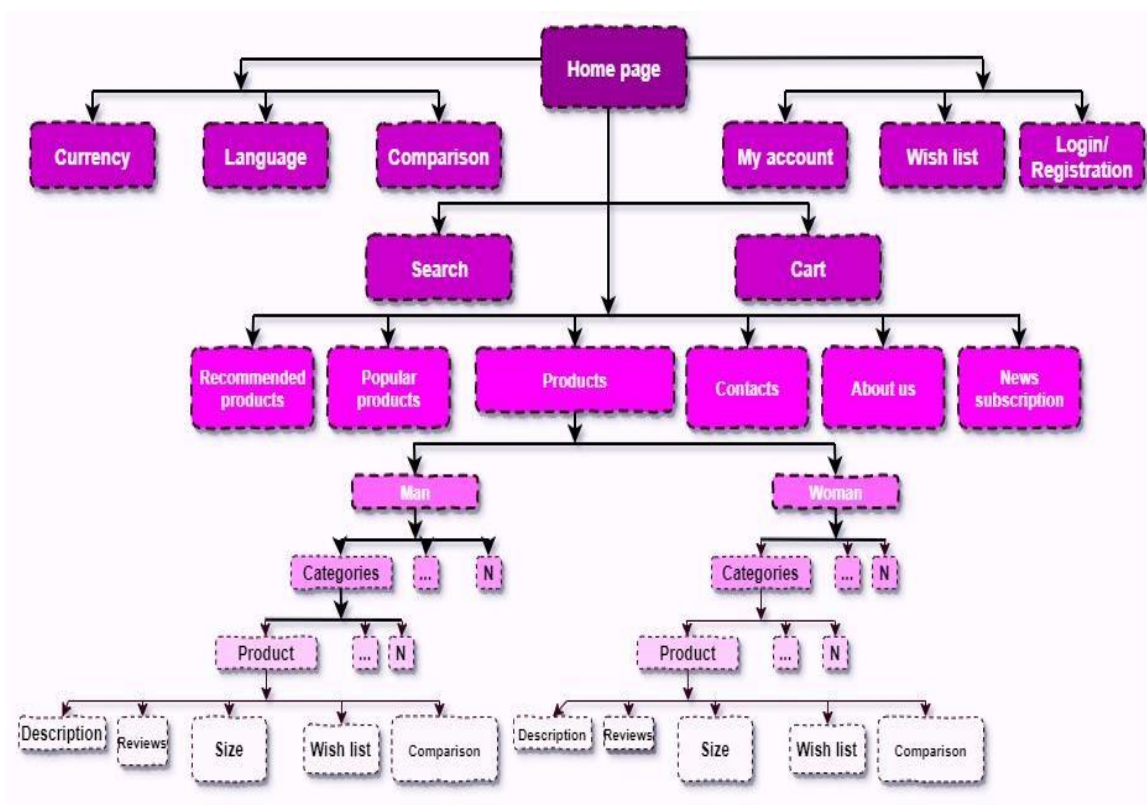


Рисунок 1.5 – Проектування структури Інтернет магазину для покупця

Інтернет-магазин, як уже зазначалося вище, схожий на магазин реального часу, з тією лише різницею, що він працює 24 години на добу та має різноманітні поділилися можливістю використовувати цю функцію.

Якщо ви проаналізуєте роботу вуличного магазину, то виявите, що всю діяльність, пов'язану з основною роботою з покупцями, тобто продаж продукції, просування продукції, закупівлі, здійснюють менеджери або менеджери-консультанти.

Так само при розробці структури інтернет-магазину було поставлено не менш важливе завдання: розробка структури адмінпанелі.

Всі необхідні операції, пов'язані з реалізацією електронної комерції та взаємодією з веб-ресурсами покупця, здійснюються через панель адміністрування.

На рисунку 1.6 показано структуру адміністративної області, придатної для введення вмісту та обліку порядку.

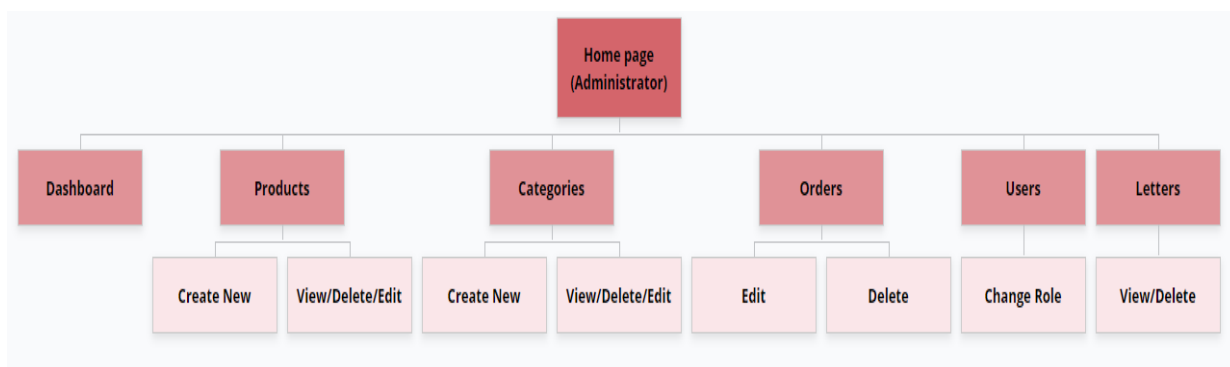


Рисунок 1.6 – Проектування структури адміністративної панелі

1.2.2. Моделювання основних принципів роботи web-сайту.

Для того, щоб сторінка була функціональною та зрозумілою, нам спочатку потрібно розробити для неї найкращу мову UML.

Мова UML — це мова візуального моделювання загального призначення, розроблена для специфікації, візуалізації, проектування та документування програмних компонентів, бізнес-процесів та інших систем.

Мова UML базується на кількох фундаментальних концепціях, які більшість програмістів і розробників можуть вивчити та застосувати за допомогою об'єктно-орієнтованого аналізу та проектування [8].

UML — це мова моделювання, а не процес розробки програмного забезпечення.

Це твердження підтверджується специфікацією UML, оскільки процес проектування описаний наступним чином:

- дає вказівки про порядок діяльності команди.
- вирішує, які артефакти розробити.
- керує окремими завданнями розробника та цілими командами.
- пропонує стандарти для моніторингування та вимірювання продукту та проектної діяльності.
- служить методом вирішення задач об'єктно-орієнтованого моделювання

систем [8].

Прикладом є RationalUnifiedProcess (RUP).

Дизайн діаграми UML в основному використовується на всіх етапах життєвого циклу бізнес-процесу та розробки прикладних програм.

Мови графічного моделювання підтримують різні типи діаграм, а також пропонують широкий спектр можливостей для опису та представлення різних аспектів складних систем, що робить цю мову універсальним засобом опису програмних проектів.

UML використовується в різних сферах застосування: у банківській справі, фінансах, Інтернеті, аерокосмічній сфері, охороні здоров'я тощо [33].

На рисунку 1.7 зображено діаграму класів інтернет-магазину «Вишиванка».

Діаграма класів включає основні класи User, Administrator, Product, Wishlist, Order, Cart, Message, їхні атрибути, операції та асоціації.

Неzareєстровані користувачі (клієнти) мають можливість:

- налаштування параметрів сторінки відповідно до побажань: зміна валюти та мови сторінки;
- перегляд каталогу продукції;
- пошук потрібного товару;
- перегляд характеристик продукту;
- визначити розмір;
- уточнення необхідної інформації про товар (зворотна форма магазину);
- додавати товар у кошик;
- створити свій акаунт.

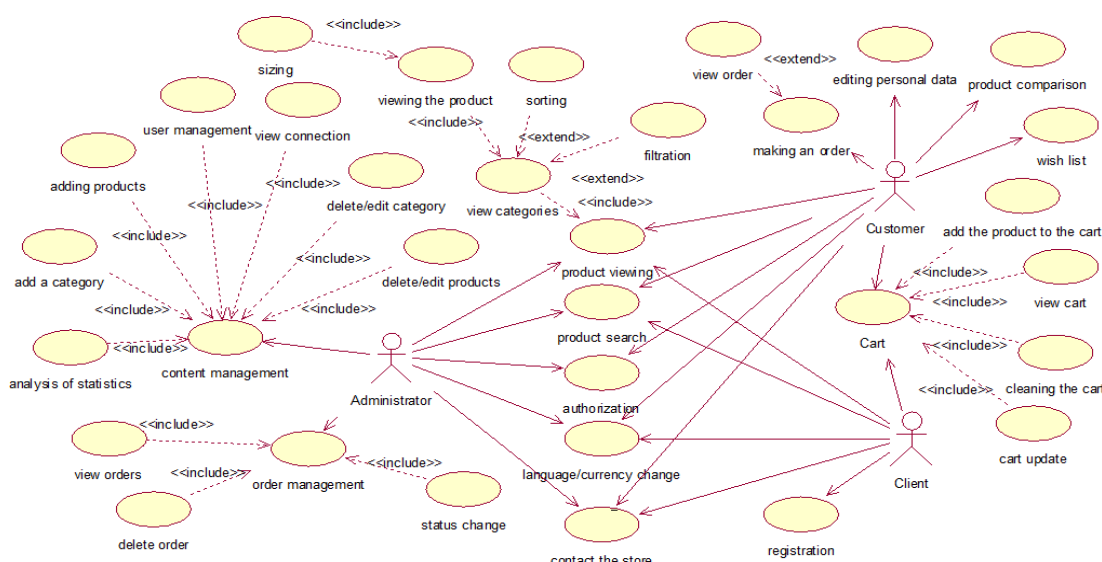


Рисунок 1.7 – Діаграма варіантів використання

У можливості зареєстрованого користувача – Customer включаються всі можливості не зареєстрованого користувача, а також додаються нові функції:

- авторизація;
- редагування профілю;
- список побажань;
- порівняльна характеристика товару;
- управління корзиною.

Адміністратор (менеджер) – Administrator має всі функції зареєстрованого користувача, а також додаються нові функції:

- керування контентом;
- керування замовленнями.

Розглянемо функцію керування контентом. До даної функції входять ще декілька, а саме: статистичний аналіз, додати категорію, товар та їх перегляд.

Перейдемо до розгляду функції керування замовленнями – видалення, змінити статус, переглянути замовлення.

Для відображення інтерактивності об'єктів поведінки системи у мові UML використовують діаграму послідовності. Зручність використання полягає в тому, що відображаються зв'язки між об'єктами в послідовному порядку.

На рисунку 1.8 зображено діаграму послідовності процесу «Оформлення замовлення». На ній є об'єкти: Customer, Web site, Web server та DB. Вертикаль лінії – час роботи конкретного об'єкта.

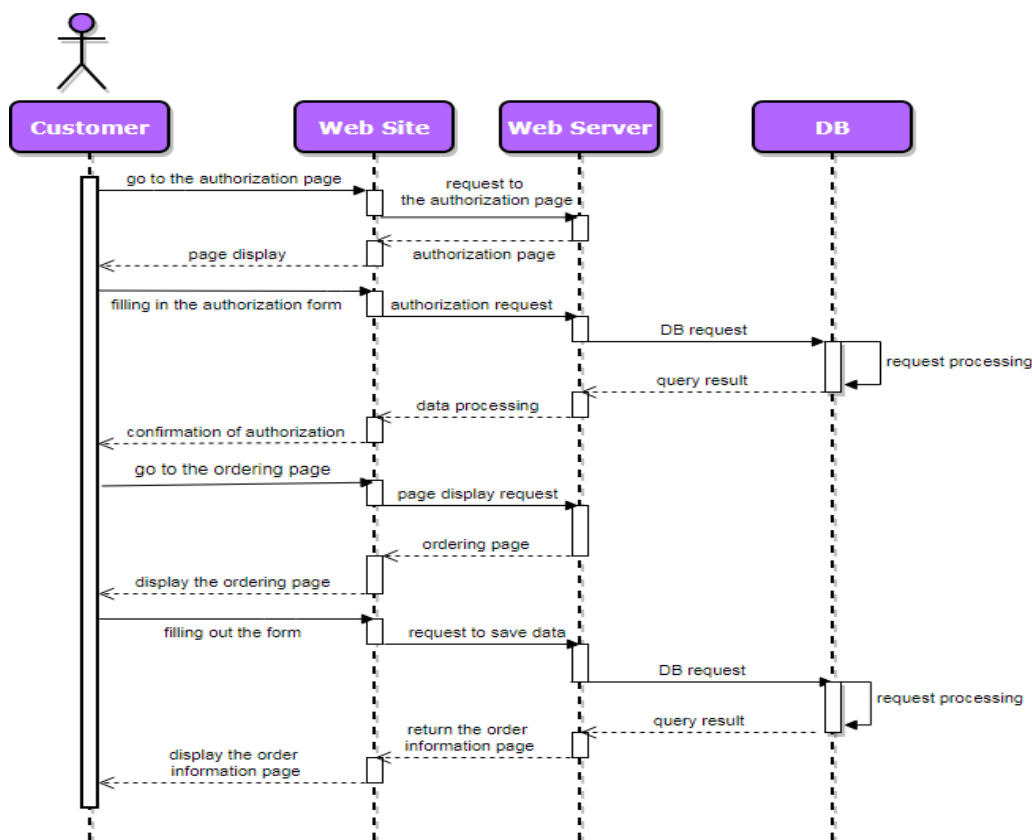


Рисунок 1.8 – Діаграма послідовності для процесу «Оформлення замовлення»

Послідовність дій проходить наступним чином: користувач натискає на кнопку авторизації. Після цього попадає на вебсайт, вебсайт передає запит на сторінку на сервер, після цього сервер видає сторінку, відображає її користувачеві, користувач здійснює заповнення форми авторизації та натискає кнопку підтвердження. Далі запит іде на веб-сервер, веб-сервер надсилає запит до бази даних, потім здійснюється обробка запиту, наступним є отримання результату з БД веб-сервером, після чого відбувається опрацювання даних, а як результат веб-сервер повертає підтвердження про успішну авторизацію. Покупець переходить на сторінку оформлення замовлення, відповідно на веб- сервер відправляється запит на відображення сторінки, а як результат запиту повертається сама сторінка оформлення замовлення і відповідно користувачеві її інтерфейс. Наступною дією

користувача є внесення даних у відповідні форми, на веб-сервер передається запит на збереження даних, після чого відбувається запит на базу даних, проходить обробка даних, на веб-сервер повертається результат про успішне внесення, а веб-сервер повертає сторінку з інформацією про замовлення, яка відображається клієнтові.

1.3 Програмна реалізація інтернет-магазину

1.3.1. Вибір та огляд програмних засобів реалізації проекту.

Вибір типу архітектури (схема взаємодії між компонентами) веб-додатків та складової моделі є одним із найважливіших, але складних у їх розробці [20 - 26]. Спосіб планування взаємодії визначає стійкість, ефективність та безпеку майбутнього веб-додатка [13].

Архітектура інтернет-магазину етнічного одягу є 3-рівнева Клієнт-Сервер-БД.

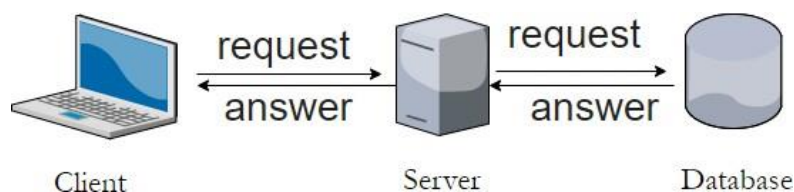


Рисунок 1.9 – Архітектура «Клієнт-сервер» інтернет-магазину

Особливістю є розташування та виконання додатку на сервері, а клієнт бачить результат. Важливо, що клієнт не залежить від ОС конкретного користувача, що робить веб-додаток міжплатформеним [26].

Веб-додаток, на стороні сервера, виконує запит та його обробку, надсилає відповідь клієнту.

Оскільки, робота браузера (клієнт) та респондента (сервер) відбувається за кодами, то при виборі мови потрібно базуватися на їх взаємодії. На стороні сервера як правило застосовують:

- мови / рамки включають, але не обмежуються ними, Ruby (Rails), JS (Node.js), Python (Django), PHP, C# , Java тощо;
- збереження стійкості даних;
- формування відповіді HTTP за URL-адресою;
- створення сторінки користувача [14].

При розробці даного проекту використовується для серверної частини – фреймворк FuelPHP, а для клієнтської частини – ReactJS [15, 16, 17].

1.3.2. Розробка основних функцій магазину.

Проект має клієнт-серверну архітектуру, тому для розробки клієнтської частини було взято framework ReactJS, а для серверної частини – framework FuelPHP. Розглянемо основні функції із сторони клієнта і з сторони сервера та їх взаємодію між собою.

На рисунку 1.10 відображено частину коду для авторизації на боці клієнта. використаємо асинхронну функцію авторизації – `async()`, щоб отримати позитивний негативний результат. Для цього потрібно ввести логін та пароль.

```

DoLogin = async () => {
  if (this.state.email && this.state.password) {
    const user = {
      email: this.state.email,
      password: this.state.password,
      login_errors: undefined
    }
    await axios.post('http://diplom:8080/server/user/login', JSON.stringify({
      email: user.email,
      password: user.password
    })))
    .then(async (res) => {
      var data = res.data
      if (data.errors) {
        this.setState({ login_errors: data.errors })
      } else if (data.usertoken) {
        cookie.save('usertoken', data.usertoken, { path: '/' })
        await this.props.setUserToken(data.usertoken)
        await axios.post('http://diplom:8080/server/user/data', JSON.stringify({
          token: cookie.load('usertoken')
        })))
        .then(async (res) => {
          this.setState({
            user_group: res.data[0].group
          })
          cookie.save('user', res.data[0], { path: '/' })
          await this.props.compareUpdate()
        })
        this.props.setLoginPopUp(false)
      }
    })
  } else {
    this.setState({ errors: "Some of inputs are empty" })
  }
}

```

Рисунок 1.10 – Метод авторизації на стороні клієнта

Реєстрація реалізована з використанням `asunc()`. Для внесення даних до БД використано POST-запит, передача даних на сервер відбувається в форматі JSON (див. рис. 1.11).

Рисунок 1.12 відображає код для отримання курсу валюти з використанням GET-запиту і конструкції `asunc/await`.

```

onSubmitRegister = async (e) => {
  e.preventDefault()

  var user = {
    email: e.target.email.value,
    password: e.target.password.value
  };
  await axios.post('http://diplom:8080/server/user/register', JSON.stringify({
    username: e.target.username.value,
    email: e.target.email.value,
    password: e.target.password.value
  })))
  .then(async (res) => {
    var data = res.data
    if (data.errors) {
      this.setState({ register_errors: data.errors })
    } else {
      this.setState({
        email: user.email,
        password: user.password,
        register_errors: undefined
      })
      await this.DoLogin();
    }
  })
}

```

Рисунок 1.11 – Метод реєстрації на стороні клієнта

```

await axios.get('https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json&valcode=USD')
  .then(res => {
    this.setState({
      usd_rate: res.data[0].rate
    })
  })

```

Рисунок 1.12 – Оновлення валюти за курсом «Національного банку України»

Частина програмного коду для додавання товару до кошика показано на рисунку 1.13.

```
onCartAdd = async (e, id, quantity, size) => {
  e.preventDefault()
  await axios.get('http://diplom:8080/server/products/single?id=' + id)
    .then(res => {
      var cart = cookie.load('cart') ? cookie.load('cart') : []
      var repeated = false
      var data = res.data
      cart.map((item, key) => {
        if (item.id === id) {
          repeated = true
          cart[key]['quantity']++
        }
      })
      if (repeated === false) {
        data.quantity = quantity ? quantity : 1
        data.size = size
        cart.push(data);
      }
      this.setState({ cart: cart, cart_count: Number(this.state.cart_count) + Number(data.new_price * data.quantity) }, () => {
        cookie.save('cart', cart, { path: '/' })
        cookie.save('cart_count', this.state.cart_count, { path: '/' })
      })
    })
}
```

Рисунок 1.13 – Функція додавання товару до корзини, на стороні клієнта

Щоб видалити товар з кошика використано функцію onCartDelete() (див. рис. 1.14).

```
onCartDelete(e, key) {
  e.preventDefault()

  var deleted = this.state.cart.splice(key, 1)

  var newcart = this.state.cart

  this.setState({
    cart: newcart,
    cart_count: Number(this.state.cart_count) - Number(deleted[0].new_price * deleted[0].quantity)
  }, function () {
    cookie.save('cart', this.state.cart, { path: '/' })
    cookie.save('cart_count', this.state.cart_count, { path: '/' })
  })
}
```

Рисунок 1.14 – Функція «Видалити товар»

На рисунку 1.15 відображено функцію «Додати товар» з використанням preventDefault(). Для цього користувач має бути зареєстрованим.

```

addToWish = async (e, id) => {
  e.preventDefault()
  var msg
  if (this.state.usertoken === undefined) {
    msg = "Login or register to add this item"
  } else {
    await axios.post('http://diplom:8080/server/wishlist/add', JSON.stringify({
      model_id: id,
      user_id: cookie.load('user').id
    })).then(res => {
      if (res.data.code === 'success') {
        msg = "You have successfully added an item to wishlist"
      } else {
        msg = "This product is already in your wishlist"
      }
    })
  }
  this.setState({
    popup_message: msg,
    popup_show: true
  })
}

```

Рисунок 1.15 – Функція «Додати товар», клієнтська частина

Щоб порівняти товари використано функцію показану на рисунку 1.16. Аналогічно використано `preventDefault()`. Потім проводимо перевірку щодо кількості товару. При перевищенні двох позицій система повідомляє про перевищення ліміту. Також перевірка унеможливорює порівняння товарів різних категорій.

```

addCompare = async (e, id, cat_id) => {
  e.preventDefault()
  if (this.state.compares.count >= 2) {
    this.setState({
      popup_show: true,
      popup_message: "Limit of items to compare is 2"
    })
  } else {
    if (cookie.load('user')) {
      if (this.state.compares.data.length === 1 && this.state.compares.data[0].category_id !== cat_id) {
        this.setState({
          popup_show: true,
          popup_message: "You can not add two products with different categories"
        })
        return
      }
      await axios.get('http://diplom:8080/server/user/add_compare?id=' + id + '&user_id=' + cookie.load('user').id)
        .then(async (res) => {
          await this.compareUpdate()
        })
    } else {
      this.setState({
        popup_show: true,
        popup_message: "Log in or register to add this product to compare list"
      })
    }
  }
}

```

Рисунок 1.16 – Функція додавання товару до списку порівняння, клієнтська частина

1.3.3. Інструкція користувача.

Результатом кваліфікаційної роботи є розроблений інтернет-магазин етнічного одягу. Вимоги до сайту: зрозумілість та зручність інтерфейсу, можливості для керування.

Головна сторінка сайту поділена на частини.

Можливість обрати мову, валюту, авторизуватись, переглянути корзину, оформити замовлення розміщені у першій частині сайту. Також передбачено пошук товарів (див. рис. 1.17).



Рисунок 1.17 – Головна сторінка

Меню, рекламний банер та коротка інформація про сайт знаходяться на 2 частині сторінки.

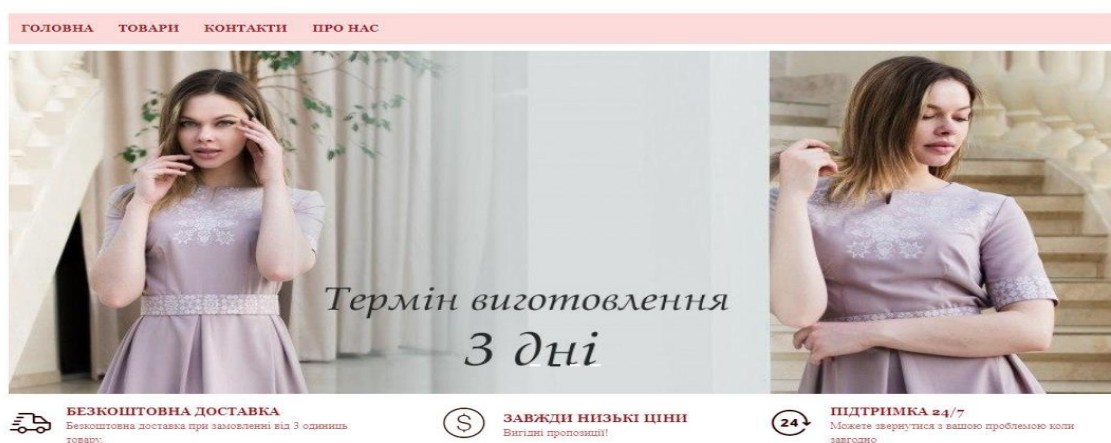


Рисунок 1.18 – Друга частина головної сторінки

Третя частина містить рекомендовану продукцію сайту та популярні товари.

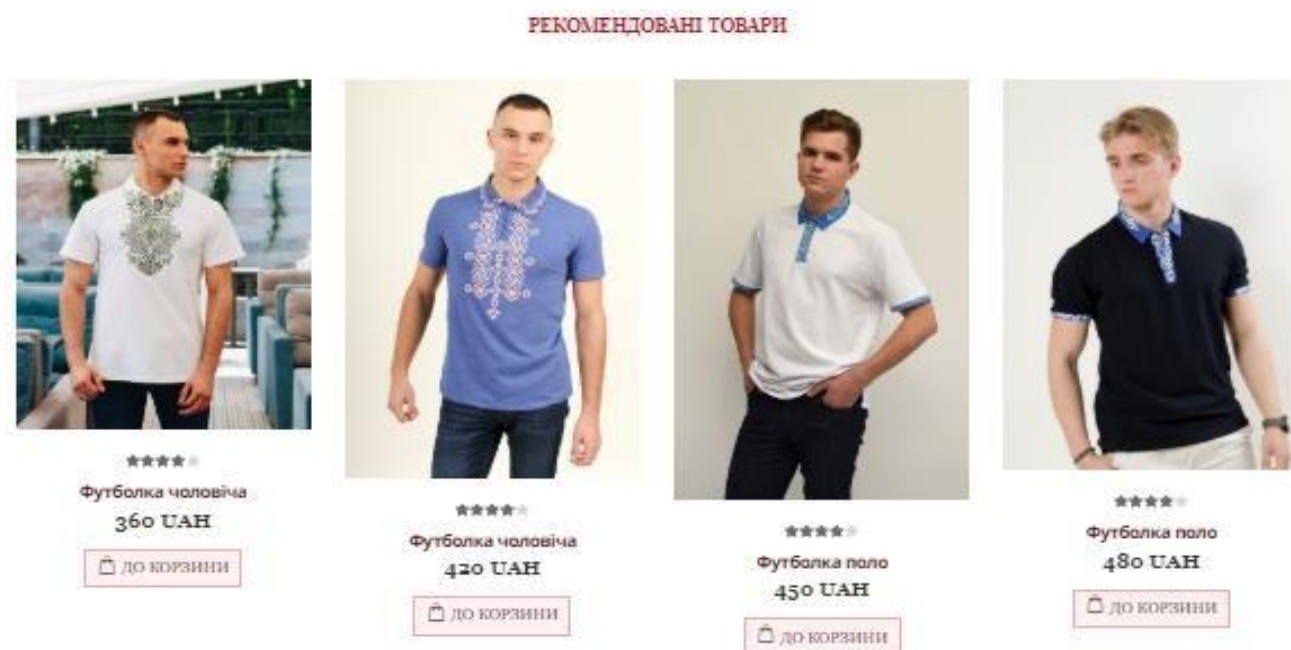


Рисунок 1.19 – Рекомендовані товари на головній сторінці

Для входу на головній сторінці відкривається форма, яку необхідно заповнити (див. рис. 1.20). Користувач має пройти реєстрацію або авторизуватись для входу на сайт.

АВТОРИЗАЦІЯ

Email address *

Password *

LOGIN

[Forgot your password?](#)

РЕЄСТРАЦІЯ

Your First and Last name *

Email address *

Password *

REGISTER

Рисунок 1.20 – Форма авторизації, реєстрації

Авторизований користувач може перейти на свою сторінку (див. рис. 1.21).

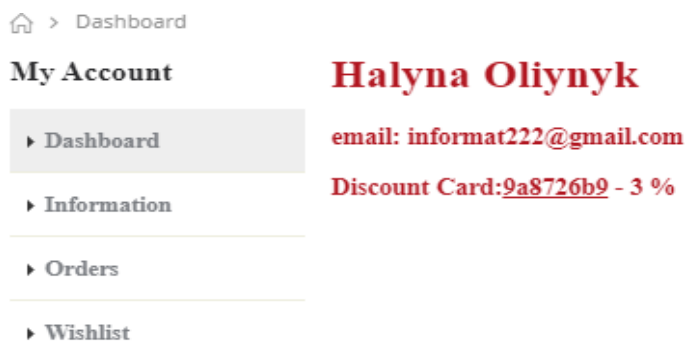


Рисунок 1.21 – Обліковий запис користувачів

Маючи свій обліковий запис, клієнт може змінити свої дані та здійснити перегляд історії замовлень (див. рис. 1.22).

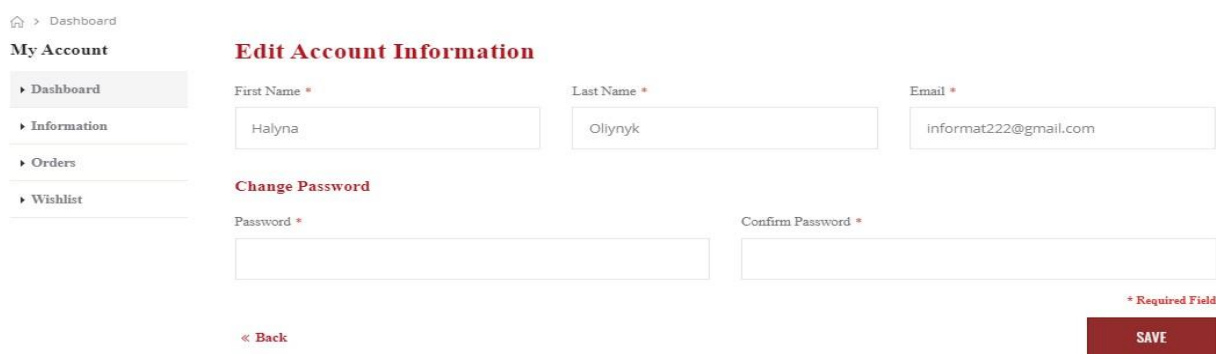


Рисунок 1.22 – Зміна особистих даних

Перелік товарів є на головній сторінці. Він поділений за категоріями. За допомогою фільтру можна вибрати товар за розміром, відсортувати по ціні тощо (див. рис. 1.23).

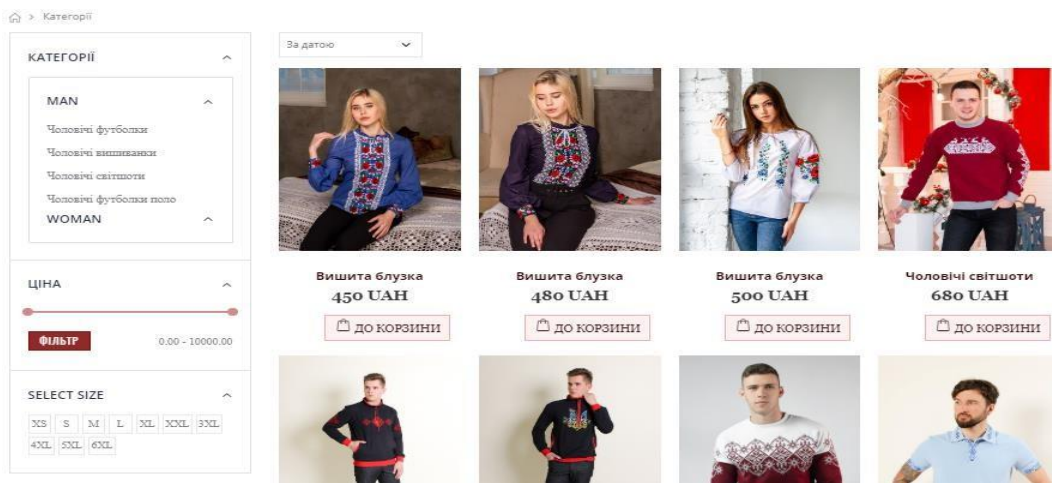


Рисунок 1.23 – Відображення товарів

Для детальної інформації про обраний товар, після натискання його зображення, відкривається вікно із нею (див. рис. 1.24).



Рисунок 1.24 – Сторінка відображення детальної інформації про товар

Наступним кроком, після вибору розміру є формування замовлення (див. рис. 1.25). Це можливо лише для авторизованого користувача. Для нових відвідувачів передбачена форма реєстрації. Також на сайті діють акції у вигляді промокодів, які дають знижку на товар.



Рисунок 1.25 – Корзина

Щоб оформити замовлення потрібно заповнити форму та обрати спосіб доставки товару.

Замовити

Ім'я *

Прізвище *

Email адреса *

Методи замовлення

☐ Фізсована
 ☐ Самовивіз

☐ За тарифом
 ☐ Нова пошта

Виберіть місто

ПІДТВЕРДИТИ

Загалом

1 PRODUCTS IN CART



Футболка чоловіча

420 UAH

Рисунок 1.26 – Заповнення форми «Оформлення замовлення»

Передбачено фіксація відгуку на сторінці сайту (див. рис. 1.27).

ДОДАЙТЕ ВАШ КОМЕНТАР

Як ви оціните даний товар? *

	1 ЗІРКА	2 ЗІРКИ	3 ЗІРКИ	4 ЗІРКИ	5 ЗІРОК
Якість	☐	☐	☐	☐	☐
Опілка	☐	☐	☐	☐	☐
Ціна	☐	☐	☐	☐	☐

Ім'я *

Email адреса *

Що ви думаєте? *

ВІДПРАВИТИ

Рисунок 1.27 – Відгуки про товар

Клієнти мають доступ до розділу інформації, де після заповнення відповідної форми, можуть отримати відповідь на виниклі запитання.

[🏠](#) > [Зв'яжіться З Нами](#)

Напишіть Нам

Ім'я *

Email адреса *

Номер телефону

Що ви думаєте? *

ВІДПРАВИТИ

Рисунок 1.28 – Форма зв'язку «Напишіть нам»

[🏠](#) > [About Us](#)



ПРО НАС

Почавши роботу як інтернет-магазин, ми з часом почали відшивати власні моделі, і на даний момент велика частина продукції яка представлена на сайті нашого власного виробництва, чим ми дуже пишаємося. У нас представлений широкий асортимент жіночих вишиванок і чоловічих вишитих сорочок, вишитих тунік, костюмів, жилетів та іншої продукції. Працюємо ми як у роздріб, так і опт. Асортимент постійно оновлюється, тому Ви завжди знайдете у нас щось для себе, або для своїх близьких, друзів, колег чи знайомих.

Також вас приємно здивують низькі ціни на нашу продукцію. Ваше замовлення буде відправлено до Вас максимально швидко, не залежно від місця Вашого проживання. Ми працюємо для Вас, і дорожимо своєю репутацією. Тому ми впевнені в правильності Вашого вибору. Хороших покупок Вам і Вашим близьким!



Рисунок 1.29 – Сторінка «Про нас»

1.4 Висновок до розділу 1

У розділі проаналізовано тему на основі літературних джерел та визначено її актуальність. Аналіз аналогів для визначення недоліків та переваг конкурентів. У процесі роботи було досліджено тему. Оптимізація веб-інтерфейсу та навігації сайту, щоб користувачеві було легше орієнтуватися у віртуальному просторі.

У майбутньому система може бути легко розширена завдяки добре продуманій архітектурі, створювати нові функції, що відповідають потребам зростання користувачів, та бути продуктивною при великих навантаженнях.

2. ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

2.1 Тестування інтерфейсу

Тестування інтернет-магазину необхідне забезпечення якості роботи системи. Без тестування можуть виникнути помилки або неполадки, які призведуть до негативного досвіду покупців та зменшення продажу в інтернет-магазині. [20, 21, 22, 23, 24, 25, 26].

В Rational Unified Process (RUP) тестування є ітеративним процесом. В основному є дві основні ролі в процесі тестування, розробник тестів та сам тестер.

Розробник формує план та модель тестування, методику випробувань, оцінює результати тестування. Тестер відповідає за виконання системного тестування.

Невід'ємною частиною RUP є артефакти – деякі продукти проєкту, породжувані або використовуються в ньому при роботі над кінцевим продуктом. В процесі тестування також створюються артефакти

План тестування – документ, що визначає стратегію тестування кожній ітерації. В ньому описано цілі та завдання процесу та використовувані стратегії. Також у плані вказано необхідні ресурси та наведено перелік тестів.

Щодо моделі тестування, то вона містить набір контрольних завдань, методик випробування, сценаріїв випробувань і очікуваних результатів, тестових скриптів і описів взаємодій тестів:

- контрольне завдання – тестові дані, умови проходження тестів, очікувані результати;
- методика – вказівки щодо контрольних тестів та їх результатів;
- сценарій тестування – це спрощений опис тесту;
- тестовий скрипт – програма, яка запускається при автоматизованому тестуванні;
- діаграма послідовностей – взаємозв'язок тестів та ПЗ.

Очевидно, що в процесі тестування виявляються його результати і дані, отримані при самому виконанні тестів. Модель робочого навантаження

використовується для моделювання зовнішніх функцій, які виконуються кінцевими користувачами, обсягів цих функцій і навантаження, що створюється цими функціями. Модель призначається для проведення навантажувального тестування.

Як ми бачимо, при найближчому розгляді, процес тестування – цілком визначений процес з певними ролями і зоною відповідальності для різних учасників проекту.

Було вирішено розпочати тестування даного проекту з сторінок авторизації та реєстрації, та перевірити реагування системи на правильно та неправильно введені паролі під час цих процесів. На початку, під час реєстрації, було введено не вірний пароль та пароль для підтвердження (див. рис 2.1).

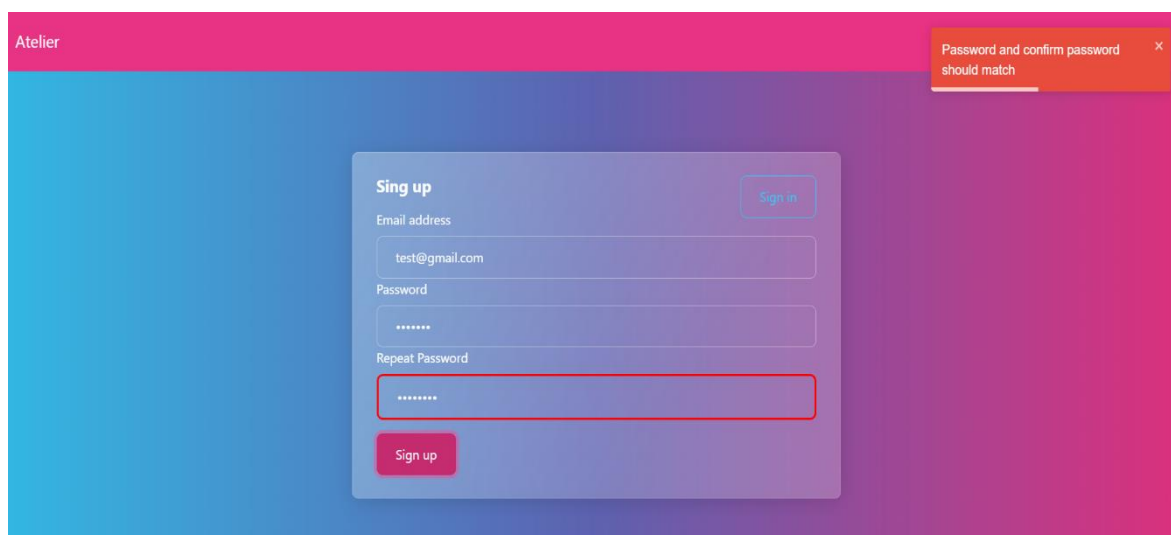


Рисунок 2.1 – Не правильно введений пароль для підтвердження

Отже, як бачимо, не можна зареєструватись, якщо пароль для підтвердження введено не правильно.

Наступним тестом було введення всі даних правильно (див. рис 2.2).

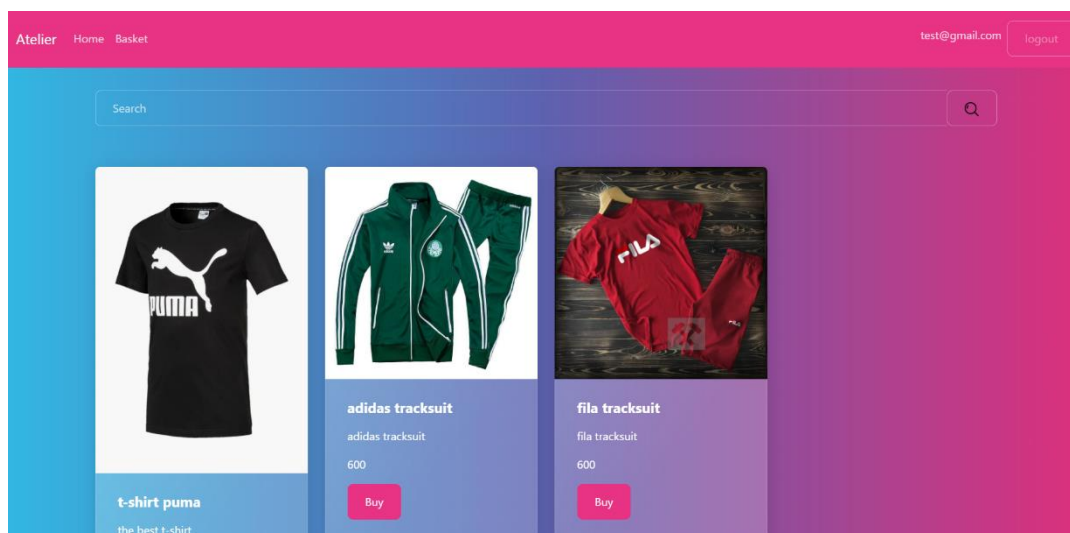


Рисунок 2.2 – Головна сторінка яку бачить користувач після авторизації

Таким чином, після авторизації, якщо користувач не був запрошений адміністратором, то він отримає роль покупця.

Наступним кроком була перевірка корзини, вона повинна бути порожня оскільки покупець ще не добавляв жодного товару в неї (див. рис 2.3).

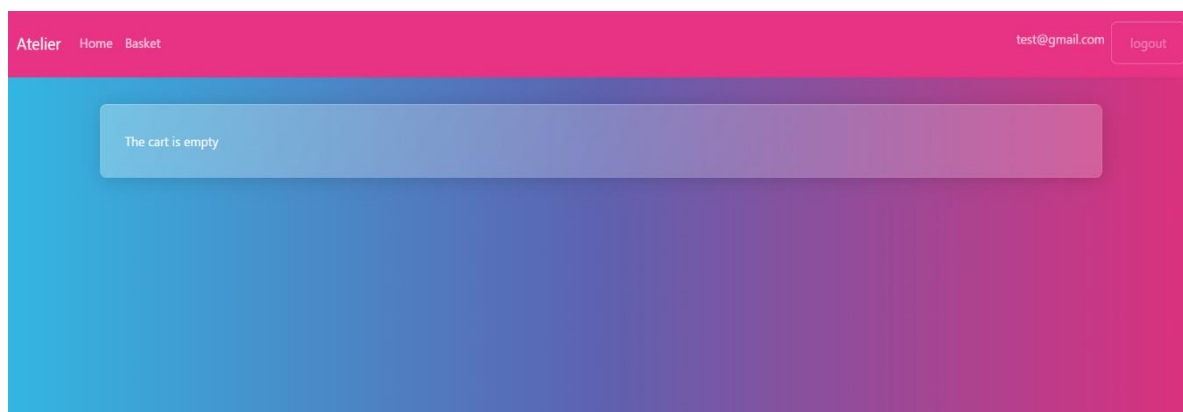


Рисунок 2.3 – Сторінка кошика, коли покупець не добавив жодного продукту

Отже, якщо користувач не додав жодного продукту в кошик, то він побачить повідомлення про це.

Впевнившись в правильності роботи корзини, було перейдено до тестування пошуку товарів (див. рис 2.4).

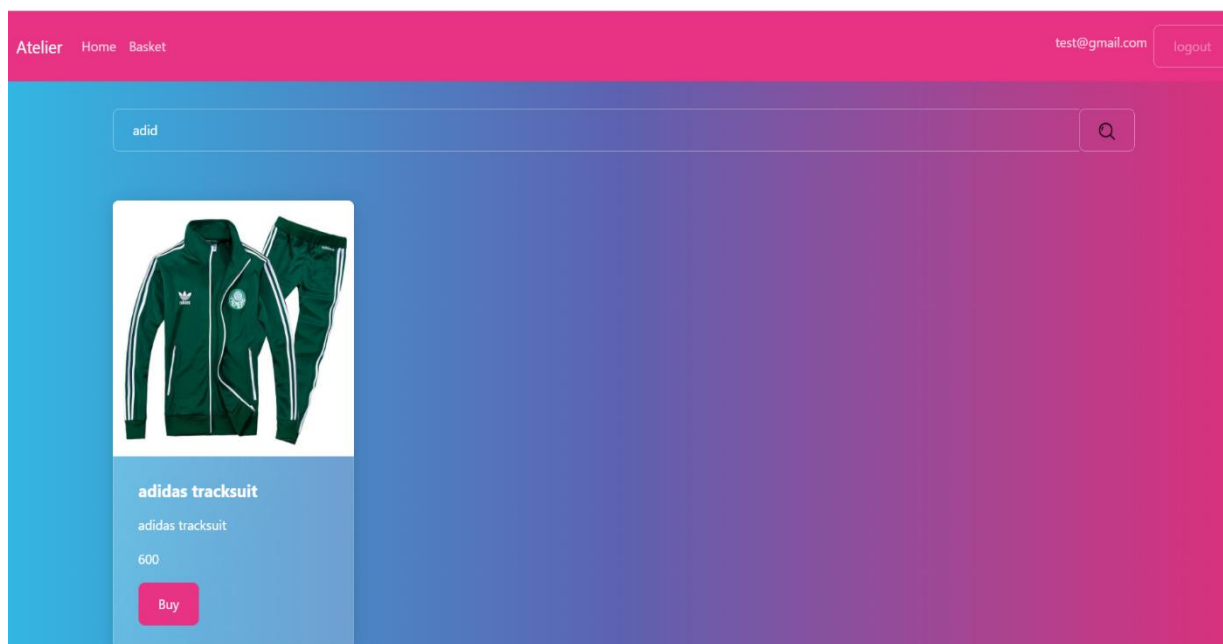


Рисунок 2.4 – Тестування пошуку товарів

Як видно, пошук товарів по ключовому слову працює правильно.

Далі додаємо товар в корзину (див. рис. 2.5)

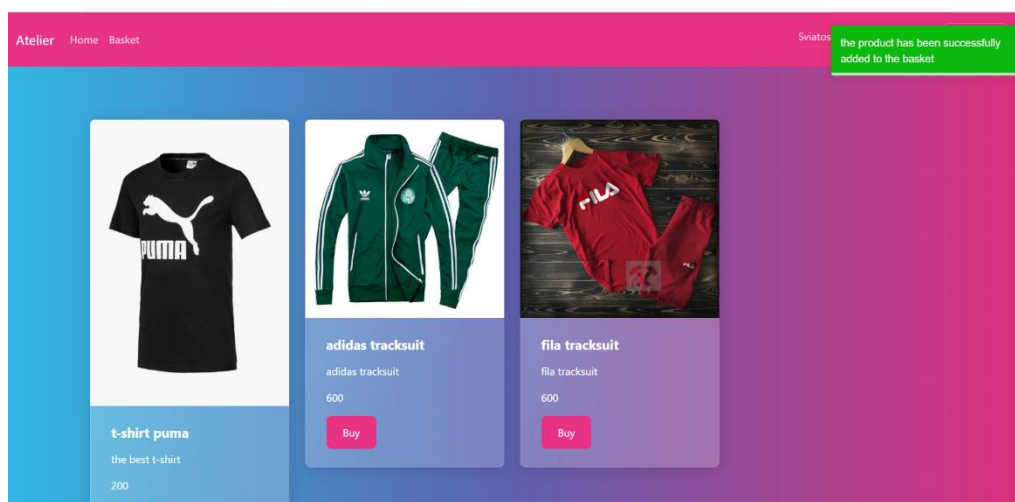


Рисунок 2.5 – Додавання товару в корзину

На рисунку відображено додавання товару в корзину, наступним кроком буде перехід на сторінку корзини, та перегляд, чи товар був добавлений (див. рис. 2.6).

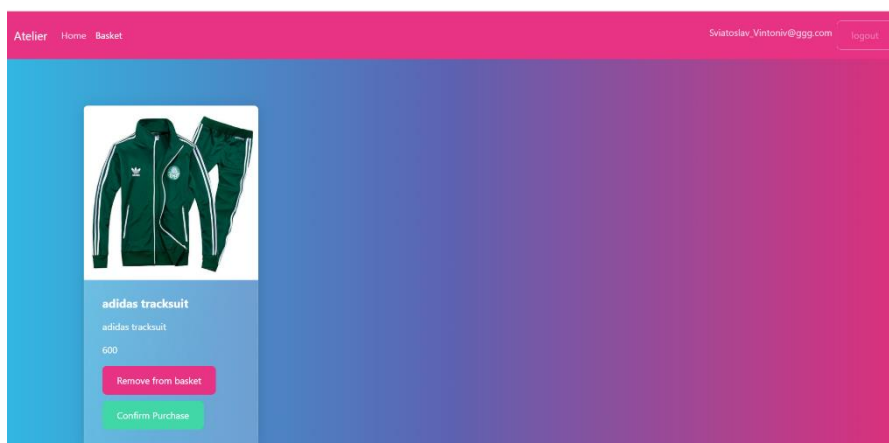


Рисунок 2.6 – Сторінка кошику

Як видно з рисунку 2.6 товар було додано в корзину, спробуємо підтвердити покупку (див. рис. 2.7).

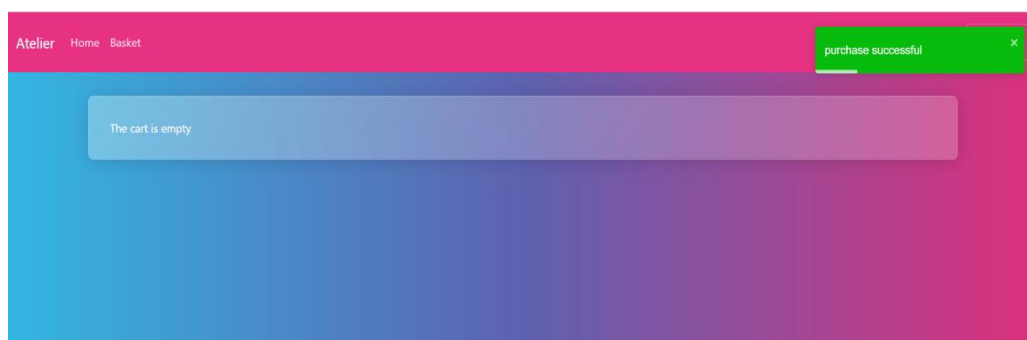


Рисунок 2.7 – Підтвердження покупки

Як можна бачити з рисунку, після підтвердження покупки можна побачити відповідне повідомлення, та те, що корзина очистилась.

2.2 Відслідковування багів

Ефективне відслідковування та управління багами є вирішальними для досягнення успіху проекту та забезпечення його якості [20, 21, 22, 23, 24, 25, 26].

У даному розділі буде детально проведений пошук багів та їх

документування у вигляді баг-репортів.

Таблиця 2.1 – Баг-репорт про реєстрацію нового облікового запису

Summary	Помилка реєстрації нового облікового запису
Severity	S.1
Priority	P.1
Description	Preconditions: - Не увійдено в обліковий запис; - Підключення до інтернету. Step to reproduce: 1. Відкрити сайт; 2. Вибрати опцію «Увійти» у гедері сайту; 3. Ввести електронну пошту та пароль у відповідні поля реєстраційної форми; 4. Пройти перевірку на робота; 5. Клацнути на кнопку «Реєстрація».
Expected Result	Новий обліковий запис буде успішно створено, користувач зможе виконувати усі передбачені функції та входити у свій обліковий запис.
Actual Result	Після натискання кнопки «Реєстрація», сторінка починає виконувати дію і зависає. Після спроби перезавантажити сторінку введені дані пропадають.

Таблиця 2.2 – Баг-репорт про завантаження меню.

Summary	Помилка завантаження меню.
Severity	S.1
Priority	P.1
Description	Preconditions: - Наявність підключення до інтернету. Step to reproduce: 1. Відкрити сайт закладу; 2. Клацнути на кнопку «Меню» на панелі навігації сайту;
Expected Result	Користувач перейде на сторінку з меню, яке успішно завантажиться з бази даних.
Actual Result	Користувач перейшов на сторінку з меню, однак меню відображається некоректно, його вигляд неструктурований, замість тексту різні символи, неправильне кодування виведеного тексту.

Таблиця 2.3 – Баг-репорт про додавання відгуку.

Summary	Помилка додавання відгуку
Severity	S.2
Priority	P.2
Description	Preconditions: - Наявне підключення до інтернету. Step to reproduce: 1. Відкрити сайт закладу; 2. Обрати пункт навігації «Контакти»; 3. Ввести необхідні дані; 4. Натиснути кнопку «Надіслати».
Expected Result	Повідомлення буде надіслано та його зможуть прочитати адміністрація закладу.
Actual Result	Відгук не приходить на скриньку.

Таблиця 2.4 – Баг-репорт про створення замовлення.

Summary	Помилка відтворення створення замовлення.
Severity	S.1
Priority	P.1
Description	Preconditions: - Користувач увійшов у обліковий запис; - Наявне підключення до інтернету. Step to reproduce: 1. Обрати пункт «Замовити онлайн» у гедері сторінки; 2. Обрати страву, її кількість та клацнути на кнопку «Додати в кошик»; 3. Перейти до кошика; 4. Оформити замовлення;
Expected Result	Замовлення буде створено, усі обрані страви будуть в ньому відображатися, автоматично виведеться сума замовлення та його можна буде оплатити.
Actual Result	Усі обрані страви були додані до кошика, підрахувалася сума замовлення, однак при натисканні на кнопку «Оформити замовлення» з'являється помилка, що необхідно додати хоча б одну позицію в кошик. Неможливо створити замовлення.

Таблиця 2.5 – Баг-репорт про додавання товару в корзину.

Summary	Помилка додавання товару з відсутніми даними
Severity	S.3
Priority	P.2
Description	Preconditions: - Відкрито адмін-панель сайту; - Наявність підключення інтернету. Step to reproduce:

Продовження таблиці 2.5 – Баг-репорт про додавання товару в корзину.

	1. Відкрити адмін-панель сайту; 2. Вибрати опцію «Меню»; 3. Вибрати одне з доступних меню; 4. Вибрати категорію меню; 5. Клацнути на кнопку «Додати», не заповнивши жодного поля.
Expected Result	Буде показана помилка, що необхідно заповнити поля.
Actual Result	Жодної помилки чи застереження не було показано. Пусту страву було додану до бази даних і вона відображається користувачам у меню після оновлення сторінки з меню.

Таблиця 2.6 – Баг-репорт про додавання нової адреси користувача.

Summary	Помилка додавання нової адреси користувача
Severity	S.2
Priority	P.2
Description	Preconditions: - Користувач увійшов в обліковий запис; - Наявність підключення інтернету. Step to reproduce: 1. Відкрити сайт закладу; 2. Увійти у свій обліковий запис; 3. Тицьнути на іконку свого профілю та вибрати пункт «Мої адреси»; 4. Тицьнути на кнопку «Додати нову адресу»; 5. Заповнити всі необхідні дані; 6. Нажати кнопку «Зберегти».
Expected Result	Нова адреса буде додана та прив'язана до профілю. При замовленні можна буде обрати цю адресу.
Actual Result	Після збереження, адреса не відображається у профілі користувача та не можна обрати її при новому замовленні.

2.3 Навантажувальне тестування

Навантажувальне тестування є важливою складовою процесу тестування, що дозволяє оцінити продуктивність та стійкість ПЗ під високим навантаженням. У випадку, коли сайт підключений до бази даних Firebase Realtime Database,

проведення навантажувального тестування бази даних є необхідним, оскільки швидкість відклику сайту залежить від продуктивності бази даних.

Для проведення навантажувального тестування було обрано браузерний ресурс – BlazeMeter. BlazeMeter – це хмарна платформа для тестування навантаження, що надає можливість тестування веб-додатків та API, з метою визначення продуктивності та масштабованості ваших додатків. BlazeMeter дозволяє створювати та запускати тестові сценарії з великою кількістю користувачів з різних точок світу, щоб перевірити, як додаток поводить себе при великому навантаженні. BlazeMeter також надає різноманітні інструменти аналізу результатів тестування, що дозволяють знайти та виправити проблеми з продуктивністю додатка.

За допомогою цього ресурсу, було проведено навантажувальне тестування двічі, перший раз з навантаженням 10 користувачів, другий раз – 50. Результат тестування показано на рисунках 2.1 і 2.2.

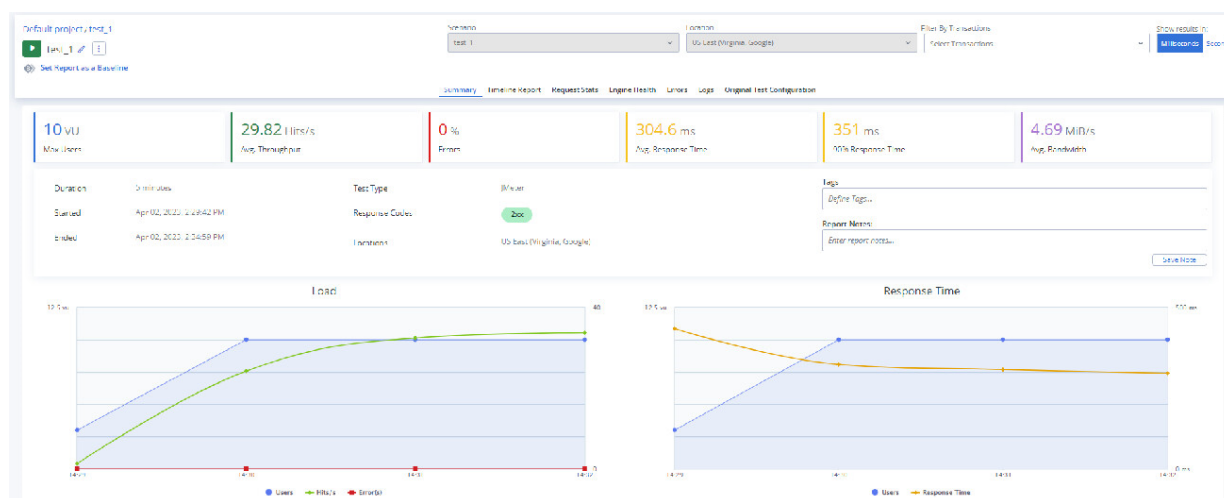


Рисунок 2.1 – Результат першого тестування

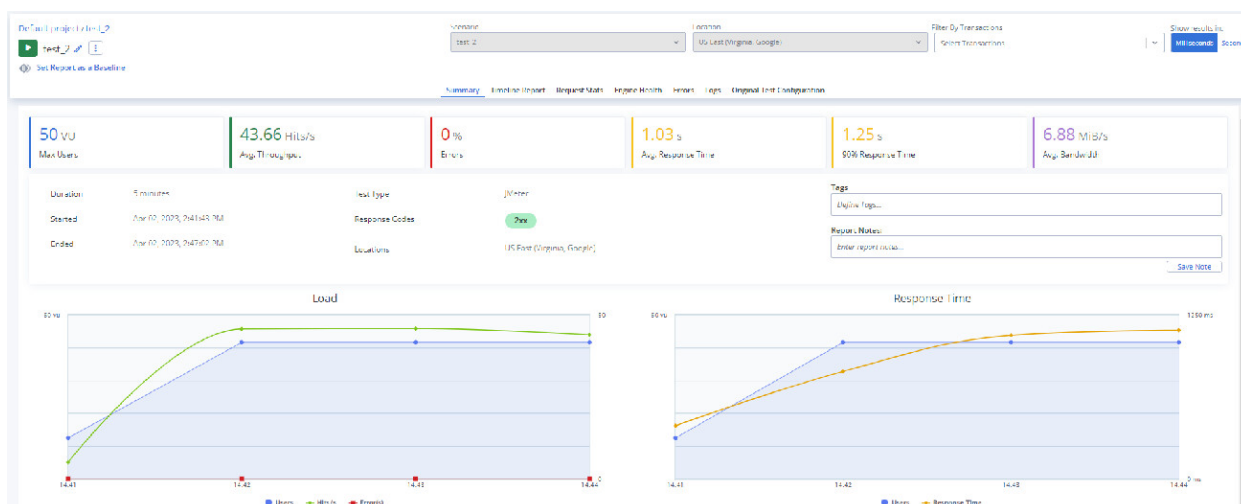


Рисунок 2.2 – Результат другого тестування

Отже, за результатами двох тестувань можна зробити висновок, що обрана база даних в реальному часі працює коректно з різним навантаженням, адже і при одночасній роботі 10 користувачів, і при одночасній роботі 50 користувачів сайт не видавав помилок. Також можна зауважити, що при збільшенні кількості користувачів збільшився і середній час реакції та відгуку, проте це збільшення є незначним та не впливає на роботу бази даних. При збільшенні кількості користувачів, збільшилась також середня пропускна здатність, що не є дивним, адже її збільшення дозволяє обслуговувати більше запитів користувачів, що допомагає забезпечити швидший доступ до бази даних при більшому навантаженні.

2.4 Автоматизоване тестування

За допомогою Selenium IDE можна легко створювати автоматизовані тестові скрипти без необхідності в програмуванні. Він дозволяє записувати дії, такі як натискання кнопок, введення тексту, переходи на сторінки, перевірку наявності елементів і багато іншого. Після запису скрипту його можна відтворити в будь-який час, щоб перевірити, чи відбувається очікуване поведінка веб-додатку.

Крім записування тестів, Selenium IDE також надає можливість редагування скриптів, додавання умов, циклів та перевірок. Він також підтримує використання змінних, що дозволяє створювати більш гнучкі тестові сценарії. Більш технічним користувачам також доступний режим скриптування за допомогою мови програмування JavaScript.

Selenium IDE може бути корисним інструментом для швидкого створення простих автоматизованих тестів та невеликих проектів. Однак, важливо враховувати, що Selenium IDE має свої обмеження і не підходить для складних тестових сценаріїв або проектів, де потрібно розширене програмування та інтеграція з іншими інструментами.

За допомогою цього розширення було складено 9 тестів, які перевіряють певний базовий функціонал сайту. У кожному першим відкривається нове вікно браузера зі встановленим розширення 1872x931 пікселі.

1. Додавання товар у кошик.

Відкривається нова сторінка, де обирається товар, підтверджується його кількість та натискається кнопка «Додати в кошик».



The screenshot shows a Selenium IDE test suite for the URL `https://pgrytsyshyn.wikisite.com/sobre/order-online/`. The test consists of five steps:

	Command	Target
1	✓ open	<code>https://pgrytsyshyn.wikisite.com/sobre</code>
2	✓ set window size	<code>1936x1056</code>
3	✓ click	<code>linkText=ЗАМОВИТИ ОНЛАЙН</code>
4	✓ click	<code>css=o_3uNQXz-focus</code>
5	✓ click	<code>css=#restaurants-orders-entity-66db8294-a16b-4bea-b662-838b8a26168f.NqoqfG.nth-child(2).oqOjCTl--typography-4-p2-m</code>

Рисунок 2.3 – Вигляд тесту додавання товару в кошик.

2. Прибирання товару з кошика.

Відкривається кошик де клацає на кнопку «Видалити товар» та після чого підтвердити дію, клацнувши на кнопку «Так».

https://pgrytyshyn.wixsite.com/sobre-online/		
Command	Target	
1 ✓ open	https://pgrytyshyn.wixsite.com/sobre-online/	
2 ✓ set window size	1972x931	
3 ✓ click	css=#restaurants-orders-entity-66db8294-a16b-4bea-b662-838ba26168f NqoqGnth-child (2) .oq3c7l--typography-10-header-ss	
4 ✓ mouse over	css=#restaurants-orders-entity-66db8294-a16b-4bea-b662-838ba26168f NqoqGnth-child (2) .oq3c7l--typography-10-header-ss	
5 ✓ mouse out	css=#restaurants-orders-entity-66db8294-a16b-4bea-b662-838ba26168f NqoqGnth-child (2) .oq3c7l--typography-10-header-ss	
6 ✓ click	css=VgltUP	
7 ✓ click	css=a___3uH2/cz-focus_s___15H2m8	
8 ✓ click	css=sZ9gVJh	
9 ✓ click	css=a___3uH2/cz-focus_s___15H2m8	
10 ✓ click	css=a7b9wD > svg	

Рисунок 2.4 – Вигляд тесту з видалення товару з кошика.

3. Написання відгуку.

У гедері, в меню навігації клацається на кнопку «Контакти», після чого сторінка скролиться донизу, де є форма у яку заповнюються відомості: ім'я, пошта, телефон та повідомлення. Потім натискається кнопка «Відправити».

https://pgrytyshyn.wixsite.com/sobre		
Command	Target	Value
1. ✓ open	https://pgrytyshyn.wixsite.com/sobre	
2. ✓ set window size	1972x931	
3. ✓ click	id=comp-hqbw4t4abel	
4. ✓ click	id=input_comp-jabqpk	
5. ✓ type	id=input_comp-jabqpk	Іван
6. ✓ click	id=input_comp-jabqpw	
7. ✓ type	id=input_comp-jabqpw	test@gmail.com
8. ✓ click	id=input_comp-jabqrb	
9. ✓ type	id=input_comp-jabqrb	+380952123123
10. ✓ click	id=textarea_comp-jabqre	
11. ✓ type	id=textarea_comp-jabqre	Чудово!
12. ✓ click	css=#comp-jabqre > .JuTaQy	

Рисунок 2.5 – Вигляд тесту написання відгуку.

4. Підписка на розсилку.

У гедері, в меню навігації клацається на кнопку «Контакти», після чого сторінка скролиться донизу, де є поле для вводу електронної пошти, де вводиться пошта та натискається кнопка «Підписатися».

https://pgrytyshyn.wixsite.com/sobre		
Command	Target	Value
1. ✓ open	https://pgrytyshyn.wixsite.com/sobre	
2. ✓ set window size	1972x931	
3. ✓ click	id=comp-hqbw4t4abel	
4. ✓ click	id=input_comp-jus4nsax	
5. ✓ type	id=input_comp-jus4nsax	test@gmail.com
6. ✓ click	id=mediaa04g845	
7. ✓ click	css=#comp-jus4nsax > .JuTaQy	

Рисунок 2.6 – Вигляд тесту підписки на розсилку.

5. Кнопка «Догори».

У гедері, в меню навігації клацається на кнопку «Контакти», після чого сторінка скролиться донизу, де є кнопка, яка скролить сторінку догори після

натискання.

https://pgrytsyshyn.wixsite.com/sobre		
	Command	Target
1	✓ open	https://pgrytsyshyn.wixsite.com/sobre
2	✓ set window size	1872x931
3	✓ click	id=comp-ihq9vovf4label
4	✓ click	css=.aizul7 > svg

Рисунок 2.7 – Вигляд тесту кнопки, що скролить догори.

6. Надсилення повідомлення в чаті.

Після відкривання вікна браузера, натискається на значок чату, що з'являється у правому нижньому куті, після чого розгортається чат, в поле повідомлення вписується текст на натискається кнопка «Надіслати».

https://pgrytsyshyn.wixsite.com/sobre			
	Command	Target	Value
1	✓ open	https://pgrytsyshyn.wixsite.com/sobre	
2	✓ set window size	1872x931	
3	✓ select frame	index=2	
4	✓ click	id=minimized-chat	
5	✓ click	css=_3nFws	
6	✓ type	css=_3nFws	Birao
7	✓ click	css=S66IV path	

Рисунок 2.8 – Вигляд тесту надсилення повідомлення в чат.

7. Додавання адреси.

У гедері тиснем на іконку профілю, де у випадаючому меню обирається пункт «Моя адреса». Після цього переходить на нову сторінку, де натискається кнопка «Додати нову адресу». З'являється модальне вікно, у якому заповнюються необхідні дані адреси та натискається кнопка «Зберегти».

https://pgrytsyshyn.wixsite.com/sobre			
	Command	Target	Value
4	✓ click	css=li:nth-child(1).aj_souU	
5	✓ select frame	index=4	
6	✓ click	css=_L26nkp > _L172js	
7	✓ select frame	relative=parent	
8	✓ select frame	index=3	
9	✓ type	id=firstName-field	test
10	✓ type	id=lastName-field	test
11	✓ type	id=company-field	test
12	✓ type	id=addressLine1-field	test
13	✓ type	id=addressLine2-field	test
14	✓ type	id=city-field	test
15	✓ type	id=zipCode-field	12321
16	✓ type	id=phone-field	+380521231231
17	✓ click	css=_L1pmwm	

Рисунок 2.9– Вигляд тесту додавання нової адреси.

8. Видалення адреси.

Клікаєм на іконку профіля, де у випадаючому меню обирається пункт «Моя адреса». Після цього переходить на нову сторінку, де натискається кнопка «Видалити адресу» та кнопка підтвердження.

https://pgrytsyshyn.wixsite.com/sobre		
	Command	Target
1	✓ open	https://pgrytsyshyn.wixsite.com/sobre
2	✓ set window size	1872x931
3	✓ mouse over	css=.xldRa
4	✓ mouse out	css=.xldRa
5	✓ click	css=.GVJl6y > svg
6	✓ click	css=li:nth-child(1) .sj_soU
7	✓ select frame	index=4
8	✓ click	css=li:nth-child(2) .hidden-mobile
9	✓ click	css=_10o0_nth-child(2) > button

Рисунок 2.10 – Вигляд тесту видалення адреси.

9. Зміна імені користувача.

Після натискання на іконку профілю, у випадаючому меню обирається пункт «Моя профіль». Після цього переходить на нову сторінку, де у полі «Ім'я» вписується нове ім'я та натискається кнопка «Оновити дані».

https://pgrytsyshyn.wixsite.com/sobre			
	Command	Target	Value
1	✓ open	https://pgrytsyshyn.wixsite.com/sobre	
2	✓ set window size	1872x931	
3	✓ mouse over	css=polygon	
4	✓ click	css=polygon	
5	✓ mouse out	css=polygon	
6	✓ click	css=li:nth-child(3) .sj_soU	
7	✓ click	id=display-name-id	
8	✓ type	id=display-name-id	pavlo1
9	✓ click	css=.owgc24j-focus.sW0Fste	

Рисунок 2.11 – Вигляд тесту зміни імені користувача.

2.5 Висновки до розділу 2

У розділі при відслідковуванні багів ПЗ було проведено пошук та відслідковування розноманітних багів, помилок чи небажаних результатів під час роботи бізнес логіки вебсайту. Було створено 6 баг-репорт, де детально задокументовано усі відомості про помилку: які передумови її виникнення повинні бути, які кроки, щоб цю помилку відтворити мають бути зроблені та чим відрізняється отриманий результат виконання частини бізнес логіки, в якій було виявлено помилку, від очікуваного результату, який закладений у дану функцію.

Під час навантажувального тестування створено тестові сценарії, які симулюють реальне навантаження на додаток, включаючи одночасний доступ багатьох користувачів і велику кількість запитів. Отримані дані про продуктивність, використання ресурсів та відгуки допомагають виявити проблеми та прийняти відповідні заходи для їх вирішення.

Таким чином, навантажувальне тестування є необхідним етапом в процесі розробки веб-додатку для забезпечення його якості та високої продуктивності. Результати тестування дозволяють виявити та виправити проблеми, що можуть виникнути під час реального використання додатку, та забезпечити задоволення користувачів від його використання.

Автоматизоване тестування використовувалося для автоматизації та прискорення процесу тестування. За допомогою інструментів, таких як Android Studio та Firebase Test Lab, були розроблені та запущені автоматизовані тести, що сприяло покращенню ефективності та точності тестування. Для проведення автоматизованого тестування ПЗ був використаний спеціальний інструмент Selenium IDE для автоматизації процесу тестування. З його допомогою було створено 10 автоматизованих тестів, які покликані тестувати основний та не надто складний функціонал сайту.

3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Управління силами та засобами об'єктів господарювання під час надзвичайних ситуацій.

Найбільш ефективний спосіб зменшення шкоди та збитків від надзвичайних ситуацій – запобігти їх виникненню, а в разі виникнення виконувати відповідні до даної ситуації заходи.

Запобігання виникненню надзвичайних ситуацій – це підготовка та реалізація комплексу заходів, спрямованих на регулювання безпеки, проведення оцінки рівнів ризику, завчасне реагування на загрозу виникнення надзвичайної ситуації на основі даних моніторингу (спостережень), експертизи, досліджень та прогнозів щодо можливого перебігу подій з метою недопущення їх переростання у надзвичайну ситуацію або пом'якшення її можливих наслідків.

Зазначені функції запобігання надзвичайним ситуаціям техногенного і природного характеру в нашій країні покликана виконувати Єдина державна система цивільного захисту (ЄДСЦЗ), затверджена Постановою Кабінету Міністрів України від 9 січня 2014 р. №11. ЄДСЦЗ включає в себе центральні та місцеві органи виконавчої влади, виконавчі органи рад, державні підприємства, установи та організації з відповідними силами і засобами, які здійснюють нагляд за забезпеченням техногенної та природної безпеки, організують проведення роботи із запобігання надзвичайним ситуаціям і реагування у разі їх виникнення з метою захисту населення і довкілля, зменшення матеріальних втрат. ЄДСЦЗ складається з постійно діючих функціональних та територіальних підсистем і має чотири рівні: загальнодержавний, регіональний, місцевий та об'єктовий. Кожен рівень ЄДСЦЗ має координуючі та постійні органи управління. Координуючими органами ЄДСЦЗ є:

на загально державному рівні:

- Державна комісія з питань техногенно-екологічної безпеки та надзвичайних ситуацій;

- Національна рада з питань безпечної життєдіяльності населення;
на регіональному рівні – комісії обласних державних адміністрацій з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
на місцевому рівні – комісія районних державних адміністрацій і виконавчих органів рад з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
на об'єктовому рівні – комісії з питань надзвичайних ситуацій об'єктів.

До систем повсякденного управління ЄДСЗР входять оснащені необхідними засобами зв'язку, оповіщення, збирання, аналізу і передачі інформації: центри управління в надзвичайних ситуаціях, оперативно-чергові служби уповноважених органів з питань надзвичайних ситуацій та цивільного захисту населення усіх рівнів; диспетчерські служби центральних та місцевих органів виконавчої влади, державних підприємств, установ та організацій.

До складу сил і засобів ЄДСЦЗ входять військові і спеціальні цивільні аварійно-рятувальні (пошуково-рятувальні) формування, які укомплектовуються з урахуванням необхідності проведення роботи в автономному режимі не менше трьох діб і перебувають у стані постійної готовності, а також недержавні (добровільні) рятувальні формування. Залежно від масштабів і особливостей надзвичайної ситуації, що прогнозується або виникла, може існувати один із таких режимів функціонування ЄДСЦЗ: повсякденної діяльності, підвищеної готовності, діяльності у надзвичайній ситуації, діяльності у надзвичайному стані. З метою ліквідації наслідків надзвичайної ситуації у мирний час може поводитися цільова мобілізація.

Ефективність функціонування систем захисту населення і територій досягається через завчасну підготовку, оперативне реагування та ефективне управління під час надзвичайних ситуацій, своєчасне відновлення життєдіяльності населення в їх зоні.

3.2 Основи охорони праці

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 – 2006 є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0 мм), для яких при використанні бокового освітлення $КПО=1,5\%$. Для штучного освітлення нормованим параметром виступає $E_{мін}$ – мінімальний рівень освітленості, та $Кп$ – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк.

В приміщенні використовуються світильники типу ЛПО. Кожен світильник комплектується двома лампами. Тобто необхідно використовувати 6 світильників із 12 працюючими лампами в них.

Приміщення, в якому працює програміст, має загальну площу 20 м^2 , висоту стелі 3 м. У приміщенні знаходиться 7 робочих місць з ПК. Кожне робоче місце обладнане робочим столом площею $1,2\text{ м}^2$, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші. Слід відзначити, що площа одного робочого місця оператора ПК не повинна бути меншою за 6 м^2 , а об'єм не менший за 20 м^3 , тобто площі та об'єму даного приміщення не вистачає для розташування семи робочих місць операторів ПК.

Аналіз умов праці показує, що у приміщенні лабораторії на програміста можуть негативно впливати наступні фізичні та психофізіологічні фактори:

- підвищена або знижена температура повітря робочої зони;
- підвищена або знижена вологість повітря;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;

- підвищена іонізація повітря;
- підвищений рівень електромагнітних випромінювань;
- нервово-психічні перевантаження (розумова перенапруга, перенапруга аналізаторів);
- фізичні перевантаження (одноманітна поза викликає статичну втоми).

Робота програміста за енерговитратами відноситься до категорії легких робіт Іа, Іб, тому повинні дотримуватися наступні вимоги згідно ДСН 3.3.6.042-99:

- оптимальна температура повітря – 22°C (допустима – 20-24°C);
- оптимальна відносна вологість – 40-60% (допустима – не більше 75%);
- швидкість руху повітря не більш 0,1 м/с.

Виміряні за допомогою приладів (психрометр Августа) температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року.

Розташовані у приміщенні 7 ПК являються джерелами тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Нормованим показником ІЧВ є гранично допустима густина потоку енергії $J_{г.д}$, Вт/м², яка встановлюється в залежності від площі опромінюваної поверхні тіла людини ($S_{опр}$). Нормовані рівні складають: $J_{г.д}=35$ Вт/м² при $S_{опр} > 50\%$; $J_{г.д}=70$ Вт/м² при $S_{опр} \sim 25-50\%$; $J_{г.д}=100$ Вт/м² при $S_{опр} < 25\%$.

Як було вказано вище, в кабінеті знаходиться сім робочих місць з ПК, кожне з яких устатковане монітором, вінчестером в системному блоці, трьома вентиляторами системи охолодження ПК та клавіатурою. Крім того поряд працює периферійна техніка. Таким чином у приміщенні мають місце шуми механічного і аеродинамічного походження, широкосмугові із аперіодичним підсиленням при роботі принтерів.

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера представлені в таблиці 4.1. Нормованим параметром невикористаного рентгенівського випромінювання виступає потужність експозиційної дози. На відстані 5 см від поверхні екрану монітору її

рівень не повинен перевищувати 100 мкР/год. Максимальний рівень рентгенівського випромінювання на робочому місці програміста зазвичай не перевищує 20 мкР/год.

На відстані 5-10 см від екрана і корпусу монітора рівні напруженості можуть досягати 140 В/м по електричній складовій, що значно перевищує допустимі значення.

ВИСНОВОК

Основною метою представленої на захист кваліфікаційної роботи бакалавра є досягнення всіх цілей та завдань, а саме розробка інтернет-магазину з продажу одягу. При виконанні даної роботи було проведено аналіз теми на основі літературних джерел та визначено її актуальність. Аналіз аналогів для визначення недоліків та переваг конкурентів. У процесі роботи було досліджено тему. Оптимізація веб-інтерфейсу та навігації сайту, щоб користувачеві було легше орієнтуватися у віртуальному просторі.

У майбутньому система може бути легко розширена завдяки добре продуманій архітектурі, створювати нові функції, що відповідають потребам зростання користувачів, та бути продуктивною при великих навантаженнях.

У зв'язку з вищевикладеним, можна дійти втішного висновку, що завданні виконані повному обсязі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ICONIX - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/ICONIX> (дата звернення: 07.12.2024).
2. JavaScript | MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 07.12.2023).
3. JavaScript documentation – DevDocs [Електронний ресурс] – Режим доступу до ресурсу: <https://devdocs.io/javascript/> (дата звернення: 07.12.2023).
4. Getting Started – React [Електронний ресурс] – Режим доступу до ресурсу: <https://legacy.reactjs.org/docs/getting-started.html> (дата звернення: 07.12.2023).
5. Quick Start – React [Електронний ресурс] – Режим доступу до ресурсу: <https://react.dev/learn> (дата звернення: 07.12.2024).
6. CSS Cascading Style Sheets | MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 07.12.2024).
7. CSS documentation – DevDocs [Електронний ресурс] – Режим доступу до ресурсу: <https://devdocs.io/css/> (дата звернення: 07.12.2024).
8. Visual Studio Code – Code Editing.Redefined [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/> (дата звернення: 07.12.2023).
9. Firebase | Google’s Mobile and Web App Development Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/> (дата звернення: 07.12.2023).
10. Firebase Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/docs> (дата звернення: 07.12.2023).
11. BlazeMeter Continuous Testing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.blazemeter.com/> (дата звернення: 07.12.2024).

12. Проектування баз даних. Хмельницький національний університет. URL: https://dn.khnu.km.ua/dn/k_default.aspx?M=k1134&T=10&lng=1_st=0 (дата звернення: 09.01.2020).

13. SciencSoft. URL: <https://www.scnsoft.com/blog/web-application-architecture> (дата звернення: 15.01.2020).

14. Web Application Architecture. URL: <https://litslink.com/blog/web-application-architecture> (дата звернення: 20.01.2020).

15. Drouyer S. FuelPHP Application Development Blueprints. Livery: Packt Publishing Ltd, 2015. 398с.

16. Fedosejev A. React.js Essentials. Livery: Packt Publishing Ltd, 2015. 208 с.

17. Stefanov S. React: Up & Running: Building Web Applications. CA: «O'Reilly Media, Inc.», 2016. 222с.

18. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. CA: «O'Reilly Media, Inc.», 2018. 834с.

19. Поняття діаграми Ганта. URL: <https://studfile.net/preview/5465642/page:25/> (дата звернення: 20.04.2020).

20. Методичні вказівки до виконання дипломної роботи освітнього рівня —бакалавр студентами усіх форм навчання для напряму підготовки 121 — Інженерія програмного забезпечення / Укладачі : Петрик М.Р., Михалик Д.М., Кінах Я.І., Гладь С.В., Цуприк Г.Б. — Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 — 28 с.

21. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 — “Інженерія програмного забезпечення” для усіх форм навчання [Текст] — Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя — 2020 — 27 с.

22. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 — Інженерія програмного забезпечення (Освітньо-професійна програма - «Інженерія програмного забезпечення») для студентів усіх форм навчання / Упор.: М.Р. Петрик, Д.М. Михалик, Я.І. Кінах, Г.Б. Цуприк - Тернопіль:

ТНТУ, 2017-38 с

23. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник. Тернопіль: : ТНТУ 2020 – 62 с.

24. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Моделювання та видобуток даних (високопродуктивні обчислення у великих алгебраїчних та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: : ТНТУ 2019 – 62 с.

25. Нога В.В., Цуприк Г.Б. Інформаційні технології при створенні автоматизованих систем для задач збору та збереження інформації. Збірник тез доповідей VII Міжнародної науково-технічної конференції молодих учених та студентів. Актуальні задачі сучасних технологій – Тернопіль 28-29 листопада 2018. — Т. : ТНТУ, 2018. — Том 2. — С. 131.

26. І.В. Бойко, М.Р. Петрик, Г.Б. Цуприк. Дискретні структури (Алгебраїчні та числові системи, комбінаторний аналіз). Навчально-методичний посібник для студентів спеціальності 121 «Інженерія програмного забезпечення», аспірантів та викладачів вищих навчальних закладів. Конспект лекцій : Навчальний посібник / І.В. Бойко, М.Р. Петрик, Г.Б. Цуприк – Тернопіль : ТНТУ , 2017. – 64 с.

27. Якість та тестування ПЗ. URL: <https://dl.tntu.edu.ua/index.php> (дата звернення: 30.05.2023).

28. ICONIX - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/ICONIX> (дата звернення: 30.05.2023).

29. Android Studio: переваги та особливості [Електронний ресурс] – Режим доступу до ресурсу: <https://qagroup.com.ua/publications/android-studio-perevagy-ta-osoblyvosti/> (дата звернення: 30.05.2024).

30. Java - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Java> (дата звернення: 30.05.2024).

31. Firebase - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Firebase> (дата звернення: 30.05.2024).

32. Лабораторна робота №3 [Електронний ресурс] – Режим доступу до ресурсу: <https://dl.tntu.edu.ua/get.php/lab20172018/lab3.pdf>.

33. Firebase Test Lab [Электронный ресурс] – Режим доступа до ресурсу: <https://firebase.google.com/docs/test-lab>.
34. What is automation testing? URL: <https://www.globalapptesting.com/blog/what-is-automation-testing>.
35. Selenium IDE. URL: <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd>.

ДОДАТКИ

ДОДАТОК А – Диск із кваліфікаційною роботою бакалавра