



Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
Кафедра Медіасистем та технологій  
Рівень вищої освіти перший (бакалаврський)  
Спеціальність 186 Видавництво та поліграфія  
Тип програми Освітньо професійна  
Освітня програма Видавничо поліграфічна справа  
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри МСТ \_\_\_\_\_

(підпис)

« 07 » червня 2021 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові Ківві Валерії Олександрівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування та розробка ридизайну сайту  
за результатами автоматизованого тестування на С#

Затверджена наказом по університету від 07 червня 2021 р. № 787 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16 червня 2021 р.

3. Вихідні дані до роботи

Методи розробки автоматизованих тестів на мові С#;

Стандарти розробки UI/UX частини сайту; Методи побудови модульних сіток

4. Перелік питань, що потрібно опрацювати в роботі

Вступ., 1 Аналіз технічного завдання 2. Аналітичний огляд літератури за темою роботи., 3  
Вибір інструментальних та програмних засобів для розробки автоматизованого тесту та  
редизайну сайту., 4. Проектування та розробка автоматизованого тесту на мові С#.,  
5. Запуск автоматизованого тесту та аналіз результатів. 6. Проведення мануального  
тестування сайту Amazon. 7. Розробка ридизайну за допомогою інструменту FIGMA. 8.  
Тестування ридизайну сайту за допомогою фокус групи. 9. Економічне обґрунтування  
роботи. Висновки., Перелік посилань, Додаток А.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)

Мета та актуальність роботи, Об'єкт та предмет роботи, Задачі кваліфікаційної роботи,  
Постановка завдання на проектування, Аналіз аналогів, Вибір інструментальних та  
програмних засобів для розробки автоматизованого тесту та ридизайну сайту, Розробка  
автотесту. Тест кейс. Вибір принципу розробки. Тестові атрибути, Запуск тесту та  
результати, Мануальне тестування. Тест кейс. Аналіз результатів., Розробка  
дизайну. Колірне рішення, Розробка дизайну. Головна сторінка сайту, Розробка  
дизайну. Сторінка пошуку, Розробка дизайну. Сторінка товару, Розробка дизайну. Сторінка  
з повідомленням про доданий товар, Розробка дизайну. Сторінка кошику, Розрахунок  
вартості виконання проекту, Висновок

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Бізюк А.В.		
Економічна частина	проф. Полозова Т.В.		

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	07.06.2021	викон.
2	Аналітичний огляд літератури за темою роботи	08.06.2021	викон.
3	Вибір інструментальних та програмних засобів для розробки автоматизованого тесту та редизайну сайту	08.06.2021	викон.
4	Проектування та розробка автоматизованого тесту на мові С#	09.06.2021	викон.
5	Запуск автоматизованого тесту та аналіз результатів	10.06.2021	викон.
6	Проведення мануального тестування сайту Amazon	11.06.2021	викон.
7	Розробка редизайну за допомогою інструменту FIGMA	12.06.2021	викон.
8	Тестування редизайну сайту за допомогою фокус групи	13.06.2021	викон.
9	Економічне обґрунтування роботи	14.06.2021	викон.

Дата видачі завдання 7 червня 2021 р.

Студент \_\_\_\_\_  
(підпис)

Ківва В.О.

Керівник роботи \_\_\_\_\_  
(підпис)

проф. Бізюк А.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 59 с., 11 ч., 42 рис., 2 табл., 1 дод., 20 джерел.

АВТОМАТИЗОВАННЕ ТЕСТУВАННЯ, ДИЗАЙН, HTML, C#, РЕДИЗАЙН, НАВИГАЦІЯ, UI/UX, SELENIUM, БРАУЗЕР, FIGMA, QA.

Метою кваліфікаційної роботи є розробка автоматизованого тесту задля перевірки UI частини сайту та створення редизайну.

Проведено аналіз технічного завдання та вихідних даних роботи.

Проведено аналіз програмних засобів, які надають можливість для створення автоматизованих тестів та також розглянуті інструменти для створення редизайну сайту. Були проаналізовані актуальні правила дизайну та зручності використання у розробці сайтів. За результатами аналізу були вибрані найбільш актуальні та доречні інструменти для створення автотесту та редизайну сайту, а також застосовані найбільш сучасні методи створення сайтів.

Розроблено 1 автотест на мові C# та редизайн і нова колірна гамма сайту. Автотест успішно пройдений та редизайн сайту створений згідно правил UI/UX сайтів. Було проведено економічне обґрунтування проекту, в результаті якого була розрахована ціна проекту 25017,124 грн.

## РЕФЕРАТ

Пояснительная записка содержит 59 с., 11 ч., 42 рис., 2 табл., 1 прилож., 20 источников.

АВТОМАТИЗОВАННОЕ ТЕСТИРОВАНИЕ, ДИЗАЙН, HTML, C#, РЕДИЗАЙН, НАВИГАЦИЯ, UI/UX, SELENIUM, БРАУЗЕР, FIGMA, QA.

Целью квалификационной работы является разработка автоматизированного теста для проверки UI части сайта и создания редизайна.

Проведено анализ технического задания и исходных данных работы.

Проведен анализ программных средств, которые предоставляют возможность для создания автоматизированных тестов и также рассмотрены инструменты для создания редизайна сайта. Были проанализированы правила дизайна и удобства использования в разработке сайтов. По результатам анализа были выбраны наиболее актуальные и уместные инструменты для создания автотеста и редизайна сайта, а также применены наиболее современные методы создания сайтов. Был разработан 1 автотест на языке C#, а также редизайн сайта и новая цветовая гамма. Автотест успешно пройден и редизайн сайта создан согласно правил UI/UX сайтов. Было проведено экономическое обоснование проекта, в результате которого была рассчитана стоимость проекта 25017,124 грн.

## ABSTRACT

Explanatory note contains 59 p., 11 pt., 42 pic., 2 tabl., 1 app., 20 sources.

AUTOMATION TESTING, DESIGN, HTML, C#, REDESIGN, NAVIGATION, UI/UX, SELENIUM, BROWSER, FIGMA, QA.

The main goal of explanatory note is developing automation test for checking the UI part of the site and making redesign.

The technical task and initial data analyze were performed.

The analyze of program resources that give an opportunity to create automation test was performed and tools for redesign were also observed. The main design developing rules and UX ruler were analyzed.

According to the analyze results the most actual and relevant tools for automation test and redesign creation were selected, the newest and most actual techniques for site creation were applied.

1 automation test in C# was developed and also site redesign and new color palette. Automation test was successfully passed and redesign was created according to UI/UX rules. The project was economically justified, as a result of which the project cost was calculated 25017,124 hrn.

## ЗМІСТ

	С.
ВСТУП.....	9
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	12
1.1 Постановка завдання на проектування.....	12
2 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ РОБОТИ.....	14
2.1 Основні завдання тестування сайту та подальшого редизайну.....	14
2.2 Аналіз сайтів-маркетплейсів, що надають схожі послуги.....	16
2.3 Аналіз тенденцій автоматизованого тестування.....	19
2.4 Огляд тенденцій web-дизайну.....	19
3 ВИБІР ІНСТРУМЕНТАЛЬНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ АВТОМАТИЗОВАНОГО ТЕСТУ ТА РЕДИЗАЙНУ САЙТУ .	24
3.1 Вибір і обґрунтування програмного забезпечення для проектування та розробки автоматизованого тесту мовою C#.....	24
3.2 Програмне забезпечення для створення редизайну сайту.....	26
3.3 Мова програмування C#.....	30
4 ПРОЕКТУВАННЯ ТА РОЗРОБКА АВТОМАТИЗОВАНОГО ТЕСТУ МОВОЮ C#.....	32
4.1 Аналіз html структури сайту Amazon.....	32
4.2 Підготовка та збір даних для розробки тесту.....	33
4.3 Розробка автоматизованого тесту.....	35
4.3.1 Вибір принципу розробки автоматизованого тесту.....	35
4.3.2 Розробка автоматизованого тесту в Visual Studio.....	37
5 ЗАПУСК АВТОМАТИЗОВАНОГО ТЕСТУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ .	39
6 ПРОВЕДЕННЯ МАНУАЛЬНОГО ТЕСТУВАННЯ САЙТУ AMAZON .....	40
7 РОЗРОБКА РЕДИЗАЙНУ САЙТУ ЗА ДОПОМОГОЮ ІНСТРУМЕНТУ FIGMA .....	45
7.1 Підбір нового кольорового та шрифтового оформлення.....	45
7.2 Вибір шрифтового оформлення.....	46
7.3 Розробка модульної сітки.....	46

7.4 Розробка макету редизайну сайту Amazon .....	46
8 ТЕСТУВАННЯ РЕДИЗАЙНУ САЙТУ ЗА ДОПОМОГОЮ ФОКУС ГРУПИ .....	51
9 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОБОТИ.....	52
9.1 Характеристика продукції та ринок збуту .....	52
9.2 Розрахунок витрат .....	52
ВИСНОВКИ .....	57
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	58
ДОДАТОК А Розробка сайту .....	<b>Error! Bookmark not defined.</b>

## ВСТУП

Мережа інтернет на сьогодні – це невід’ємна частина нашого життя, а сайти – це основні компоненти цієї мережі. Це спосіб передачі інформації користувачу у доступній, зручній та зрозумілій формі, тобто сайти це основа взаємодії користувача з мережею Інтернет. У наш час якість та зручність сайту – це те, що привабить та утримає користувача. Довіра користувача до продукту – це одна із головних цілей сайту, адже це один із основних інструментів покращення, підтримання або ж погіршення репутації компанії та продукту.

Конкуренція неймовірна, адже існують 1000 різноманітних сайтів і щоб захопити увагу та прихильність відвідувача, потрібно слідувати усім основам UI/UX розробки, чітко та зрозуміло демонструвати тематику та ціль сайту, підтримувати його в актуальному стані, обновлювати його згідно теперішнім трендам та обов’язково враховувати зворотній зв’язок користувача.

Якісний, швидкий та зручний сайт дуже сильно економить час користувача, а як ми знаємо, час на сьогодні – це дуже цінна річ. Коли люди звертаються до Інтернету, вони очікують отримати задоволення та користь від проведеного часу, а за це відповідають саме сайти.

А ось за якість сайтів та додатків відповідає саме веб тестування. Тестування – це один із найважливіших етапів створення та підтримки сайту, адже від нього залежить подальший успіх та розвиток сайту. Основна мета тестування – перевірка відповідності роботи та UI частини продукту до вимог замовника, а також перевірка доречності та стабільності продукту. Тестування сайтів використовується для перевірки того, наскільки довго завантажуються сайт, наскільки коректно він відображається у різних браузерах та різних розмірах вікна браузера, наскільки коректна його Html структура та в якій мірі сайт відповідає своїй меті та очікування користувача. Існує декілька видів та методів тестування. Тестування поділяється на автоматизоване та мануальне.

Для тестування сайтів найбільш доречні такі види:

– функціональне тестування - даний етап, дозволяє перевірити відповідність функціональних можливостей сайту з технічним завданням. Також, перевіряється робота сайту і його функцій на програмні помилки коду;

– навантажувальне тестування. Процес тестування навантаження, проводиться методом емуляції одночасного використання сайту великою кількістю користувачів. Також, перевіряється скільки і яких ресурсів вимагає сайт при великих навантаженнях;

– тестування інтерфейсу і перевірка юзабіліті. Якість дизайну і інтерфейсу додатку або сайту дуже важливі. Дружній інтерфейс, який не вимагає великої кількості часу, що знайти ту чи іншу функцію на сайті, збільшує ймовірність підвищення поведінкових факторів, що позитивно впливає як на ранжирування сайту в пошукових системах, так і на ймовірність здійснення цільової дії користувачем;

– перевірка безпеки. Кожен сайт, особливо якщо він має на увазі транзакційні дії, повинен бути захищений від будь-яких варіантів злому ззовні. Для цього, проводиться детальне тестування безпеки сайту, на предмет виявлення можливих sql ін'єкцій, ddos-атак і інших способів взлому сайту.

За результатами тестування можна виявити помилки у проектуванні сайту та на основі них зробити редизайн. На сьогодні тренди змінюються дуже швидко, тому час від часу кожен сайт потребує оновлення, тобто редизайну. Це можуть бути зовнішні поліпшення або глибокі зміни, які пов'язані з поліпшенням юзабіліті і реалізації нових функцій. Існує декілька видів редизайну, а саме: візуальний, функціональний, технічний та глибокий редизайн. В результаті такої модернізації, сайт стане більш актуальним, трендовим та популярним за рахунок того, що його зручність та якість значно зросте. В результаті цього була вибрана така тема кваліфікаційної роботи: «Проектування та розробка редизайну сайту на основі результатів автоматизованого тестування на C#»

Метою кваліфікаційної роботи є проектування та розробка редизайну сайту Amazon на основі автоматизованого тесту на мові C#. Amazon це глобальна платформа для придбання різноманітних товарів, проте дизайн їх сайту, зручність та прозорість використання та призначення різних елементів на сайті доволі застаріла.

За допомогою проектування автоматизованого тесту досконало досліджується та перевіряється html структура сайту, її логічність та коректність.

Для досягнення заданої мети необхідно вирішити такі завдання:

- проаналізувати технічне завдання;
- провести аналіз аналогів сайтів-маркетплейсів для виявлення актуальних трендів та тенденцій в дизайні додатку;
- провести аналітичний огляд літератури за темою роботи;
- обрати інструментальні (програмні) засоби розробки;
- проаналізувати html структуру сайту для створення стабільного автоматизованого тесту;
- розробити автоматизований тест за допомогою мови C#;
- проаналізувати результати тестування;
- підібрати доречну та актуальну кольорову гамму та шрифтове рішення;
- зробити редизайн додатку;
- протестувати сайт після редизайну;
- розрахувати вартість проектування та розробки з урахуванням усіх можливих потреб.

## 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

Основним завданням є проектування та розробка редизайну сайту на основі результатів автоматизованого тесту на мові C#.

Щоб виконати проектування та розробку редизайну сайту та автоматизованого тесту, необхідно виконати наступні завдання:

- а) вибрати інструментальні та програмні засоби розробки;
- б) проаналізувати та спроектувати структуру тесту;
- в) створити автоматизований тест;
- г) проаналізувати результати тестування;
- д) розробити редизайн інтерфейсу;
- е) протестувати ресурс після редизайну.

### 1.1 Постановка завдання на проектування

У процесі бакалаврської роботи потрібно виконати збір даних для створення автоматизованого тестування, розробити автоматизований тест для перевірки базового шляху користувача на сайті Amazon, провести етап мануального тестування, проаналізувати результати та виявити помилки, розробити редизайн сайту та протестувати його за допомогою фокус групи.

Для розробки автоматизованого тесту потрібно виконати етапи підготовки:

- аналіз Html частини сайту;
- аналіз результатів та помилок;
- визначення базового шляху користувача на сайті;
- розробка тестового сценарію;
- визначення методу розробки автоматизованого тесту;
- збір даних та підготовка XPath, Css запитів ;
- вибір інструменту для взаємодії із браузером;
- налаштування роботи Visual Studio та встановлення Nuget пакетів

для коректної роботи з автоматизованим тестуванням.

Процес розробки автоматизованого тесту та аналіз результату буде складатись з:

- створення класів та методів згідно базовому шляху користувача на сайті;
- застосування тестових атрибутів для коректної структури тесту;
- створення основної частини тесту;
- запуск тесту;
- аналіз результатів та виявлення помилок.

Процес мануального тестування буде містити:

- розробка тестового сценарію для мануального тестування;
- проходження тестового сценарію;
- аналіз результатів;
- виявлення помилок та недоліків.

Процес розробки редизайну міститиме:

- вибір інструментів для підбору кольорової гами;
- вибір інструментів для розробки макету сайту;
- підбір актуальної, нової кольорової гами;
- розробка модульної сітки;
- розробка макету сайту.

Процес тестування фокус групою буде складатися із:

- створення тестового сценарію;
- проходження сценарію фокус групою;
- відгуки та їх аналіз.

## 2 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ РОБОТИ

### 2.1 Основні завдання тестування сайту та подальшого редизайну

Коли людина хоче вирішити якусь свою проблему, дізнатися щось нове або просто гарно провести час вона йде до мережі інтернет та відвідує сайти. Під час цього процесу користувач очікує, що йому не потрібно буде вирішувати додаткові проблеми, наприклад, розбиратися з незрозумілим меню або структурою подачі інформації. Якщо на сторінці буде присутня плутанина із, здавалося б, звичайними елементами сайту або вони будуть розміщені в несподіваних та незвичних для користувача місцях – він, швидше за все, не знайде як вирішити свою проблему і покине сайт. Зараз, коли конкуренція серед різних ресурсів дуже висока, таку ситуацію допускати не можна.

Користувачі отримують більше задоволення та користі проводячи час на якісному сайті, без помилок та із зручним та зрозумілим інтерфейсом. Щоб визначити наскільки сайт відповідає очікуванням користувача та основним правилам UI/UX розробки сайтів, проводиться тестування чорного та білого ящика (ручне тестування). В ході такого тестування інженер оцінює наскільки сайт задовольняє потреби користувача, а саме цільової аудиторії.

В ході цього інженер складає список помилок, які потребують корегування. Найпоширеніші з них наведені нижче:

- дуже багато або дуже мало інформації на сайті. Надлишок інформації призведе до того, що користувач витратить дуже багато часу та терпіння на пошук того, що йому потрібно і може зовсім не досягнути своєї цілі;

- надмірна кількість зображень та анімації. Ця помилка особливо сильно впливає на конверсію інформаційних сайтів, адже користувач очікує отримати багато корисної інформації на подібних сайтах та не потребує перегляду великої кількості зображень;

- неординарне розміщення елементів, хедеру, меню та інших частин сайту. Існують сталі правила розробки сайтів і дуже великі відхилення від

них можуть погано вплинути на популярність сайту, якщо це не ресурс для більш вузької цільової аудиторії;

– непрозорість елементів сайту, тобто відсутність логічних пояснень для комфортного та зрозумілого користування сайтами. Якщо кнопки та елементи сайту не будуть явно підписані або ж якось підкреслені та не буде зрозуміло для чого потрібен той чи інший елемент, то користувач дуже швидко втомиться шукати правильний елемент і вибере інший ресурс для задоволення потреб;

– занадто багаторівнева навігація. Це може призвести до того, що задля доступу до потрібної інформації потрібно буде пройти дуже довгий шлях, що призведе до зниження кількості користувачів в силу складності та заплутаності сайту.

Саме тестування сайту допоможе уникнути або виправити ці моменти.

Проте ручне тестування може зайняти дуже багато часу та з певної точки зору буде мати погрішність в силу виконання тестування людиною. В такому разі доречніше використовувати автоматизований метод тестування. Це забезпечить стабільність тестування, чіткі та прозорі результати, регулярне тестування сайту. Більш того в ході проектування та розробки тесту інженер аналізує та вивчає html структуру сайту та робить висновки щодо доречності певних елементів та складності структури сайту. Для найефективнішого результату потрібно комбінувати ручне та автоматизоване тестування.

За результатами тестування потрібно провести виправлення помилок та редизайн сайту за необхідності. Редизайн позитивно впливає на імідж продукту та конверсію сайту, адже після якісного редизайну сайт стає зрозумілішим, більш трендовим та сучасним.

Можна виділити такі ознаки якісного та сучасного сайту, який направлений на комфорт та зручність використання користувачем:

- якісне текстове наповнення;
- зрозуміла та чітка навігація;

- коректні акценти на елементах та блоках сайту;
- дизайн, який відповідає тематиці продукту та водночас трендам теперішнього часу;
- зрозумілий та простий пошук на сайті;
- інтуїтивність елементів сайту. користувач одразу розуміє який елемент за що відповідає та може легко пересуватися по сайту без додаткових зусиль;
- кросбраузерність сайту;
- підказки до використання елементів (у разі відсутності можливості спростити навігацію та покращити прозорість використання елементів сайту);
- розташування блоків сайту згідно їх правилам;
- безпека та конфіденціальність даних у разі наявності елементів реєстрації користувача або оплати товару;
- відсутність орфографічних та логічних помилок в контенті сайту. такі помилки вказують недбалість власників продукту до свого сайту, що створює враження недбалості у відношенні до клієнту та користувача;
- актуальність та унікальність контенту на сайті.

## 2.2 Аналіз сайтів-маркетплейсів, що надають схожі послуги

Було розглянуто 4 сайти навчальних закладів та їх дизайн, які працюють в Україні за схожим направленням. Проаналізував їх, були виявлені такі гідності та недоліки:

а) сайт «Rozetka». Серед його плюсів можна виявити: стриманість дизайну та зрозуміла навігація, чіткий опис категорій в меню, відсутність надмірної кількості реклами, відсутність непомітних та дрібних елементів, ціль яких не одразу зрозуміла. З недоліків можна виділити не дуже помітну кнопку для додавання товару до корзини, непомітне вікно для фільтрації товарів на сторінці, деякі елементи занадто світло сірого кольору, тому на білому фоні їх погано видно. Сторінка сайту представлена на рисунку 2.1;

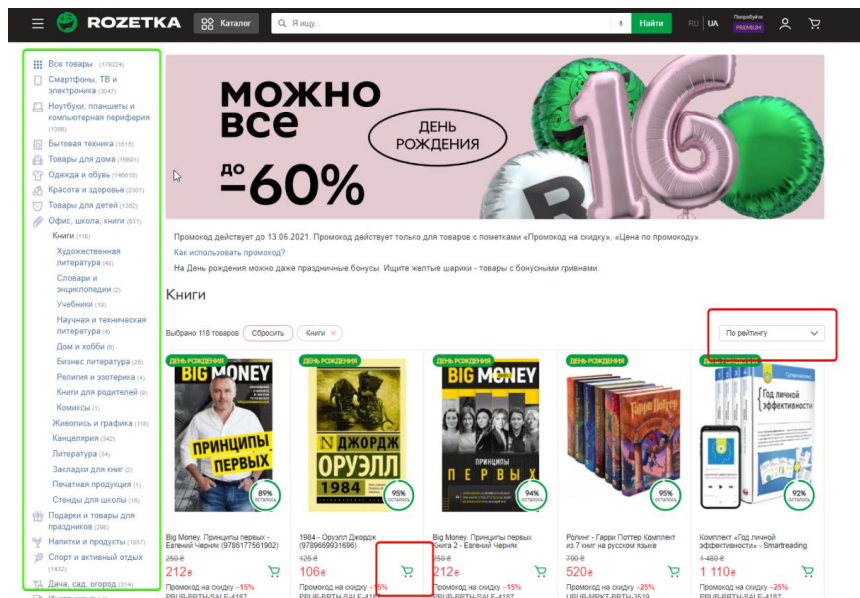


Рисунок 2.1 – Сторінка сайту маркетплейсу «Rozetka»

б) сайт «Prom UA» Сайт має стриманий дизайн та кольорову гаму, зручний пошук та дуже детальну класифікацію товарів. Проте сайт має багатов недоліків, такі як непрозорість розташування меню, велика кількість реклами та рекомендацій та дуже яскравий акцент на них, дрібність елементів у меню та хедері, блок меню ніяк не виділений і виглядає непомітним та недоречним (рис. 2.2);

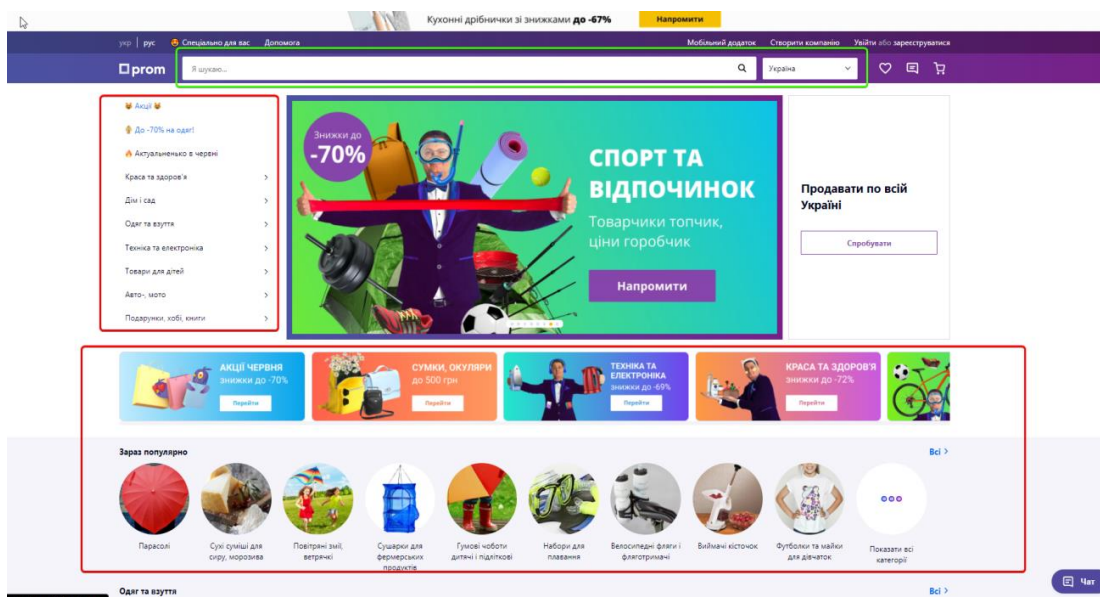


Рисунок 2.2 – Головна сторінка сайту «Prom UA»

в) сайт «ЕВАУ» має стриманий проте застарілий дизайн, гарний та помітний пошук, але дрібні та непомітні елементи меню та елементи в хедері (рис. 2.3);

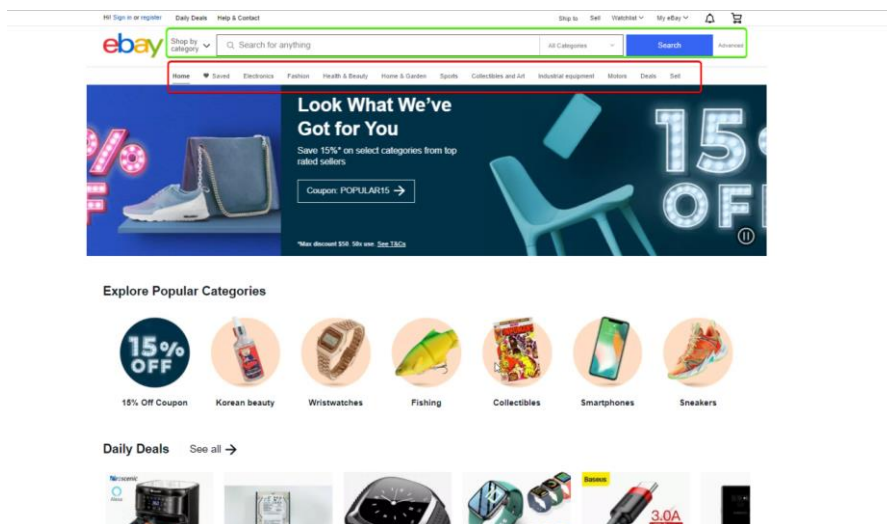


Рисунок 2.3 – Головна сторінка сайту «ЕВАУ»

г) сайт для придбання товарів для handmade творчості «ETSY». Проста навігація та стриманий дизайн, доречні рекомендації та відсутність реклами є несумнівним плюсом. Проте елементи навігації занадто маленькі та не дуже деталізовані (рис. 2.4);

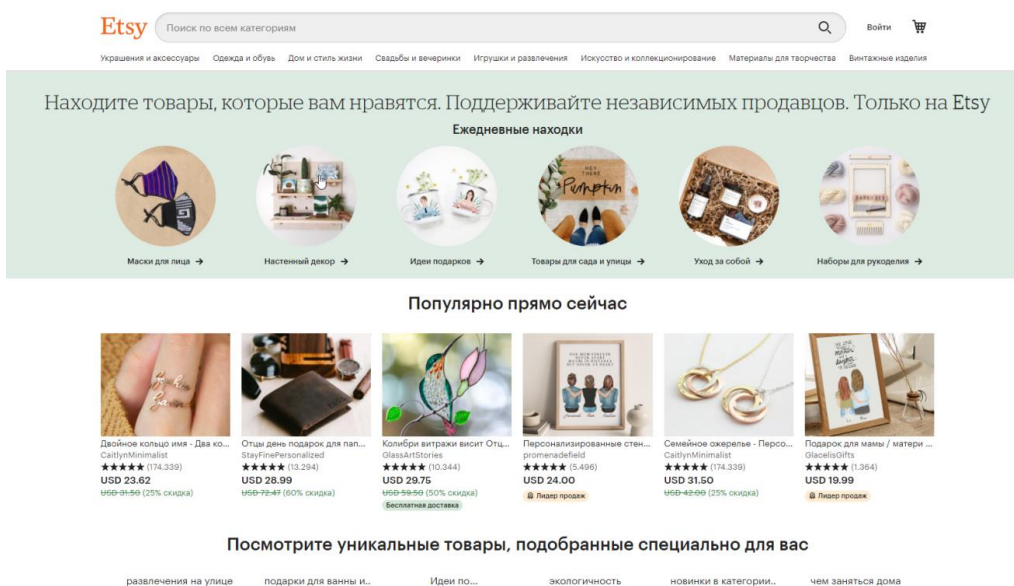


Рисунок 2.4 – Головна сторінка сайту «ETSY»

В результаті аналізу аналогів були виокремлені ознаки якісного сайту-маркетплейсу:

- зручна, зрозуміла та детальна навігація;
- логічне розташування блоків сайту згідно правилам UX дизайну;
- дизайн стриманий, світлий фон та чітке виокремлення елементів навігації, тому ще це основою сайту-маркетплейсу;
- доречні акценти на основному контенті сайту;
- читабельний шрифт;
- відсутність дрібних кнопок та блоків.

### 2.3 Аналіз тенденцій автоматизованого тестування

Автоматизація на сьогодні дуже швидко розвивається. Основна перевага автоматизованого тестування в тому, що QA інженери можуть сконцентруватися на проведенні нових експериментів та розробці тестів, ніж на повторному тестуванні. Також вона дозволяє QA інженерам більш ефективно розпоряджатися своїм часом та роботою, отримувати точні та зрозумілі результати тестування. Завдяки автоматизації можна отримувати безперервний зворотній зв'язок щодо стану продукту та досягти повного покриття продукту тестами без витрати великої кількості часу.

На сьогодні з кожним днем збільшується кількість організацій, які впроваджують автоматизоване тестування у процеси своєї роботи. Загальний рівень автоматизації тестів продовжує рости. Проте автоматизоване тестування не виключає мануальне тестування, тобто ручне. Основна тенденція в тому, щоб завдяки досвіду та розвитку сфери тестування досягти ідеального балансу автоматизованого тестування та ручного.

### 2.4 Огляд тенденцій web-дизайну

Тренди веб дизайну 2021 року йдуть нога в ногу з часом та відповідають сучасним потребам та бажанням оточуючого світу.

На сьогоднішній день трендами дизайну являються такі [6]:

– яскраві кольори та простота. Все більше та більше компаній намагаються бути помітнішими на тлі безлічі конкурентів, тренд на використання яскравих і сміливих кольорів в дизайні веб-сайтів був актуальний та залишиться таким ще довго. Яскраві і насичені кольори не тільки занурюють в себе, а й привертають увагу, і багато онлайн-брендів цим активно користуються. Враховуючи конкуренцію на сьогоднішній день, щоб сайт помітили – потрібно виділитися та використовувати нестандартні рішення (рис. 2.5);



Рисунок 2.5 – Приклад сайту з яскравим та простим дизайном

– анімація та динамічні ілюстрації. У наступному році особлива увага буде приділена мобільній анімації. Орієнтація на мобільні екрани (mobile first) залишається трендом веб-дизайну, і буде тільки посилюватися. Анімація привертає увагу до сайту та продукту в цілому а ілюстрації, в свою чергу, можуть істотно полегшити задачу по демонстрації потенційним клієнтам того, як користуватися продуктом або допомогти з оглядом функцій сервісу;

– асиметричні макети. Для розробки більшості сайтів використовуються сітка, межі якої окреслюються внутрішніми і зовнішніми відступами, що робить будь-яку сторінку максимально комфортною для сприйняття. Проте такий стандартний макет розробки приводить до того, що більшість сайтів виглядає нудно та однаково. Тому в 2021 році саме

асиметричні структури дозволять багатьом брендам висловити свою індивідуальність і навіть бруталність в повній мірі, створюючи химерні, незвичні людському оку та екстравагантні роботи (рис. 2.6);

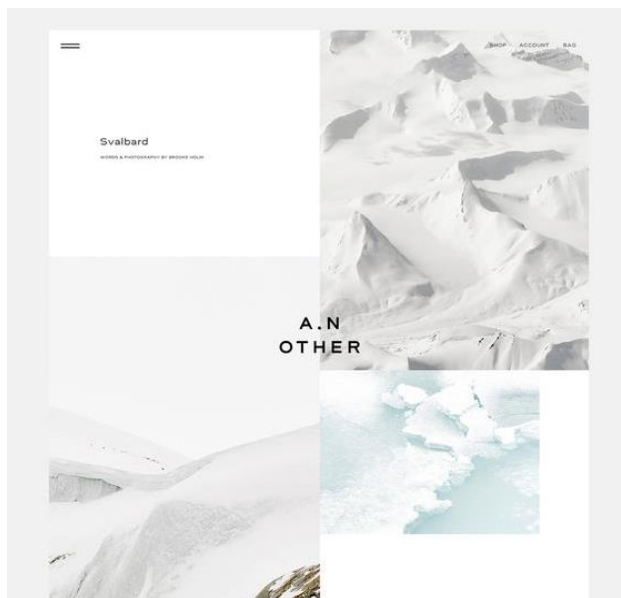


Рисунок 2.6 – Асиметричний дизайн сайту

– намальовані від руки зображення. По мірі того, як все більше число веб-майстрів починає сприймати сайти як щось більше, ніж просто адресу в інтернеті, мальовані елементи, які роблять ці самі сайти унікальними і незвичайними, будуть зустрічатися все частіше. Такі ілюстрації надають сайту приємної привабливості та якості продукту, тому що на сьогодні більшість малобюджетних сайтів використовують лише фотографії та прості графічні об'єкти, тому використання мальованих ілюстрацій підвищить прихильність користувача до сайту (рис. 2.7);

– органічний дизайн. Сучасна тенденція на екологічний образ життя та захист природи прямо впливає на веб-дизайн тренди. Люди стають більш прихильні до природи, її захисту, а у випадку веб дизайну до її кольорів та спокою. Тож використання теплих землястих кольорів, природних форм і необроблених органічних текстур зроблять сайт трендовим та комфортним для користувача [6] (рис. 2.8);

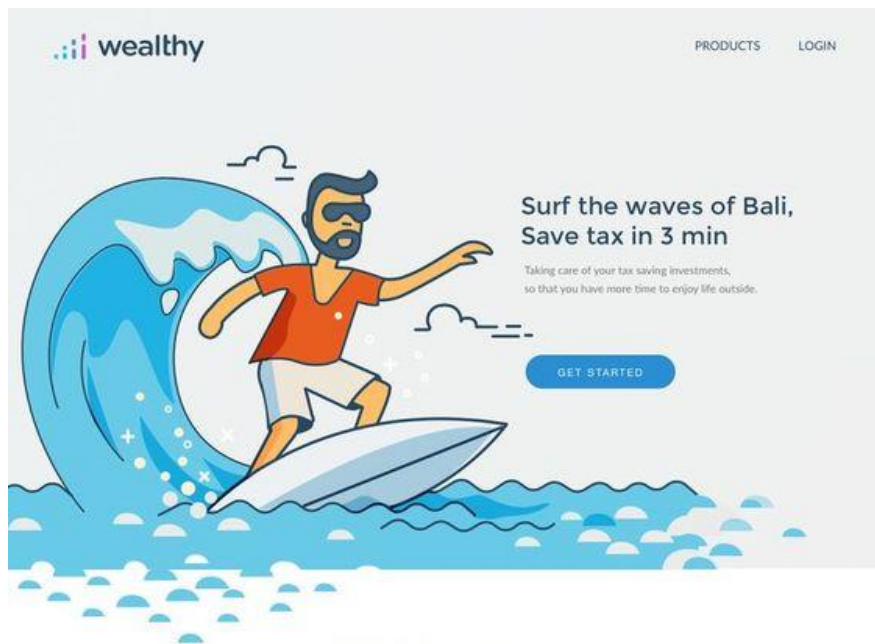


Рисунок 2.7 – Намальовані від руки зображення

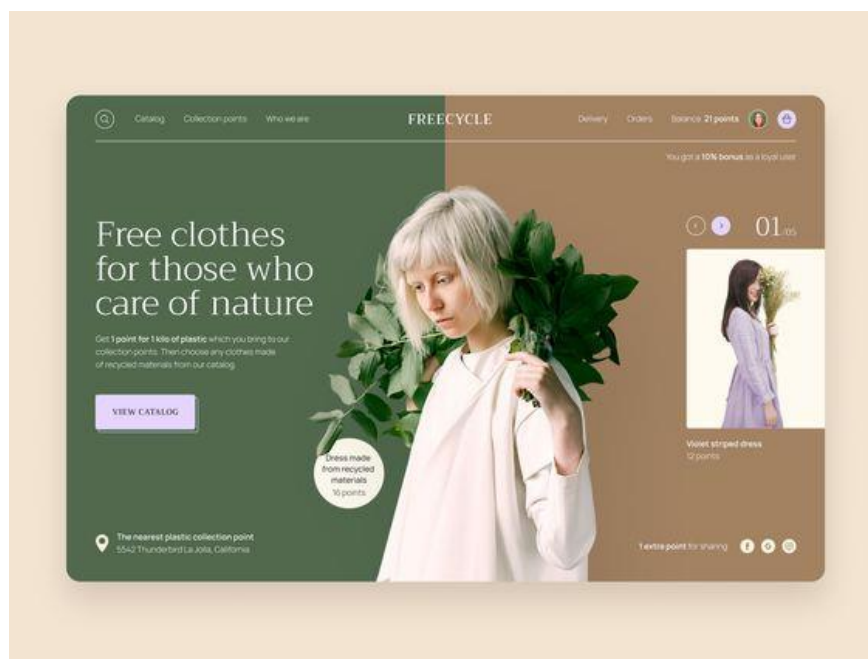


Рисунок 2.8 – Приклад органічного дизайну

– голосовий інтерфейс. Голосові чат-боти і віртуальні помічники – один із найпопулярніших технологічних трендів UX / UI 2020-2021 років.

Впровадження в дизайн сайту голосового інтерфейсу допоможе виділитися серед конкурентів і стати лідером у своїй ніші за рахунок розширення охоплення. Говорити безпосередньо в пристрій набагато

зручніше, ніж набирати запити вручну. І якщо веб-розробники адаптують свої інтерфейси під голосову взаємодію, то зможуть привернути увагу більш зайнятої аудиторії.

Екрани пристроїв зменшуються в розмірі, і на них залишається все менше вільного місця, а голосові моделі не дуже вибагливі в просторі і персоналізують доставку контенту так, як не може жоден текст.

При редизайні сайту-маркетплейсу Amazon були взяті до уваги усі основні тенденції, які являються доречними для даної тематики.

## 3 ВИБІР ІНСТРУМЕНТАЛЬНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ АВТОМАТИЗОВАНОГО ТЕСТУ ТА РЕДИЗАЙНУ САЙТУ

### 3.1 Вибір і обґрунтування програмного забезпечення для проектування та розробки автоматизованого тесту мовою C#

Вибір середовища розробки програмного забезпечення для проектування та розробки автоматизованого тесту мовою C# виконувався на підставі аналіз джерел в мережі Інтернет. Існує багато різних IDE для комфортної розробки автоматизованих тестів, нижче будуть розглянуті деякі з них:

а) ProjectRider – це середовище розробки для роботи з платформою C#. Має як переваги так і недоліки. До основних переваг можна віднести те, це середовище є частиною великого комплексу програм, які працюють в єдиному циклі та завдяки такій можливості можна чітко та логічно організувати увесь життєвий цикл розробки ПЗ. Також наявність великої кількості допоміжних інструментів для покращення коду спрощують процес розробки та покращують якість коду. Також існує можливість інтеграції з такими інструментами як GIT, TFS та інше. До недоліків можна віднести новизну середовища, що призводить до недосконалості системи та великої кількості помилок у роботі. Більш того – це платне середовище, що зменшує коло його потенціальних користувачів;

б) Visual Studio – одне з найпопулярніших середовищ розробки. Основними перевагами можна вважати те, що це офіційне середовище для розробки ПЗ. Воно має як платні, так і безкоштовні версії, функціонал яких дозволяє проектувати повноцінне програмне забезпечення без втрати якості коду. Це середовище дуже гнучке в плані налаштувань його функцій та покращення за допомогою різних плагінів. Visual Studio відмінно підходить для розробки програмного забезпечення на мові програмування C#, тому що ця мова, як і середовище розробки були створені в Microsoft, тому сумісність цих елементів найбільша та найбільш якісна та функціональна. Проте основним недоліком є те, що це середовище розробки має велику кількість

неочевидного функціоналу і для продуктивної та гарної взаємодії із нею потрібно освоїти велику кількість інформації та знань по роботі із Visual Studio. Інтерфейс продемонстрований далі (рис. 3.1);

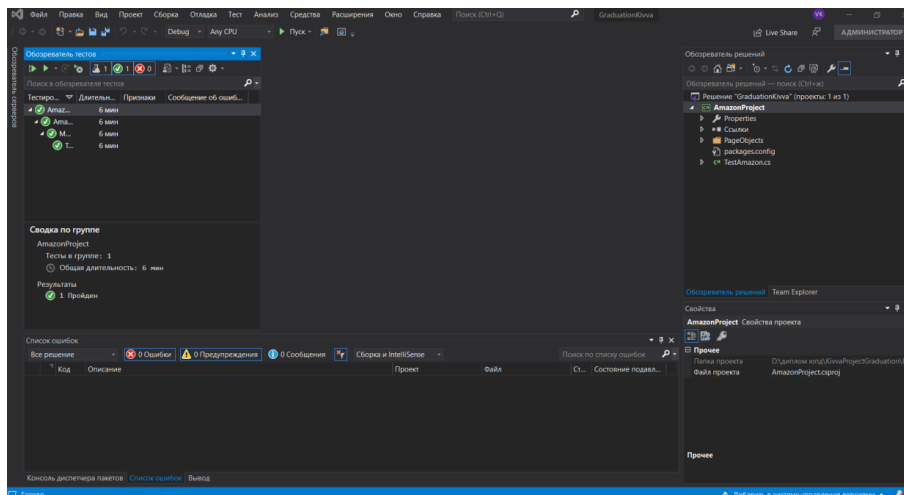


Рисунок 3.1 – Visual Studio

в) Eclipse – мультимовна середовище розробки. Має велику кількість плагінів, що значно розширює її можливості. Велике ком'юніті, завдяки цьому легко знайти рішення проблеми у роботі із середовищем. Великий діапазон функціоналу та безкоштовність продукту впливає на збільшення аудиторії користувачів. Проте до недоліків можна віднести велику кількість помилок у роботі та те, що це середовище більш направлене на розробку на мові програмування Java, аніж C#, що в даному випадку недолік.

Для досягнення цілей бакалаврської роботи, створення автоматизованого тесту, було обране середовище розробки Visual Studio враховуючі всі переваги та сумісність мови та середовища, це найоптимальніший варіант для розробки тесту.

Драйвер для взаємодії автоматизованого тесту з браузером обирався з умови швидкої та зручної роботи. Найпопулярніший та найдосконаліший драйвер для взаємодії із браузером на сьогодні це Selenium WebDriver. Це програмна бібліотека, яка надає можливість розроблювати тести або програми, які керують браузером. Драйвер – це програма (програмна

бібліотека), яка дозволяє іншим програмам/системам взаємодіяти один з одним. За допомогою селеніума можна викликати команди браузера та дозволяти автоматизованим тестам керувати ним. Селеніум надає широкий функціонал, завдяки якому можна керувати функціями браузера, проте селеніум веб драйвер не має інтерфейсу, тому для взаємодії із ним потрібно використовувати сторонні продукти, в даному випадку Visual Studio.

### 3.2 Програмне забезпечення для створення редизайну сайту

Програмне забезпечення для підбору кольорів для розробки гармонічного дизайну базується на теоретичних засадах колірних схем. Серед них можна виділити такі:

а) Colorscheme – інструмент для підбору кольору, який має багато додаткових можливостей. Наприклад даний сайт надає можливість одразу оцінити, як буде виглядати ваш сайт з темною та світлою темою. Інструмент має простий та зрозумілий інтерфейс, який дозволить будь-кому підібрати доречні кольори для свого продукту із збереженням балансу та гармонії між ними (рис. 3.2);

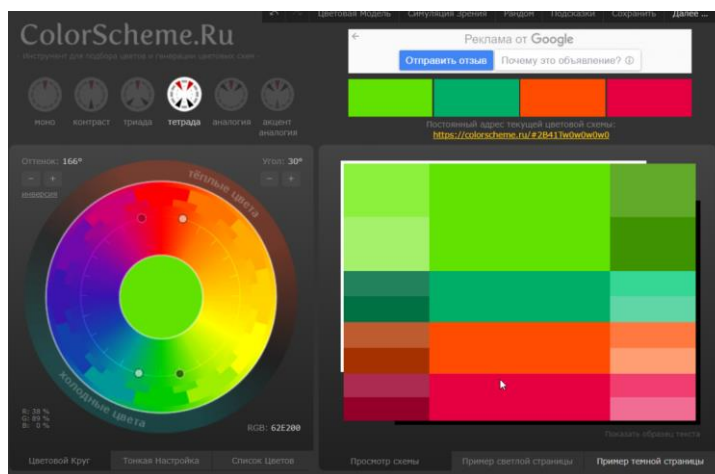


Рисунок 3.2 – інструмент Colorscheme

б) ColourCode – онлайн інструмент для генерації кольорових схем. Має 8 варіацій генерації схем, дуже зрозумілий та доступний функціонал,

можливість додавати проміжні кольори в необмеженій кількості, що буде корисно для градієнтної техніки у дизайні. Більш того надає можливість завантажувати генеровані схеми та працює в інтеграції із такими системами як GitHub, Imprint (рис. 3.3);

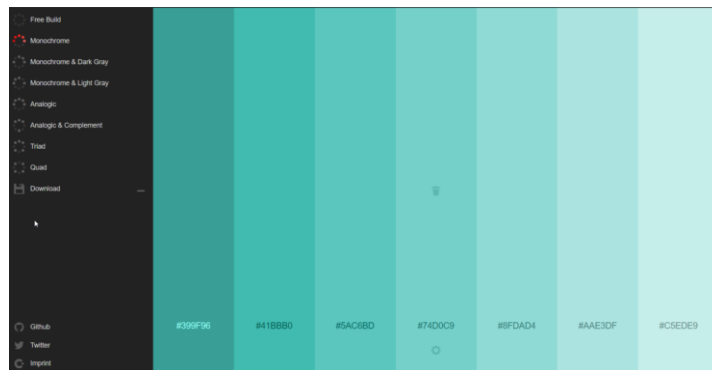


Рисунок 3.3 – інтерфейс інструменту ColourCode

в) Color Hunter – простий інструмент для генерації добірок кольорів із зображень та фото. Також має бібліотеку вже генерованих сполучень. Якщо дизайнер черпав натхнення з конкретного фото, то простим рухом він може загрузити це фото на колор хантер та отримати палітру кольорів, які на ньому присутні. Існує можливість завантажити готові палітри до сервісу Pinterest, що також є великим майданчиком для пошуку ідей та натхнення для дизайнерів при створенні сайтів (рис. 3.4).

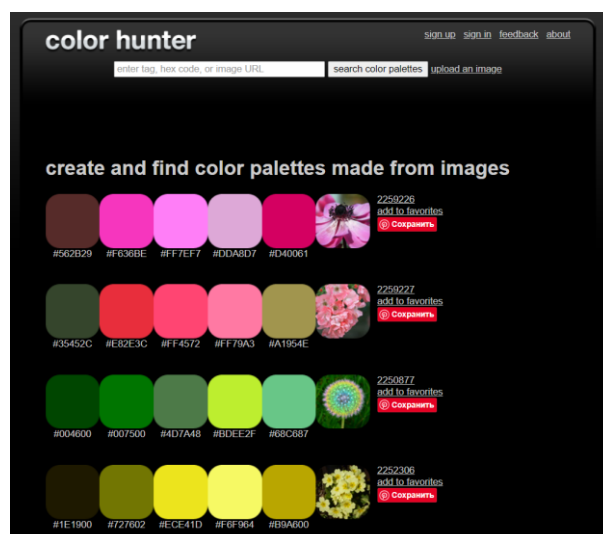


Рисунок 3.4 – інтерфейс інструменту Color Hunter

Для створення колірної схеми створюваного ресурсу були використані сервіси ColourCode та ColourScheme.

Програмне забезпечення для створення дизайну макетів має забезпечувати опрацювання графіки та підтримувати роботу із шарами та об'єктами. Були розглянуті такі програмні продукти:

а) Adobe Photoshop - графічний редактор, розроблений для редагування растрових та векторних об'єктів, створення графічних інтерфейсів, модульних сіток, логотипів та інших елементів. Adobe Photoshop має широкий функціонал, який надає можливість створювати будь-які графічні об'єкти. Photoshop - це найбільш потужна і функціональна програма в своєму класі. Програма дозволяє ретушувати зображення і додавати дефолтні та кастомні фільтри та спецефекти, комбінувати знімки та об'єкти, додавати текстові елементи, керувати та налаштовувати кольори, необхідні для створення дизайну, регулювати різні показники зображень за допомогою кривих та іншого функціоналу. Гарно працює в інтеграції із різними додатковими пристроями, такими як графічні планшети тощо. Photoshop підійде як досвідченим, так і починаючим дизайнерам та представникам інших професій. За замовчуванням він налаштований дуже якісно і дозволяє будь-кому майже одразу почати створювати дизайн. З іншого ж боку ця програма містить неймовірну кількість налаштувань, які дозволяють повністю налаштувати програму під конкретного користувача. Проте для більш конкретних змін, налаштувань та освоєння нових, складних функцій, потрібно вивчити багато нюансів та команд. Більш того Photoshop – це платна програма, вона потребує завантаження на локальний комп'ютер та займає досить багато місця в пам'яті;

б) Figma – це безкоштовний інструмент для створення графічних інтерфейсів. Існує також і платна версія, проте її функціонал майже не відрізняється від безкоштовної, що надає можливість створювати трендові та привабливі дизайни кожному. Фігма знаходиться у стані постійного розвитку, регулярно поліпшує та оновлює свій функціонал, що з кожним

днем робить її найбільш доцільною програмою для створення прототипів графічних інтерфейсів.

До основних переваг даного інструменту можна включити те, що працювати у Figma можна безпосередньо у браузері без втрати широкого функціоналу. Це значно спрощує процес створення дизайну, адже дизайнер не повинен мати дуже потужний ноутбук чи комп'ютер для роботи з цим інструментом, чого не можна сказати про роботу з Adobe Photoshop. Figma дуже швидкий та гнучкий інструмент, що надає можливість працювати із додатком Google Fonts (шрифти). Це вирішує проблему несумісності використаних у певному дизайні шрифтів із шрифтами на пристрої конкретного користувача.

Створення прототипів графічних інтерфейсів у Figma прискорюється наявністю вже готових кнопок, форм та іншого. Ви можете шукати та використовувати будь-які готові елементи у своїх дизайнах та немає необхідності витрачати час на їх створення з нуля.

Існує також функція роботи із копіями. Можна налаштувати Figma так, що змінюючи один суб'єкт конкретної кнопки наприклад будуть змінюватися усі її копії в різних частинах дизайну. З цією функцією неможливо пропустити та не оновити той чи інший елемент, а також не потрібно витрачати час на зміну кожного екземпляру певного об'єкту.

Figma надає можливість встановлювати зв'язки між сторінками та елементами. Якщо існує потреба продемонструвати взаємодію елементів, то фігма найкраще пасує для таких цілей. Можна вказати усі зв'язки, тоді клік на кнопку/сторінку, буде відкриватися зв'язаний з цією кнопкою/сторінкою елемент. Завдяки цьому можна одразу корегувати та налаштовувати дизайн до усіх вимог та потреб користувача.

Figma має дуже зручний, зрозумілий та простий інтерфейс (рис. 3.5).

Для створення дизайн-макетів був обраний інструмент Figma, адже це дозволить продемонструвати зв'язок елементів, тобто увесь шлях навігації по сайту, а також цей інструмент має гнучкий та зрозумілий інтерфейс і не

потребує встановлення на локальний пристрій, що спрощує та прискорює редизайн сайту.

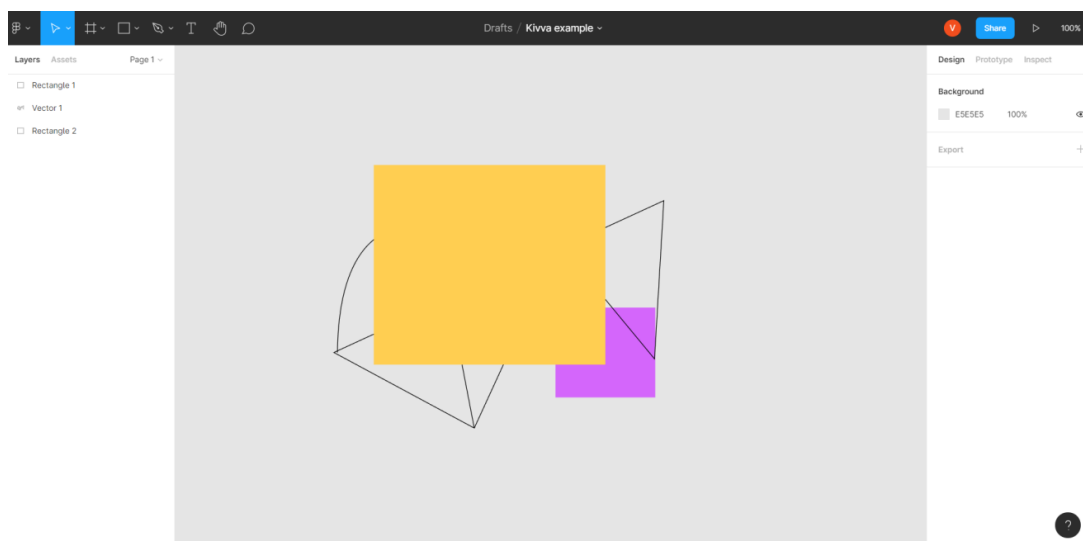


Рисунок 3.5 – Інтерфейс інструменту Figma

### 3.3 Мова програмування C#

Мова програмування C# одна з найпопулярніших мов для розробки програмного забезпечення. Ця мова існує з 2002, тому завдяки цьому більшість проблем та недоліків були виправлені та сама мова була оновлена і покращена багато разів. На C# пишуться різноманітні додатки, як невеликі десктопні так і величезні веб-сервіси.

Мова програмування C# була створена для роботи із фреймворком .NET.

.NET – це потужна платформа для створення додатків. Має безліч переваг, такі як:

- підтримка декількох мов програмування. Це дозволяє створювати різні модулі одного продукту на різних мовах;

- кросплатформеність – на цій платформі можна розроблювати додатки для різноманітних платформ, таких як Windows, MacOS, Linux, Android, iOS, Tizen;

- велика та потужна бібліотека класів. Бібліотека класів допомагає створювати додатки швидко та якісніше;

– додатки розроблені на цій платформі дуже велику продуктивність у роботі.

Як C# так і .NET мають таку функцію, як автоматичний збір сміття. Тобто після виконання коду нам не потрібно турбуватися про звільнення місця на нашому комп'ютері, адже функція збору сміття зробить це сама.

## 4 ПРОЕКТУВАННЯ ТА РОЗРОБКА АВТОМАТИЗОВАНОГО ТЕСТУ МОВОЮ С#

### 4.1 Аналіз html структури сайту Amazon

Перед створенням автоматизованого тесту для тестування сайту, потрібно проаналізувати його html/css код, виявити можливі ризики та проблеми та загалом дослідити структуру для подальшої праці із кодом.

Подібна перевірка виявить дублікати та помилки у html коді, дублювання контенту або зайві елементи, наявність яких сповільнює роботу сайту, швидкість відображення елементів та в результаті погіршує конверсію сайту.

Конверсія сайту – це відношення загального числа його відвідувачів до числа користувачів, які наприклад здійснили покупку або добавили товар до кошика.

Для аналізу html структури сайту та виявлення помилок були застосовані сервіси, які створені безпосередньо для такого аналізу.

Сервіс HTML web аналіз допомагає виявити помилки та проаналізувати загальні розміри скриптів та файлів, адаптивність верстки.

Також був використаний сервіс NU HTML Checker. Він виявив помилки в конкретних строках та файлах, наявність спам елементів.

В результаті аналізу головної сторінки сайту Amazon було виявлено:

- використання застарілого протоколу HTTP/1.1. HTTP/2 – це нова версія застарілого протоколу. Завдяки цьому підвищується швидкість завантаження сайтів. HTTP/1 текстовий протокол, тоді він менш компактний та ефективний. Він не стискає HTML заголовки. На сьогодні усі сучасні браузери підтримують роботу по протоколу HTTP/2;

- відсутня адаптивна верстка;

- перевищена кількість посилань на сторінці – 309 посилань. Це занадто багато для сторінки;

- перевищення кількості скриптів – 133;

– розмір скриптів на сторінці – 231258 байт. Це занадто багато, та було б краще винести їх до окремого файлу;

– кількість вбудованих CSS-стилів – 19416 байт. Це дуже багато, краще було в винести в сторонній файл.

Проаналізувавши структуру сайту були дослідженні шляхи для створення автоматизованого тесту, адже автоматизований тест керує запитам та діями браузера завдяки html коду сайту.

#### 4.2 Підготовка та збір даних для розробки тесту

Для створення автоматизованого тестування був створений мануальний тест кейс.

Test Case – це тестовий артефакт, який відповідає за опис та виконання певного тестового сценарію та перевірку необхідного функціоналу або UI частини сайту. Тестові сценарії проектуються QA інженерами (quality assurance – контроль якості). Тест кейс описує основні кроки по сайту, а також описує результат, який очікується після виконання певного кроку.

Для тестування основних елементів сайту Amazon було вибрано таке флоу:

Головна сторінка – форма пошуку + “С# book” – сторінка з результатами пошуку – сторінка товару – додавання друкованої версії книги до кошику – валідація кошику.

В результаті був створений тест кейс – табл. 4.1.

Для керування елементами сайту були створені xpath запити до кожного необхідного елемента. XPath – це потужна мова, яка часто використовується для парсингу веб-сайтів. Вона дозволяє звертатися до вузлів чи вираховувати значення з XML та HTML. Схожі функції використовують CSS селектори, але XPath дозволяє робити набагато більше.

Таблиця 4.1 – Тест кейс

Перейти на головну сторінку сайту Амазон за посиланням <a href="https://www.amazon.com/">https://www.amazon.com/</a>	Головна сторінка сайту відкрита.
Ввести в пошукове поле запит “с# book”. Натиснути кнопку пошуку	Пошук відбувся успішно. Сторінка з результатами пошуку відображається.
Перевірити заголовок сторінки	Заголовок сторінки відповідає пошуковому запиту – “с# book”
Вибрати першу книжку в списку пошуку, Paperback версію	Сторінка товару відкрита
Додати книжку до кошику	Товар доданий до кошику та сторінка кошику відкрита
Перевірити, що кількість доданих елементів у кошику дорівнює 1	У кошик доданий 1 товар. Цифра 1 відображається на іконці кошику в хедері сайту.

Використовуючи вирази шляхів, ця мова вибирає вузли або набори вузлів у XML. Бібліотека стандартних функцій допомагає користувачам розуміти та працювати з елементами та атрибутами XML. XPath дозволяє різним типам виразів отримувати відповідну інформацію з документа XML. XPath адресує певну частину документа. Він моделює XML-документ як дерево вузлів. Вираз XPath - це техніка навігації та вибору вузлів з документа. Вирази XPath можна використовувати в C#, C ++, Python, Java, JavaScript, PHP, XML-схемі та багатьох інших мовах. Вираз XPath посилається на шаблон для вибору набору вузлів.

Xpath запити та пошук за ID для головної сторінки:

- поле пошуку: `By.Id("twotabsearchtextbox");`
- кнопка пошуку в пошуковому полі: `By.XPath("//div[contains(@class, 'nav-search-submit')]")`.

Xpath запити для сторінки з результатами пошуку:

- кнопка Paperback для 1го елемента на сторінці пошуку: `By.XPath("//div[@data-component-type='s-search-result'" + "//div[not(contains(data-component-type, 'sp-sponsored result'))]/a[contains(text(), 'Paperback')]");`
- заголовок сторінки: `"//span[contains(text(), '_text_')]".`

Xpath запити та пошук за ID для сторінки товару:

– кнопка перегляду всіх опцій товару: `By.Id("buybox-see-all-buying-choices-announce");`

– кнопка додавання товару до кошику: `By.XPath("//input[@value = '1']/input[@name='submit.addToCart']");`

Xpath та CSS запити для сторінки кошику:

– кнопка переходу до корзини в хедері: `By.CssSelector("#nav-cart");`

– заголовок сторінки кошику: `By.XPath("//h1[contains(text(),'Shopping Cart')]");`

Загальні Xpath (для хедеру): лічильник кількості товарів у кошику на іконці в хедері сайту: `By.XPath("//span[@id='nav-cart-count' and contains(text(),'1')]");`

В результаті спроектованих Xpath, Css запитів автоматизований тест коректно керуватиме роботою браузеру та сайту. На основі цих запитів Selenium буде звертатися безпосередньо до елементів сайту. Для створення автоматизованого тесту за допомогою мови C# буде створений код з використанням спроектованих запитів.

## 4.3 Розробка автоматизованого тесту

### 4.3.1 Вибір принципу розробки автоматизованого тесту

Для розробки автоматизованого тесту сайту Amazon на мові C# був обраний підхід з використанням PageObjects. Page objects моделюють частини програми або сайту, які піддаються тестуванню. Цей принцип допомагає легко підтримувати автоматизовані тести та зробити їх максимально гнучкими та конфігурабельними.

Щоб змінити якусь частину коду, потрібно лиш змінити 1 місце в коді та усі пов'язані елементи, які використовують Page Object автоматично зміняться без потреби додаткових виправлень. Цей метод допомагає розділити технічні деталі, елементи в браузері, сторінки, функціонал певної

групи. Також логічно структурувати автоматизований тест, щоб зробити логіку тесту прозорою та зрозумілою для будь-кого, хто буде підтримувати та модифікувати тест.

Цей підхід також робить код більш компактним та зручним, зменшує дублікати елементів в коді. Даний підхід допоможе зробити автоматизовані тести стабільними та керованими. Інженер витратить менше часу на підтримку, тому продуктивність команди та швидкість виконання роботи значно збільшиться.

Автоматизований тест сайту Amazon має розділення за сторінками та логікою. Структура тесту з використанням методу Page Object виглядатиме таким чином:

- General – загальний клас, який містить перевірку хедеру сайту;
- MainPage – клас, який містить запити та методи, які виконуються на головній сторінці сайту;
- SearchPage – клас, який містить запити та методи, які виконуються на сторінці пошуку на сайті;
- ProductPage – клас, який містить запити та методи, які виконуються на продуктовій сторінці сайту;
- ShoppingCart – клас, який містить запити та методи, які виконуються на сторінці кошику сайту (рис. 4.1).

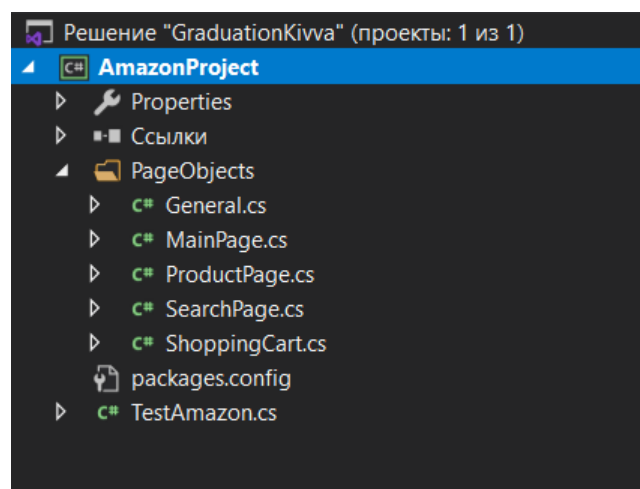


Рисунок 4.1 – Структура файлів проекту

### 4.3.2 Розробка автоматизованого тесту в Visual Studio

Розробка автоматизованого тесту починається із створення нового проекту та рішення у Visual Studio.

Для коректної роботи проекту, взаємодії із браузером та створення UI тестів загалом потрібно додати до проекту Nuget пакети (бібліотеки).

NuGet – це додаток до Visual Studio, який дозволяє швидко додавати та модифікувати сторонні бібліотеки та інструменти, які полегшать створення проекту. Тобто це певний код, створений іншими розробниками, яким вони поділилися для спрощення роботи всієї спільноти розробників.

Для створення автоматизованого тесту на C# з використанням Selenium були встановлені наступні пакети:

- MSTest.TestFramework;
- MSTest.TestAdapter;
- Selenium.WebDriver;
- Selenium.Support;
- Selenium.WebDriver.ChromeDriver.

Кожен з цих пакетів відповідає за функціонал тесту і створення автоматизованого тесту без них буде неможливим. В результаті цих оновлень можна створювати тест на мові програмування C#.

Спершу був створений клас, в якому буде тіло нашого тесту – MyTestAmazon. Цей клас містить звернення до інших елементів(класів) та відповідає за виконання самого тесту. В даному проекті був використаний MSTest, то ми задаємо наш тест за допомогою атрибутів [TestClass], [TestInitialize()], [DataRow], [DataTestMethod] та [TestCleanup()]. Кожен тестовий клас повинен мати атрибут TestClass. Щоб вказати на те, що тест параметризований, було додано атрибут [DataTestMethod] в який можна передавати різноманітні примітивні параметри за допомогою атрибуту [DataRow]. В автоматизованому тесті для C# в атрибуті [DataRow] було передано 1 значення – “c# book”.

[TestInitialize()] – цей атрибут потрібен для того, щоб запустити певну функцію перед виконанням самого тесту. В автоматизованому тесті тут було проініціалізовано функцію запуску драйверу браузера.

[TestCleanup()] – це атрибут має протилежні функції до атрибуту [TestInitialize()] та виконується після виконання самого тесту. В автоматизованому тесті тут передана функція, яка закриває усі екземпляри браузера та драйверів браузера. Це потрібно обов'язково робити для подальшого коректного виконання тесту/тестів, адже при наявності активного екземпляру браузера може виникнути конфлікт у виконанні наступних тестів. Слідуючи методу Page Objects були створені класи для позначення функціоналу кожної сторінки. Приклад наведений далі (рис 4.2), (рис 4.3). Більше прикладів наведено у Додатку А.

```
1  using System;
2  using AmazonWebDriverProject;
3  using OpenQA.Selenium;
4  using OpenQA.Selenium.Support.UI;
5
6
7  namespace AmazonProject.PageObjects
8  {
9      class SearchPage
10     {
11         private readonly IWebDriver _driver;
12         private readonly WebDriverWait _wait;
13
14         [Constructor]
15         public SearchPage(IWebDriver driver)
16         {
17             _driver = driver;
18             _wait = new WebDriverWait(driver, TimeSpan.FromSeconds(60));
19
20             public By doPaperbackLink = By.XPath("//div[@data-component-type='s-search-result']" +
21             "/div[not(contains(data-component-type, 'sp-sponsored-result'))]/a[contains(text(), 'Paperback')]");
22             public string pageHeader = "//span[contains(text(), '_text_')]";
23
24             [Constructor]
25             public void verifyPageHeader(string searchEntry)
26             {
27                 wait.Until(condition =>
28                 {
29                     try
30                     {
31                         var replaceHeader = pageHeader.Replace("_text_", searchEntry);
32                         var header = _driver.FindElement(By.XPath(replaceHeader));
33                         return header.Displayed;
34                     }
35                     catch (Exception)
36                     {
37                         return false;
38                     }
39                 });
40             }
41         }
42     }
43 }
```

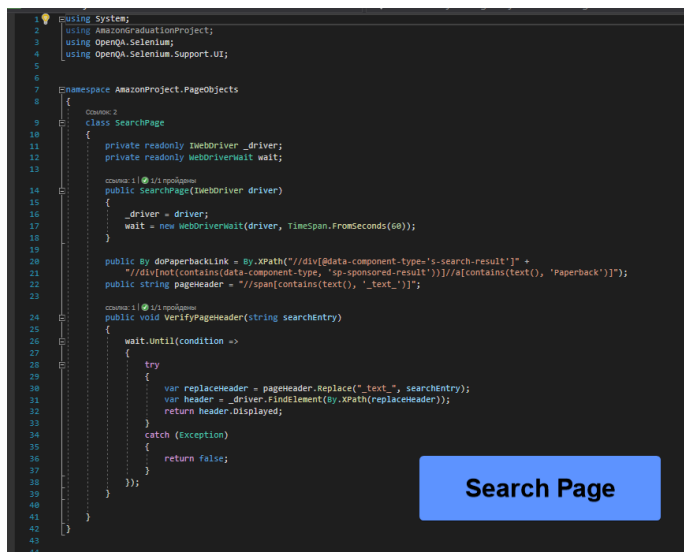


Рисунок 4.2 – Клас SearchPage

```
1  using System;
2  using OpenQA.Selenium;
3  using OpenQA.Selenium.Support.UI;
4
5
6  namespace AmazonProject.PageObjects
7  {
8      class ProductPage
9      {
10         private readonly IWebDriver _driver;
11         [Constructor]
12         public ProductPage(IWebDriver driver)
13         {
14             _driver = driver;
15
16             [Constructor]
17             public void ClickCartButton()
18             {
19                 try
20                 {
21                     var seeAllChoices = _driver.FindElement(By.Id("button-see-all-buying-choices-announce"));
22                     var addToCartButton = _driver.FindElement(By.XPath("//input[@value = '1']/input[@name='submit.addToCart']"));
23                     seeAllChoices.Click();
24                     addToCartButton.Click();
25                 }
26                 catch (Exception)
27                 {
28                     var addToCartButton = _driver.FindElement(By.Id("add-to-cart-button"));
29                     addToCartButton.Click();
30                 }
31             }
32         }
33     }
```

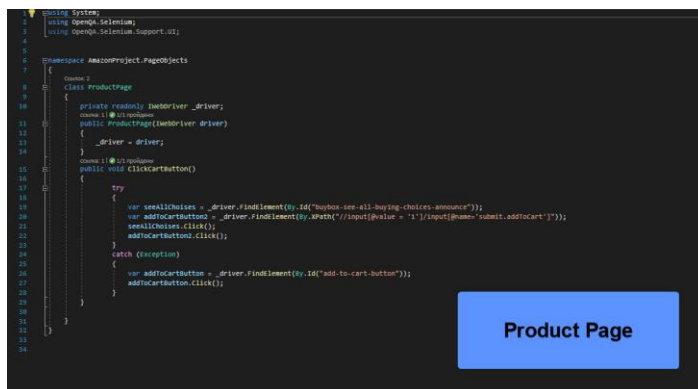


Рисунок 4.3 – Клас ProductPage

## 5 ЗАПУСК АВТОМАТИЗОВАНОГО ТЕСТУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Запуск тесту був виконаний на браузері Chrome. Тест був успішно пройдений, проте цей результат потребував більш складних запитів та коду, тому що сайт Amazon перевантажений зайвими елементами у HTML структурі, які не використовуються або застарілі і не виконують свої функції. Це поскладнювало процес керування елементами сайту за допомогою автотесту, адже знаходження елементів за запитом виконувалось довше, ніж могло би. Сторінки сайту завантажувалися та відображалися довше за рахунок перевантаженості сайту зайвими скриптами та елементами.

В результаті проходження тесту декілька разів та виведення середнього цілого був розрахований середній час проходження класичного шляху користувача на сайті (не враховуючи людський фактор, а лише механічні дії) – 8 хвилин. Автотестом було пройдено 5 сторінок сайту Amazon і враховуючи відсутність людського фактору затримки та роздумування на певних сторінках, 8 хвилин – це занадто довго для класичного шляху користувача. Це відбулося через велику кількість елементів та реклами на сайті-маркетплейсі, що призводить до складної структури HTML/CSS сайту.

Результат виконання тесту наведений на рис 5.1.

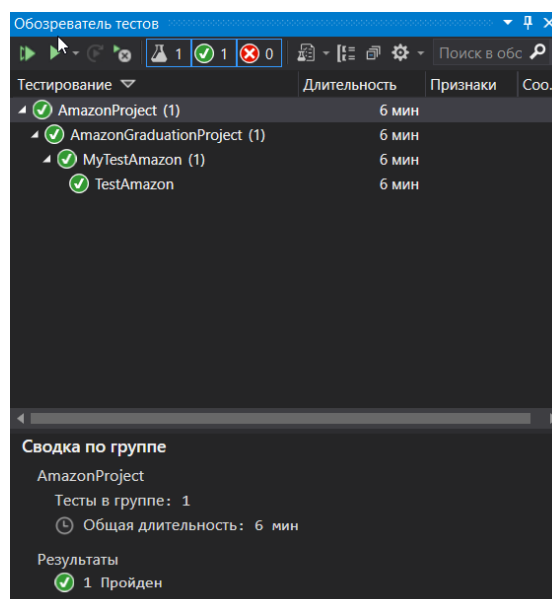


Рисунок 5.1 – Результат виконання автотесту

## 6 ПРОВЕДЕННЯ МАНУАЛЬНОГО ТЕСТУВАННЯ САЙТУ AMAZON

З метою виявлення візуальних недоліків та з метою аналізу зручності та трендовості сайту Amazon було проведено мануальне тестування сайту. Мануальне тестування – це перевірка функціоналу або UI сайту QA інженером, як кінцевим користувачем. Для здійснення такого тестування повинна бути підготовча база знань в області UI/UX або чіткі задокументовані вимоги до продукту.

Для мануального етапу тестування був спроектований тестовий сценарій на перевірку візуальної частини сайту, а також зручності використання. Тестовий сценарій був розроблений з урахуванням перевірок наявності елементів, їх відображення, акценти на сайті, розташування елементів тощо.

В результаті був створений тест кейс:

Таблиця 6.1 – Тест кейс

Перейти на головну сторінку сайту Амазон за посиланням <a href="https://www.amazon.com/">https://www.amazon.com/</a>	Головна сторінка сайту відкрита.
Перевірити наявність хедеру та розміщення елементів у ньому	Хедер присутній. Елементи розміщені рівномірно та чітко проакцентовані
Перевірити розміщення блоків тіла сайту	Блоки чітко розмежовані. Текст на блоках чіткий та читабельний. Блоки розміщені логічно та відсутня велика кількість зайвих промо блоків.
Ввести в пошукове поле запит “с# book”. Натиснути кнопку пошуку	Пошук відбувся успішно. Сторінка з результатами пошуку відображається.
Перевірити розташування та розмежування товарів на сторінці з результатами пошуку	Товари чітко розмежовані та логічно розподілені
Перевірити короткий опис товару для присутніх результатів пошуку	Опис товару читабельний та усі деталі читабельного розміру та контрастні по відношенню до фону сторінки
Перевірити заголовок сторінки	Заголовок сторінки відповідає пошуковому запиту – “с# book”
Перевірити наявність фільтрів та їх розмежування	Фільтри наявні та чітко та логічно розмежовані кольором або границями

## Продовження таблиці 6.1

Перевірити наявність сортування	Функціонал для сортування наявний та візуально помітний та логічно розміщений на сторінці
Вибрати першу книжку в списку пошуку, Paperback версію	Сторінка товару відкрита
Перевірити акценти на сторінці та відсутність зайвої спам інформації	Головні акценти зроблені на описанні товару, назві та кнопці Додати до кошику
Перевірити наявність кнопки Back to results	Кнопка присутня та контрастна по відношенню до фону
Перевірити наявність дрібних та нечитабельних елементів на сторінці товару	Дрібні, нечитабельні та зайві елементи відсутні на сторінці
Додати книжку до кошику	Товар доданий до кошику та сторінка кошику відкрита
Перевірити наявність повідомлення про доданий товар	Повідомлення присутнє та яскраво виділене на сторінці
Перейти до кошику та перевірити, що кількість доданих елементів у кошику дорівнює 1	У кошик доданий 1 товар. Цифра 1 відображається на іконці кошику в хедері сайту.
Перевірити відображення товару у списку кошику	Товар відображається у основній частині сайту та зрозуміло виділений на сторінці
Перевірити наявність кнопки Proceed to checkout та її акцентність	Кнопка присутня та контрастно виділена розміром та кольором на сторінці

В результаті мануального тестування було виявлено ряд помилок.

Помилки та недоліки на головній сторінці сайту:

- розмежування блоків неочевидне та залежить від кольору динамічного фону, через це майже повністю зливається із світлими фонами;
- надписи в кожному блоці погано видно через відсутність розділення зображень товарів та опису категорії;
- наявність дрібних і нечитабельних підписів категорій у блоках;
- кнопки “Shop now”, “See more”, “Explore now” нечитабельні та блідні, на світлому фоні майже непомітні;
- велика кількість блоків з рекомендаціями на сторінці та щільність розміщення елементів утворюють враження завантаженості та брудності сайту (рис 6.1-6.2).

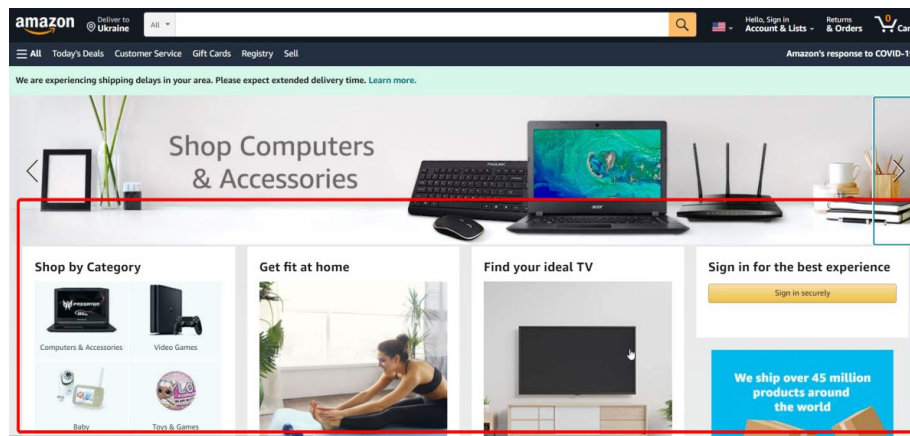


Рисунок 6.1 – Відсутність виділення блоків та дрібні надписи на головній сторінці сайту

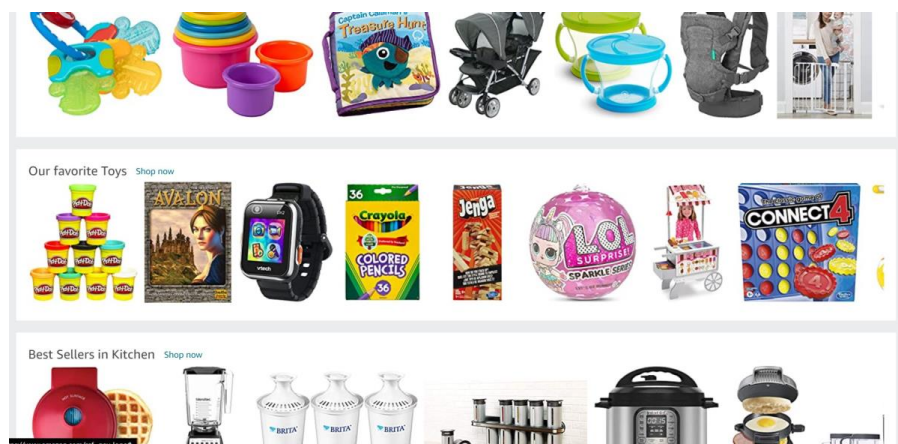


Рисунок 6.2 – Приклад хаотичного розміщення рекомендацій (товарів) без підпису на головній сторінці сайту

Помилки та недоліки на сторінці з результатами пошуку:

- Dropdown сортування блідний, дрібний та крихітний по відношенню до інших елементів;
- фільтри не розділені, з'являється відчуття цільного тексту, хоча це різні пункти фільтру;
- відсутність розподілення блоків товарів;
- відсутність акцентів та відчуття суцільного хаосу на сторінці (рис. 6.3).

Помилки та недоліки на сторінці товару:

- величезна кількість як дрібних так і великих елементів, що збиває з пантелику користувача;

- наявність великої кількості дрібних текстових елементів, які не виділені кольором або розміром;
- недоречне розміщення нелогічних рекомендацій в блоці під хедером;
- деталі продукту знаходяться під 4ма блоками з рекламою та рекомендаціями;
- блок для додавання елемента в корзину або покупки містить велику кількість хаотично розміщених елементів, які виділені різними кольорами, відтінками, а також більшість з них занадто дрібні та нечитабельні (рис. 6.4).

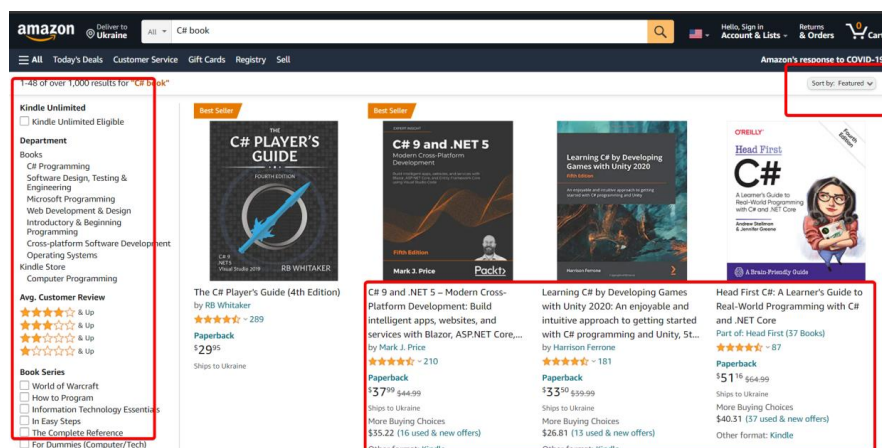


Рисунок 6.3 – Приклад помилок на сторінці з результатами пошуку

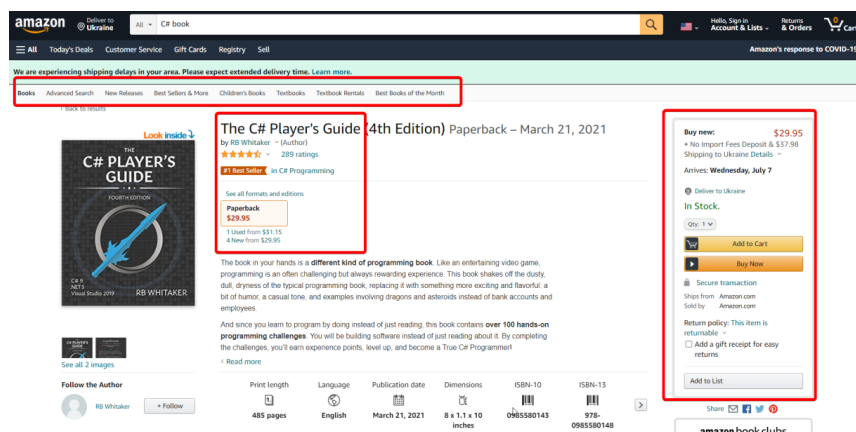


Рисунок 6.4 – Приклад помилок на сторінці товару

Помилки та недоліки на сторінці з повідомленням про доданий товар:

- дрібне та не акцентне повідомлення про доданий товар. Рекомендації на сторінці більш помітні, ніж повідомлення;

– панель справа містить дрібні та нечитабельні елементи, виглядає недоречно та старомодно. Виглядає ніби зайвий елемент або помилка (рис. 6.5).

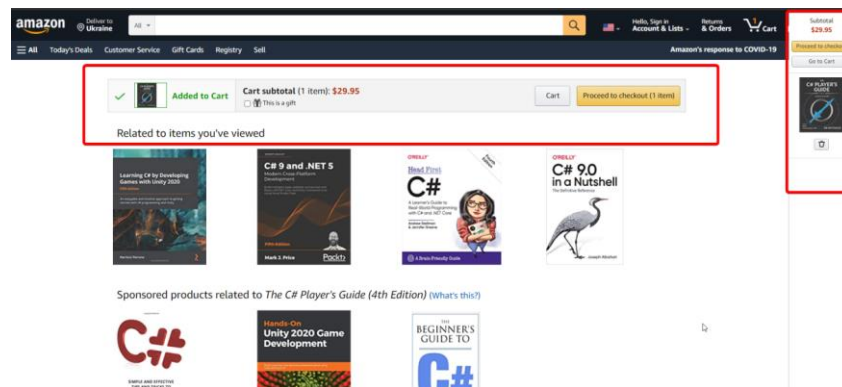


Рисунок 6.5 – Приклад помилок на сторінці з повідомленням про доданий товар

Помилки та недоліки на сторінці кошику:

- дрібні елементи в описі товару;
- кнопка для оформлення замовлення майже однакового розміру та однакового кольору із рекомендаціями;
- блок з товарами у корзині подібний за кольором до фону, тому відсутнє контрастне розподілення (рис. 6.6).

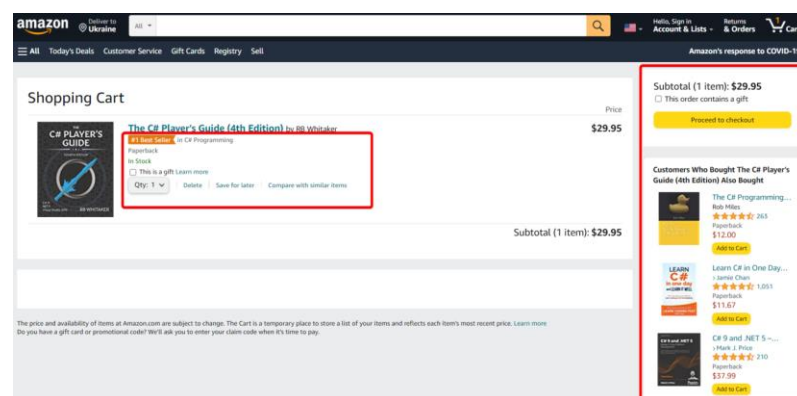


Рисунок 6.6 – Приклад помилок на сторінці кошику

В результаті двох видів тестування та аналізі помилок та недоліків був розроблений редизайн сайту Amazon.

## 7 РОЗРОБКА РЕДИЗАЙНУ САЙТУ ЗА ДОПОМОГОЮ ІНСТРУМЕНТУ FIGMA

### 7.1 Підбір нового кольорового та шрифтового оформлення

Редизайн – це певне оновлення розміщення елементів та блоків, логіки сайту і звісно кольорового оформлення. Підбір доречної та актуальної кольорової схеми це один із найважливіших етапів розробки сайту. Колір може як підкреслити, так і зіпсувати презентацію продукту та сайт в цілому. Колір впливає на комфорт користувача при знаходженні на сайті. Колір повинен бути контрастним, проте збалансованим і приємним оку.

Кольорова гама повинна бути доречною для конкретного продукту. Враховувати цільову аудиторію та її потреби.

Для редизайну сайту Amazon були вибрані такі кольори – рис 7.1.



Рисунок 7.1 – Кольорова палітра сайту Amazon після редизайну

Усі кольори пастельних та приглушених відтінків, що не навантажає око користувача, адже сайт-маркетплейс значить, що людина буде проводити багато часу вибираючи товари та приймаючи рішення. Більш того така кольорова палітра деякими моментами подібна до палітри старої версії сайту, що позитивно вплине на реакцію користувача на редизайн, адже Amazon це дуже багаторічна платформа та для деяких людей такі кольори звичні та асоціюються саме з цією платформою.

Синій колір асоціюється із спокоєм та відпочинком. Також синій дає відчуття сили та лідерства. Більш того синій надає лояльності до продукту

або сайту. Користувачу комфортно проводити свій час на сайті. Синій колір гарно комбінується із жовтим та помаранчевим. Ці кольори відповідають за щастя та радість. Також був використаний сірий нейтральний колір щоб уникнути перевантаження дизайну кольорами.

## 7.2 Вибір шрифтового оформлення

В якості основної гарнітури було вибрано шрифт Inter Bold та Medium.

Цей шрифт був створений на основі шрифту Roboto, проте його зробили більш зручним та читабельним. Це сімейство шрифтів було створено саме для використання у розробці графічних інтерфейсів.

## 7.3 Розробка модульної сітки

Модульна сітка – це система розміщення елементів на сайті, яка базується на подібних колонках та строках . Модульні сітки допомагають утримувати увагу користувача та підвищують читабельність та якість сприйняття тексту та елементів на сторінці. Розміщення блоків та тексту виглядає збалансовано. Модульна сітка структурує усе, що розміщено на сайті для комфортного проектування та розробки дизайну. Модульна сітка відповідає за розміщення елементів та дотримання єдиного формату.

В ході роботи була розроблена 10-ти колонна модульна сітка для редизайну сайту Amazon (рис 7.1).

## 7.4 Розробка макету редизайну сайту Amazon

Основною метою редизайну сайту є покращення, полегшення та оновлення дизайну сайту згідно сучасним трендам, потребам та правилам зручності використання. Для сайту маркетплейсу найважливіше це бути зрозумілим, мати прозору логіку та простий лаконічний дизайн із стриманими, пастельними кольорами.



Рисунок 7.2 – Модульна сітка

Був спроектований простий графічний інтерфейс, присутня достатня кількість “повітря” між блоками та елементами сайту, що запобігає перевантаженості та допомагає концентрувати увагу на основних елементах.

Для сайтів маркетплейсів головна мета – це продати товар, та зарекомендувати якнайбільше додаткових товарів, проте не перевантажити та не втомити користувача надмірної кількістю інформації.

Коли користувач заходить на сайт маркетплейс в нього одна мета – знайти та придбати конкретний товар. Тому поле пошуку є одним із найважливіших елементів на подібному сайті (рис. 7.3).

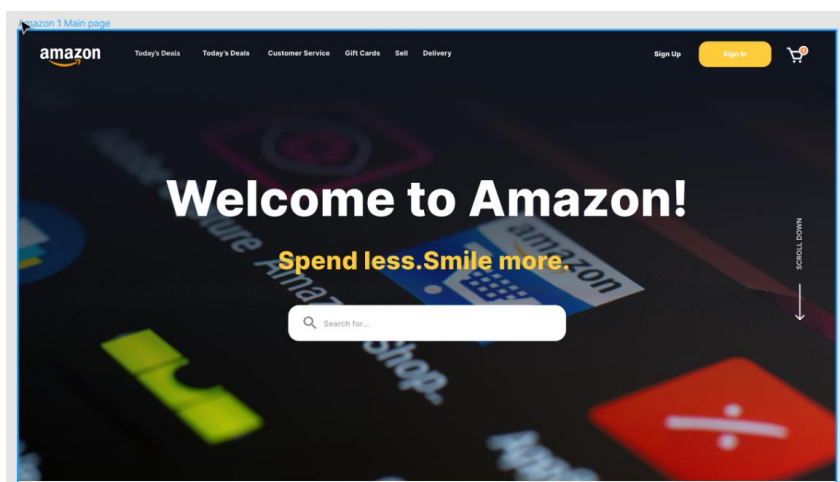


Рисунок 7.3 – Головна сторінка сайту. Поле пошуку

Фільтрація на сайті маркетплейсі – це дуже важливий елемент, який відповідає за зручність користування сайтом. Блок фільтрації був логічно та графічно розділений на складові, що допоможе користувачу швидко та зрозуміло знаходити потрібний фільтр (рис. 7.4).

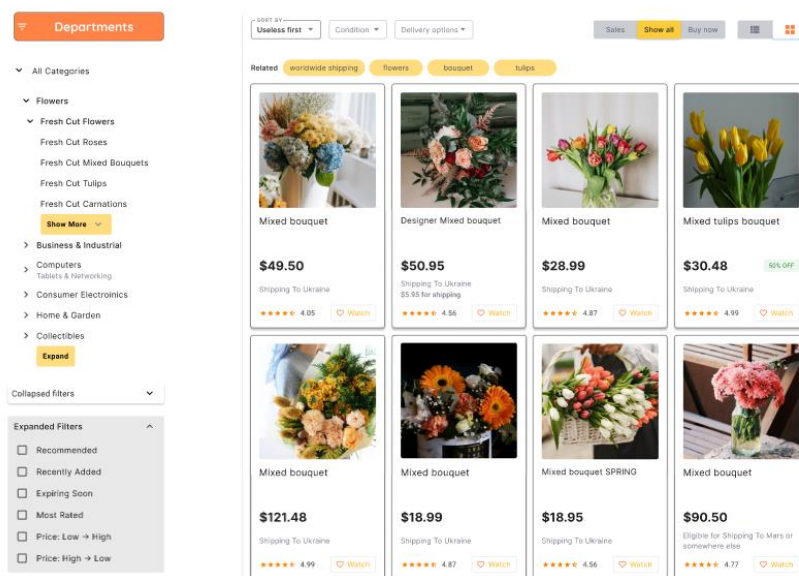


Рисунок 7.4 – Фільтрація на сайті

Сортування на сторінці з результатами пошуку відіграє важливу роль, адже кожен користувач звик до певного типу перегляду сайтів маркетплейсів або наприклад користувач хоче швидко продивитися список товарів не вдаючися у деталі. В цьому випадку йому допомагає сортування. Тому кнопки сортування були розроблені яскравими та помітними із прозорим функціоналом (рис. 7.5).

Сторінка товару була розроблена без зайвих елементів, з чітко виділеними акцентами та прозорим функціоналом тих чи інших елементів.

Кнопки для додавання в корзину або оформлення замовлення виділені кольором. Рекомендації для користувача не відволікають увагу від товару та його опису. Вони оформленні у стриманому, лаконічному вигляді та не беруть на себе основний акцент на сторінці. Опис товару знаходиться одразу поруч із товаром, а не унизу сторінки, тому користувач одразу може дізнатися усю необхідну інформацію про товар без додаткового пошуку (рис. 7.6).

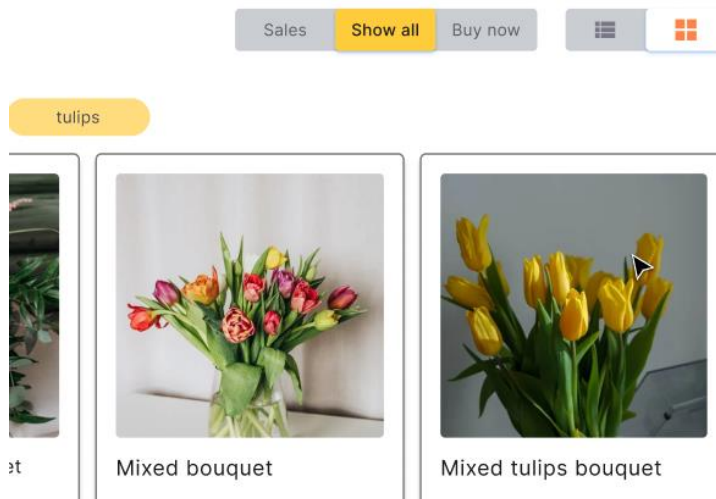


Рисунок 7.5 – Сортування на сайті

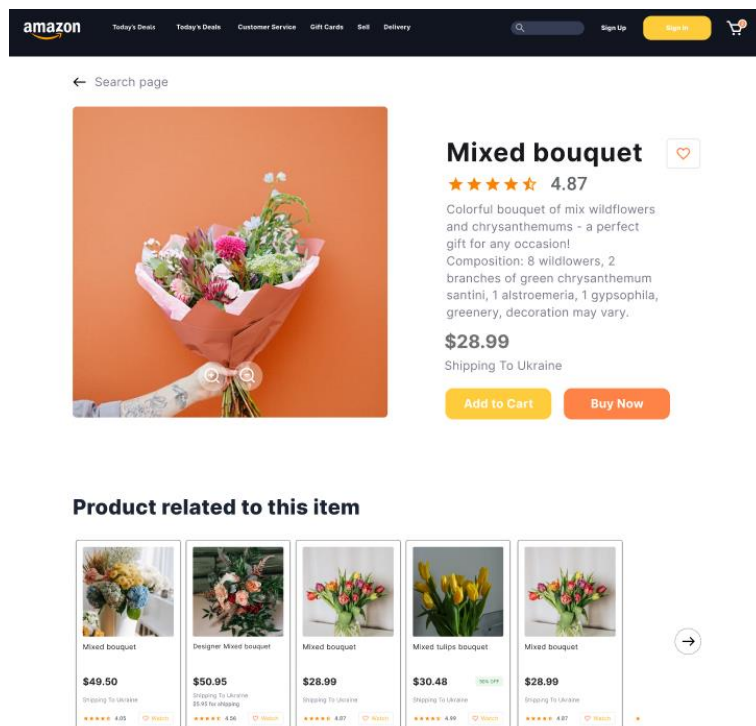


Рисунок 7.6 – Сторінка товару

На сторінці із повідомленням про доданий товар повідомлення великого розміру, виділене кольором та границями із читабельним шрифтом та у вибраній кольоровій гамі дозволяє користувачу сфокусуватись саме на ньому. Усі додаткові рекомендації знаходяться у нижчій частині сторінки та поділені по категоріям (рис. 7.7).

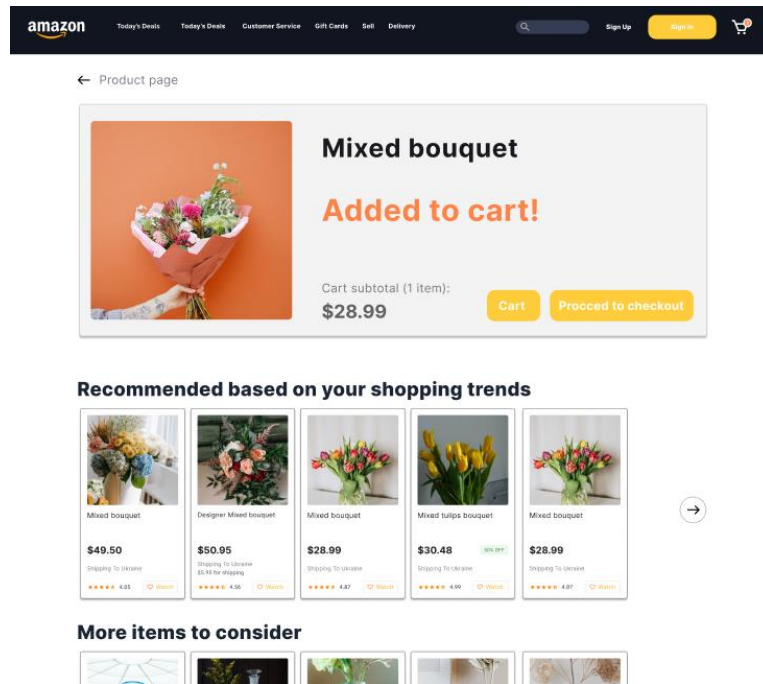


Рисунок 7.7 – Сторінка із повідомленням про доданий товар

Більше прикладів редизайну наведені у додатку А.

## 8 ТЕСТУВАННЯ РЕДИЗАЙНУ САЙТУ ЗА ДОПОМОГОЮ ФОКУС ГРУПИ

Фокус група – це метод дослідження, коли запрошується група користувачів для тестування певного продукту та відгуку про цей продукт після.

Найкраще включати у фокус групу цільову аудиторію продукту, адже тоді оцінка тестування буде наймеш суб'єктивною.

Було проведено мануальне тестування редизайну сайту фокус групою із 10 людей за тестовим сценарієм та отримано такі висновки:

- елементи на сайті читабельні та зрозумілі;
- кольорове оформлення сайту відповідає тенденціям 2021 року та правилам комбінування кольорів;
- поле пошуку помітне та зручне;
- хедер сайту лаконічний та усі його елементи читабельні;
- результати пошуку (товари) розміщені симетрично та вдало. опис товарів читабельний та функціонал додавання товарів до кошика прозорий та зрозумілий;
- сторінка кошика містить список чітко розподілених елементів (товарів);
- кнопка для оформлення замовлення акцентно виділена та помітна;
- ціна товару як на сторінці пошуку так і на сторінці кошика читабельна, контрастна до фону та розміщена зрозуміло та логічно відносно інших блоків сайту;
- головна сторінка сайту не перевантажена додатками, рекламою та рекомендаціями, проте все одно надає можливість порекомендувати товари користувачу, але у більш лаконічній та зрозумілій формі.

## 9 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОБОТИ

### 9.1 Характеристика продукції та ринок збуту

У результаті виконання кваліфікаційної роботи створений автоматизований тест до сайту Amazon та створений його редизайн. Головне завдання тестування сайту – визначити усі недоліки та помилки у дизайні сайту, виокремити те, що може заважати користувачу. Основне завдання редизайну – це оновити кольорове рішення на сайті, виправити помилки та зробити сайт більш комфортним та зрозумілим для користувача.

Оновлення дизайну сайту гарно впливає на репутацію продукту. Користувач бачить, що розробники та власники продукту оновлюють та піклуються про сайт та продукт, а це означає, що власник продукту зможе надати гарний сервіс.

### 9.2 Розрахунок витрат

У розробку автоматизованого тесту та редизайн за результатами тестування включають такі етапи:

- перший етап - проектування, на якому проводиться аналіз HTML структури сайту, а також збираються та підготовлюються дані для створення автоматизованого тесту;

- етап розробки автоматизованого тесту;
- етап проектування та проведення мануального тестування сайту;
- етап аналізу результатів;
- етап проектування редизайну сайту (підбір оновленого кольорового рішення, шрифтів, розробка модульної сітки);
- етап розробки макету редизайну сайту.

У собівартість розробки веб-сайту входять наступні статті витрат:

- основна заробітна плата;
- єдиний соціальний внесок;

- витрати на електроенергію;
- витрати на обслуговування устаткування.

Тестування та розробку редизайну сайту проводять два фахівці: QA інженер та дизайнер. Зарплата QA інженера становить 17000 грн/міс, дизайнера 12000 грн/міс. При цьому тривалість робочого дня кожного з них становить 8 годин. Автоматизований тест та редизайн сайту розроблялися 14 днів. Розрахунок основної заробітної плати наведено в таблиці 9.1.

Ставка єдиного соціального внеску становить 22% від величини основної заробітної плати:

$$V_{\text{соц}} = ЗП_{\text{осн}} \cdot 0,22,$$
$$12960 * 0,22 = 2851,2 \text{ грн},$$

де  $ЗП_{\text{осн}}$  – основна заробітна плата.

До інших витрат слід віднести витрати на електроенергію. Витрати на електроенергію розраховуються виходячи з споживаної потужності пристрою і тарифу на електроенергію.

У даному випадку передбачається використання 2 комп'ютерів з потужністю 0,5 кВт/год. Вартість одного кВт/години електроенергії дорівнює 1,619 грн.

Час використання обладнання в процесі розробки ( $T_{\text{об}}$ ) розраховується за формулою:

$$T_{\text{об}} = T_{\text{р}} \cdot C, \tag{9.1}$$
$$T_{\text{об}} = 14 * 8 = 112 \text{ годин},$$

де  $T_{\text{р}}$  – необхідна кількість днів для розробки;

$C$  – кількість робочих годин на добу.

Таблиця 9.1 – Розрахунок основної заробітної плати

Етапи	Вид робіт	Виконавець		Середня заробітна плата в день, грн	Дні на виконання	Трудо-місткість	Заробітна плата, грн.
		Кількість	Посада				
1. Проектування	Аналіз HTML структури сайту. Збір даних для створення автоматизованого тесту;	1	QA інженер	680	1	1	680
2. Розробка тесту	Розробки автоматизованого тесту	1	QA інженер	680	4	2	5440
3. Мануальне тестування сайту	Проведення мануального тестування	1	QA інженер	680	1	1	680
4. Аналіз результатів.	Аналіз результатів та проектування редизайну сайту	1	QA інженер	680	2	1	1360
5. Проектування редизайну	Проектування редизайну сайту	1	Дизайнер	480	2	1	960
6. Розробка редизайну	Розробка редизайну сайту	1	Дизайнер	480	4	2	3840
Всього					14	8	12960

Звідки, плата за електроенергію (Ел) складає:

$$Eл = T_{об} \cdot P \cdot Tar, \quad (9.2)$$

$$Eл = 112 * 0,5 * 2 * 1,619 = 181,328 \text{ грн.},$$

де  $P$  – потужність;

$Tar$  – ціна однієї кВт/години електроенергії.

Витрати на обслуговування ЕОМ ( $B_{ЕОМ}$ ) визначаються з вартості ЕОМ і часу її експлуатації, після закінчення якого, вона підлягає заміні (звичайно

цей час не перевищує 3-х років), протягом року ЕОМ використовує 254 робочих дні:

$$V_{\text{ЕОМ}} = (4200 / (3 * 8 * 254)) * 64 = 44,09 \text{ грн.}$$

Проект впроваджується на 1 компанію, тому собівартість розробки становить 16036,618 грн.:

$$16036,618 : 1 = 16036,618 \text{ грн.}$$

Розрахуємо суму прибутку від реалізації розробки (виходячи з рівня рентабельності 30 %):

$$16036,618 * 0,3 = 4810,9854 \text{ грн.}$$

Також розраховуючи ціну кінцевого продукту, слід урахувати ПДВ (20% від ціни без ПДВ). Податок на додану вартість (ПДВ) – це непрямий податок, який входить в ціну товарів (робіт, послуг) та сплачується покупцем, але його облік та перерахування до державного бюджету здійснює продавець (податковий агент).

Розрахуємо ціну розробки сайту без податку на додану вартість (ПДВ):

$$16036,618 + 4810,9854 = 20847,6034 \text{ грн.}$$

Розрахуємо суму ПДВ, вона дорівнює 20% від ціни без ПДВ:

$$20847,6034 * 0,2 = 4169,5206 \text{ грн.}$$

Розрахуємо ціну сайту з урахуванням ПДВ:

$$20847,6034 + 4169,5206 = 25017,124 \text{ грн.}$$

Розрахунок ціни сайту представлений в таблиці 9.2.

Таблиця 9.2 – Розрахунок ціни

№	Показник	Сума, грн.
1	Заробітна плата	12960,00
2	Єдиний соціальний внесок	2508,00
3	Витрати на обслуговування ЕОМ	44,09
4	Плата за електроенергію	181,33
5	Собівартість розробки сайту	16036,618
6	Прибуток (30%)	4810,9854
7	Ціна без ПДВ	20847,6034
8	ПДВ (20%)	4169,5206
9	Ціна з урахуванням ПДВ	25017,124

Таким чином, повна вартість проектування автоматизованого тесту та редизайну сайту складе приблизно 25017,124 грн, час виконання замовлення складе 14 днів для команди з одного QA інженера і одного дизайнера. Очікувана сума прибутку складе 4810,985 грн.

## ВИСНОВКИ

В ході кваліфікаційної роботи був спроектований та розроблений автоматизований тест сайту-маркетплейсу Amazon за допомогою Visual Studio, Selenium та мови програмування C# та виконаний редизайн сайту згідно трендам 2021 року та правилам розробки якісних графічних інтерфейсів.

Результатом роботи є актуальний, трендовий та зручний інтерфейс сайту, вільний від зайвих дрібних елементів та з оновленою кольоровою гамою.

В ході роботи:

- був проведений аналіз сайтів аналогів
- обрано мову програмування для розробки автоматизованого тесту сайту;
- зроблений вибір інструментальних та програмних засобів для розробки автоматизованого тесту;
- зроблений вибір інструментальних та програмних засобів для редизайну графічного інтерфейсу сайту;
- розроблена нова модульна сітка;
- обрана оновлена кольорова гамма;
- розроблено редизайн та навігаційну структуру;
- створені макети сторінок та встановлені зв'язки між елементами інтерфейсу;
- проведено тестування сайту фокус групою та зроблені висновки про редизайн сайту.

В економічній частині був зроблений розрахунок вартості розробки автоматизованого тесту та редизайну сайту з урахуванням основної заробітної плати, єдиного соціального внеску, витрат на електроенергію, витрат на обслуговування та рекламу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Полозова Т.В. Комплекс навчально-методичного забезпечення навчальної дисципліни "Економіка і бізнес" підготовки бакалавр спеціальність 186 - Видавництво та поліграфія, спеціалізація "Видавничо-поліграфічна справа". Харків, ХНУРЕ. 2017. 382 с.

2. Методичні рекомендації до виконання економічної частини дипломних проектів, робіт для студентів денної та заочної форми навчання усіх спеціальностей / Л.В. Соколова, О.І. Горбач, С.В. Гришко, Є.В. Діденко, Л.В. Левченко, Г.М. Путятіна, В.Г. Харченко. Харків: ХНУРЕ, 2015. 49 с.

3. Методичні вказівки з виконання кваліфікаційної роботи для студентів денної та заочної форм навчання першого (бакалаврського) рівня вищої освіти спеціальності 186 "Видавництво та поліграфія" / В.П. Ткаченко, А.В. Бізюк, О.В. Вовк та ін. Харків : ХНУРЕ, 2021. 68 с.

4. Держстандарт України ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання.

5. Вимоги безпеки під час роботи на ПК. URL: <https://sites.google.com/site/ohoronaprasi44/33-vimogi-bezpeki-pid-cas-roboti-na-pk/> (дата звернення: 03.06.2021).

6. Тренди веб-дизайну 2021. URL: <https://redstone.media/тренди-веб-дизайну-2021> (дата звернення: 04.06.2021).

7. Use the MSTest framework in unit tests. URL: <https://docs.microsoft.com/en-us/visualstudio/test/using-microsoft-visualstudio-testtools-unittesting-members-in-unit-tests?view=vs-2019> (дата звернення: 25.05.2021).

8. Testing Tools in Visual Studio. URL: <https://docs.microsoft.com/en-us/visualstudio/test/?view=vs-2019> (дата звернення: 25.05.2021).

9. Основні тренди веб дизайну. URL: <https://webtune.com.ua/statti/dyzajn/osnovni-trendy-veb-dyzajnu/> (дата звернення: 25.05.2021).

10. Page Object Models. URL: [https://www.selenium.dev/documentation/en/guidelines\\_and\\_recommendations/page\\_object\\_models/](https://www.selenium.dev/documentation/en/guidelines_and_recommendations/page_object_models/) (дата звернення: 25.05.2021).

11. The Selenium Browser Automation Project. URL: <https://www.selenium.dev/documentation/en/> (дата звернення: 25.05.2021).
12. C# Documentation. URL: <https://docs.microsoft.com/en-us/dotnet/csharp> (дата звернення: 25.05.2021).
13. Nuget packages. URL: <https://www.nuget.org/packages> (дата звернення: 25.05.2021).
14. Great Figma Plugins for Designers. URL: <https://uxplanet.org/10-great-figma-plugins-for-designers-b904c16f9d8f> (дата звернення: 25.05.2021).
15. What is figma?. URL: <https://webdesign.tutsplus.com/articles/what-is-figma--cms-32272> (дата звернення: 25.05.2021).
16. The Power of Figma as a Design Tool. URL: <https://www.toptal.com/designers/ui/figma-design-tool> (дата звернення: 25.05.2021).
17. Лекція 11. Тестування програмного продукту. URL: <http://lib.mdpu.org.ua/e-book/vstup/L11.htm> (дата звернення: 25.05.2021).
18. Best Color Palette Generators. URL: <https://htmlcolorcodes.com/resources/best-color-palette-generators/> (дата звернення: 25.05.2021).
19. Чем Figma лучше Photoshop для разработки веб-интерфейсов . URL: <https://habr.com/ru/company/skillbox/blog/359258/> (дата звернення: 25.05.2021).
20. Шарко М.В., Мешкова-Кравченко Н.В., Радкевич О.М. Економіка підприємства: навч. посіб. для студ. ВНЗ. Ч. 1. Херсон: Херсон. нац. техн. ун-т., 2014. 434 с.
21. Dadkhah, M., Jazi, M., & Lyashenko, V. (2014). Prediction of phishing websites using classification algorithms based on weight of web pages characteristics. *Journal of Mathematics and Technology*, 5(2), 24-35.
22. Dadkhah, M., Jazi, M. D., Mobarakeh, M. S., & Lyashenko, V. (2015). Developing expert system in order to detect the journal phishing attacks. *Journal of Mathematics and Technology*, 6(1), 70-73.
23. Omarov, M., Tikhaya, T., & Lyashenko, V. (2019). Internet marketing metrics visualization methodology for related search queries. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), 2277-2281.

24. Baranova, V., Zeleniy, O., Deineko, Z., & Lyashenko, V. (2019, October). Stochastic Frontier Analysis and Wavelet Ideology in the Study of Emergence of Threats in the Financial Markets. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 341-344). IEEE.

25. Omarov M., Tikhaya T., & Lyashenko, V. Use of Wavelet Techniques in the Study of Internet Marketing Metrics. Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering, 20, 157-163.

26. Baranova, V., Zeleniy, O., Deineko, Z., Bielcheva, G., & Lyashenko, V. (2019, October). Wavelet Coherence as a Tool for Studying of Economic Dynamics in Infocommunication Systems. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 336-340). IEEE.

27. Дурняк Б. В., Ткаченко В. П., Чеботарьова І. Б. Стандарти в поліграфії та видавничій справі: довідник //Львів: Українська академія друкарства. – 2011.

28. Кипень, Н. Ю., Бокарева, Ю. С., & Дейнеко, Ж. В. (2016). Исследование особенностей плоского и материал-дизайна в UI-интерфейсах.

29. Бокарева, Ю. С., & Дейнеко, Ж. В. (2015). Исследование влияния цвета при проектировании целевых страниц LandingPage (Doctoral dissertation, HTMT).

30. Кулишова, Н. Е., Чеботарева, И. Б., Ткаченко, В. Ф., & Гурьева, Н. С. (2013). Поддержка стабильности цвета в открытых полиграфических системах: Монография.