

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач випускової кафедри  
\_\_\_\_\_ Аліна САВЧЕНКО

«\_\_\_»\_\_\_\_\_2023 р.

# КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР  
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ  
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

**Тема: «Вебсайт інтернет-магазину “StyleShop” на платформі React.js з  
використанням CSS та HTML»**

Виконавець: Едуард ЧЕРНІЦЬКИЙ

Керівник: к.т.н., доцент Олена ТОЛСТІКОВА

Нормоконтролер: к.т.н., доцент Олена ТОЛСТІКОВА

КИЇВ 2023

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій  
Кафедра комп'ютерних інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:  
завідувач кафедри КІТ  
Аліна САВЧЕНКО  
(підпис)  
«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

## ЗАВДАННЯ на виконання кваліфікаційної роботи *Черніцького Едуарда Владиславовича*

(ПІБ випускника )

1. Тема роботи: «Вебсайт інтернет-магазину “StyleShop” на платформі React.js з використанням CSS та HTML» затверджена наказом ректора № 623/ст від 01.05.2023р.

2. Термін виконання роботи: з 15 травня 2023 року по 25 червня 2023 року.

3. Вихідні дані до роботи: вебсайт інтернет-магазину “StyleShop” на платформі React.js з використанням текстового редактору Visual Studio Code.

4. Зміст пояснювальної записки: 1. Огляд та аналіз предметної області. 2. Особливості Frontend розробки. 3. Створення вебсайту інтернет-магазину “StyleShop”.

5. Перелік обов'язкового ілюстративного матеріалу: 1. Особливості Frontend розробки. 2. Вибір програмного забезпечення для розробки. 3. Застосування платформи React.js 4. Верстка постійних елементів інтернет-магазину. 5. Верстка навібару та банеру головної сторінки “StyleShop”. 6. Верстка додаткових розділів головної сторінки. 7. Верстка сторінки обраного товару.

8. Верстка інших сторінок. 9. Розробка функціональної складової вебсайту.

10. Вивантаження “StyleShop” на хостинг.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Огляд та аналіз предметної області. Написання 1 розділу, представлення керівнику.	15.05.2023- 20.05.2023	
2.	Вибір та опис використаних технологій. Написання 2 розділу, представлення керівнику.	22.05.2023- 26.05.2023	
3.	Створення вебсайту інтернет-магазину “StyleShop”. Написання 3 розділу, представлення керівнику.	29.05.2023- 02.06.2023	
4.	Тестування продукту. Хостинг проекту, представлення керівнику.	05.06.2023- 09.06.2023	
5.	Загальне редагування та друк пояснювальної записки.	10.06.2023- 12.06.2023	
6.	Проходження нормоконтролю, перепліт пояснювальної записки.	12.06.2023- 15.06.2023	
7.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	16.06.2023- 18.06.2023	

7. Дата видачі завдання \_\_\_\_\_ 15.05.2023р.

Керівник кваліфікаційної роботи

\_\_\_\_\_

Олена ТОЛСТІКОВА

(підпис керівника)

Завдання прийняв до виконання

\_\_\_\_\_

Едуард ЧЕРНІЦЬКИЙ

(підпис випускника)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Вебсайт інтернет-магазину “StyleShop” на платформі React.js з використанням CSS та HTML» містить: 66 сторінок, 43 рисунки, 17 інформаційних джерел.

**Об’єкт дослідження** – вебсайт інтернет-магазину.

**Предмет дослідження** – вебсайт інтернет-магазину “StyleShop” на платформі React.js з використанням CSS та HTML.

**Мета кваліфікаційної роботи** – розробити інтернет-магазин брендovих речей «StyleShop» на платформі React з використанням CSS та HTML за допомогою текстового редактору Visual Studio Code.

**Методи дослідження** – логічний, синтезу, аналізу, порівняльний, обробка літературних джерел.

Результати кваліфікаційної роботи рекомендується використовувати для загального ознайомлення з основними характеристиками та перевагами вебсайту "StyleShop" на платформі ReactJS. Вони можуть бути корисними для розуміння технологій, використаних у створенні сайту, його функціональності та можливостей.

Для розробки вебсайту інтернет-магазину “StyleShop” було завантажено та використано програмне забезпечення VS code, а також безкоштовна API документація, різноманітні фотографії товарів з доступних джерел, готові матеріали та шаблони.

FRONTEND, HTML, CSS, JAVASCRIPT, REACT.JS, REDUX TOOLKIT, VS CODE, SINGLE PAGE APPLICATION, ПРОЕКТУВАННЯ, РОЗРОБКА, ТЕСТУВАННЯ, ХОСТИНГ, ІНТЕРНЕТ-МАГАЗИН.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	6
ВСТУП .....	7
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1. Розвиток технологій та передумови розробки інтернет-магазинів .....	9
1.2. Перший інтернет-магазин .....	12
1.3. Інтернет-магазини в 21 столітті .....	14
1.4. Переваги та недоліки інтернет-магазину .....	16
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	18
РОЗДІЛ 2. ОСОБЛИВОСТІ FRONTEND РОЗРОБКИ .....	19
2.1. Методи підходу до розробки та основи Frontend .....	19
2.2. Вибір програмного забезпечення .....	23
2.3. Мова розмітки, стилі та програмування на React.js .....	34
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	43
РОЗДІЛ 3. СТВОРЕННЯ ВЕБСАЙТУ ІНТЕРНЕТ-МАГАЗИНУ “STYLESHOP” .....	45
3.1. Верстка макету .....	45
3.2. Розробка функціональної складової вебсайту, тестування .....	54
3.3. Хостинг .....	60
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	63
ВИСНОВКИ .....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	66

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

VS code	–	Текстовий редактор
HTML	–	Мова гіпертекстової розмітки
CSS	–	Каскадні таблиці стилів
FRAMEWORK	–	Програмна середовище, яке спрощує та прискорює створення програмного продукту
REACT	–	Javascript бібліотека, для створення інтерфейсів користувача
JSON	–	Текстовий формат обміну даними заснований на Javascript
IT	–	Інформаційні технології
ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
XML	–	Розширювана мова розмітки
API	–	Інтерфейс програмування додатків
CRM	–	Управління взаємовідносинами з клієнтами

## ВСТУП

У сучасному світі інтернет-магазини стають все популярнішими і зручнішими для покупців. Однак, щоб залучити більше клієнтів та забезпечити комфортну покупку, потрібно не просто мати сайт, але і зрозуміти, наскільки важливим є його візуальне оформлення. Розробка веб-сайту інтернет-магазину "StyleShop" на платформі React.js з використанням CSS та HTML, забезпечить відвідувачам сайту комфортну інтернет-покупку, що в свою чергу забезпечить успішність інтернет-магазину. Дослідження з вивчення впливу візуального оформлення сайту на настрої та емоції користувача, є актуальним в контексті залучення уваги та збільшення продажів. Одна з цілей це створити практичне візуальне оформлення сайту з різними товарами, категоріями та цікавою тематикою. Для розробки були обрані технології, які забезпечують швидке та ефективне створення веб-сайтів зі сучасним дизайном [2].

Для ефективного здійснення етапів розробки потрібно відповідально підійти до вибору програмного забезпечення для створення інтернет-магазину та бібліотек для оптимізації роботи. Саме тому до одного із розділів дипломного проекту включення питання вибору найбільш ефективного ПЗ та відповідних бібліотек для створення вебсайту інтернет-магазину "StyleShop".

**Актуальність** теми кваліфікаційної роботи «Вебсайт інтернет-магазину "StyleShop" на платформі React.js з використанням CSS та HTML» ґрунтується на тому, що вона дозволяє дослідити вплив візуального оформлення сайту та простоту користування, а також дозволить розглянути питання проектування веб-сайтів з точки зору сучасних веб-технологій та тенденцій веб-дизайну.

**Об'єктом дослідження** кваліфікаційної роботи є вебсайт.

**Предмет дослідження** – вебсайт інтернет-магазину "StyleShop" на платформі React.js з використанням CSS та HTML.

**Мета кваліфікаційної роботи** – розробити інтернет-магазин брендових речей «StyleShop» на платформі React з використанням CSS та HTML за допомогою текстового редактору Visual Studio Code.

Відповідно до поставленої мети роботи визначено основні **завдання дослідження**:

- провести комплексний аналіз методичної літератури стосовно архітектури та побудови інтернет-магазинів;
- поглибити знання про принципи веб-розробки на платформі React;
- проаналізувати особливості різних програмних забезпечень для написання коду та обрати найефективніший;
- розглянути можливі бібліотеки, вибрати необхідні для розробки;
- створити вебсайт інтернет-магазину “StyleShop”.

Для досягнення поставленої мети й виконання завдань використано наступні методи: логічний, синтезу, аналізу, порівняльний, обробка літературних джерел та проектування.

**Наукова новизна** роботи інтернет-магазину "StyleShop" полягає в унікальному досвіді покупок, поєднуючи сучасні технології з елегантним дизайном та інтуїтивно зрозумілим інтерфейсом. Цей інноваційний проект використовує передові функціональні можливості ReactJS для створення потужного та ефективного вебсайту, який відповідає потребам покупців.

**Практичне значення отриманих результатів.** Результати кваліфікаційної роботи рекомендується використовувати для загального ознайомлення з основними характеристиками та перевагами вебсайту "StyleShop" на платформі ReactJS. Вони можуть бути корисними для розуміння технологій, використаних у створенні сайту, його функціональності та можливостей. Отримані результати демонструють практичне застосування ReactJS для створення високоякісних інтернет-магазинів, що допомагає покупцям здійснювати зручні та швидкі покупки, а також підвищує ефективність та конкурентоспроможність бізнесу.

Завдяки швидкості та реактивності "StyleShop", покупці можуть насолоджуватися швидким та безперебійним переглядом товарів, миттєвим оновленням кошика покупок та швидким оформленням замовлення. Це покращує загальний досвід покупки і сприяє залученню та утриманню клієнтів.



## РОЗДІЛ 1

### ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1. Розвиток технологій та передумови розробки інтернет-магазинів

Люди звикли до інтернет-магазинів як до чогось самоочевидного, простого та доступного кожному. Все більше людей робить покупки, не виходячи з дому і навіть не встаючи від комп'ютера. Інтернет-магазини набули популярності і за останні кілька років їх стало дуже багато. Та коли з'явився перший сервіс і як розвивалася ця сфера?

Спочатку треба розібратися – з чого складається інтернет-магазин? Домен, хостинг, сайт, каталог товарів, cgm, cms, оплата, доставка. І що з цього було найпершим? Ось тут дивлячись від чого починати рахувати, і дивлячись що з усього цього найголовніше. Більшість джерел, а їх напрочуд мало, починають літопис інтернет-торгівлі з SABRE - американської системи резервування авіаквитків - це 1960-ті роки.



Рис. 1.1. Система резервування авіаквитків SABRE в дії

Кафедра КІТ				НАУ 23 33 34 000 ПЗ			
	ПІБ	Підпис	Дат	РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	Літ.	Аркуш	Аркушів
Розроб.	Черніцький Е.В.					9	10
Керівник	Толстікова О.В.				ТП-415Б - 122		
Н.Контр.	Толстікова О.В.						

Через цю систему приймали замовлення, а також аналізували продаж та формували ідеальну вартість квитка на конкретний момент часу. Такий собі аналог сучасної CRM. Система, яка й досі живе – в Аерофлоті, зокрема. Популярною SABRE стала завдяки тому, що навчилася заробляти на економних громадянах, підвищуючи або знижуючи для них вартість, залежно від часу до польоту та залишку. Ну і, ясна річ, прискоривши процес придбання квитків.

Наступний етап телетекст. Почалася ця справа в сімдесяті (Ceefax, 1974, Великобританія). Спочатку користувачеві лише виводилася текстова інформація. Але потім телевизор підключили до телефону та модему, і користувач поступово став активним покупцем. Якісний прорив у цій сфері судилося зробити французькому Мінітелю (1980-ті роки). Сприяло цьому становище Франції, яка не зав'язана на холодній війні, на відміну від двох полюсів, і досить централізована для того, щоб поширити свою технологію на всю країну. France Telecom – монополіст телефонного зв'язку – роздавала апарати Мінітель кожному жителю безкоштовно. А за користування послугами через цей апарат стягувалася абонентська плата.



Рис. 1.2. Мінітель – французька інформаційна система, 1980-ті роки

Як функціонував інтернет-магазин у Мінителі? Покупцеві надавався каталог товарів, в якому він міг зробити замовлення. До кожного клієнта був прив'язаний особистий рахунок, яким можна було розплачуватися за здійснене замовлення, іншими словами, брати кредит, який потім погашався при внесенні абонентської плати. Це ще не класична онлайн-оплата, але дуже схоже.



Рис. 1.3. Сторінка сайту в епоху розквіту Мінитель

Далі очікувано Мінитель прийшов у глухий кут, вичерпавши ресурси телефонного зв'язку. Ніякі надбудови не допомогли (серед причин не на останньому місці вплив незграбної держави як власника компанії). Щоправда, варто визнати, що безвихідь продовжувалася досить довго за рахунок французів-пенсіонерів, яким було звично користуватися такими послугами. France Telecom намагалася кілька разів згорнути проект.

У результаті закритий він тільки в 2012 р., втім функція чату для мільйона користувачів, що залишився, досі присутня

## 1.2. Перший інтернет-магазин

Більшість вважає саме amazon.com першим повноцінним інтернет-магазином, який зараз можна скрізь спостерігати (домен 1994 року). Але є також і меншість, яка приписує першість books.com і нібито його творцю Чарльзу Стеку - домен від 1992 року, до речі, найстаріший домен датується 1985-м. Жодних подробиць про самого Стека знайти не вдалося.

Натомість домен належить, і, можливо, належав компанії Barnes and Noble – дуже старій та найбільшій американській компанії з продажу книг.



Рис. 1.4. Компанія з продажу книг - Barnes and Noble

Пара цікавих нюансів у зв'язку із цим:

- Амазон Джеффа Безоса розпочав свою торгівлю з книг. Barnes and Noble, звичайно, теж, в інтернеті навіть;
- Book.com веб-архів фіксує активність у 1996 р., а на amazon.com лише у 1998 р. але навряд чи там, і там продажі почалися так пізно;
- Безос та власники VN перебувають у стані постійної конкуренції;





Рис. 1.5. Голова Amazon Джеф Безос

До 1990 року у Сполучених Штатах діяв регламент, згідно з яким використання Інтернету в комерційних цілях було заборонено. Авторство цього документа належить Національному науковому фонду США. Не виключено, що подібне обмеження було накладено у зв'язку зі специфікою Мережі, яка спочатку створювалася як військова мережа, а пізніше - як міжуніверситетський ресурс.

Після 1990 року ситуація докорінно змінилася: великі приватні компанії отримали дозвіл на використання Мережі. Ще через деякий час уряд країни взагалі передав контроль за Інтернетом у приватні руки. Роком народження інтернет-магазинів вважається 1994: Джефф Безос, господар книгарні під назвою Amazon, задумався над розширенням свого бізнесу. Вже у липні 1995 року було запущено сайт магазину.

Спочатку на Amazon можна було придбати лише книги, але поступово асортимент розширювався. Вже за чотири роки оборотні кошти магазину оцінювалися в кілька сотень тисяч доларів. Сьогодні цей сервіс - один із найбільших інтернет-магазинів у світі з багатомільйонним грошовим обігом.

### 1.3. Інтернет-магазини в 21 столітті

Завдяки швидкому технологічному розвитку та зростанню популярності Інтернету, електронна торгівля виявила неймовірну потужність та вплив на сучасне суспільство. Зараз інтернет-магазин представляє собою віртуальну платформу, де товари та послуги продаються через Інтернет. Він дозволяє покупцям замовляти товари та здійснювати оплату онлайн, не виходячи зі зручності своїх домівок. Інтернет-магазини можуть пропонувати широкий спектр товарів, від одягу та електроніки до продуктів харчування та послуг.

Однією з найбільших переваг інтернет-магазинів є зручність та доступність для споживачів. Вони працюють цілодобово, 7 днів на тиждень, що дає можливість покупцям зробити покупку у будь-який зручний для них час. Крім того, вони забезпечують широкий вибір товарів, конкурентні ціни та зручні методи оплати. Наприклад, покупці можуть використовувати кредитні картки, електронні гаманці, банківські перекази та інші електронні методи оплати.

Існує два типи інтернет-магазинів і відрізняються видом торгівлі. Перший тип магазинів, що продають товар зі свого складу. Нерідко такі магазини існують як додаткова платформа збуту товарів великої торгової мережі, наприклад. Нерідко в інтернет-магазинах вартість товарів значно відрізняється від вартості в реальних магазинах мережі. А другий тип – це магазини, що продають товар інших магазинів або людей. Це може бути торгівля всередині країни або міжнародна торгівля. Таким чином інтернет-магазин заробляє на комісії, яку платять продавці за виставлення товару. Тут інтернет-магазин виступає гарантом угоди між продавцем та покупцем;

Крім цього, можна розділити інтернет-магазини за способом продажу:

- Аукціон системи;
- Фіксована ціна товару.

В останньому випадку, доставка або спочатку включена у вартість, або розраховується окремо, залежно від способу та/або термінів доставки.

За останні роки спостерігається значний ріст популярності інтернет магазинів, а також зростання обсягів електронної торгівлі. Необхідно підкреслити деякі ключові фактори, що сприяють розвитку інтернет-магазинів у 21 столітті:

1. Технологічний прогрес: Швидкий розвиток технологій, особливо інтернету, мобільних пристроїв та бездротового зв'язку, дозволив створити потужні та зручні платформи для онлайн-торгівлі Висота – 50 сантиметрів, ширина – 1 сантиметр.

2. Зручність та швидкість: Інтернет-магазини пропонують покупцям зручність та швидкість замовлення товарів. Замість того, щоб витратити час на пошуки товарів у фізичних магазинах, покупці можуть просто переглянути асортимент і зробити покупку в кілька кліків.

3. Глобальний доступ: Інтернет-магазини дозволяють покупцям отримати доступ до товарів і послуг з будь-якої точки світу. Це дає можливість зробити покупки з-за кордону або замовити ексклюзивні товари, які не є доступними в місцевих магазинах.

4. Розширені можливості маркетингу: Інтернет-магазини використовують різноманітні цифрові маркетингові стратегії, такі як реклама в соціальних медіа, пошукова оптимізація та email-маркетинг, для залучення нових клієнтів і збільшення продажів.

5. Конкурентні ціни та знижки: Завдяки низьким витратам на оренду приміщень та зменшенню проміжних ланок, інтернет-магазини здатні пропонувати конкурентоспроможні ціни. Крім того, вони часто надають знижки та акції, що привертають увагу покупців.

Хоча інтернет-магазини мають багато переваг, вони також стикаються з викликами. Для успішного функціонування вони повинні забезпечувати безпеку платежів, надійну доставку товарів та якість обслуговування клієнтів. Крім того, конкуренція у сфері електронної торгівлі постійно зростає, що вимагає постійного оновлення та удосконалення сервісів.

#### 1.4. Переваги та недоліки інтернет-магазину

Покупки в інтернеті стають все більш популярними, що є причиною виникнення все більшої кількості інтернет-магазинів які мають свої переваги, а також і недоліки. З переваг можна виділити наступне:

– Зручність та доступність: Покупці можуть здійснювати покупки в будь-який зручний для них час, не виходячи з дому або офісу. Вони мають доступ до широкого асортименту товарів та послуг, які можна замовити з будь-якої точки світу;

– Широкий вибір: Інтернет-магазини часто пропонують більший вибір товарів, ніж фізичні магазини. Покупці можуть легко порівнювати ціни та характеристики товарів з різних джерел, що дозволяє їм зробити краще обгрунтований вибір;

– Комфортна покупка: Відсутність черг, тиску продавців або обмежень часу дає покупцям можливість зробити покупку без поспіху та стресу. Вони також можуть швидко знайти необхідну інформацію про товари, відгуки попередніх покупців та поради експертів;



Рис. 1.6. Черги покупців до магазину Apple



– Конкурентні ціни та знижки: у багатьох випадках, інтернет-магазини пропонують конкурентоспроможні ціни, оскільки вони мають менше витрат на оренду приміщень та управління персоналом. Крім того, вони часто надають знижки, акції та промокоди для привернення покупців.

Недоліки інтернет-магазину:

– Відсутність можливості спробувати або перевірити товар: Відсутність можливості фізично побачити або перевірити товар до покупки може бути недоліком. Покупці можуть не мати достатньої інформації про якість або розмір товару, що може призвести до незадоволення після отримання;

– Проблеми з доставкою: Інтернет-магазини можуть стикатися з проблемами доставки, такими як затримки, пошкодження товарів або недоставка. Це може призвести до нестабільної та непостійної якості обслуговування, особливо при замовленні з-за кордону;

– Безпека та приватність: Інтернет-магазини вимагають введення особистої інформації, такої як номер кредитної карти або адреса доставки. Це створює ризик зловживання та крадіжки особистих даних, якщо магазин не забезпечує належний рівень безпеки;

– Відсутність особистого контакту: Для деяких покупців важливо отримати особистий контакт з продавцем, отримати консультацію або отримати підтримку після покупки. В інтернет-магазинах такий прямий особистий контакт може бути відсутнім.

Інтернет-магазини мають багато переваг, але також пов'язані з певними недоліками. Кожен покупець повинен особисто для себе вирішити, які фактори важливі для нього і які ризики він готові прийняти, перш ніж здійснювати покупки в інтернет-магазинах [4].

## ВИСНОВКИ ДО РОЗДІЛУ 1

В першому розділі було детально розглянуто передумови створенню інтернет-магазинів. Описані ключові події розвитку технологій, зазначено назви компаній і прізвища людей, завдяки яким це стало можливо. Особлива увага зверталася на технологічні зміни та ідеї завдяки яким у подальшому сформувалась можливість прогресувати і перетворити інтернет-магазини в сучасні, інноваційні та зручні платформи електронної торгівлі.

Розвиток інтернет-магазинів почався з появи системи SABRE в 1960-х роках, яка дозволяла резервувати авіаквитки та аналізувати продажі. В 1970-х роках з'явилися перші послуги телетексту, які дозволяли користувачам здійснювати покупки через телевізор і підключення до телефону та модему. Французький проект Мінітель, який почався в 1980-х роках, також відіграв важливу роль у розвитку інтернет-магазинів. В рамках цього проекту користувачам надавали каталог товарів та можливість замовлення з оплатою через особистий рахунок. Перший повноцінний інтернет-магазин вважають Amazon.com, який був запущений у 1994 році і спочатку спеціалізувався на продажу книг. Згодом асортимент магазину розширився значно.

Швидкий технологічний прогрес, зокрема розвиток інтернету, мобільних пристроїв та бездротового зв'язку, створив потужні та зручні платформи для онлайн-торгівлі. Інтернет-магазини надають покупцям зручність та швидкість замовлення товарів, глобальний доступ до товарів і послуг, розширені можливості маркетингу, а також конкурентні ціни та знижки. Проте, вони також стикаються з викликами, такими як забезпечення безпеки платежів, надійна доставка товарів та якість обслуговування клієнтів. Конкуренція в цій сфері постійно зростає, що вимагає постійного оновлення та удосконалення сервісів.

## РОЗДІЛ 2

### ОСОБЛИВОСТІ FRONTEND РОЗРОБКИ

#### 2.1. Методи підходу до розробки та основи Frontend

Слово "фронтенд" все частіше можна зустріти не тільки на просторах мережі, а й у розмові у звичайних дружніх тусовках. Давайте докладно розберемося, що означає фронтенд, яке його відмінність від бекенда.

Frontend - це публічна частина web-додатків (вебсайтів), з якою користувач може взаємодіяти та контактувати безпосередньо. У Frontend входить відображення функціональних завдань, інтерфейсу користувача, що виконуються на стороні клієнта, а також обробка користувальницьких запитів. По суті, фронтенд це все те, що бачить користувач при відкритті web-сторінки.

У свою чергу, web-додаток - клієнт-серверний додаток, в якому клієнтом виступає в основному браузер, а сервером - web-сервер. Логіка web-додатка розподілена між сервером та клієнтом, зберігання даних здійснюється переважно на сервері, обмін інформацією відбувається через мережу. Простіше кажучи, це те, що бачить користувач і які дії виконує щоразу, коли підключається до Інтернету і відкриває будь-який браузер.

Frontend-розробка - це робота зі створення публічної частини web-додатку, з якою безпосередньо контактує користувач, та функціоналу, який зазвичай виконується на стороні клієнта. Тобто, фронтенд розробник працює над тим, щоб на сайті кожна кнопка, іконка, текст і вікно не тільки стояли на своєму місці, не перекривали один одного і виглядали цілісно (це веб-верстка), а й щоб вони виконували своє пряме призначення. робили якусь дію.

Компоненти фронтенд розробки:

– HTML (HyperText Markup Language) - це мова розмітки всіх

елементів та документів на сторінці, їх взаємодія у структурі сторінки;

Кафедра КІТ

НАУ 23 33 34 000 ПЗ

				Літ.	Аркуш	Аркушів
Розроб.	ПІБ				19	25
Керівник	Черніцький Е.В.			РОЗДІЛ 2. ОСОБЛИВОСТІ FRONTEND РОЗРОБКИ		
	Толстікова О.В.					
Н.Контр.	Толстікова О.В.					
ТП-415Б - 122						

– CSS (Cascading Style Sheets) – це мова характеристики та стилізація зовнішнього вигляду документа. За допомогою CSS-коду браузер розуміє, як необхідно відображати елементи. CSS створює шрифти, кольори, визначає розташування блоків сайту та інше. Також адаптує той самий документ у різних стилях, виводить передачу на екран або для читання голосом;

– Javascript — мова, створена для оживлення веб-сторінок. Завдання JavaScript - відгукуватися на дії користувача, обробляти натискання клавіш, переміщення курсору, кліки мишкою. JavaScript також дає можливість вводити повідомлення, надсилати запити на сервер, а також завантажувати дані без перезавантаження сторінки, і так далі;

Вся фронтенд розробка виконується на стороні користувача, вона не менш важлива ніж бекенд розробка, тому що це те, що користувач бачить і з чим взаємодіє. Основне завдання фронтенд розробки - це пов'язати представлені дизайнером графічні макети Web-додатки (сторінки сайту) з бекендом і при необхідності реалізувати обчислювальний функціонал на стороні користувача. Основні технології в арсеналі фронтендера це HTML, CSS та JavaScript [1].

Бекенд (англ. Back-end) - програмно-апаратна частина сервісу. Невидима частина сайту. Тобто бекенд - це все те, що відбувається на стороні сервера і що залишається невидимим користувачеві (сам сервер теж є частиною бекенда, тільки апаратною). Звідси і назва front – це видиме спереду, back – це те, що приховано ззаду, невидиме. Бекенд-розробка несе відповідальність за те, що приховано від очей користувача і працює на сервері. Наприклад: backend створює бази даних і програми, які будуть записувати інформацію в базу; шифрує паролі і цінну інформацію; налаштовує доступи і систему резервного копіювання даних; створює програми, що обробляють інформацію, невидиму користувачеві. [2]

Якщо ж клієнт оплачує покупку в інтернеті: вводить дані картки, натискає "оплатити" і бачить напис "ваш платіж прийнятий в обробку" - це

був фронтенд. Те, як рухаються його гроші всередині мережі і те, як його замовлення надходить до магазину – це бекенд. Відповідно, коли магазин бачить повідомлення про те, що надійшло замовлення, а гроші зарахувалися на рахунок — це знову робота фронтенду.

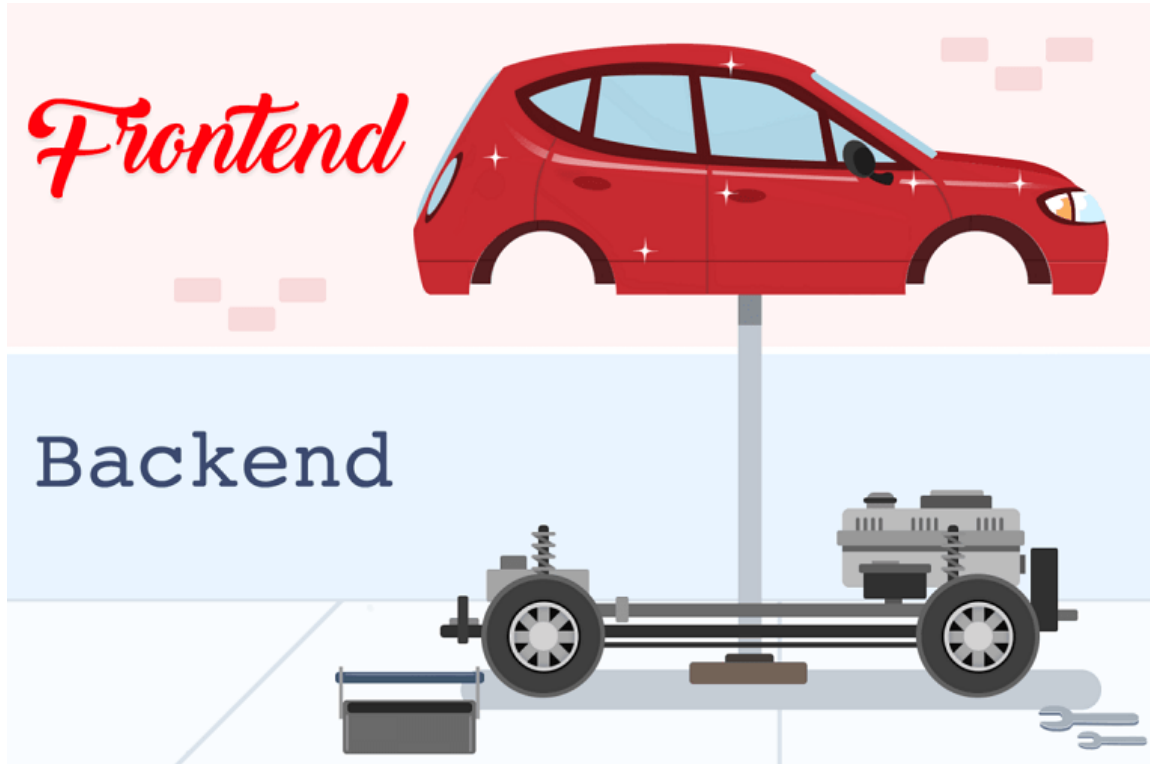


Рис. 2.1. Наглядний приклад різниці між Frontend та Backend розробкою

Мов програмування для бекенд кілька: PHP, Ruby, Python або Node.js. Для бекенд-розробки потрібні системи управління базами даних: MySQL, PostgreSQL, SQLite або MongoDB.

Фронтенд і бекенд - це варіант архітектури програмного забезпечення. Терміни з'явилися в програмній інженерії внаслідок розвитку принципу поділу відповідальності між зовнішнім поданням і внутрішньої реалізацією. Backend для надання своєї функції реалізує API, яке використовує frontend. Таким чином frontend розробнику не потрібно знати особливостей реалізації сервера, а backend розробнику - реалізацію frontend.

Розглянемо процес взаємодії frontend та backend:

- фронтенд відправляє інформацію користувача в бекенд;
- інформація обробляється;

– інформація повертається назад, прийнявши цілісну форму та виконавши опрацьований запит.

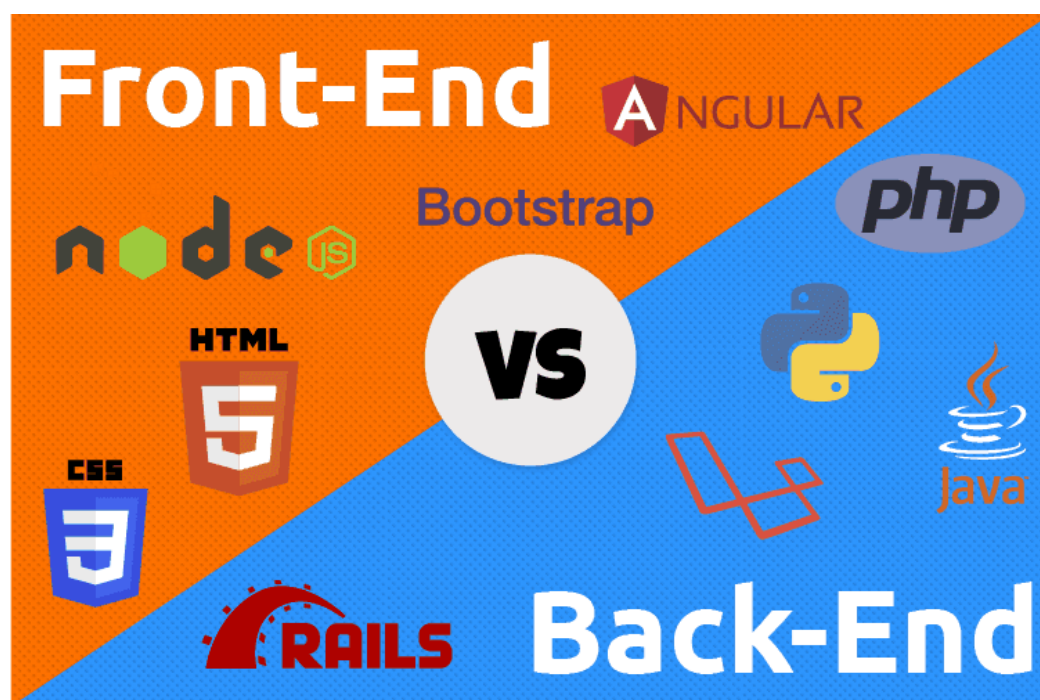


Рис. 2.2. Приклад технологій, які використовуються тому чи іншому напрямках

Варіанти взаємодії frontend та backend:

- HTTP-запит відправляється на сервер, сервер у процесі пошуку інформації, вбудовує її в шаблон і повертає назад у вигляді HTML-сторінки;
- випадок із застосуванням інструментарію AJAX (Asynchronous JavaScript and XML). У цьому випадку запит надсилає JavaScript, який завантажений в браузер, відповідь надходить у форматі XML або JSON [3];
- односторінкові програми, які завантажують дані без поновлення сторінок. Це робиться за допомогою AJAX або фреймворків Angular та Ember;
- Ember або бібліотека React надають допомогу у використанні програми в клієнтській частині та на сервері. Frontend та backend взаємодіють через AJAX та HTML-код, який обробляється на сервері.

Робота та обов'язки frontend та backend девелоперів найчастіше розділені, але іноді виникає необхідність у програміста вирішувати проблеми як на стороні сервера, так і в клієнтській частині.

## **2.2. Вибір програмного забезпечення**

В сфері програмування існує безліч різноманітних програмних продуктів, спеціально створених для виконання різних завдань. Кожен з цих програмних засобів має свою специфіку і застосування. Деякі з них призначені для розробки веб-додатків, інші - для написання мобільних додатків, а є й такі, що допомагають розробляти настільні програми.

Вибір правильного програмного забезпечення для розробки проекту є критично важливим етапом. Від цього вибору залежить якість і швидкість розробки проекту. Для виконання поставлених завдань кваліфікаційної роботи було складено перелік програмних засобів з метою аналізу та порівняння їх можливостей. На основі результатів такого дослідження буде визначено продукт, за допомогою якого планується розробити вебсайт інтернет-магазину “StyleShop”.

Visual Studio Code (VS Code) - безкоштовний редактор коду, який підтримує багато мов програмування, включаючи JavaScript та React. VS Code має багато корисних розширень та інтеграцій з іншими інструментами, такими як Git. VS Code створений компанією Microsoft, який є одним з найбільш популярних інструментів розробки для JavaScript і React.

Особливості VS Code для розробки React-додатків:

- Підтримка React та JSX-синтаксису: VS Code має вбудовану підтримку для React та JSX, що полегшує розробку React-додатків;
- Розширення та плагіни: VS Code має велику кількість розширень та плагінів, які дозволяють налаштувати редактор коду для потреб розробки React-додатків;
- Налагоджувач (Debugger): VS Code має потужний налагоджувач, який дозволяє відстежувати та досліджувати код, що допомагає виявляти та виправляти помилки;

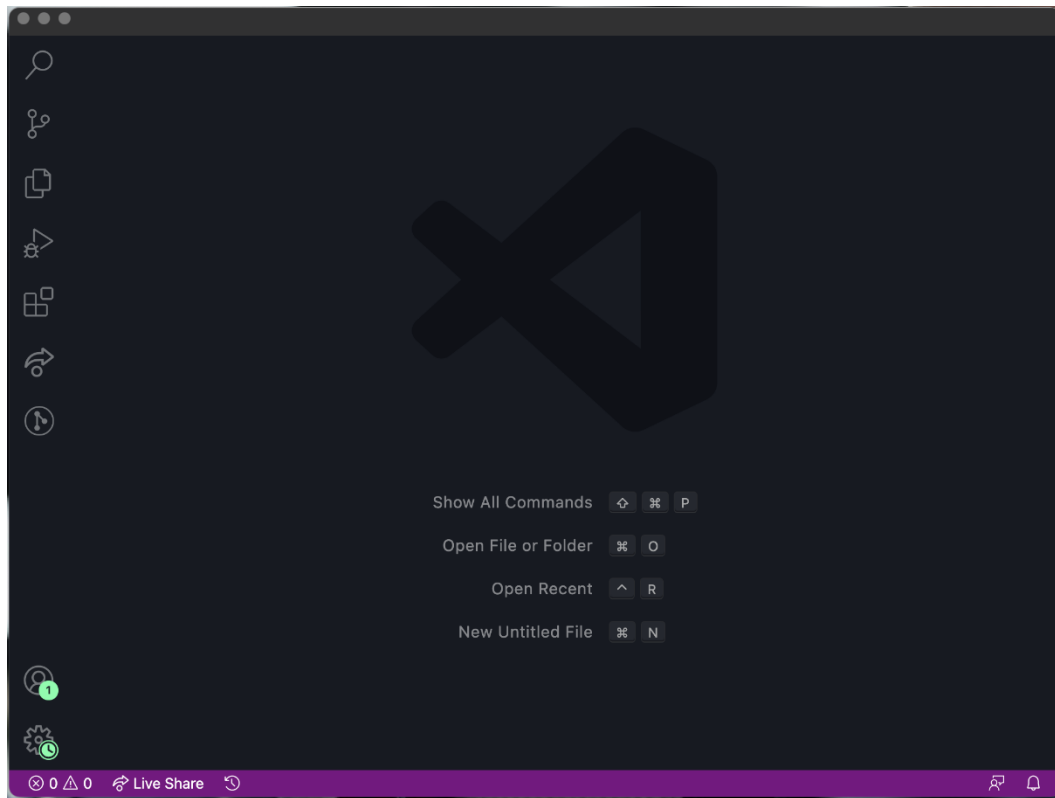


Рис. 2.3. Інтерфейс текстовий редактору Visual Studio Code

– Git-інтеграція: VS Code має вбудовану підтримку Git, що дозволяє працювати з репозиторіями, контролювати версії та ділитись кодом з іншими розробниками;

– Live Server: VS Code має плагін, який називається Live Server, що дозволяє запускати веб-додатки на локальному сервері та автоматично оновлювати сторінку при зміні коду;

– IntelliSense: VS Code має функціональність IntelliSense, яка допомагає розробникам автоматично доповнювати код, зокрема при роботі з React-компонентами.

Редактор згортання коду (Code folding): VS Code дозволяє згорнути рядки коду, що допомагає зберігати оглядовість проекту при роботі з великими файлами. Крім того, VS Code є багатоплатформовим інструментом, що підтримує роботу на Windows, macOS та Linux. Він має зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє просто розгорнути проект та почати роботу над ним.



Для роботи з React, VS Code також має плагіни, які дозволяють легко встановити та настроїти найнеобхідніші інструменти, такі як ESLint, Prettier, React Developer Tools тощо. Ще однією з переваг VS Code є можливість інтеграції з різними інструментами, такими як розширення GitLens, яке дозволяє відслідковувати зміни в коді та авторів цих змін, та розширення Bracket Pair Colorizer 2, яке дозволяє виділяти пари дужок різними кольорами, що полегшує читання коду та розуміння його структури [9].

Загалом, VS Code - це потужний, простий та зручний інструмент розробки для React-додатків, який має вбудовану підтримку для React та JSX, велику кількість розширень та плагінів, та інтеграцію з Git та іншими інструментами.

WebStorm - інтегроване середовище розробки для JavaScript та інших мов програмування. WebStorm підтримує React та має багато корисних інструментів для розробки веб-додатків. WebStorm добре розуміє структуру проектів та допоможе з будь-якими аспектами написання коду. Автодоповнення коду, безпечний рефакторинг, постійний пошук потенційних проблем та підказки щодо їх виправлення завжди будуть під рукою. Один з головних плюсів IDE у тому, що він поєднує усі необхідні інструменти. Крім того WebStorm використовується для налагодження та тестування клієнтського коду та програм на Node.js, і для роботи з системою контролю версій. Зручно користуватися інтегрованими лінтерами, інструментами збирання, терміналом та HTTP-клієнтом [10].

Webstorm дозволяє швидко рухатися за кодом незалежно від розмірів проекту. Шукати файли, класи або символи, щоб переглянути всі результати в одному місці. Переходити до визначення функцій, методів, змінних, компонентів або класів, знаходити їх використання буквально в пару кліків. Є можливість налаштувати інтерфейс WebStorm на свій смак, за допомогою різних тем і плагінів. Можна зберегти налаштування та використовувати їх в інших інсталяціях WebStorm.

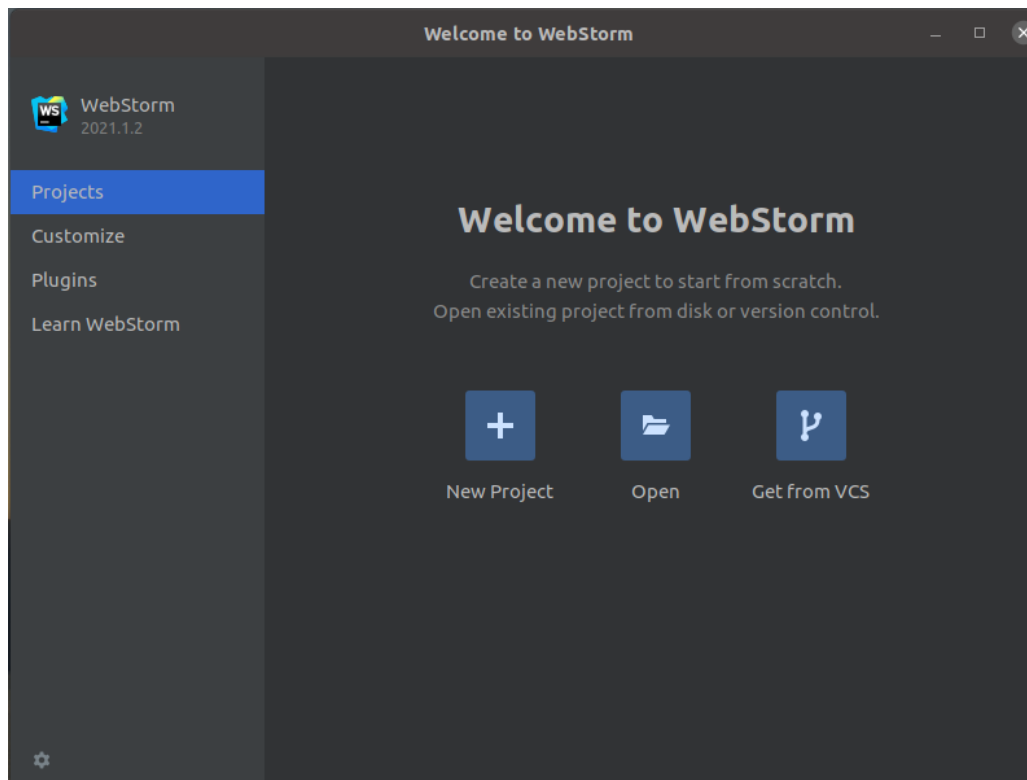


Рис. 2.4. Інтерфейс інтегрованого середовища розробки WebStorm

IDE пропонує безліч спеціальних можливостей для різних потреб. Вона дозволяє налаштовувати кольори елементів інтерфейсу, змінювати розмір вікон і шрифтів у редакторі, коригувати колірну схему у разі порушеного сприйняття кольору, налаштовувати сполучення клавіш і багато іншого. Також підтримуються інструменти читання з екрана. При потребі швидко відкрити файл інтегрований LightEdit, з ним більше не доведеться створювати або завантажувати цілий проект. Файл відкриється у полегшеному текстовому редакторі, який підтримує підсвічування синтаксису, автоматичне збереження та інші базові можливості. Після закінчення редагування можна буде переключитися назад до проекту всього в пару кліків. Перегляд файлів проекту, переходи до файлів і фрагментів коду, які недавно відкривались або редагувались - WebStorm запам'ятовує, з чим велася робота, і дозволяє швидко повернутися назад. Для навігації використовуються вкладки або клавіші. У WebStorm є прев'ю для статичних HTML-файлів. Коли редагується HTML-код або пов'язані CSS і JavaScript-файли, зміни зберігаються і оновлюються автоматично. Якщо треба

переглянути документацію для символу, то для цього не потрібно виходити з IDE. З WebStorm можна навести або поставити курсор миші на потрібний символ, натиснути Ctrl+Q і розробник зможе переглянути відповідну інформацію (рис. 2.5). Також WebStorm підказує параметри для методів і функцій, що викликаються.

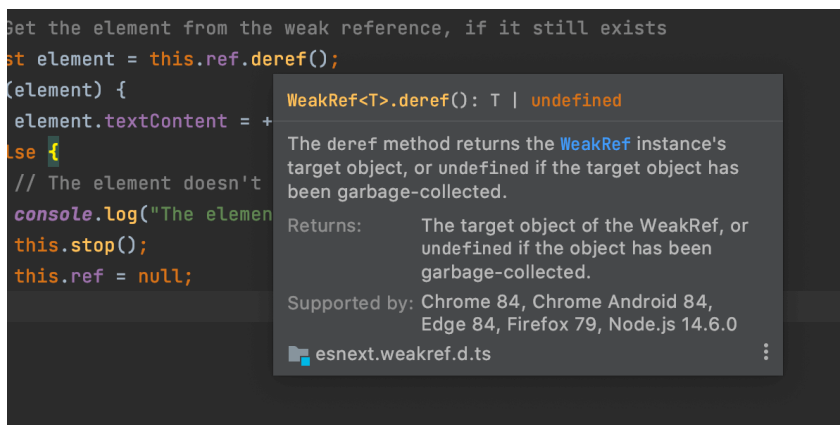


Рис. 2.5. Швидкий перегляд документації у WebStorm

Atom - безкоштовний редактор коду (рис. 2.6), який має багато розширень для розробки React-додатків та інших веб-додатків. Atom підтримує багато мов програмування та має інтеграцію з Git. Програма має всі функції, які можна очікувати від редактора коду, включаючи підсвічування синтаксису, автовизначення мов, автозаповнення тексту, можливість використовувати кілька панелей та зберігати проект у кількох папках, підтримка фрагментів коду та потужний інструмент пошуку. А головна властивість – це модульність середовища розробки, до якого можна додавати додаткові функції, встановлюючи додаткові пакети. Ще GitHub включає систему керування для Git, так що можна публікувати будь-який контент з використанням платформи GitHub. Atom – це мультиплатформний (Windows, Linux та Mac) інструмент, що дозволяє робити проекти крос-платформними. Налаштування IDE можна назвати основною перевагою програми. На момент написання було доступно понад 2 000 пакетів та 600 тем. Враховуючи численні налаштування програми та кількість вже включених до неї функцій, її безперечно можна назвати одним із найкращих

доступних на сьогодні інструментів веб-розробки. На додаток до всього, програма виділяється перед іншими подібними альтернативами тим, що не займає багато місця на комп'ютері [11].

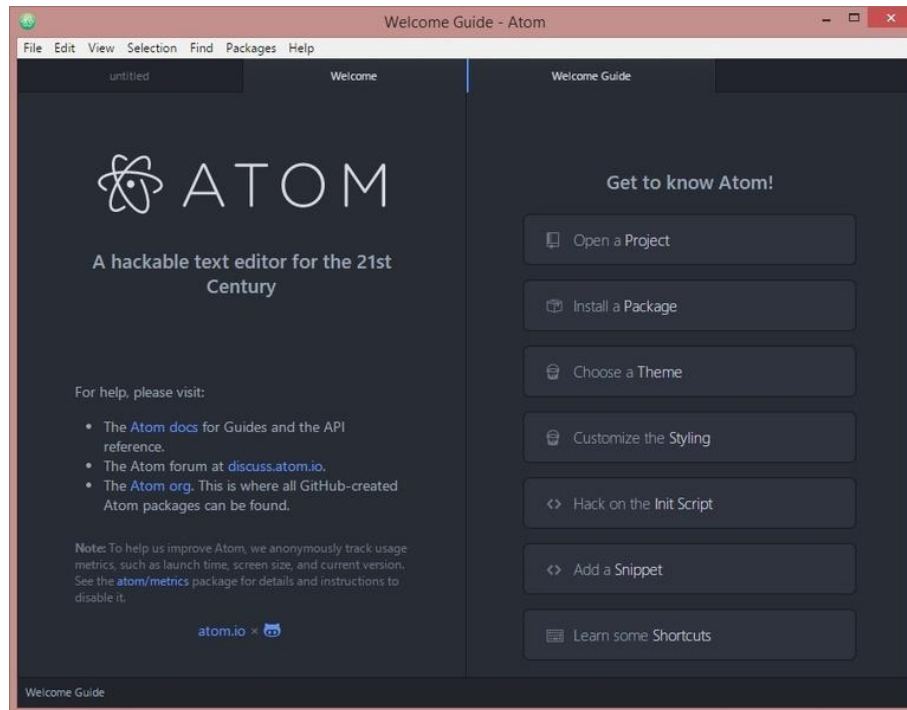


Рис. 2.6. Текстовий редактор Atom

З переваг можна виділити наступне:

– поставляється з менеджером пакетів інтегрований, за допомогою якого ви можете легко встановлювати або змінювати плагіни (або навіть створювати свої власні);

– вільне програмне забезпечення. Більшість функціональних можливостей Atom виражена у вигляді пакетів, до яких є змога безкоштовно отримати доступ до його бета-версії. Але тепер, з випуском першої стабільної версії, GitHub вирішив випустити решту Atom: ядро програми, що управляє справами пакети, еі оболонка Atom і його рамки заснований на Chromium, безкоштовному браузері Google;

– Atom допомагає швидко і гнучко писати код завдяки своєму розумному автозаповнення;

– має файловий браузер, за допомогою якого відкривається один файл, весь проект або кілька проектів в одному вікні;

- пошук і заміна тексту (під час написання) у файлі або у всіх інших проектах.

- налаштування зовнішнього вигляду Atom (шрифт, кольори вікон і т. д.) через кілька доступних тем.

З недоліків можна виділити те, що на початку 2022 року розробники закрили проект. Надалі текстовий редактор не отримуватиме оновлень і на цьому його шлях припиняється, але найближчий час він все ще буде актуальним.

Sublime Text - кросплатформовий текстовий редактор (рис. 2.7) для написання програмного коду різними мовами програмування (Groovy, Erlang, C+, Java і т.д.), а також верстки веб-документів. Але розповсюджується за платною ліцензією ПЗ, і має безкоштовну версію лише для ознайомлення. За рахунок широкої функціональності, яку можна розширювати шляхом підключення плагінів, а також кросплатформенності Sublime Text популярний серед програмістів та веб-розробників.

Ключові можливості:

- інтерфейс, що налаштовується;
- множинне виділення;
- автоматичне наповнення;
- підсвічування синтаксису та висока контрастність;
- сніпети;
- підключення плагінів;
- роздільне редагування;
- швидкодія.

Але треба виділити, чого справді бракує текстовому редактору:

- Графічний інтерфейс для створення колірних схем. Редагувати XML-файл, щоб оформити під себе - це вкрай незручно, забирає купу часу;

- Вставлення зображення через провідник Windows. Тобто. Коли вказується зображення, і програма автоматично підставляє в тег <img> шлях до нього, а також ширину та висоту [12];

- Додавання слів до словника для перевірки орфографії;
- Клікабельність посилань.

З недоліків можна виділити наступне:

- час завантаження програми. Якщо порівнювати Sublime Text, Notepad++ та Rapid PHP, то Notepad++ відкривається просто миттєво, Sublime Text приблизно за 1 секунду, а Rapid PHP секунди за 3;
- Платність. Коштує 59 доларів. З одного боку, не мало для текстового редактора, з іншого — вона того варта. Хоча можна скористатися і вічним тріалом (періодично при збереженні файлів вискакує пропозиція придбати).



Рис. 2.7. Текстовий редактор Sublime Text

Visual Studio - це інтегроване середовище розробки (IDE) від Microsoft (рис. 2.8), яке підтримує розробку в різних мовах програмування, включаючи JavaScript. Visual Studio надає різноманітні інструменти для розробки React-додатків, такі як налагоджувач (debugger), засоби створення та редагування компонентів, підтримку JSX-синтаксису та багато іншого.

Функціональна структура середовища включає:

- редактор вихідного коду, який включає безліч додаткових функцій, як автодоповнення IntelliSense, рефакторинг коду тощо;
- налагоджувач коду;

- редактор форм, призначений для спрощеного конструювання графічних інтерфейсів;
- веб-редактор;
- дизайнер класів;
- дизайнерсхем баз даних.

Visual Studio також дозволяє створювати та підключати сторонні доповнення (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (Subversion та VisualSourceSafe), додавання нових наборів інструментів (для редагування та візуального проектування коду на предметно-орієнтованих мовах) програмування або інструментів інших аспектів процесу розробки програмного забезпечення).

Комерційні версії у порядку зростання ціни: Visual Studio Professional, Visual Studio Premium та Visual Studio Ultimate.

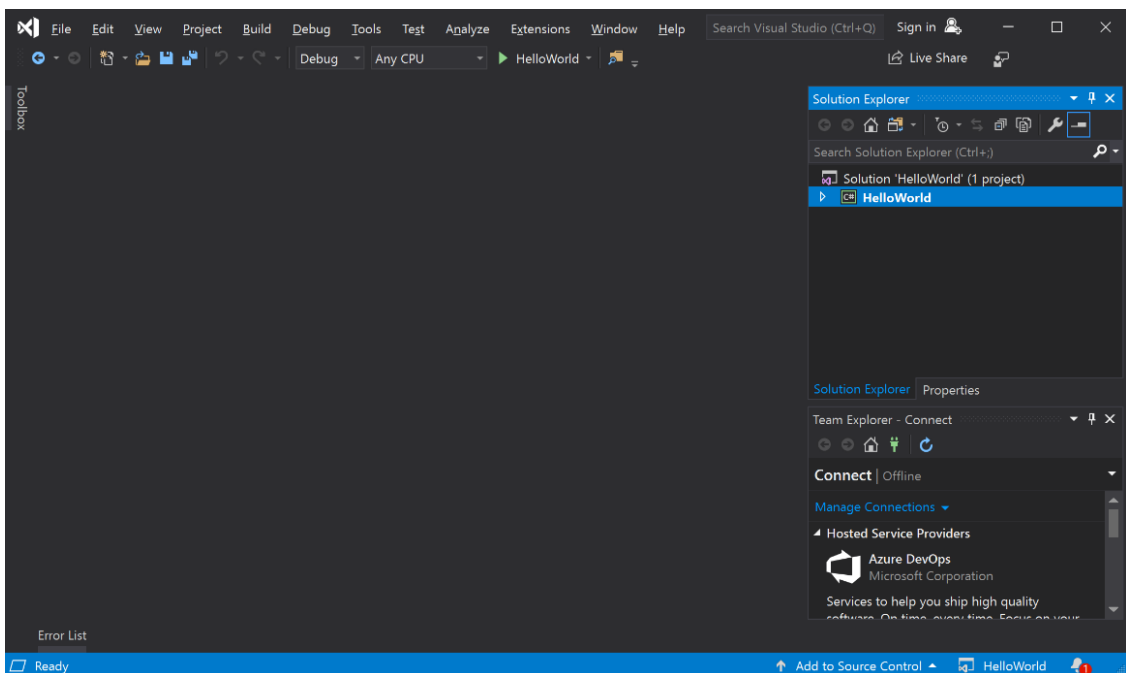


Рис. 2.8. Інтерфейс IDE Visual Studio

Інтегроване середовище розробки (IntegratedDevelopmentEnvironment - IDE) Visual Studio пропонує ряд високорівневих функціональних можливостей, що виходять за рамки базового керування кодом. Нижче наведено основні переваги IDE-середовища Visual Studio:

– Вбудований Web-сервер. Для обслуговування Web-дodatка ASP.NET необхідний Web-сервер, який чекатиме на Web-запити і обробляти відповідні сторінки. Наявність у Visual Studio інтегрованого Web-сервера дозволяє запускати Web-сайт прямо з середовища проектування, а також підвищує безпеку, за винятком ймовірності отримання доступу до тестового Web-сайту з якогось зовнішнього комп'ютера, оскільки тестовий сервер може приймати з'єднання тільки з локального комп'ютера;

– Підтримка багатьох мов під час розробки. Visual Studio дозволяє писати код своєю мовою або будь-якою іншою мовою, використовуючи весь час той самий інтерфейс (IDE). Більш того, Visual Studio також ще дозволяє створювати Web-сторінки різними мовами, але поміщати їх все в один і той же Web-дodatок. Єдиним обмеженням є те, що в кожній Web-сторінці можна використовувати лише якусь одну мову (очевидно, що інакше проблем при компіляції було б просто не уникнути);

– Менше за код для написання. Для створення більшості програм потрібна пристойна кількість стандартного стереотипного коду та Web-сторінки ASP. NET тому не виняток. Наприклад, додавання Web-елемента керування, приєднання обробників подій та коригування форматування потребує встановлення в розмітці сторінки ряду деталей. У Visual Studio такі деталі встановлюються автоматично;

– Інтуїтивний стиль кодування. За промовчанням Visual Studio форматує код у міру його введення, автоматично вставляючи необхідні відступи та застосовуючи колірне кодування для виділення елементів типу коментарів. Такі незначні відмінності роблять код більш зручним для читання і менш схильні до помилок. Застосовувані Visual Studio параметри форматування можна навіть налаштовувати, що дуже зручно у випадках, коли розробник віддає перевагу іншому стилю розміщення дужок (наприклад, стиль K&R, при якому дужка, що відкриває, розміщується у того ж рядка, що і оголошення, якому вона передує);



– Вища швидкість розробки. Багато функціональних можливостей Visual Studio спрямовані на те, щоб допомагати розробнику виконувати свою роботу якнайшвидше. Зручні функції, на зразок функції IntelliSense (яка вміє перехоплювати помилки та пропонувати правильні варіанти), функції пошуку та заміни (яка дозволяє відшукувати ключові слова як в одному файлі, так і у всьому проекті) та функції автоматичного додавання та видалення коментарів (яка може тимчасово приховувати блоки коду), дозволяють розробнику працювати швидко та ефективно;

– Можливості налагодження. Пропоновані Visual Studio інструменти налагодження є найкращим засобом для відстеження загадкових помилок і діагностування дивної поведінки. Розробник може виконувати свій код за рядком за раз, встановлювати інтелектуальні точки переривання, за бажанням зберігаючи їх для використання в майбутньому, і у будь-який час переглядати поточну інформацію з пам'яті.

Visual Studio також має безліч інших функцій: можливість управління проектом; вбудована функція керування вихідним кодом; можливість рефакторизації коду; потужна модель розширюваності. Більше того, у разі використання Visual Studio 2008 Team System розробник отримує розширені можливості для модульного тестування, спільної роботи та керування версіями коду (що значно більше за те, що пропонується в більш простих інструментах на зразок Visual SourceSafe) [13].

Як недолік можна відзначити неможливість відладчика (Microsoft Visual Studio Debugger) відстежувати у коді режиму ядра. Налагодження в Windows у режимі ядра в загальному випадку виконується під час використання WinDbg, KD або SoftICE.

Для подальшого дослідження в кваліфікаційній роботі потрібно створити макет майбутнього сайту та зверстати його за допомогою мови гіпертекстової розмітки (HTML) та стилізувати за допомогою каскадної таблиці стилів (CSS) з використанням програми Visual Studio Code – легкого

і безкоштовного текстового редактора (рис. 2.9) вихідного коду та програми Figma (рис. 2.10) - багатofункціонального графічного редактора.

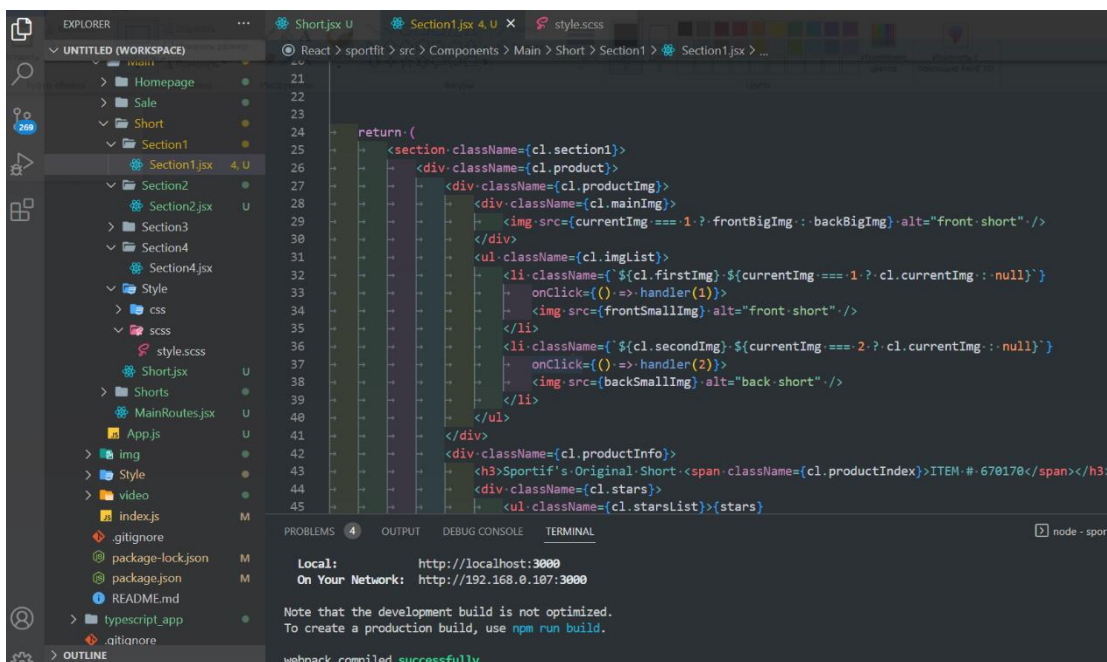


Рис. 2.9. Visual Studio Code з початковою структурою HTML

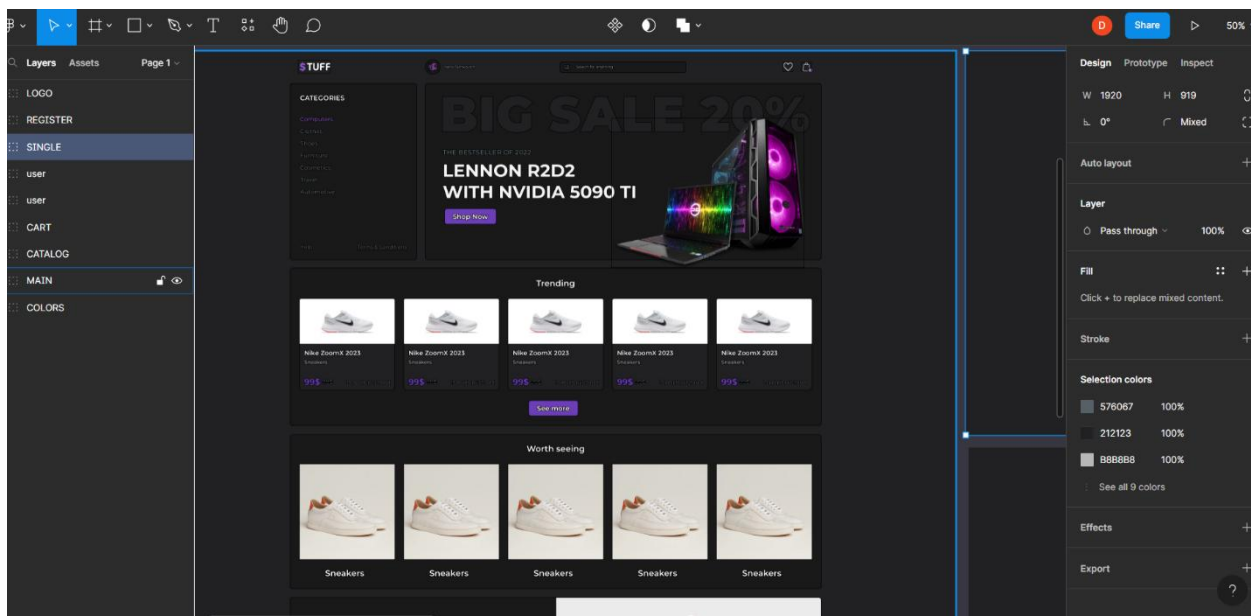


Рис. 2.10. Програма Figma

### 2.3. Мова розмітки, стилі та програмування на React.js

HTML - це мова гіпертекстової розмітки. Відповідає за структуру і зміст сторінки. Вона складається з тегів, а тег складається з імені, укладеного між знаками «менше» і «більше». Наприклад: `<h1>`, `<p>` і `<ul>`.

Використовується розширення \*.html (рис. 2.11). Воно дає зрозуміти браузеру, що всередині файлу знаходиться код веб-сторінки. Браузер розбирає його структуру, визначає взаємне розташування елементів і візуалізує їх.

```
<html>
  <head>
    <meta charset="utf-8">
    <title> HTML Документ </title>
  </head>
  <body>
    <p> Даний текст залишиться без змін,
      <b> а цей текст буде напівжирним,
      <i> а цей - ще й курсивним </i>.
    </b>
  </p>
</body>
</html>
```

Рис. 2.11. Базова структура пустої вебсторінки з розширенням .html

Веб-стандарти HTML описані в специфікаціях. Це головне джерело знань і для браузерів, і для розробників. Наразі використовується версія HTML 5.3 [6].

CSS - це каскадні таблиці стилів, які використовуються для стилізації розмітки. Наприклад: зробити текст червоним, вирівняти по ліву сторону, округлити кути зображення, пограти зі шрифтами, зробити відступи більше або менше [7].

Тег <div> відповідає за формування блоку. Припустимо, нам потрібно щоб всі <div> на сторінці отримали однакові властивості. Для цього треба підключити до сторінки стилі. Поки вони не підключені ніякого ефекту від них не буде. Стилї можна вбудувати всередині HTML-коду сторінки (інлайново) або підключити їх як зовнішній файл. Найчастіше стилі підключають із зовнішнього файлу з розширенням \*.css. Так розмітка відділяється від стилізації, знижується ймовірність помилки, зовнішній файл

кеширується і не завантажується повторно. Для підключення з зовнішнього файлу використовується тег `<link>`.

1. **Inline стилі** – додавання в html коду стилів через атрибут `STYLE`.
2. Збереження CSS коду в html документі у тезі `<STYLE>`, що теж розміщується в `<HEAD>`.
3. Внесення CSS коду в окремий документ з розширенням `.css` та підключенням в `<HEAD>` за допомогою тега `<LINK>`.
4. Підключення CSS коду в html за допомогою конструкції `@import`.

**Найпоширеніший 3-ій спосіб підключення:**  
(підключення стилів в розділі `<HEAD>` )

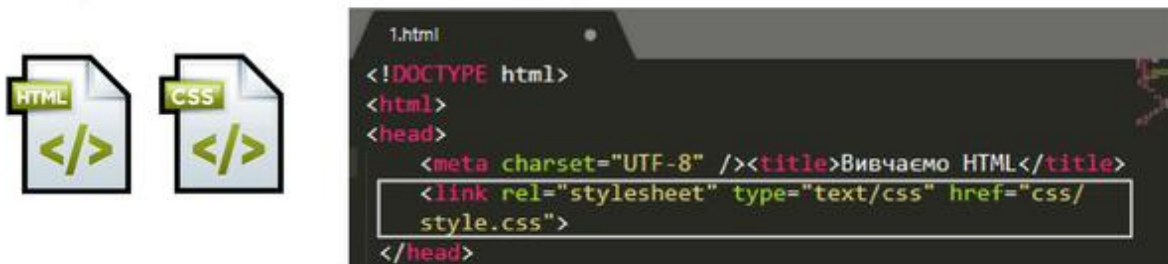


Рис. 2.12. Варіанти підключення CSS стилів до HTML коду

Та сторінка на чистому HTML виглядають «мертвими», інакше качучи статичними. Щоб оживити HTML-сторінки застосовується мова програмування Javascript, яка додає динамічність та інтерактивність сторінці, обробку дій користувача або цікаві візуальні ефекти.

JavaScript - Браузерна мова програмування для фронтенд-розробки. Її підтримують всі популярні браузері. Кожен сайт, кожне веб-додаток, яким рядовий користувач користується містить JavaScript-код. Його часто рекомендують для вивчення новачкам, тому що він досить простий, але містить всі фундаментальні речі: об'єктно-орієнтовану модель і структури даних. З його допомогою можна навіть писати картини на канвасі. Можливості застосування JavaScript обмежуються тільки фантазією розробника і підтримкою браузерів.

Ключові аспекти, пов'язані з JavaScript:

– функціональність на клієнтському боці: JavaScript дозволяє оживити HTML-сторінки та забезпечити взаємодію з користувачем. Ви можете створювати динамічні елементи, обробляти події, маніпулювати DOM-структурою, виконувати анімацію та багато іншого;

– кросплатформеність: JavaScript підтримується всіма сучасними браузерами і працює на різних операційних системах, таких як Windows, macOS, Linux тощо. Це дозволяє розробникам створювати веб-додатки, які працюють на різних пристроях і платформах;

– простота вивчення: JavaScript є відносно простою мовою програмування, що робить її доступною для новачків. Вона має простий синтаксис, схожий на мови, такі як C і Java, тому розробники, які вже знайомі з цими мовами, можуть швидко адаптуватися до JavaScript;

– широкий екосистема і підтримка: JavaScript має велику спільноту розробників та багато розширень, бібліотек і фреймворків, які полегшують розробку веб-додатків. Наприклад, React.js, Angular, Vue.js та інші фреймворки дозволяють створювати складні інтерфейси з використанням компонентного підходу.

Незважаючи на переваги, JavaScript також має деякі недоліки:

– безпека: JavaScript виконується на стороні клієнта, що робить його вразливим до атак, таких як внедрення коду (injection) і перехоплення даних (data interception). Однак, існують методи та практики, які можуть зменшити ризики безпеки;

– швидкодія: JavaScript виконується в середовищі браузера, що може вплинути на його продуктивність, особливо при обробці великої кількості даних. Оптимізація коду та використання ефективних алгоритмів можуть допомогти вирішити цю проблему;

– залежність від браузера: JavaScript може вести себе по-різному в різних браузерах, оскільки має питання сумісності та реалізації стандартів. Проте, багато бібліотек і фреймворків допомагають забезпечити сумісність та розробку, що працює на багатьох браузерах.

Недивлячись на ці недоліки, JavaScript є незамінною мовою для фронтенд-розробки та динамізації веб-сторінок, яка постійно розвивається та має широкі можливості. Використання JavaScript також поширене у розробці серверної частини веб-додатків за допомогою платформи Node.js. Node.js

дозволяє виконувати JavaScript-код на сервері, що розширює можливості мови та надає зручний інструментарій для роботи з мережевими операціями, базами даних та іншими складовими серверних додатків. Однак, із зростанням складності веб-додатків та підвищенням вимог до швидкодії та ефективності, виникає потреба у використанні фреймворків, наприклад React.js.

React.js, розроблений Facebook, є одним з найпопулярніших фреймворків для розробки інтерфейсів користувача. Він базується на JavaScript і використовує концепцію компонентів, що дозволяє створювати перевикористовувані та масштабовані інтерфейсні елементи.

Перехід від JavaScript до React.js може принести багато переваг. Фреймворк надає зручний спосіб організації коду, зокрема шляхом розділення його на окремі компоненти, що полегшує розробку та підтримку додатків. Крім того, React.js має ефективний механізм оновлення інтерфейсу лише при зміні потрібних даних, що дозволяє покращити продуктивність та реактивність додатку.

Компоненти — основні будівельні блоки React. Якщо є досвід праці з Angular, то можна помітити подібність компонентів до директив. Іншими словами можна вважати, що компоненти схожі на віджети або ж модулі. Але якщо більш точно висловлюватися про компонент то це набір з HTML, CSS та JS, який має деякі внутрішні особливості, які властиві тільки компонентам. Іноді порівнюють компоненти React з пиріжками з начинкою, тільки зі світу веб-технологій. Вони можуть містити все, чого розробник тільки забажає, й бути огорнуті в оболонку, що легко компонується. Компоненти реалізуються на чистому JavaScript, або ж використовують JSX [5].

Що робить React таким зручним для побудови інтерфейсів? Перш за все через те, що дані скоріше всього отримуються від батьківського компонента, або вони вже знаходяться у дочірньому компоненті.

Якщо спробувати відтворити Twitter сторінку (рис. 2.13) засобами React, то необхідно розбити її на різноманітні компоненти (обведені в рамку).



Треба зауважити, що вони можуть містити вкладені дочірні компоненти. Але як вибрати, що має бути компонентом, а що ні?



Рис. 2.13. Twitter сторінка поділена на компоненти

Перше, що треба зробити – це уявити кордони кожного компонента (і підкомпонента) в макеті та дати їм імена. Якби робота була з дизайнерами, цілком можливо, що вони вже якось називають компоненти. Наприклад, шари Figma часто підказує імена для React-компонентів. Це схоже на те, як ви вирішуєте, чи треба створити нову функцію чи об'єкт [14]. Можна застосувати принцип єдиного обов'язку: кожний компонент в ідеалі повинен займатися якимось одним завданням. Якщо функціонал компонента збільшується з плином часу, його слід розбити на дрібніші підкомпоненти.

Припустимо, компонент зліва (у рожевій рамці) «UserInfo». Усередині компонента «UserInfo», розташовано інший компонент у (помаранчевій рамці), його можна назвати «UserImages». Спосіб, у який батьківський

компонент працює з дочірнім компонентом «UserInfo», або навпаки — робота зі станом «state», як власним, так і дочірнього компонента «UserImages». Якщо є потреба використовувати будь-яку частину даних з батьківського компонента у дочірньому, то ці дані передаються до дочірнього компонента у якості атрибуту. У цьому прикладі передається компонента «UserImages» і всі зображення, які знаходяться у компоненті «UserInfo». Ієрархія батьківських та дочірніх елементів спрощує керування даними, тому розробник точно знає де дані знаходяться, а тому нема потреби маніпулювати ними в іншому місці.

У React, щоб створити компонент, необхідно спочатку створити клас або функцію, яка буде представляти цей компонент.

```
function MyComponent() {  
  return <h1>Hello, world!</h1>;  
}
```

Рис. 2.14. Приклад коду створення функціональної компоненти

У прикладі (рис. 2.14) створено функцію MyComponent, яка повертає JSX-елемент «<h1>Hello, world!</h1>». JSX - це розширення JavaScript, яке дозволяє використовувати HTML-подібний синтаксис для створення компонентів. Важливо пам'ятати, що JSX не є чистим HTML або XML, це всього лише синтаксичний цукерка над JavaScript.

JSX-елементи не можуть бути відразу відображені в браузері. Щоб перетворити JSX-елементи в HTML-код, React використовує Virtual DOM - віртуальне представлення дерева DOM, яке дозволяє ефективно оновлювати сторінку без повного перерендеринга всього дерева DOM.

```
ReactDOM.render(  
  <MyComponent />,  
  document.getElementById('root')  
);
```

Рис. 2.15. Приклад коду використання методу ReactDOM.render() для відображення компонента



Для відображення компонента на сторінці використовується метод «ReactDOM.render()» (рис. 2.15). Цей метод приймає два аргументи: елемент, який ми хочемо відобразити, та DOM-елемент, в якому необхідно відобразити цей елемент. У прикладі передається компонент «<MyComponent />» як перший аргумент методу «ReactDOM.render()», а також вказується «document.getElementById('root')» як другий аргумент, щоб вказати, де на сторінці необхідно відобразити компонент.

Тепер, коли запускається цей код у браузері, компонент «MyComponent» буде відображено на сторінці. Кожен раз, коли буде змінюватися стан компонента, React буде перерендерювати його віртуальне представлення та оновлювати тільки необхідні частини DOM, що дозволяє реалізувати ефективно оновлення сторінки та зменшує навантаження на браузер. Завдяки використанню віртуального DOM, React дозволяє ефективно взаємодіяти зі сторінкою, не перерендерюючи всю сторінку кожного разу, коли змінюється стан компонента. Це дозволяє створювати швидкі та ефективні веб-додатки, які працюють з максимальною продуктивністю та без перезавантаження сторінки.

Головними принципами роботи з ReactJS та компонентами залишаються:

- Компонентний підхід: у ReactJS всі елементи веб-сторінки представлені компонентами. Компоненти - це віддільні блоки, які містять у собі HTML-код, JavaScript-логіку та CSS-стили. Це дозволяє розбити сторінку на окремі компоненти та повторно використовувати їх у різних місцях веб-додатку;

- Віртуальний DOM: у ReactJS використовується віртуальний DOM, який є копією реального DOM та зберігається у пам'яті. Кожен раз, коли змінюється стан компонента, ReactJS перерендерює тільки необхідні частини віртуального DOM та оновлює тільки необхідні елементи на сторінці. Це

дозволяє зменшити навантаження на браузер та зробити веб-додаток більш продуктивним;

– Єдиний напрямок даних: у ReactJS єдиний напрямок даних - від батьківського компонента до дочірнього. Це означає, що стан компонента може змінюватись тільки за допомогою взаємодії з батьківським компонентом або за допомогою виклику зовнішніх функцій. Це дозволяє зробити веб-додаток більш передбачуваним та легше відслідковувати зміни стану.

Життєвий цикл компонента: кожен компонент у ReactJS має життєвий цикл, який складається з певних етапів: відображення компонента, оновлення стану, видалення компонента.

React - це потужна бібліотека для створення веб-інтерфейсів, яка базується на компонентах, JSX, Virtual DOM, render та ReactDOM.render. Ось деякі з переваг та недоліків використання цих інструментів у React:

З переваг React можна зазначити, що JSX дозволяє змішувати HTML-подібний код з JavaScript, що полегшує розробку та підтримку компонентів. Virtual DOM дозволяє ефективно оновлювати сторінку без повного перерендеринга всього дерева DOM, що зменшує витрати на обробку та збереження даних. Застосування методу «ReactDOM.render()» дозволяє швидко та просто відображати компоненти на сторінці. Компоненти дозволяють розділити веб-інтерфейс на декілька частин, які можна з легкістю підтримувати та перевикористовувати. React підтримує розробку як на класах, так і на функціях, що дозволяє розробникам вибирати той підхід, який більше підходить для їх потреб.

З недоліків платформи виділити те, що використання JSX може призвести до ситуації, коли розробник може змішати логіку та відображення, що робить код складнішим для розуміння та підтримки. Virtual DOM може займати більше пам'яті, ніж звичайний DOM, що може призвести до збільшення витрат на пам'ять та обробку. Застосування методу «ReactDOM.render()» може призвести до збільшення часу завантаження

сторінки, якщо на сторінці знаходиться багато компонентів, які необхідно відобразити. Компоненти можуть бути менш ефективними в роботі з даними, які змінюються часто, так як React повинен повністю перерендерювати компонент.

## ВИСНОВКИ ДО РОЗДІЛУ 2

Веб-програмування є невід'ємною частиною сучасного технологічного світу. Цей процес передбачає створення вебсайтів за допомогою використання різноманітних стеків технологій, інструментів та застосунків. В даному розділі було розглянуто два методи підходу розробки, основи Frontend, вибір програмного забезпечення, мова розмітки, стилі та розробка на React.js.

Розглянувши різні інтегровані середовища розробки та текстові редактори, для створення вебсайту інтернет-магазину “StyleShop”, було обрано Visual Studio Code. Оскільки VS Code є відмінним вибором завдяки своїм потужним інструментам та розширенням, спеціально розробленим для роботи з React-проектами. Він має інтеграцію з React DevTools, що дозволяє зручно налагоджувати та аналізувати компоненти React. Крім того, VS Code підтримує JSX-синтаксис, автодоповнення та перевірку помилок, що полегшує написання та редагування коду React.

Для ефективнішої розробки у створенні інтернет-магазину застосовується фреймворк React.js, він є одним з найпопулярніших фреймворків для фронтенд розробки завдяки своїй ефективності, повторному використанню компонентів та зручному віртуальному DOM, що забезпечує швидку та плавну роботу веб-додатків. Його гнучка архітектура дозволяє зручно організувати код, забезпечуючи його легку підтримку та розширення.

Фреймворк React.js у поєднанні з редактором VS Code створюють ідеальну комбінацію для розробки веб-додатків. Використовуючи React.js, можуть ефективно створювати повторно використовувані компоненти та

організувати логіку додатка, а VS Code надає потужні інструменти, такі як автодоповнення, перевірку помилок та налагоджувач, що спрощують процес розробки. Ця комбінація забезпечує зручність, продуктивність та швидкість в розробці веб-додатків з використанням React.js.

## РОЗДІЛ 3

# СТВОРЕННЯ ВЕБСАЙТУ ІНТЕРНЕТ-МАГАЗИНУ “STYLESHOP”

### 3.1. Верстка макету

Розробка інтернет магазину “StyleShop” складається з 3 етапів і починається з верстки макету, який вже спроектований веб-дизайнером у застосунку Figma. Верстка макету є не менш важливою ніж наступні етапи розробки, оскільки вона відіграє ключову роль у створенні привабливого інтернет-магазину "StyleShop".

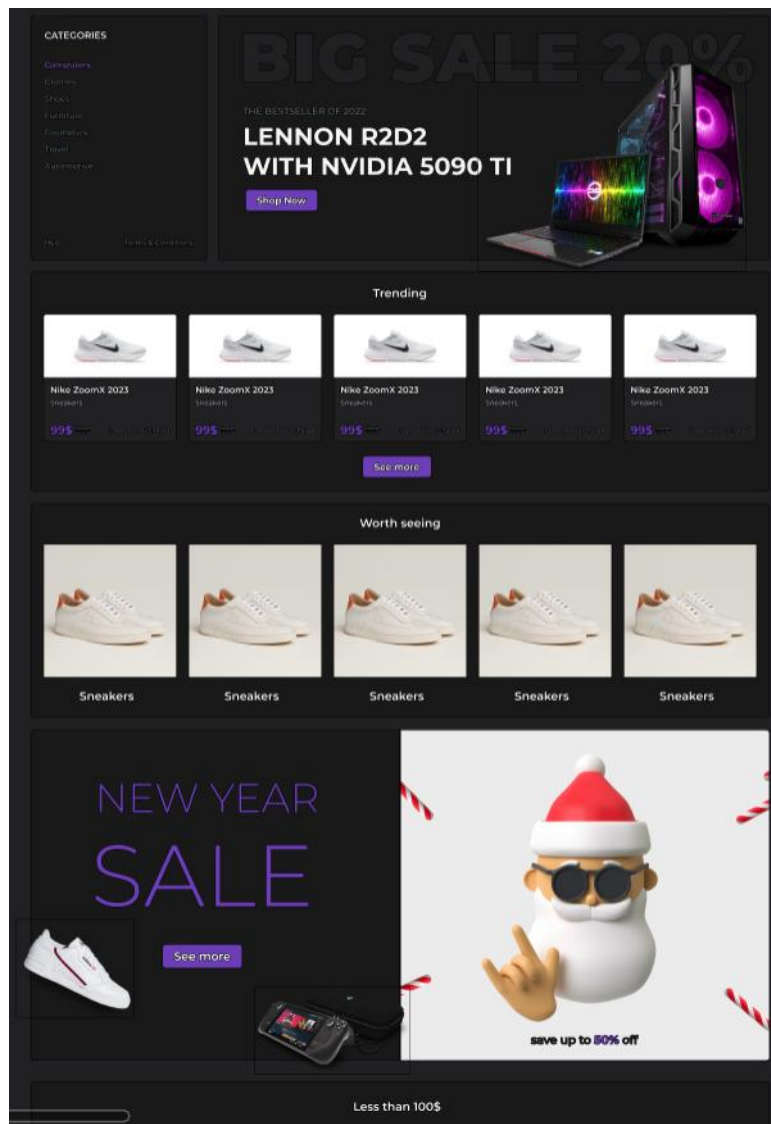


Рис. 3.1. Макет головної сторінки вебсайту

Кафедра КІТ

НАУ 23 33 34 000 ПЗ

	ПІБ			Лім.	Аркуш	Аркушіє
Розроб.	Черніцький Е.В.				45	23
Керівник	Толстікова О.В.			ТП-415Б - 122		
Н.Контр.	Толстікова О.В.					

РОЗДІЛ 3. СТВОРЕННЯ  
ВЕБСАЙТУ ІНТЕРНЕТ-  
МАГАЗИНУ “STYLESHOP”

ТП-415Б - 122

Верстка макету є необхідним етапом, що допомагає перетворити дизайнерські концепції у функціональний та стильний інтернет-магазин. Вона забезпечує візуальну привабливість, функціональність та оптимальну продуктивність, які є ключовими факторами для забезпечення успіху магазину та задоволення потреб клієнтів.

Оскільки макет складається з кількох сторінок, верстка розпочинається з головної сторінки(рис. 3.1), зокрема з повторно використовуваних компонентів, тобто ті, які повторюються на кожній сторінці. Першим повторюваним компонентом є Header.jsx(рис. 3.2), тобто шапка інтернет-магазину.

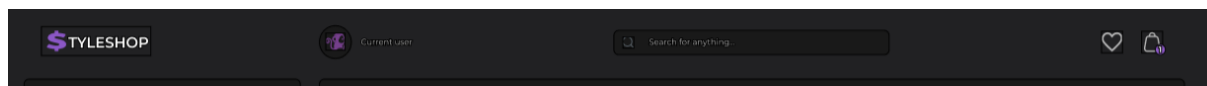


Рис. 3.2. Верстка компоненти Header.jsx

Header.jsx залишається незмінною компонентою інтернет магазину на будь-якій сторінці, яка складається з логотипу StyleShop, аватару користувача, його ім'я та прізвище, пошукове поле товарів, іконки вподобайка та кошик. Перша іконка відповідає за товари що сподобалися користувачу, а друга відповідає за товари що користувач обрав для замовлення. Зареєстрований користувач має змогу завантажити будь-яку фотографію на свій розсуд, якщо ж фотографія відсутня або користувач не увійшов в акаунт - буде відображатися фотографія за замовчуванням.

Наступний повторно використовуваний компонент є Footer.jsx (рис. 3.3), тобто підвал магазину. Футер є важливою частиною веб-сторінки, яка знаходиться внизу і надає користувачам додаткову інформацію, посилання та контактні дані. Він сприяє створенню повного враження про магазин та покращує його зручність використання.

Основною метою розробки футера є функціональність та естетичність, він має доповнювати загальний дизайн магазину та забезпечувати користувачів необхідною інформацією. Футер складається з логотипу

інтернет-магазину “StyleShop”, прізвище розробника і посилання на соціальні сторінки інтернет-магазину.

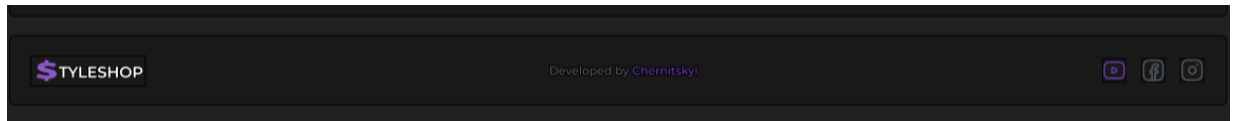


Рис. 3.3. Верстка компоненти Footer.jsx

Після завершення роботи Footer.jsx, верстка продовжується компонентою Navbar.jsx (рис. 3.4). Навбар є одним з ключових елементів веб-сторінки, який забезпечує основну навігацію та дозволяє користувачам швидко досягати до різних розділів і функціональності магазину.

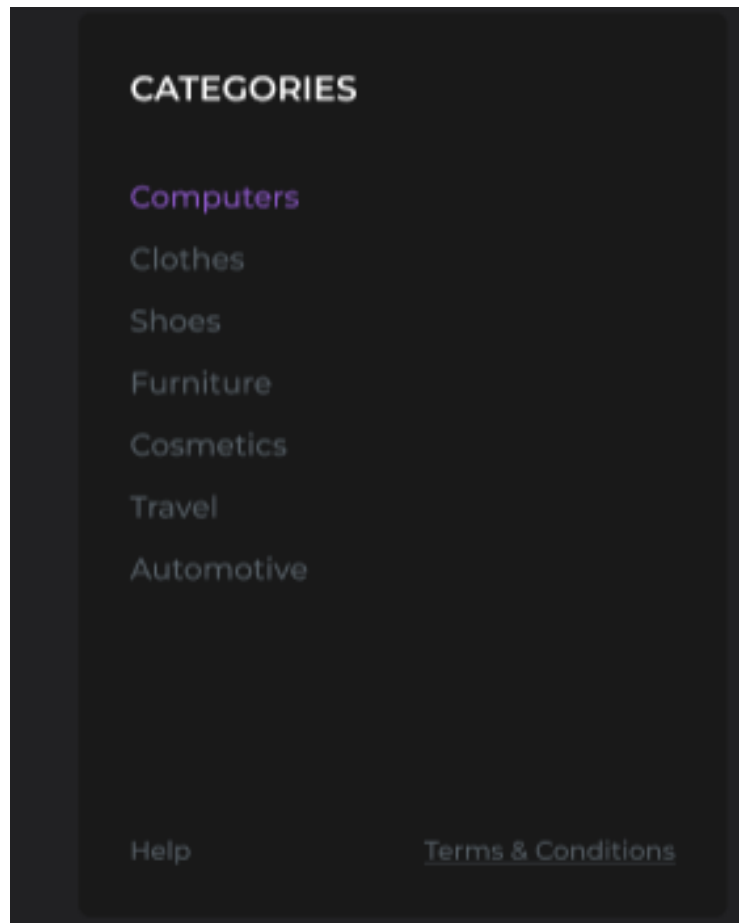


Рис. 3.4. Верстка компоненти Navbar.jsx

Головною метою під час верстки компоненти є розробка привабливого та функціонального навбару, який забезпечить зручну навігацію та покращить загальний взаємодію користувачів з магазином. Навігаційна

панель складається таких категорій: комп'ютери, одяг, взуття, мебель, косметика, подорож, та автомобільні аксесуари.

Остання повторно використовувана компонента головної сторінки є банер (рис. 3.5).



Рис. 3.5. Верстка банеру головної сторінки

Банер і перший компонент головної сторінки є ключовим елементом, який визначає загальний вигляд та функціональні можливості сторінки. Верстка цього компоненту є важливим завданням, оскільки він відповідає за перше враження користувачів та створює їхню першу взаємодію з магазином. Основною метою розробки першого компоненту є привернення уваги користувачів, надання їм важливої інформації та стимулювання до подальшої навігації по сайту. В цьому контексті верстка включає в себе створення привабливого дизайну, правильне розміщення важливих елементів та забезпечення зручної навігації.

Після успішної верстки банеру необхідно розмістити 2 маркетингові розділи “Trending” (рис. 3.6) та “Worth seeing” (рис. 3.7), в інтернет-магазині під банером.

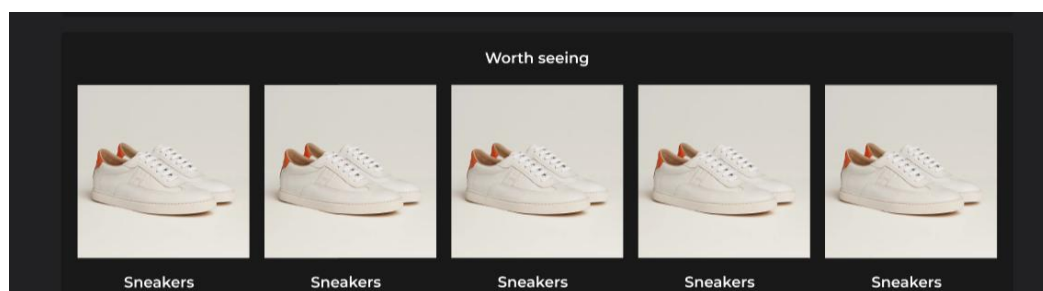


Рис. 3.6. Верстка компоненти рекомендацій



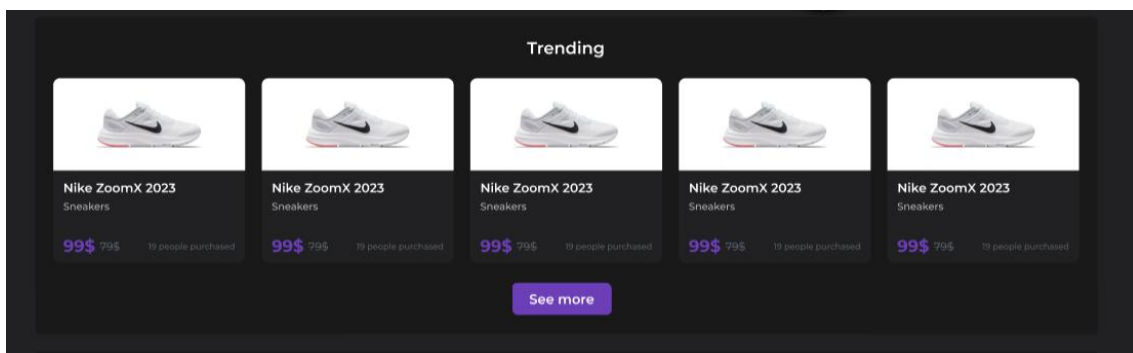


Рис. 3.7. Верстка компоненти з трендовими речами головної сторінки

Ці розділи привертають увагу відвідувачів та стимулюють їх до покупок. Вони представляють актуальні, популярні та модні товари, які відповідають сучасним тенденціям і попиту споживачів. Дані розділи дозволяють відвідувачам швидко знайти актуальні продукти, які є популярними серед інших покупців. Це створює відчуття сьогоденності і релевантності у магазині, що спонукає покупців до більш активних дій.

Крім того, присутність розділу "Trending" допомагає підтримувати імідж магазину як місця, де можна знайти найновіші і найбільш популярні товари.

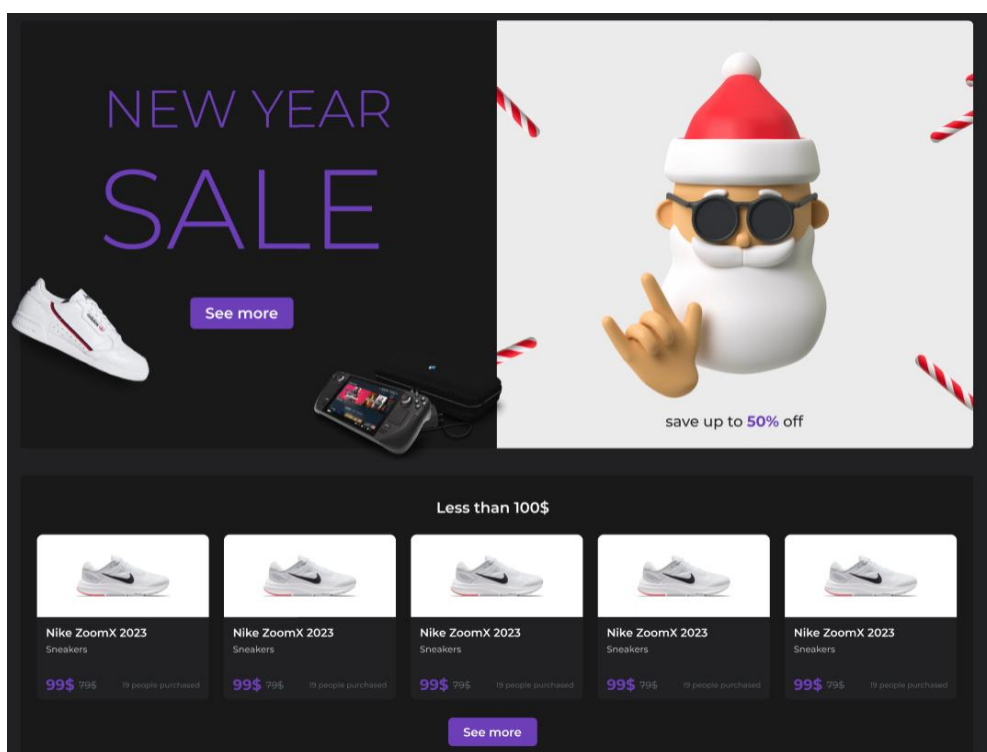


Рис. 3.8. Верстка останньої компоненти головної сторінки

На момент верстки вебсайту усі картинки однакові, оскільки відсутнє підключення до бази даних товарів, яке реалізується згодом, в розділі функціональне розширення інтернет-магазину.

Завершує верстку головної сторінки досить об'ємна компонента з банером та розділом товарів «Less than 100\$» (рис. 3.8). Банер має особливу вагу у контексті привертання уваги та стимулювання продажів. А компонента пропонує відвідувачам особливу можливість знайти товари за вигідними цінами, що не перевищують 100 доларів.

Банер, що супроводжує цей розділ, містить привабливий графічний контент та промоційні акції. Він має за мету привернути увагу відвідувачів та спонукати їх до перегляда та придбання товарів за доступними цінами. Цей розділ дає змогу швидко знайти бюджетні варіанти товарів без необхідності витратити багато часу на пошук.

Верстка головної сторінки завершена, та це лише половина роботи, оскільки наступний етап – сторінка товару(рис. 3.9).

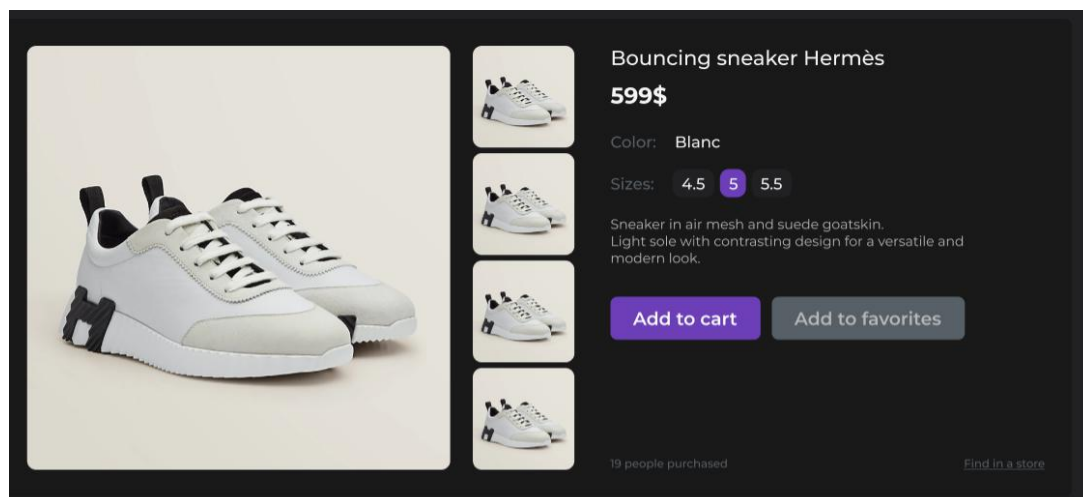


Рис. 3.9. Верстка сторінки обраного товару

Сторінка відіграє ключову роль у переконанні відвідувачів зробити покупку, надаючи їм детальну інформацію про конкретний товар та вражаючи їх якісним дизайном. На сторінці товару розміщена велика фотографія та галерея зображень, що демонструють товар з різних кутів та

деталей. Це дає можливість відвідувачам краще оцінити зовнішній вигляд та якість товару перед покупкою. Також на цій сторінці в залежності від товару можуть бути представлені додаткові зображення, відеоогляди або ілюстрації, які ще більше допомагають уявити собі товар. Опис товару є ще одним важливим компонентом сторінки товару. Він містить докладні відомості про характеристики, матеріали, розміри, кольори та інші важливі деталі, що допомагають відвідувачам зробити обґрунтований вибір. Крім того, опис може містити інформацію про бренд, виробника, гарантійні умови та інші фактори, які можуть вплинути на рішення покупки.

Під обраним товаром завжди присутній розділ «Related products» (рис. 3.10). Він відіграє важливу роль у залученні уваги та заохоченні покупців до додаткових покупок. Цей розділ містить список товарів, які пов'язані або доповнюють головний товар, що переглядається.

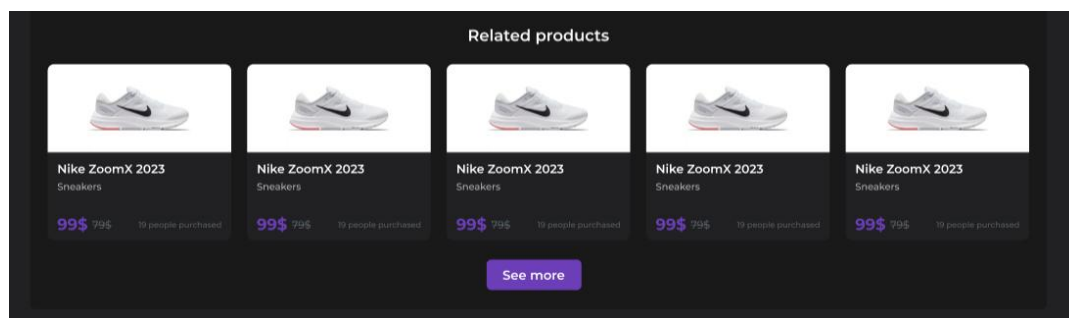


Рис. 3.10. Верстка розділу «Related products»

Розділ "Related Products" створює можливості для крос-продажу та збільшення середнього чеку. Він пропонує покупцям додаткові варіанти товарів, які можуть їх зацікавити на основі їхніх переглядів, попередніх покупок або рекомендаційних алгоритмів. Це дозволяє стимулювати покупців робити додаткові покупки, розширюючи їхні можливості і варіанти вибору.

Розділ "Related Products" містить мініатюрні зображення товарів, назви, ціни і спеціальні пропозиції, які привертають увагу покупців. Додаткові товари можуть бути пов'язані з основним товаром за категоріями, характеристиками, стилем, колекціями або іншими параметрами. Це

допомагає покупцям знайти подібні або додаткові товари, які можуть відповідати їхнім потребам і бажанням.

Наступним етапом верстки стає реєстраційне вікно та форма авторизації (рис. 3.11). Форма реєстрації надає можливість користувачам створити обліковий запис і отримати доступ до різноманітних функцій та переваг. Форма відіграє важливу роль у зборі необхідної інформації від користувачів, що дозволяє підтримувати зв'язок з ними, забезпечувати персоналізований досвід та забезпечувати безпеку даних.

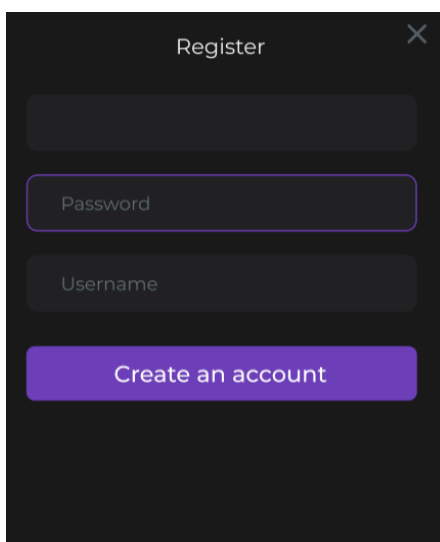
A dark-themed mobile registration form titled "Register" with a close button (X) in the top right corner. It features three input fields: a top field (likely for email), a "Password" field, and a "Username" field. Below the fields is a prominent purple button labeled "Create an account".

Рис. 3.11. Верстка форми авторизації користувача

Форма реєстрації містить поля, які заповнюються користувачами, такі як ім'я, прізвище, електронна пошта, пароль та посилання на картинку для створення аватару. Ця інформація є необхідною для створення облікового запису і забезпечення зручності замовлення та доставки товарів.

Передостання сторінка, яку необхідно зверстати – це розділ товарів по категорії (рис. 3.12). Сторінка товарів по категорії надає користувачам можливість швидко знайти і переглянути товари з конкретної категорії. Ця сторінка є ефективним інструментом для навігації по асортименту магазину та забезпечення зручності вибору. На сторінці товарів по категорії розміщені мініатюрні зображення товарів, назви, ціни та інші важливі характеристики. Це дозволяє користувачам швидко оцінити доступні товари та знайти той, який їх зацікавив.

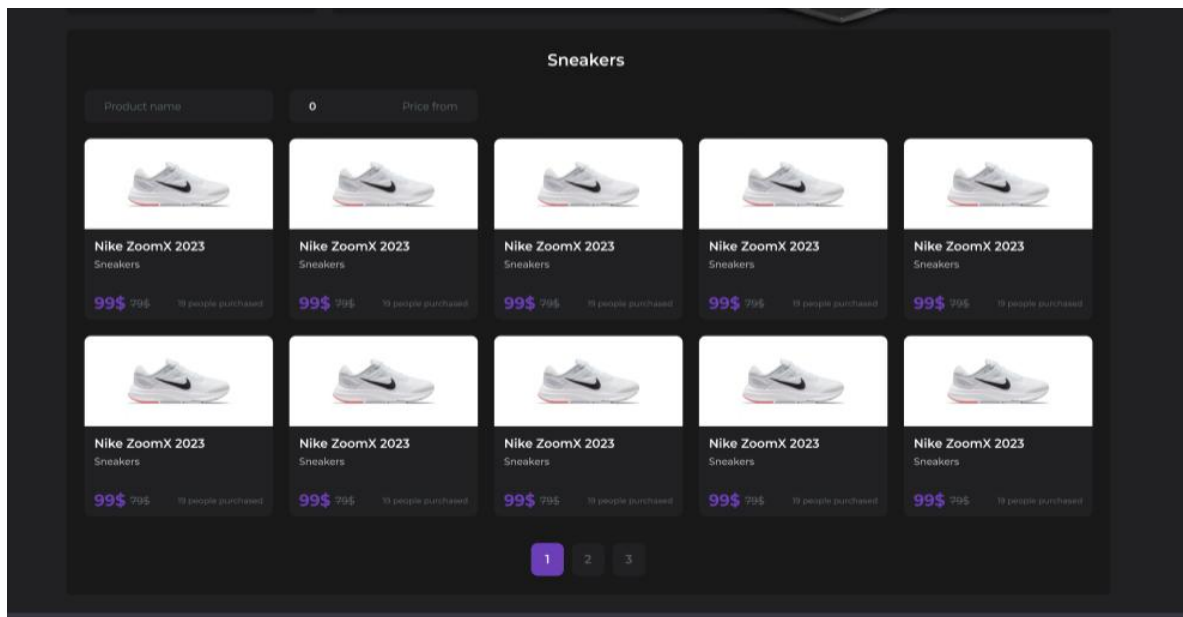


Рис. 3.12. Верстка сторінки товарів по категорії

Зручне сортування та фільтрація товарів також є важливою частиною верстки сторінки, що допомагає користувачам швидко знайти товари за певними критеріями, такими як ціна, розмір, бренд, оцінка та інші параметри. Товари представлені у логічному порядку та легко переглядаються без зайвих зусиль. Відображення товарів у вигляді сітки. Також можна використовувати пагінацію, щоб забезпечити зручний перехід між сторінками з багатьма товарами.

Останнім етапом верстки інтернет-магазину “StyleShop” залишилася верстка сторінки оформлення замовлень (рис. 3.13), або її ще називають корзина. Сторінка корзини та оформлення замовлень надає користувачам можливість переглядати, редагувати і оформляти свої замовлення перед завершенням покупки. Ця сторінка має важливу роль у зручності та безпеці процесу оформлення замовлення для користувачів. На сторінці корзини користувач зазвичай бачить список товарів, які він додав до своєї корзини. Для кожного товару відображені зображення, назва, ціна, кількість та сума. Користувачі мають можливість змінити кількість товару або видалити товари. Ефективна верстка сторінки корзини та оформлення замовлень полягає в її простоті, зрозумілості та зручності використання, а інтерфейс інтуїтивно зрозумілий.

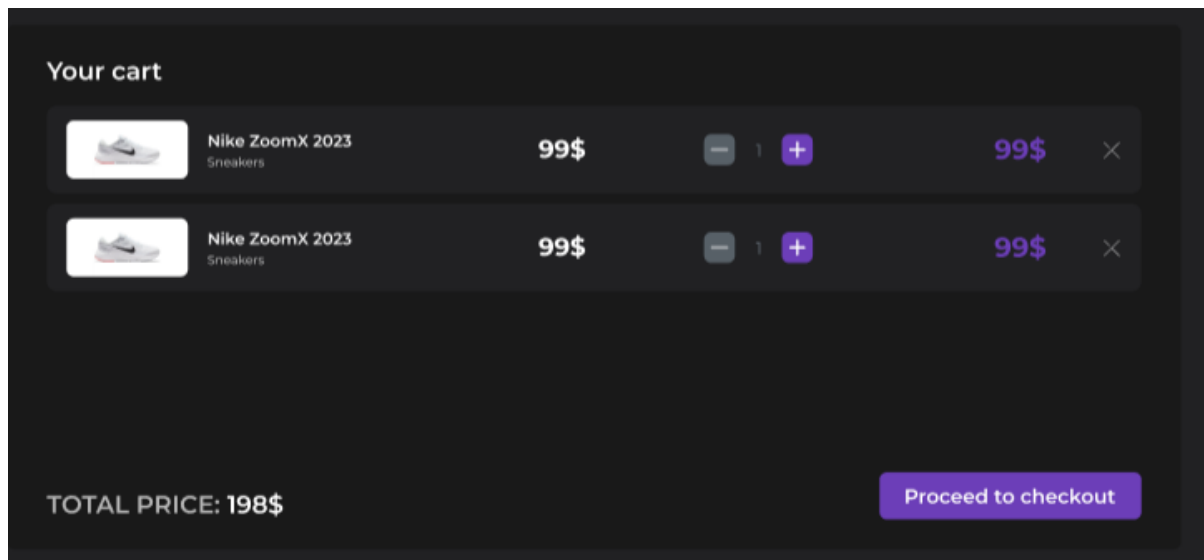


Рис. 3.13. Верстка корзини користувача, сторінка оформлення замовлень.

### 3.2. Розробка функціональної складової вебсайту, тестування

Другий етап розробки інтернет-магазину "StyleShop" – розробка функціональної складової вебсайту. Це досить складний процес, оскільки потребує спеціальних навичок, знань мови програмування Javascript та технологій розробки у фреймворку React.js для створення потужного, інтерактивного та розширюваного вебсайту.

Перш за все, інтернет-магазину необхідний API (Application Programming Interface) – це набір визначень та протоколів, які дозволяють магазину "StyleShop" взаємодіяти з сервером, щоб отримувати та надсилати дані. Це може включати отримання інформації про товари, замовлення, користувачів, оплату тощо, а також надсилання нових замовлень або оновлення даних на сервері [16]. Усі ці функціональності, які надає API, допоможуть магазину "StyleShop" покращити свою ефективність, забезпечити зручний користувацький досвід та розширити можливості у веб-середовищі. Зазвичай, якісний API платний, але для кваліфікаційної роботи було знайдено безкоштовний сервіс Platzi Fake Store API (рис. 3.14). Безкоштовний сервіс надає https посилання, яким можна користуватися для написання запитів на сервер.

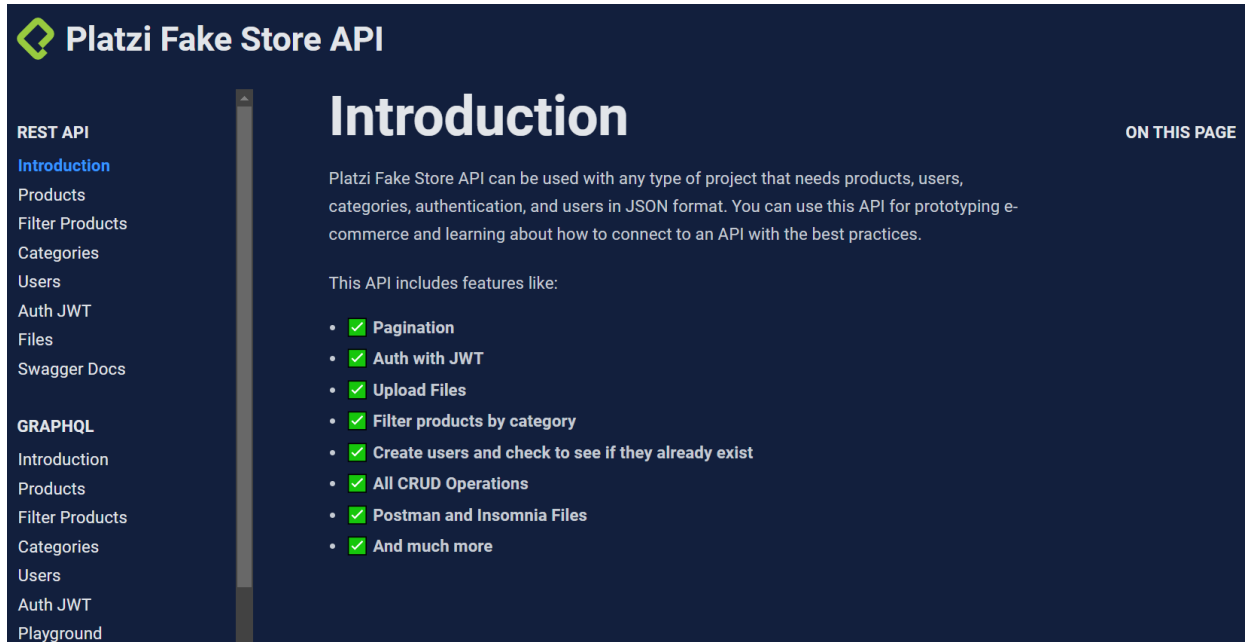


Рис. 3.14. Сервіс з безкоштовним API

Оскільки верстка вебсайту вже готова і замість товарів в магазині знаходяться «заглушки», то наступним кроком розробки є реалізація get запиту на сервер, тобто зробити запит товарів на сервер і у відповідь отримати список справжніх товарів, що знаходяться у базі даних на сервері.

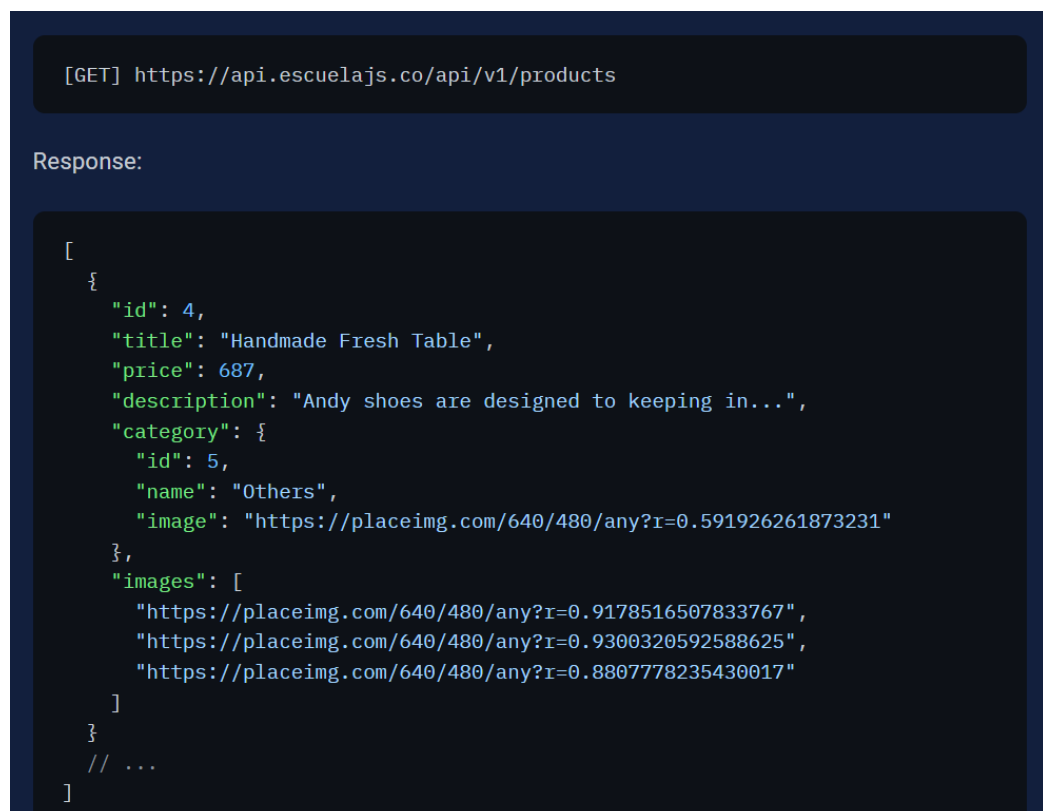


Рис. 3.15. Вигляд Get-запиту і відповідь серверу



У відповіді сервера міститься багато цінної інформації (рис. 3.15): id – унікальний ідентифікатор елемента, назва товару, картинки різних розмірів, ціна, опис, кількість людей, що придбали товар, ціна. Тепер необхідно відобразити товари замість «заглушок» (рис. 3.16). Завдяки запиту до сервера, магазин отримує найсвіжішу інформацію про товари. Це дозволяє показувати користувачам актуальні дані про товари, що допомагає покращити користувацький досвід.

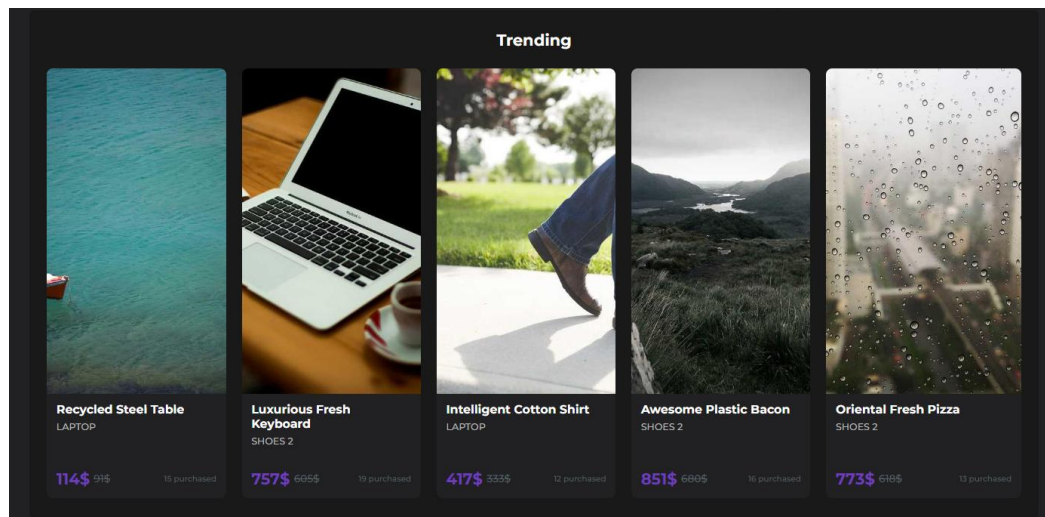


Рис. 3.16. Нові товари розділу “Trending”

Запит на товар у тренді товар виконано, але на сайті ще є інші розділи, наприклад, розділ за категоріями “Worth to see”. Запит по категоріях до API сервера інтернет-магазину є важливою функціональністю, яка дозволяє класифікувати та організовувати товари за різними категоріями. Цей тип запиту дозволяє інтернет-магазину пропонувати користувачам зручний спосіб навігації та пошуку товарів, відповідно до їхніх інтересів і вимог.

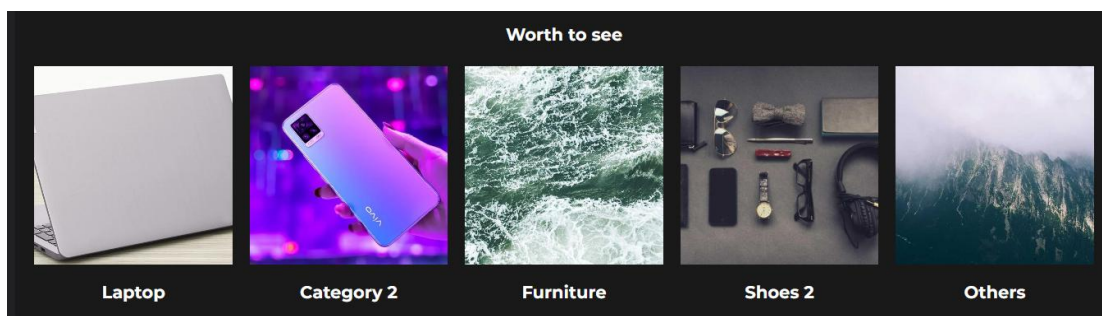


Рис. 3.17. Нові товари розділу “Worth to see”



Завдяки категоріям, користувачі можуть швидко знайти потрібні товари, переходячи до конкретної категорії, що відповідає їхнім потребам. Це полегшує їм пошук і забезпечує зручний досвід покупки. Виконуючи схожий запит до серверу, як на товари, тільки по категоріям, сервер повертає дані про категорії які виводяться у компоненту розділу “Worth to see” (рис. 3.17).

Запит по категорії "less than 100\$" до API сервера інтернет-магазину StyleShop є корисним інструментом для користувачів, які шукають доступні товари в межах певного бюджету. Цей тип запиту дозволяє класифікувати товари за ціновим діапазоном та надає ряд переваг для покупців. Це особливо важливо для тих, хто хоче знайти вигідні пропозиції, розпродажі або товари зі знижкою. Категорія "less than 100\$" (рис. 3.18) може містити широкий асортимент товарів, що дозволяє покупцям мати більше варіантів для вибору.

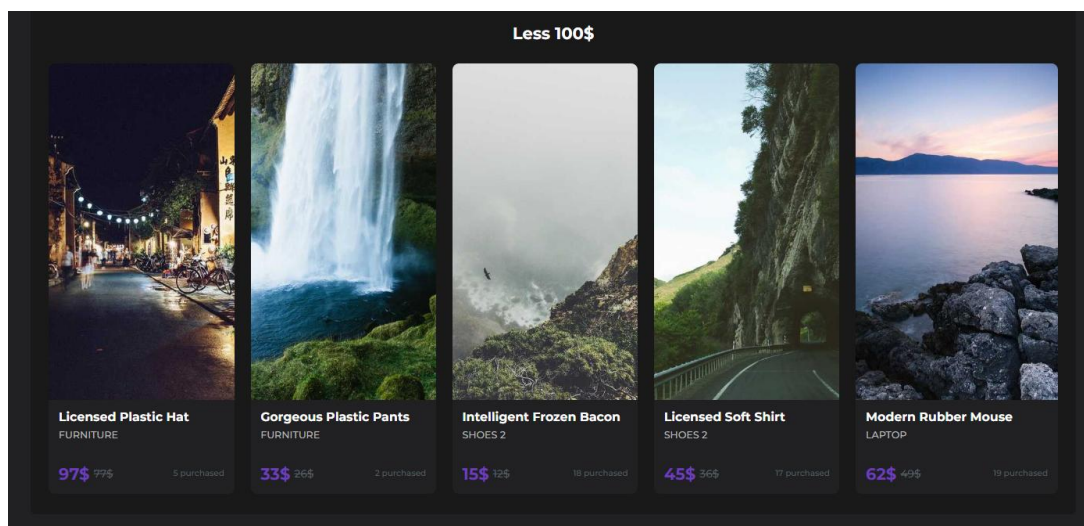


Рис. 3.18. Нові товари розділу “Less 100\$”

Вона охоплює товари з різних категорій, включаючи одяг, аксесуари, електроніку, тощо, що робить її привабливою для різноманітних потреб та вподобань користувачів. А запит робиться на всі товари, коли сервер повертає дані, функція з боку клієнта обробляє дані і фільтрує за ціною меншою за 100\$, все що вдалося відфільтрувати виводиться в компоненті.

Останній запит буде відсилатися на сервер для категорії категорії "related" на сторінці обраного товару (рис. 3.19). Категорія стане корисним

інструментом для користувачів, які шукають пов'язані або супутні товари до конкретного продукту.

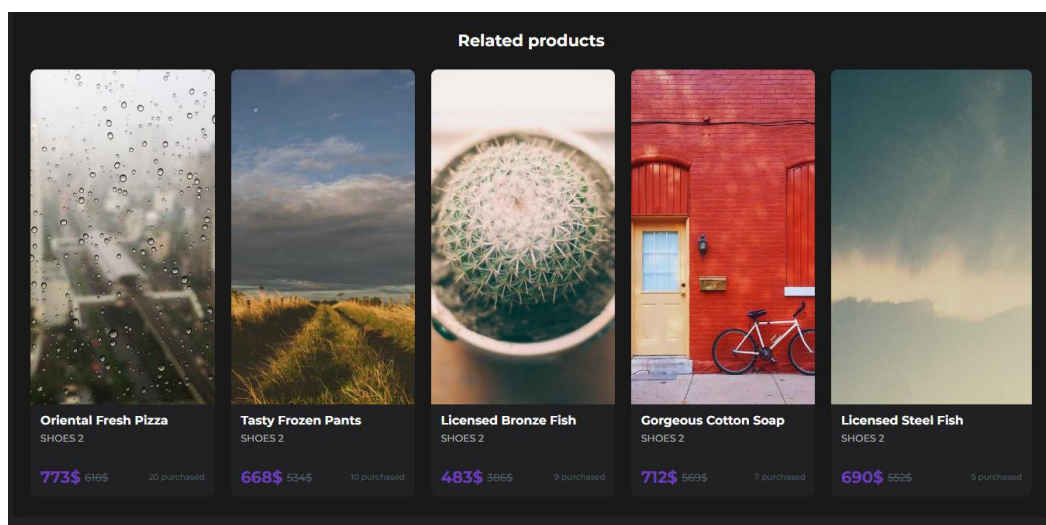


Рис. 3.19. Вивід товарів розділу “Related products” на сторінці обраного товару

Наприклад, якщо користувач переглядає смартфон, запит по категорії "related" може вивести на екран аксесуари, чохли, зарядні пристрої або навушники, які сумісні з цим смартфоном. Таким чином, користувачі отримують більше варіантів для вибору та можуть знайти додаткові товари, які можуть їх зацікавити. Категорія "related" допомагає просувати й продавати додаткові товари. Вона показує покупцям доповнення до їхніх покупок, що може спонукати їх до додаткових покупок. Наприклад, якщо користувач купує плаття, запит по категорії "related" може вивести на екран взуття, прикраси або сумки, які гармоніюють з цим платтям, сприяючи збільшенню обсягу продажів.

Наступною, не менш цікавою особливістю магазину стане пошук по всьому асортименту товарів (рис. 3.20). Поле пошуку дозволяє користувачам швидко знайти той товар, який їх цікавить. Вони можуть ввести ключові слова, опис або навіть артикул товару і отримати відповідні результати. Це економить час та зусилля, оскільки користувачам не потрібно переглядати всі сторінки або категорії, щоб знайти потрібний товар. Поле пошуку може бути

налаштоване для використання різних алгоритмів і методів, що дозволяє поліпшити якість пошуку та точність результатів.

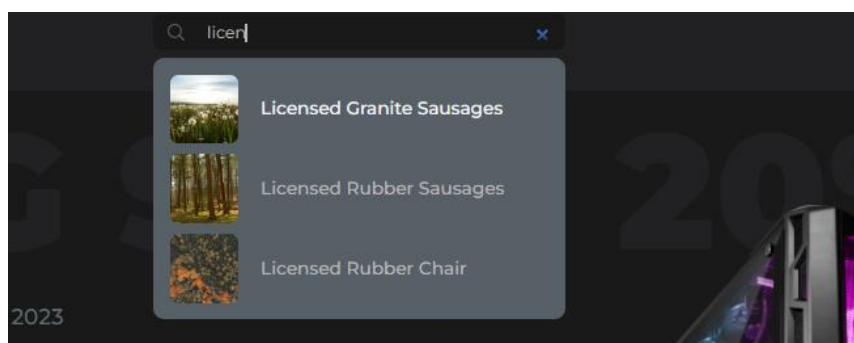


Рис. 3.20. Поле пошуку товарів в інтернет-магазині

Це допомагає зменшити кількість невідповідних або непотрібних результатів і дозволяє користувачам швидко знайти саме те, що вони шукають.

Останнім елементом розширення функціоналу вебсайту стала авторизація та реєстрація користувача через JSON webtoken запит. Використання token запиту на API сервера дозволяє забезпечити безпеку обміну даними між клієнтською стороною та сервером. Під час авторизації користувача, сервер генерує унікальний токен, який передається назад до клієнта. Цей токен може бути використаний для ідентифікації та перевірки автентичності користувача при подальших запитах. Такий підхід допомагає запобігти несанкціонованому доступу до особистої інформації користувачів. Використання token запиту дозволяє зберігати інформацію про авторизацію на стороні клієнта, що полегшує надання доступу до облікового запису користувача без необхідності повторно вводити логін та пароль при кожному запиті до сервера. Це забезпечує зручність та швидкість у використанні сайту, оскільки користувачам не потрібно постійно вводити свої облікові дані. Завдяки token запиту, сервер може зберігати інформацію про користувача і надавати персоналізовані дані та функціонал. Наприклад, після авторизації користувача, сайт може відображати історію покупок, улюблені товари, персоналізовані рекомендації та інші налаштування, що підвищують рівень задоволення від використання сайту.

### 3.3. Хостинг

Існує кілька безкоштовних платформ, на яких можна розмістити свій інтернет-магазин. Ось кілька сервісів, разом з їх перевагами та недоліками:

– WooCommerce: Це безкоштовний плагін для WordPress, який дозволяє перетворити ваш сайт на повноцінний інтернет-магазин. Він має велику спільноту користувачів та розширень, що дозволяє розширити можливості вашого магазину. Однак, для використання WooCommerce потрібен хостинг та доменне ім'я, а також необхідно мати базові знання WordPress;

– Shopify: Ця платформа пропонує безкоштовний пробний період та легкий у використанні інтерфейс. Вона має багато готових тем і розширень, що дозволяє швидко створити інтернет-магазин. Однак, після закінчення пробного періоду, Shopify стягує щомісячну плату, а також комісію за кожен продаж. Крім того, ви обмежені можливостями кастомізації інтернет-магазину, оскільки весь код знаходиться на сервері Shopify;

– OpenCart: Це безкоштовна платформа з відкритим вихідним кодом, яка надає широкі можливості для налаштування та розширення інтернет-магазину. Вона має багато готових шаблонів та модулів, що дозволяє створити унікальний дизайн і розширити функціонал вашого магазину. Однак, для використання OpenCart вам потрібно мати технічні знання для налаштування та управління платформою;

– Netlify: дозволяє швидко розгорнути інтернет-магазин із сховища Git. Процес автоматизований та простий у використанні. Netlify пропонує безкоштовний хостинг, що дозволяє розмістити інтернет-магазин без додаткових витрат. Платформа використовує глобальну мережу доставки контенту (CDN), що забезпечує швидку загрузку вашого сайту для користувачів з різних регіонів світу. Netlify легко масштабується, що дозволяє вашому інтернет-магазину зростати разом із збільшенням трафіку та популярності.

Серед недоліків Netlify можна виділити обмежену функціональність, оскільки Netlify є хмарною платформою, немає можливості самостійно керувати сервером, що обмежує гнучкість у виборі конфігурацій та розширень сервера. Netlify інтегрується з різними сторонніми сервісами, такими як системи керування версіями, CI/CD і інші. Однак, це може створити залежність від цих сервісів та обмежити користувача у виборі [15].

Проаналізувавши безкоштовні платформи, на основі описаних переваг та недоліків було обрано сервіс Netlify (рис. 3.21) як платформу для хостингу інтернет-магазину “StyleShop”.

Щоб захостити вебсайт на Netlify, було виконано наступні кроки:

- необхідно зайти на веб-сайт Netlify і створити безкоштовний акаунт.

Можна використовувати свій обліковий запис Google або GitHub для швидкого входу;

- після входу в акаунт сервіс дозволяє додати свій проект, натиснувши кнопку "New site from Git" або "New site from ZIP". Обирається потрібна опція залежно від того, де звідки завантажується проект;

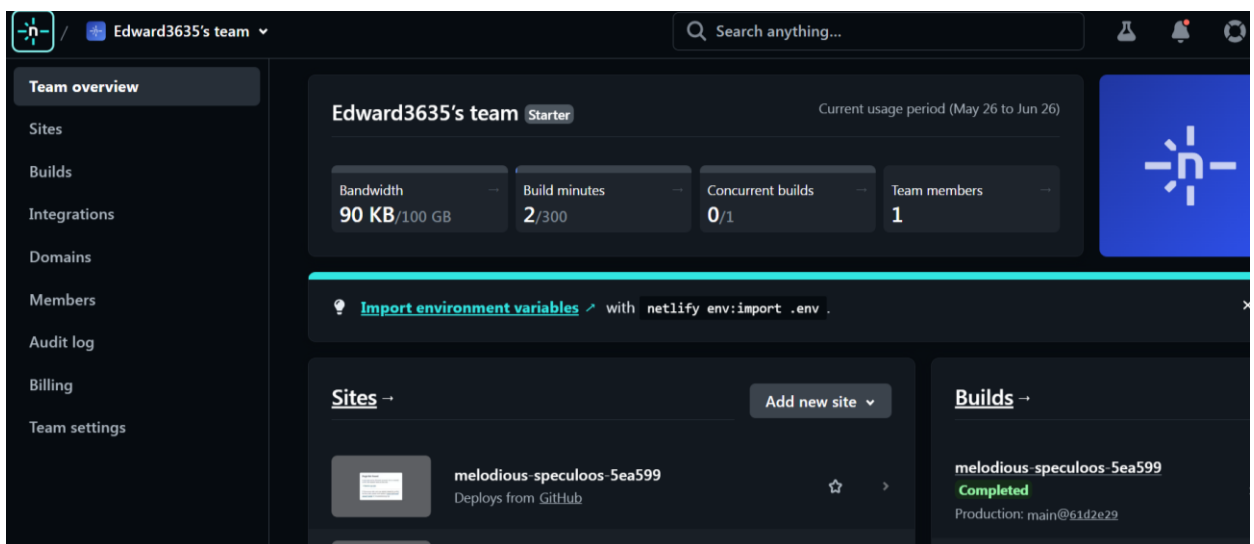


Рис. 3.21. Платформа Netlify

- далі відбувається підключення репозиторію або завантаження ZIP-архіву: Оскільки проект знаходиться на Git, обирається відповідний Git-сервіс та під'єднується свій репозиторій;

- налаштування сайту: після підключення проекту були налаштовані основні параметри вебсайту, такі як доменне ім'я, налаштування збірки (build settings) та налаштування розгортання (deployment settings);
- запуск процесу розгортання: Netlify автоматично збирає проект, будує його та розгортає на своїх серверах;
- після початку розгортання сервіс демонструє живий лог розгортання, де будуть показані будь-які помилки або повідомлення про успіх. Цей процес відбувається досить швидко;
- перевірка роботи: після завершення розгортання вебсайт буде доступний за зазначеною розробником доменною назвою. Він відкривається в браузері, все працює належним чином.

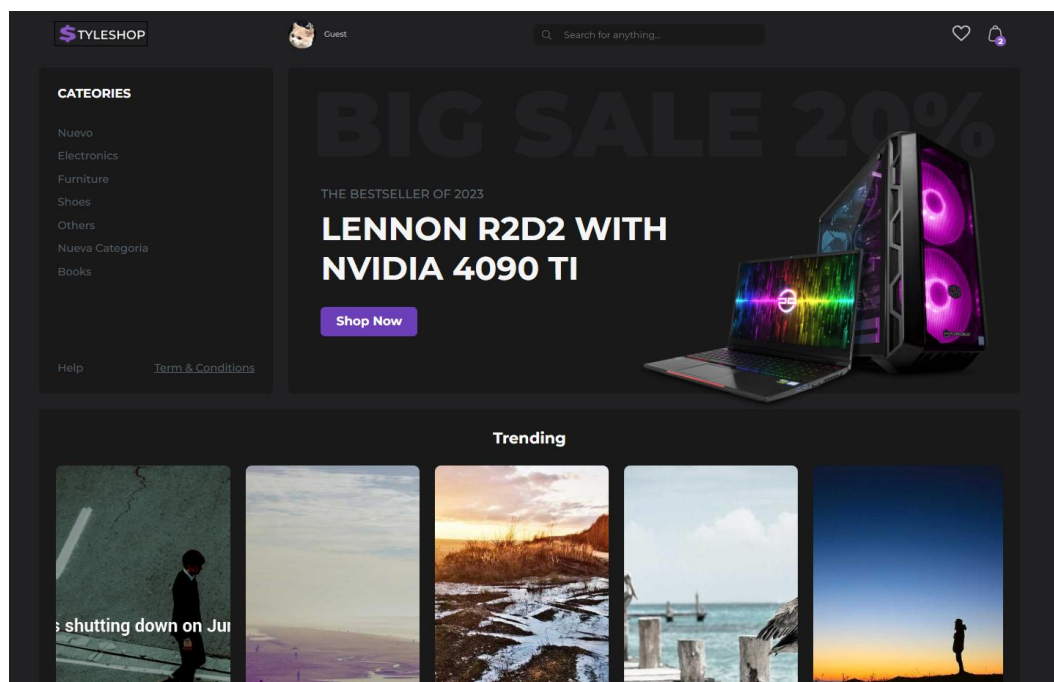


Рис. 3.22. Інтернет-магазин “StyleShop” на хостингу

Netlify має простий і зручний процес розгортання, що робить його популярним серед розробників. Він також надає безкоштовний план з обмеженнями, але можливості розширення і платні плани доступні для більш вимогливих проектів.

## ВИСНОВКИ ДО РОЗДІЛУ 3

Створення інтернет-магазину "StyleShop" - це складний та працездатний процес, який вимагає детального планування та продуманої роботи над кожною складовою. Розробка вебсайту поділена на 3 етапи: верстка макету, розширення функціональності магазину та тестування, і хостинг як завершальний етап розробки кваліфікаційної роботи.

Старт написання вебсайту інтернет-магазину "StyleShop" розпочався з верстки макету та побудови структури сторінок на HTML. Використання платформи React дозволяє ефективно керувати станом додатка, забезпечуючи швидку та зручну роботу з даними. Для розробки складніших функцій в інтернет-магазині "StyleShop" використовувались компоненти React, які дозволяють створювати багатофункціональні елементи та взаємодію між ними. Спеціальні компоненти React дозволяють реалізувати такі функції, як динамічне завантаження товарів, фільтрація за категоріями, розрахунок ціни залежно від обраних параметрів, кошик покупок з можливістю додавання та видалення товарів, а також забезпечують інтерактивну взаємодію з користувачем, наприклад, форми замовлення та оплати.

Завдяки використанню CSS вдалося стилізувати інтерфейс "StyleShop", додати привабливість, розміщення елементів на сторінці, вирівнювання, анімацію та інші декоративні ефекти, щоб створити зручне та естетично привабливе оточення для користувачів.

В завершальному етапі розробки вебсайт "StyleShop" вимагає ретельного тестування на працездатність і помилки. Після успішного тестування відбувається вивантаження роботи на хостинг в інтернет під загальнодоступний домен. В результаті клієнт, що відвідає "StyleShop" отримує потужний та привабливий інтернет-магазин брендів речей з використанням найновіших технологій веб-розробки.

Процеси, описані у цьому розділі, є завершальним і дають можливість зробити загальні висновки про отриманні результати.



## ВИСНОВКИ

Завдяки швидкому технологічному розвитку та зростанню популярності Інтернету, електронна торгівля виявила неймовірну потужність та вплив на сучасне суспільство. Зараз інтернет-магазин представляє собою віртуальну платформу, де товари та послуги продаються через Інтернет. Він дозволяє покупцям замовляти товари та здійснювати оплату онлайн, не виходячи зі зручності своїх домівок. Інтернет-магазини можуть пропонувати широкий спектр товарів, від одягу та електроніки до продуктів харчування та послуг.

У кваліфікаційній роботі детально описано ключові події в історії розвитку технологій і передумови створення перших інтернет-магазинів, були згадані прізвища людей і назви компаній, які вплинули на розвиток інтернет-магазинів. Проаналізувавши різні середовища розробки було обрано популярний і ефективний текстовий редактор VS Code для проектування та розробки вебсайту на платформі React.js. Робота розроблялася з використанням сучасних технологій та методологій, що дозволило досягти високої якості та продуктивності.

У розробці веб-сайту інтернет-магазину першим і не менш важливим етапом стала верстка сучасного та привабливого дизайну вебсайту для покупок онлайн. Для ефективнішої розробки у створенні інтернет-магазину, на другому етапі розробки було прийнято рішення застосувати фреймворк React.js для розширення функціоналу, він є одним з найпопулярніших фреймворків для фронтенд розробки. Фреймворк React.js у поєднанні з редактором VS Code створюють ідеальну комбінацію для розробки високоефективних, структурованих та масштабних веб-додатків. Використання технологій React.js на мові програмування Javascript дозволило створювати інтерактивні елементи, функціональні кнопки, взаємодію з користувачем, та забезпечувати відповідність на різних пристроях, а робота з HTML та CSS посприяла створенню привабливих та естетичних інтерфейсів вебсайту.



На фінальному етапі розробки, коли вебсайт інтернет-магазину “StyleShop” функціонує, він пройшов тестування на різноманітні помилки взаємодії клієнт-серверної архітектури, працездатності з клієнтом, і після виправлення неочікуваних багів та недоліків головною задачею стає хостинг сайту. Проаналізувавши кілька хостинг сайтів, остаточний вибір пав на безкоштовний онлайн ресурс. Шляхом нескладних але довготривалих кроків реєстрації, завантаження проєкту, налаштувань домену і активізації сайту на сервері, тепер кожний користувач інтернету має змогу відвідати і зробити замовлення на високоякісному та ефективному вебсайті “StyleShop”, який задовольняє потреби та очікування клієнтів, а також забезпечує безпеку та захист персональних даних користувачів.

Також, у кваліфікаційній роботі було враховано вимоги до зручності та зрозумілості інтерфейсу користувача, щоб забезпечити максимальний комфорт та зручність під час покупок. Було створено простий та зрозумілий інтерфейс, що дозволяє легко здійснювати покупки та швидко знаходити потрібні товари. Крім того у процесі розробки було використано сучасні інструменти та технології, такі як Git для контролю версій коду, Webpack для збірки та оптимізації проєкту, та багато інших. Інтернет-магазин забезпечує максимальну зручність під час покупок та високу якість обслуговування. Для збереження особистої інформації було використано надійний протокол HTTPS, що забезпечує захист персональних даних користувачів від несанкціонованого доступу [8]. Розробка інтернет-магазину брендових речей StyleShop є успішним прикладом застосування сучасних технологій та методологій розробки програмного забезпечення.

Результати кваліфікаційної роботи можуть бути використані для загального ознайомлення з основними характеристиками та перевагами вебсайту "StyleShop" на платформі ReactJS. Вони можуть бути корисними для розуміння технологій, використаних у створенні сайту, його функціональності та можливостей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Харенко О. О. Дослідження поняття веб-технологій і ролі front-end розробки в формуванні сучасного веб-простору / О. Харенко, С. Одокієнко, М. Люта – Київ: КНУТД, 2021. - С. 95-101.
2. Бородкіна І., Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів. / І. Бородкіна, Г. Бородкін. 2018. – 590 с.
3. Medium - Мікросервісна архітектура. [Електронний ресурс] Режим доступу: <https://medium.com/@IvanZmerzlyi/microservices-architecture-461687045b3d>
4. Краус К.М. Електронна комерція та інтернет-торгівля / К. Краус, Н. Краус, О. Манжура. – Київ: Видавництво ТОВ «Аграр Медіа Груп», 2021. – с. 34-39.
5. Reactjs ReactDOM. [Електронний ресурс] Режим доступу: <https://legacy.reactjs.org/docs/react-dom.html>
6. Офіційна документація HTML веб-розробка. [Електронний ресурс] Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTML>
7. Офіційна документація CSS веб-розробка. [Електронний ресурс] Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/CSS>
8. Офіційна документація HTTPS основні поняття. [Електронний ресурс] Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
9. Положення про дипломні роботи та проекти. [Електронний ресурс]. Режим доступу: <https://kit.nau.edu.ua/upload/dyplom.pdf>.
10. WebStorm. [Електронний ресурс] Режим доступу: [https://www.jetbrains.com/help/webstorm/getting-started-with-webstorm.html#ws\\_getting\\_started\\_user\\_interface\\_overview](https://www.jetbrains.com/help/webstorm/getting-started-with-webstorm.html#ws_getting_started_user_interface_overview)
11. Atom. [Електронний ресурс] Режим доступу: [https://linuxhint.com/atom\\_text\\_editor\\_tutorial/](https://linuxhint.com/atom_text_editor_tutorial/)

12. Sublime Text. [Электронный ресурс] Режим доступа:  
<https://docs.sublimetext.io/guide/#history>
13. Visual Studio Code. [Электронный ресурс] Режим доступа:  
<https://code.visualstudio.com/docs/getstarted/tips-and-tricks>
14. Visual Studio. [Электронный ресурс] Режим доступа:  
<https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
15. Figma Pros and Cons. [Электронный ресурс] Режим доступа:  
<https://medium.com/gizalab/pros-cons-of-using-figma-317115f762be>
16. Netlify. [Электронный ресурс] Режим доступа:  
<https://otakoyi.software/blog/overview-of-benefits-and-drawbacks-of-netlify-cms>
17. Reduxjs/toolkit. [Электронный ресурс] Режим доступа:  
<https://dev.to/dawnind/ditch-redux-use-reduxjs-toolkit-3h02>