

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ  
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

“\_\_” \_\_\_\_\_ 2023 р.

# КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ  
СИСТЕМИ ТА ТЕХНОЛОГІЇ”

**Тема: “ Web-сайт покупки побутових товарів з використання  
новітніх технологій”**

**Виконавець** Березін Андрій Валерійович

**Керівник:** к.т.н., доцент Моденов Юрій Борисович

**Нормоконтролер:** Ігор РАЙЧЕВ

(підпис)

**Київ 2023**

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Аліна САВЧЕНКО

“\_\_” \_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи студента**

**Березіна Андрія Валерійовича**

- 1. Тема роботи:** «Web-сайт покупки побутових товарів з використання новітніх технологій» затверджена наказом ректора від “29”вересня 2023р. за № 1976/ст
- 2. Термін виконання роботи:** 02.10.2023 – 31.12.2023.
- 3. Вихідні дані до роботи:** створення web-сайту покупки побутових товарів.
- 4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):** опис та аналіз вирішуваної проблеми, технологій реалізації та результатів, висновки.
- 5. Перелік обов'язкового графічного матеріалу:** слайди презентації MS PowerPoint.

## 6. Календарний план-графік

<i>№ з/п</i>	<i>Завдання</i>	<i>Термін виконання</i>	<i>Підпис керівника</i>
1	Аналіз та дослідження предметної області	02.10.2023р. – 15.10.2023р.	
2	Опрацювання літературних джерел, що відносяться до теми дипломного проекту	15.10.2023р. – 01.11.2023р.	
3	Аналіз технологій для вирішення поставленої задачі	01.11.2023р. – 15.11.2023р.	
4	Розробка web-сайту	15.12.2023р. – 01.12.2023р.	
5	Покриття розробленого софту мануальним тестуванням	01.12.2023р. – 05.12.2023р.	
6	Оформлення пояснювальної записки	05.12.2023р. – 10.12.2023р.	
7	Підготовка презентації та доповіді	10.12.2023р. – 17.12.2023р.	

Дата видачі завдання: 02.10.2023 р.

Керівник дипломного проекту \_\_\_\_\_ Юрій МОДЕНОВ  
(підпис керівника)

Завдання прийняла до виконання \_\_\_\_\_ Андрій БЕРЕЗІН  
(підпис випускника)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «WEB-сайт покупки побутових товарів» представлена на 66 сторінках, містить 17 рисунків, 6 наукових джерел.

**Ключові слова:** WEB-КОМЕРЦІЯ, WEB-САЙТ, FRONT-END, BACK-END, СТРУКТУРА, КОРИСТУВАЧ, ПРОЕКТУВАННЯ.

**Об'єкт дослідження:** сучасні технології розробки для створення комерційного бізнесу

**Предмет дослідження:** розробка WEB-сайту для покупки побутових товарів за допомогою сучасних WEB-технологій

**Мета дипломного проекту:** розробити WEB-сайт для покупки побутових товарів

**Метод дослідження:** аналіз існуючих сайтів WEB-комерції, аналіз та порівняння сучасних технологій та процесів розробки для створення інтернет-магазину

**Результат проекту:** розроблено WEB-сайт для покупки побутових товарів. WEB-сайт залучає нових клієнтів за рахунок своєї доступності, зручності та простоти у використанні.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

**HTML** - (англ. Hyper Text Markup Language) – мова розмітки гіпертексту

**БД** – база даних

**CSS**(від англ. Cascading Style Sheets) – це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду

**IDE** – Інтегроване середовище розробки

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТ.....	11
1.1 Доцільність проблеми в реальному та майбутньому часі.....	11
1.2 Аналіз та підбір технологій для процесу реалізації поставленої задачі..	12
1.3 Front-end частина. Підбір актуальних технологій на основі вимог поставленої задачі.....	17
1.4 Back-end частина. Підбір актуальних технологій на основі вимог поставленої задачі.....	23
1.5. Психологічний аналіз користувачів інтернет магазинів.....	31
1.6. Завдання дослідження.....	36
1.7. Висновок.....	37
РОЗДІЛ 2 АНАЛІЗ СЕРЕДОВИЩ РОЗРОБКИ ТА КЛЮЧОВІ ЕТАПИ СТВОРЕННЯ WEB-САЙТА.....	38
2.1. Підбір середовища розробки.....	38
2.2. Ключові пункти створення веб-сайту.....	49
2.3. Висновки.....	54
РОЗДІЛ 3 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ ЧАСТИНИ WEB-САЙТУ З МЕТОЮ ЗДІЙСНЕННЯ ПРОДАЖУ.....	55
3.1. Концепція сайту.....	55
3.2. Структура веб-сторінки.....	55
3.3. Висновок.....	64

## ВСТУП

Люди сучасності використовують мережу Інтернет щоденно з різною метою, таких як спілкування з близькими, ведення роботи, бізнесу та вирішення повсякденних справ. Основна перевага Інтернету полягає в його постійній доступності, яка охоплює будь-який час, будь-яке місце та будь-який куточок планети. Це дуже зручно, особливо для тих, хто прагне розвивати свій бізнес, оскільки існує можливість шукати партнерів у будь-якій країні та підтримувати з ними зв'язок у будь-який момент.

З інтенсивним розвитком мережі Інтернет людям доводиться швидко адаптуватися до цих змін і мати можливість здобувати знання та навички використання, наприклад, електронною поштою, інтернет-магазинами, сучасними месенджерами тощо.

Сучасний Інтернет все більше розвиває свої можливості як маркетплейс, перетворюючи концепції обслуговування клієнтів та модифікуючи поведінку споживача, а також їх сприйняття процесу покупки. Цим активно користуються комерційні підприємства, і найбільш розвинені серед них вже давно мають власні інтернет-магазини чи веб-сторінки. Їх завдання - приваблювати, утримувати та взаємодіяти з користувачем за допомогою інноваційного інтернет-маркетингу. Тепер покупець може насолоджуватися перебуванням у своєму улюбленому магазині навіть поза межами власного дому, достатньо мати доступ в Інтернет.

В сучасному світі використання Інтернету стає не лише засобом забезпечення повсякденних потреб, але і ключовим фактором для досягнення професійних та особистих цілей. Завдяки поширенню мережевих технологій індивіди мають можливість швидко отримувати інформацію, здійснювати віртуальні покупки та знаходити рішення широкого спектру завдань.

Суттєвий вплив Інтернету на сферу бізнесу виявляється в швидкості та доступності взаємодії. Можливість проводити онлайн-зустрічі, обмінюватися документами та укладати угоди без фізичного присутності стає дорогоцінною

перевагою для підприємців. Це також розширює можливості у веденні міжнародного бізнесу, оскільки електронний простір дозволяє легко знаходити потенційних партнерів чи клієнтів у будь-якій точці світу.

Розвиток мережевих технологій визначає нові стандарти взаємодії між компаніями та споживачами. Віртуальні магазини та онлайн-платформи не лише пропонують продукцію чи послуги, але й створюють інтерактивне середовище для спілкування з аудиторією. Компанії використовують соціальні мережі, месенджери та інші комунікаційні канали для побудови взаємовідносин зі споживачами, надаючи персоналізовані пропозиції та отримуючи зворотний зв'язок.

Цей тренд також впливає на сприйняття процесу покупки. Клієнти отримують можливість детально вивчати товар чи послугу, звертаючись до відгуків та рейтингів онлайн. Освоюючи інтернет-простір, покупці можуть придбати не лише товари, але і послуги, організувати подорожі, навчання чи отримати консультації експертів в різних галузях.

Це сучасне використання Інтернету створює нові можливості та виклики, вимагаючи від людей постійного вдосконалення навичок у використанні технологій та адаптації до стрімкого темпу змін. Причина цієї ситуації пояснюється, головним чином, стрімким підйомом популярності мережі Інтернет серед людей, а також неймовірним темпом сучасного життя, що супроводжується постійною зайнятістю та прагненням економити час і гроші в усьому.

Послуги в інтернет-бізнесі мають ряд переваг, до яких відноситься:

- Порівняно з перебуванням у фізичному магазині, обсяг інформації про спеціальні пропозиції, акції та сам продукт в Інтернет-магазині значно більший, що дозволяє покупцеві ефективніше використовувати час у віртуальному просторі, що є вигідним як для нього, так і для бізнесу;
- Віртуальний магазин представляє собою автоматизовану систему, яка може приймати будь-яку кількість клієнтів у будь-який час, що не можливо у реальних магазинах, де обслуговуванням займаються живі люди, що потребують відпочинку;
- Автоматизовані програми можуть виконувати бухгалтерські операції, розраховувати витрати та доходи, аналізувати обсяги продажів та надавати статистичні висновки для бізнесу, що економить час і гроші



підприємства, надаючи можливість розвиватися та зосереджувати ресурси на покращенні якості, розширенні та привабленні нових користувачів; • Веб-магазин може легко додавати нові товари, функціональність, змінювати дизайн та впроваджувати нові стратегії та концепції бізнесу з мінімальними зусиллями та витратами часу. Його гнучкість і легкість адаптації дозволяють власникові вибирати курс розвитку свого продукту; • Рост попиту завжди призводить до збільшення пропозиції. Чим більше людей користується Інтернетом, тим більше та різноманітніше стає бізнес, який активно адаптується до нових стандартів.

Цей тренд розширює можливості та висуває завдання перед сучасними людьми, щоб постійно вдосконалювати свої технологічні навички та адаптуватися до стрімкого розвитку.

Разом з тим, що Інтернет став важливою частиною нашого повсякденного життя, це також створює нові виклики та можливості для суспільства та бізнесу. З поширенням цифрових технологій і Інтернету зростає значущість онлайн-присутності для кожного. Індивіди, підприємства та уряди стають залежними від цифрових засобів для вирішення своїх потреб та завдань.

Однією з важливих тенденцій є електронна комерція, яка продовжує трансформувати спосіб, яким ми купуємо та продаємо товари. Онлайн-магазини, ринки та електронні платформи розширюють свою функціональність, роблячи ширший асортимент товарів і послуг доступним для глобального споживача. Це створює можливості для підприємств різного масштабу привертати клієнтів із різних кінців світу та розширювати свою аудиторію.

Також, варто відзначити вплив Інтернету на спосіб здобуття знань та навичок. Онлайн-освіта стає все більш популярною, надаючи можливість навчатися в будь-який зручний час та місце. Люди можуть отримувати нові знання, покращувати свої вміння та адаптуватися до змін швидше, ніж коли-небудь раніше.

Зростання кількості віртуальних спільнот і мережевих зв'язків також визначає сучасну діалектику Інтернету. Соціальні мережі, блоги та форуми стають платформами для обміну ідеями, взаємодії та створення нових цифрових культур.

Усе це свідчить про те, що Інтернет і далі буде грати ключову роль у формуванні нашого світу. З одного боку, він полегшує доступ до інформації та послуг, з іншого - ставить перед нами виклики, пов'язані з цифровою безпекою, конфіденційністю та етикою в онлайн-середовищі. Тому важливо розвивати та вдосконалювати свої навички для ефективного використання цього цифрового ресурсу в сучасному світі.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Доцільність проблеми в реальному та майбутньому часі

Спричинення наукової революції у XVI столітті призвело до виникнення численних інновацій, які спрощують та покращують життя людей. На мою думку, найбільш впливовим відкриттям виявилось наприкінці XX століття, забезпечивши можливість мільярдам людей об'єднатися та взаємодіяти, незважаючи на просторові та мовні бар'єри. Тут йдеться про глобальну мережу Інтернет.

Ця мережа не лише надала можливість спілкуватися з людьми по всьому світу, але й дозволила отримувати будь-яку інформацію всього за кілька кліків. Саме ця здатність стала одним із ключових аспектів комп'ютерної глобалізації. Пошук інформації в Інтернеті здійснюється через взаємодію клієнтської частини з сервером. Зазвичай звичайний користувач виступає як клієнтська частина, яка подає запит серверу. Сервер обробляє цей запит і висилає відповідь назад клієнту. Простий приклад запиту включає ключові слова теми, що цікавить користувача.

Чи буде мережа Інтернет використовуватися людьми у майбутньому? Розглядаючи досягнення провідних винахідників, таких як Білл Гейтс та Ілон Маск, можна з упевненістю стверджувати, що актуальність Інтернету в майбутньому не зменшиться, так само, як і всіх сервісів, які в ньому використовуються.

Виникнення та розвиток Інтернету визначають сучасний спосіб життя, перетворюючи спосіб, яким ми взаємодіємо та сприймаємо світ. Спільність Інтернету дозволяє нам не лише обмінюватися інформацією, а й знаходити спільні інтереси та вирішувати глобальні проблеми.

<b>Кафедра КІТ (47)</b>				<b>НАУ 23 01 01 000 ПЗ</b>			
<i>Виконав</i>	<i>Березін А.В.</i>			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	<i>Літера</i>	<i>аркуш</i>	<i>аркуші</i>
<i>Керівник</i>	<i>Моденов Ю.Б.</i>					11	27
<i>Консульт.</i>					<b>УС-211М 122</b>		
<i>Н-контрол.</i>	<i>Райчев І. Е.</i>						

Сервіси Інтернету надають неймовірні можливості для підприємництва та комерції. Від веб-магазинів до платформ для фрілансу та технологій блокчейн, Інтернет стає платформою для новаторських ідей та високотехнологічного розвитку. Такий розвиток дозволяє підприємствам пристосовуватися до змін, бути конкурентоспроможними та швидко реагувати на потреби споживачів.

Однією з ключових переваг Інтернету є його роль в освіті та доступі до знань. Інтернет робить навчання доступним та ефективним для мільйонів людей навколо світу. Від відкритих онлайн-курсів до ресурсів для самостійного вивчення, Інтернет відкриває двері для навчання та розвитку.

Майбутнє Інтернету, ймовірно, буде ще більш здійсненим і різноманітним. Розробники та інноватори продовжать створювати нові технології, які змінюватимуть наше життя. Інтернет речей (IoT), розширена реальність (AR), віртуальна реальність (VR) - це лише кілька напрямків, які об'єднують технології та Інтернет для створення ще більш захопливого та зручного середовища для спілкування та розвитку. Таким чином, майбутнє Інтернету залишається захоплюючим майданчиком для дослідження, творчості та досягнення нових висот.

## **1.2 Аналіз та підбір технологій для процесу реалізації поставленої задачі**

Для успішного користування Інтернетом створено спеціальні програми, які полегшують перегляд веб-сторінок без великих зусиль. Веб-браузери мають зрозумілий та зручний інтерфейс для користувачів з різним рівнем знань та компетенції. Один з найпопулярніших веб-браузерів, розроблений Google, називається Google Chrome. Він об'єднує зручність для користувача та можливості для розробників програмного забезпечення. Проте велика кількість користувачів сама по собі не гарантує якість програмного продукту. Тому важливо ознайомитись з декількома браузерами, вибрати той, який підтримує всі сучасні технології розробки та є зручним для програміста.

З моменту винайдення веб-браузерів відбулися значні зміни у способі, яким ми взаємодіємо з інтернетом. Сьогодні ці програми стали невід'ємною частиною нашого онлайн-життя, надаючи нам можливість легко та ефективно користуватися різноманітними сервісами та ресурсами.

Google Chrome є безкоштовним браузером, розробленим компанією Google, і відрізняється великою швидкістю порівняно з конкурентами. Він має зручне сортування вкладок і вбудовані розширення для перекладу веб-сторінок з іноземних мов. Також в Google Chrome є окрема платформа з розширеннями, що дозволяє користувачеві кастомізувати браузер під свої потреби, не нав'язуючи непотрібні розширення, що може вплинути на його швидкодію. Завдяки своїй швидкості та функціональності, став одним з найпопулярніших браузерів. Його можливість кастомізації та інтеграція з численними розширеннями надають користувачам гнучкість у використанні та налаштуванні згідно з їхніми потребами.

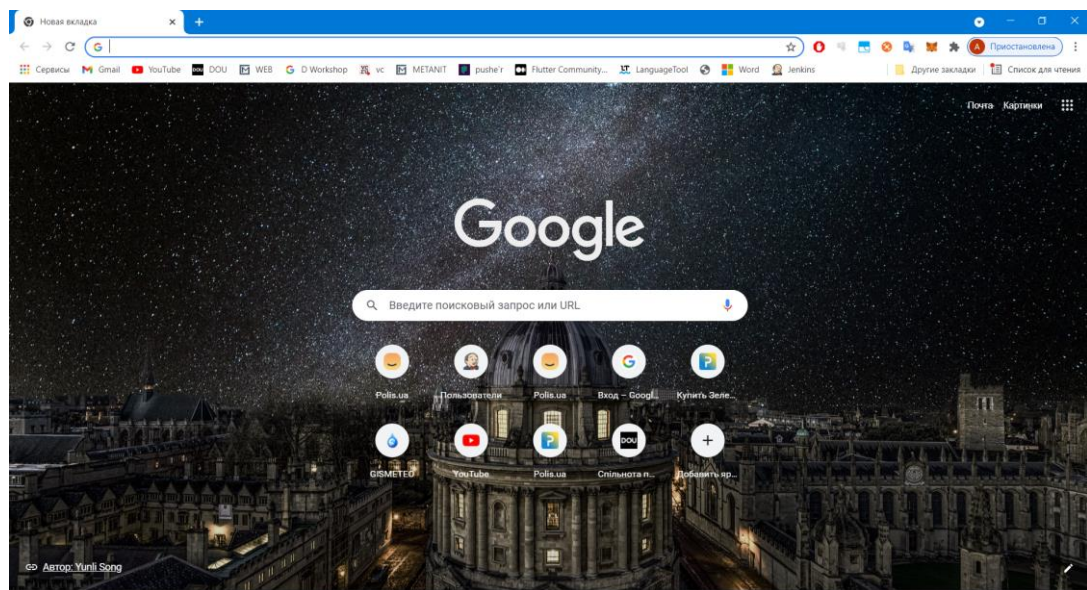


Рис. 1.1. Головна сторінка браузера «Google Chrome».

Опера - це веб-браузер, створений компанією Opera Software. Він відрізняється легкістю використання для апаратної частини та відповідає стандартам HTML, маючи вбудований VPN.

Microsoft Edge (під кодовою назвою "Spartan") - це найновіший інтернет-браузер від Microsoft, представлений в операційній системі Windows 10 як заміна Internet Explorer. Він є сучасним, легким та економить ресурси, підтримує чорнильне введення, додатки для Firefox та Chrome, має швидке відображення сторінок, інтеграцію з Cortana, автоматичне заповнення форм, приватний перегляд та розширені функції безпеки, і вже не включає ActiveX.

Mozilla Firefox є вільним веб-браузером із відкритим кодом, розробленим Mozilla Foundation та її філією Mozilla Corporation.

У 2017 році у Firefox була впроваджена нова технологія, відома як Quantum, яка спрямована на розвиток паралельності та створення більш інтуїтивного інтерфейсу користувача. Браузер підтримується на операційних системах Windows 7 та новіших, macOS та Linux. Також існують неофіційні порти для різних Unix-подібних систем, таких як FreeBSD, OpenBSD, NetBSD, illumos, Solaris Unix, а також версії для мобільних платформ Android та iOS. З серед функціональностей можна виділити:

розширений набір інструментів для зручного та безпечного перегляду Інтернету. Однією з ключових особливостей є можливість встановлення різноманітних розширень, які дозволяють користувачам налаштовувати браузер під свої потреби.

Нова технологія Quantum значно покращила продуктивність та швидкість браузера, забезпечуючи більш ефективне використання ресурсів комп'ютера. Безпека також знаходиться на високому рівні завдяки вбудованому захисту від шкідливих програм та можливості приватного перегляду.

Firefox підтримує сучасні веб-стандарты і забезпечує користувачам стабільний та надійний інтернет-досвід. Багатофункціональність браузера дозволяє ефективно взаємодіяти з великою кількістю вкладок, використовувати закладки для зручного доступу до улюблених сайтів і використовувати різноманітні інструменти для налаштування інтерфейсу.

Firefox продовжує зберігати свою популярність серед користувачів, надаючи їм зручний, швидкий та безпечний інтернет-браузер, який можна налаштовувати під особисті потреби та вподобання.

Програмне забезпечення, яке розповсюджується безкоштовно, і відповідно, його вихідний код доступний для огляду громадськістю. Це дозволяє будь-кому, хто бажає, аналізувати код на предмет можливих вразливостей у системі безпеки. Крім цього, у Mozilla існує система компенсації за виявлення помилок у безпеці: кожна реальна критична помилка, про яку повідомляється, винагороджується сумою у \$3000 за кожне сповіщення, а також футболкою Mozilla. Це сприяє підвищенню рівня довіри до програмного продукту, адже відкритий доступ до вихідного коду створює можливість для спільної перевірки та вдосконалення безпекових стандартів. Учасники, які знаходять та повідомляють про потенційні ризики, не лише сприяють загальній безпеці, але й мають змогу матеріально винагороджуватися за свій внесок.

Цей підхід до безпеки створює відчуття відкритості та співпраці, а також підкреслює важливість спільних зусиль у забезпеченні цифрової безпеки. Такий відкритий підхід допомагає забезпечити надійність програми та заохочує спільноту до активної участі в створенні безпечного та надійного програмного забезпечення

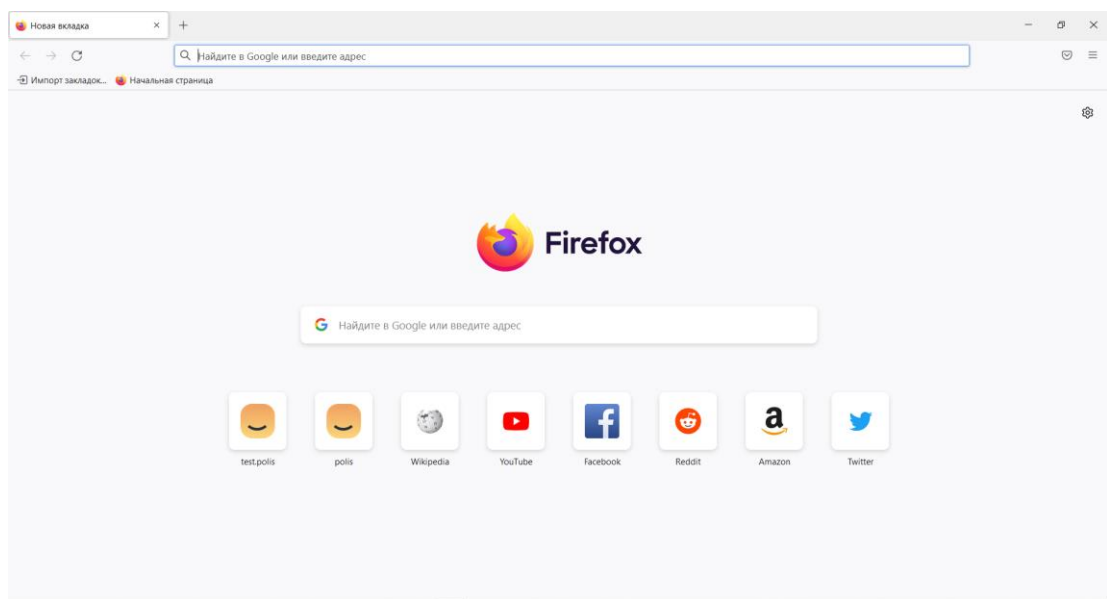


Рис. 1.2. Головна сторінка браузера «Mozilla Firefox».

Порівняльна таблиця веб-браузерів

Суть дослідження	1'st	Last
Швидкодія: холодний старт(одразу після ввімкнення комп'ютера)		
Швидкодія: старт після використання комп'ютерної системи		
Швидкодія: загрузка сторінок без кешу		
Швидкодія: загрузка сторінок з кешем		
Використання оперативної пам'яті: одна відкрита вкладка		
Використання оперативної пам'яті: 10 відкритих вкладок		
Відповідність стандартам: HTML5		
Відповідність стандартам: CSS3		
Продуктивність JavaScript: SunSpider benchmark		
Продуктивність JavaScript: Dromaeo benchmark		
Продуктивність JavaScript: Speed-Battle Benchmark		
Продуктивність JavaScript: Peacekeeper Benchmark		
Продуктивність JavaScript: Aggregate Benchmark		



### **1.3 Front-end частина. Підбір актуальних технологій на основі вимог поставленої задачі**

Мова розмітки гіпертекстових документів, або HTML (від англійської HyperText Markup Language), є текстовою мовою розмітки, заснованою на SGML (загальній мові розмітки). Головна мета HTML - це маркування документів, які містять текст, зображення, гіперпосилання і інші елементи. HTML-документи є основою Всесвітньої павутини і відображаються за допомогою веб-браузерів.

Крім відображуваної інформації, HTML-документи включають додаткові метадані, такі як мова тексту, автор документа та краткий опис. Розробка мови розмітки була поруч з консорціумом W3C, і остання версія, 4.01, призначена бути заміненою розширеною мовою розмітки гіпертексту, відомою як XHTML.

Дані у форматі HTML схожі на звичайні текстові файли, за винятком того, що деякі символи в них, так звані теги, інтерпретуються як розмітка. Розмітка надає документу визначену тегами структуру, таку як параграфи, розділи, абзаци, списки, малюнки, таблиці, колонтитули, індекси, зміст тощо. У межах кожного блоку можна змінювати шрифт, розмір символів, колір тексту, виділяти текст курсивом чи робити його напівжирним. Однією з ключових особливостей HTML є здатність використовувати гіперзв'язки, завдяки яким можна робити посилання та переходи з поточної веб-сторінки на інші документи, як локальні (документи поточного сервера), так і ті, що розташовані на серверах у найвіддаленіших регіонах світу. Мова HTML також дозволяє вставляти в документи зображення, звук, відео і інші мультимедійні елементи. Відображення HTML-документу здійснюється за допомогою веб-оглядача, такого як Internet Explorer, Netscape та інші.

Початкова мова розмітки HTML розроблялася за стандартом SGML, але зараз все частіше використовується XHTML, яка, в основному, аналогічна звичайному HTML, за винятком деяких синтаксичних відмінностей, введених стандартом XML.

HTML відкриває широкі можливості для створення веб-сторінок та забезпечення їх взаємодії з користувачем. За допомогою HTML можна не лише

створювати текстовий контент, але і вбудовувати зображення, відео, аудіо, форми для збору даних, гіперпосилання для навігації між сторінками та багато іншого.

Технологія HTML продовжує розвиватися, із новими версіями, які вносять покращення та доповнення для оптимізації веб-розробки. Наприклад, HTML5 впроваджує нові елементи, які полегшують роботу з мультимедійним контентом, графікою та анімацією.

Використання стандартів HTML забезпечує сумісність веб-сайтів із різними браузерами, забезпечуючи зручний та однаковий перегляд для всіх користувачів. Також, HTML іграє ключову роль у впровадженні принципів доступності, забезпечуючи зручний доступ до веб-контенту для всіх користувачів, включаючи людей із різними видами обмежень.

Завдяки HTML, веб-розробники можуть ефективно створювати різноманітні та інтерактивні веб-застосунки, що відкриває шлях для нових можливостей та ідей у сфері розробки веб-контенту.

CSS, що визначає каскадні таблиці стилів, представляє собою спеціалізовану мову, яка використовується для визначення вигляду документів, написаних мовами розмітки даних, такими як HTML, XHTML та XML. Головна перевага використання CSS полягає в можливості відокремлення вмісту сторінки від її форматування. Це дозволяє поліпшити сприйняття та доступність контенту, забезпечуючи більшу гнучкість та контроль над відображенням в різних умовах.

CSS забезпечує можливість відображення одного документа в різних стилях та створення різних дизайнів сторінки для різних пристроїв. Наприклад, дизайн для екрану може бути розроблений для великої ширини, при цьому друкування не включає меню, а на смартфоні меню розташоване внизу, під контентом.

Використання CSS також призводить до скорочення часу завантаження сторінок сайту за рахунок виокремлення правил відображення в окремий CSS-файл. Браузер завантажує лише структуру документа та збережені на сторінці дані, один раз завантажуючи та кешуючи стильові правила.

Однією з основних переваг використання CSS є також легкість зміни дизайну, оскільки необхідно внести зміни лише в CSS-файл, не редагуючи кожен окрему сторінку.

CSS є невід'ємною складовою розробки веб-сторінок і грає важливу роль у покращенні користувацького досвіду. Додатково до раніше вказаних переваг використання CSS, важливо відзначити його роль у реалізації адаптивного дизайну. Адаптивний дизайн забезпечує оптимальний вигляд веб-сторінки на різних пристроях та розмірах екрану, враховуючи їхні особливості.

CSS пропонує медіа-запити, які дозволяють визначати стилі для конкретних умов відображення, таких як ширина екрану, розміри пристрою чи орієнтація. Це дозволяє створювати веб-сайти, які ефективно працюють як на настільних комп'ютерах, так і на мобільних пристроях, покращуючи універсальність та зручність використання.

Окрім того, CSS використовується для анімації та створення переходів між станами елементів веб-сторінки. Це дозволяє розробникам створювати динамічні та привабливі візуальні ефекти, покращуючи враження від взаємодії з веб-сайтом.

Узагальнюючи, CSS є потужним інструментом, який не лише полегшує форматування веб-документів, але й сприяє створенню високоякісних та адаптивних інтерфейсів для користувачів.

Додаткові Опції Стилю. Наприклад, за допомогою CSS-форматування можна налаштувати відображення меню завжди при прокручуванні сторінки, або вилучити підкреслення в посиланнях. Це дозволяє створювати складний та деталізований дизайн.

Правила в CSS працюють за принципом каскаду (порядку, важливості). Це дозволяє отримати передбачуваний результат, коли до одного елемента одночасно застосовуються кілька стилів.

Всі CSS-норми складаються із селектора та блоку декларацій (вкладеного у фігурні дужки). У середині блоку декларацій може знаходитися одне або кілька визначень, розділених крапкою з комою. Визначення - це лінія, складена з css-атрибуту та його значення.

Кожне правило розпочинається з селектора (індикатора), який вказує на ті html-елементи, до яких ми будемо застосовувати css-правило. У блоку декларацій відбувається найцікавіше - ми встановлюємо правила відображення вибраних нами елементів, визначаємо їх атрибути - розмір, колір, контури, відступи, положення на екрані і так далі.

Розширені Можливості Стилюзації. Прикладом може служити використання CSS-розмітки для того, щоб меню завжди лишалося видимим при прокручуванні сторінки або прибрати підкреслення з посилань. Це відкриває двері для створення складних та вдосконалених дизайнерських рішень.

Механізми CSS працюють в режимі каскаду, де порядок та важливість визначають результат. Цей підхід дозволяє отримувати очікуваний стиль, навіть якщо до одного елемента застосовано кілька стилів одночасно.

Усі правила CSS складаються з селектора та блоку оголошень, який включається в фігурні дужки. В межах блоку оголошень може знаходитися одне чи кілька оголошень, розділених крапкою з комою. Оголошення – це стрічка, що включає css-властивість та її значення.

Кожне правило розпочинається з селектора, який вказує на ті html-елементи, до яких застосовуються css-правила. У блоку оголошень відбувається цікавий процес - ми встановлюємо правила відображення обраних нами елементів, визначаємо їх характеристики, такі як розмір, колір, форма, відступи, положення на екрані і багато іншого.

Загалом, використання CSS надає не тільки зручності при форматуванні веб-документів, а й створює можливості для створення якісних та адаптивних інтерфейсів для користувачів.

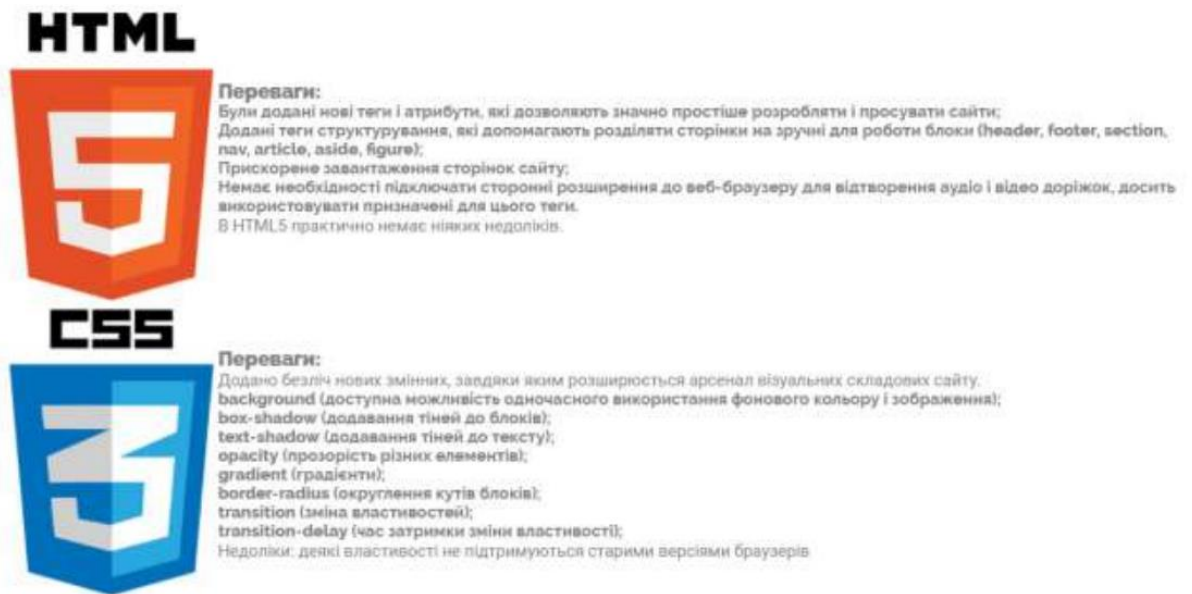


Рис. 1.3. Переваги технологій HTML5 та CSS3

**JAVA SCRIPT** - мова програмування, яка дозволяє впроваджувати складні рішення в web-документах. Вона сприяє зробленню веб-сторінок більш інтерактивними та обробляє дії користувачів. Це об'єктно-орієнтована мова для клієнтського застосування, яка взаємодіє з дизайном сайту. JavaScript отримав додаткову популярність серед розробників після виникнення AJAX-технології, що привело до нової етапу розвитку веб-сайтів. JavaScript, часто абревіюється як JS, представляє собою потужний інструмент у світі веб-розробки. Ця мова програмування дозволяє втілити різноманітні складні рішення в web-документах, надаючи можливість створювати вражаючі, високоінтерактивні веб-сторінки та ефективно обробляти дії користувачів. JavaScript використовується як об'єктно-орієнтована мова для клієнтської розробки та активно взаємодіє з дизайном сайтів. Після виникнення AJAX-технології, JavaScript отримав ще більше визнання, відкриваючи нові можливості у веб-розробці.

Переваги JavaScript:

1. Швидкість. JS часто виконується безпосередньо в браузері клієнта, що робить його дуже швидким. Без зовнішніх ресурсів він не уповільнюється через виклики до сервера. Також, всі основні браузери підтримують JIT-компіляцію для JS, що усуває необхідність у попередній компіляції коду.

2. Простота. Синтаксис JavaScript був взятий за основу у Java і він порівняно простий для вивчення у порівнянні з іншими популярними мовами, такими як C++.

3. Популярність. JavaScript практично необхідний в Інтернеті, а впровадження технології Node.js робить його популярним і на серверному рівні. Багато ресурсів присвячено вивченню цієї мови, і кількість проектів, які використовують JavaScript, постійно зростає. 4. Сумісність - На відміну від PHP та інших мов сценаріїв, JavaScript можна впроваджувати в будь-яку веб-сторінку. JS можна використовувати в різноманітних програмах завдяки підтримці інших мов, таких як Pearl та PHP.

5. Ефективність. JavaScript працює в браузері клієнта, що робить його дуже швидким. Він не сповільнюється через виклики до сервера, особливо при відсутності зовнішніх ресурсів. Техніка JIT-компіляції дозволяє виконувати код без попередньої компіляції.

6. Легкість навчання. Синтаксис JavaScript був вплинутий Java і відзначається відносною легкістю вивчення порівняно з іншими мовами, такими як C++.

7. Універсальність. JavaScript можна впроваджувати в будь-яку веб-сторінку, що надає йому високий рівень сумісності. Також, завдяки підтримці інших мов, включаючи Pearl та PHP, вона знаходить застосування в різних типах програм.

JavaScript залишається ключовим інструментом у розробці веб-додатків, привносячи в них динаміку та інтерактивність.



#### Переваги JavaScript

- Жоден сучасний браузер не обходиться без підтримки JavaScript.
  - З використанням написаних на JavaScript плагінів та скриптів упорається навіть не фахівець.
- Корисні функціональні налаштування.
- Мова, що постійно вдосконалюється - зараз розробляється бета-варіація проекту, JavaScript2.
  - Взаємодія з програмою може здійснюватися навіть через текстові редактори – Microsoft Office та Open Office.
  - Перспектива використання мови у процесі навчання програмування та інформатики.

Рис. 1.4. Переваги технології JavaScript

## 1.4 Back-end частина. Підбір актуальних технологій на основі вимог поставленої задачі

Spring представляє собою фреймворк рівня для розробки додатків на мові Java/J2EE, заснований на вихідних кодах, які були опубліковані у книзі "Expert One-on-One J2EE Design and Development" Джонсона Родома.

Які завдання виконує і що пропонує Spring? Цей фреймворк надає значну функціональність. Яка ж мета призначення Spring?

Спершу слід визначити обсяг дій, які охоплює Spring. Навіть якщо горизонт Spring досить широкий, важливо розуміти, на що саме він впливає та що залишається за його межами.

Основною метою Spring є спрощення використання J2EE та поширення добрих практик програмування. Це досягається за допомогою реалізації моделі програмування на основі POJO, яка застосовна у різноманітних середовищах.

Spring не створює щось абсолютно нове. Отже, у Spring ви не знайдете пакетів для ведення журналів, пулів з'єднань або координатора розподілених транзакцій. Усі ці аспекти забезпечуються відкритими проектами (наприклад, Commons Logging або Commons DBCP) або серверами застосувань. З тієї самої причини рівень відображення об'єктів (O/R mapping) також не входить у функціонал Spring. Для цієї задачі можна використовувати TopLink, Hibernate або JDO.

Spring не має на меті полегшити використання існуючих технологій. Наприклад, хоча він не охоплює координацію низькорівневих транзакцій, він надає абстракцію на рівні JTA або будь-якої іншої транзакційної стратегії.

Spring не конкурує відкрито з іншими проектами до того часу, поки він не може запропонувати щось нове. У деяких областях, таких як контейнер IoC та AOP фреймворк, в Spring є пряма конкуренція, але варто відзначити, що Spring був піонером у цих сферах.

В розробці програмного забезпечення на Java, фреймворк Spring є однією з ключових технологій, що забезпечує багатоцільові можливості та допомагає вирішувати різноманітні завдання. Давайте розглянемо деякі нові аспекти Spring:

1. Модулі та розширення Spring: Spring має ряд модулів, які розширюють його базовий функціонал. Наприклад, Spring Security забезпечує механізми автентифікації та авторизації, а Spring Boot дозволяє швидко створювати самостійні, готові до використання додатки.

2. Інтеграція з іншими технологіями: Spring легко інтегрується з іншими популярними технологіями. Наприклад, взаємодія з базами даних здійснюється через Spring Data, а для роботи з веб-технологіями використовується Spring MVC.

3. Мікросервісна архітектура: Spring ідеально підходить для створення мікросервісних додатків. За допомогою Spring Cloud та інших інструментів, ви можете легко реалізувати розподілені системи, забезпечуючи масштабованість та гнучкість.

4. Спрощення конфігурації: Введення анотацій та JavaConfig у Spring дозволяє спростити конфігурацію додатків. Це дозволяє розробникам зосередитися на бізнес-логіці, зменшуючи кількість необхідних XML-конфігурацій.

5. Підтримка реактивного програмування: Spring включає в себе підтримку реактивного програмування з використанням Project Reactor. Це дозволяє ефективно взаємодіяти з великим обсягом даних та забезпечує високу швидкодію додатків.

Ці аспекти вказують на те, що Spring продовжує розвиватися, адаптуючись до нових викликів та надаючи розробникам ефективні інструменти для розробки сучасних додатків.

Spring володіє перевагами внутрішньої стабільності. Всі програмісти ділять однаковий підхід, адже основні принципи беруться з "Expert One-on-One J2EE Design and Development" Джона Родсона. Основні концепції, такі як IoC, використовуються в різних контекстах.

Spring є переносним між серверами застосувань. Зрозуміло, що переносимість завжди супроводжується труднощами і ризиками, але розробники Spring стараються уникати платформено-залежного або нестандартного підходу, забезпечуючи підтримку для WebLogic, Tomcat, Resin, JBoss, Jetty, Geronimo, WebSphere та інших серверів застосувань. Підхід на основі POJO дозволяє використовувати особливості конкретного середовища, не втрачаючи при цьому переносимості.



Переваги Spring, за словами Рода Джонсона, полягають в його унікальності з кількох причин:

1. Spring охоплює багато сфер, які інші популярні фреймворки не торкаються, фокусуючись на можливостях управління бізнес-об'єктами.

2. Він одночасно рівневий і модульний, дозволяючи використовувати його частини ізольовано або в єдності, забезпечуючи внутрішню послідовність.

3. Spring створений для полегшення написання тестового коду, що робить його ідеальним для test-driven проектів.

4. Як інтеграційна технологія, Spring набуває популярності через визнання його ролі відомими вендорами.



Рис. 1.5. Переваги технології Spring

Отже, Spring є потенційним універсальним інструментарієм, який охоплює більшість аспектів інфраструктури типових додатків.

Однією з суттєвих переваг Spring є його активна спільнота та велика кількість ресурсів для навчання. Розробники можуть легко знаходити підтримку, допомогу та матеріали для вивчення фреймворку, що значно полегшує процес впровадження та розвитку проектів на основі Spring.

Крім того, Spring Framework постійно оновлюється, враховуючи сучасні тенденції та вимоги ринку розробки програмного забезпечення. Такі оновлення включають в себе вдосконалення ефективності, додавання нових можливостей та інтеграцію з іншими передовими технологіями.

Багатофункціональність Spring також дозволяє розробникам ефективно працювати з різними аспектами проєктів, такими як робота з базами даних, взаємодія зі сторонніми службами та забезпечення безпеки додатків. Це робить Spring ідеальним вибором для великих і складних систем.

Нарешті, розширюваність Spring дає можливість розширювати функціональність додатків, додаючи або модифікуючи компоненти фреймворку. Це дозволяє розробникам створювати налагоджувані та індивідуальні рішення для конкретних потреб їх проєктів.

## **Docker**

Docker був представлений в 2013 році і став швидко заслуженим лідером в індустрії контейнеризації. Однак коріння Docker можна відстежити до попередніх технологій та концепцій, таких як LXC (Linux Containers) і FreeBSD Jails, які надавали віртуалізацію на рівні операційної системи.

Прорив Docker полягав у тому, що він забезпечив простий та однорідний інтерфейс для створення, розгортання та управління контейнерами. Він використовує технологію контейнеризації, яка дозволяє ізолювати та виконувати програми разом із всіма їх залежностями та конфігураціями.

Задачі Docker:

1. Контейнеризація. Docker забезпечує легку та ефективну ізоляцію додатків та їх залежностей в контейнерах. Це дозволяє розробникам створювати, переносити та розгортати додатки на різних середовищах без змін у коді.

2. Портативність. Оскільки контейнери містять усі необхідні ресурси для виконання додатків, вони можуть працювати консистентно на будь-якій системі або хості, де встановлено Docker.

3. Швидкість та Легкість. Контейнери запускаються швидше та вимагають менше ресурсів порівняно з традиційними віртуальними машинами. Docker дозволяє ефективно використовувати ресурси хост-системи.

4. Масштабованість. Docker дозволяє легко масштабувати додатки, запускаючи їх у контейнерах та оркеструючи їх за допомогою інструментів, таких як Docker Compose та Kubernetes.

5. Інтеграція та Автоматизація. Docker інтегрується з багатьма іншими інструментами для автоматизації процесів розробки, тестування та розгортання, що сприяє покращенню швидкості та ефективності розробки.

За допомогою Docker розробники отримали потужний інструмент для створення, розгортання та управління додатками, що прискорило розробку програмного забезпечення та полегшило роботу в інтерфейсах розгортання на різних платформах.

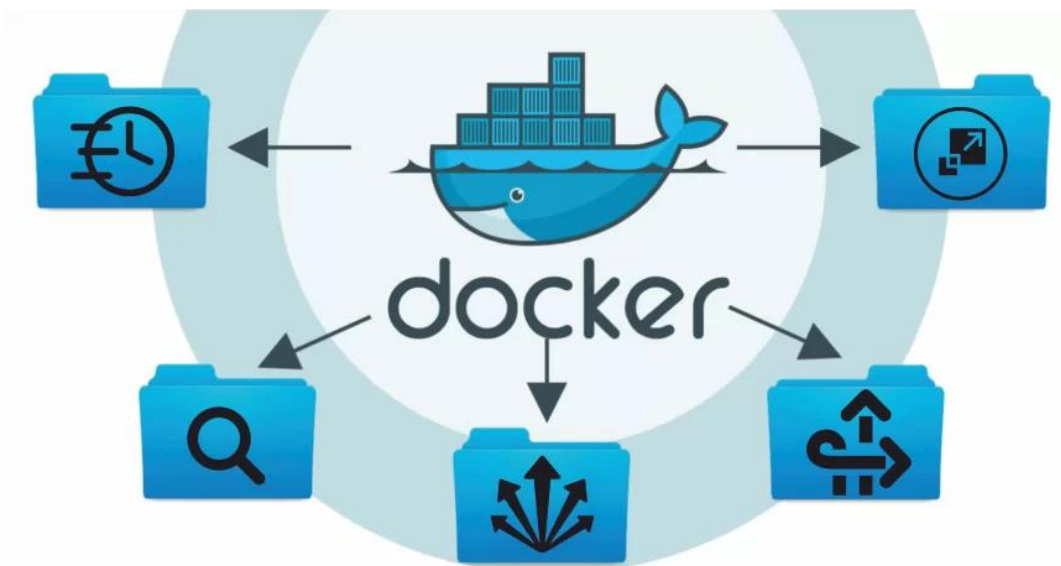


Рис. 1.6. Спрощене ілюстрація роботи Docker

### **Rabbit Mq**

RabbitMQ — це розподілений брокер повідомлень, який вперше був представлений в 2007 році компанією Rabbit Technologies Ltd. Його розробка була розпочата Джоном О'Харо та Алексом Хіндмаршем. У 2010 році компанію було придбано VMware, а пізніше у 2013 — компанією Pivotal Software.

Основними ідеями розробки RabbitMQ були стандартизація інтерфейсів для спрощення комунікації між різними компонентами системи та забезпечення масштабованості систем обробки повідомлень. Розробники мали на меті створити ефективний, легкий у використанні та розширюваний брокер повідомлень.

Основні задачі RabbitMQ:

1. Посередництво повідомлень. RabbitMQ використовує патерн "посередник" для обміну повідомленнями між різними частинами системи. Він служить як надійний канал для обміну інформацією між компонентами додатка.

2. Маршрутизація повідомлень. Брокер може маршрутизувати повідомлення до відповідних компонентів системи в залежності від правил конфігурації. Це дозволяє ефективно розподіляти та обробляти повідомлення.

3. Гарантована доставка. RabbitMQ забезпечує механізми для гарантованої доставки повідомлень, що робить його надійним інструментом для взаємодії між різними компонентами систем.

4. Масштабованість. Брокер може бути легко масштабований, дозволяючи обробляти великий обсяг повідомлень та підтримувати потреби розширюваності систем.

5. Підтримка різних протоколів. RabbitMQ підтримує різні протоколи для взаємодії, включаючи AMQP (Advanced Message Queuing Protocol), що робить його універсальним рішенням для різних застосувань.

6. Інтеграція. Завдяки широкому спектру бібліотек та інструментів, RabbitMQ легко інтегрується з різноманітним програмним забезпеченням та мовами програмування.

За своєю природою RabbitMQ є потужним та гнучким брокером повідомлень, який допомагає створювати розподілені системи з ефективними механізмами обміну даними між компонентами.

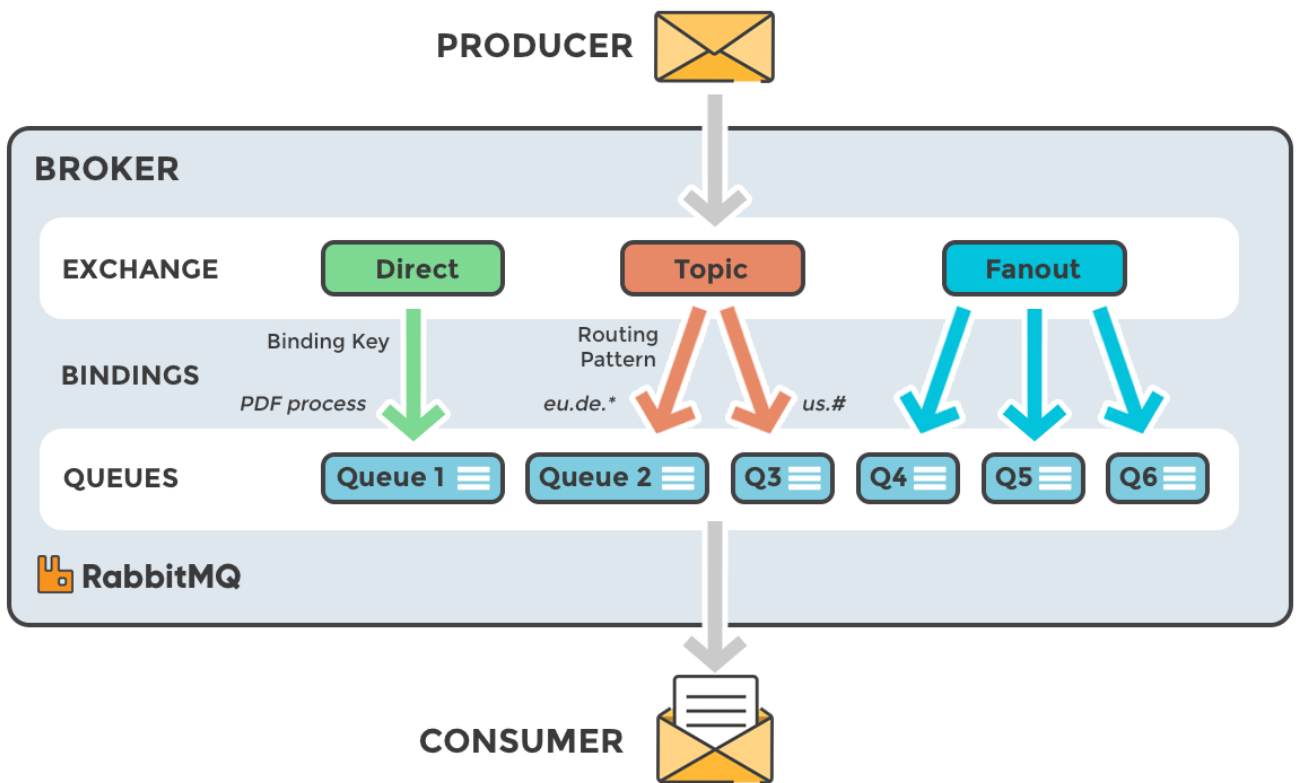


Рис. 1.7. Спрощене ілюстрація роботи Rabbit Mq

## Kubernetes

Kubernetes, часто називаний "K8s", з'явився на світ як відкрите програмне забезпечення для автоматизації, розгортання та керування контейнеризованими додатками. Його історія почалася в компанії Google, коли вони стали зіткнутися з великими та складними системами, які вимагали ефективного управління.

В 2014 році Google представив свій внутрішній інструмент Borg, який вони використовували для керування своїми контейнерами. Пізніше того ж року Google випустив Kubernetes як відкрите програмне забезпечення, надаючи йому універсальність та гнучкість для спільного використання.

Проект швидко отримав підтримку від широкого співтовариства і став частиною Cloud Native Computing Foundation (CNCF), забезпечуючи йому нейтральність та відкритий розвиток. З часом Kubernetes став стандартом для оркестрації контейнерів та знаходить застосування в різних індустріях та компаніях.

Задачі Kubernetes:

1. Оркестрація контейнерів. Kubernetes дозволяє розгорнути та керувати великою кількістю контейнерів, забезпечуючи автоматичне розподілення ресурсів та балансування навантаження.

2. Автоматизація розгортання. Kubernetes автоматизує процес розгортання додатків, забезпечуючи швидкі та стандартизовані процеси в усіх середовищах.

3. Масштабування. Kubernetes дозволяє легко масштабувати додатки, як вгору, так і вниз, в залежності від потреб навантаження.

4. Самовідновлення. Платформа має вбудовані механізми для самовідновлення, які автоматично відновлюють працездатність системи в разі відмови чи помилок.

5. Моніторинг та логування. Kubernetes забезпечує інтеграцію з інструментами моніторингу та збору логів для спрощення відстеження та аналізу роботи додатків.

6. Декларативне конфігурування. Конфігурація Kubernetes відбувається через декларативні файли, що полегшує автоматизацію та контроль версій.

7. Швидка реакція на зміни. Kubernetes дозволяє автоматично адаптуватися до змін у середовищі, розгортання нових версій додатків та оновлення конфігурації.

Завдяки своїм функціональним можливостям та відкритому характеру, Kubernetes став країною технологією для розгортання та керування сучасними додатками в умовах хмарної інфраструктури та контейнеризації.

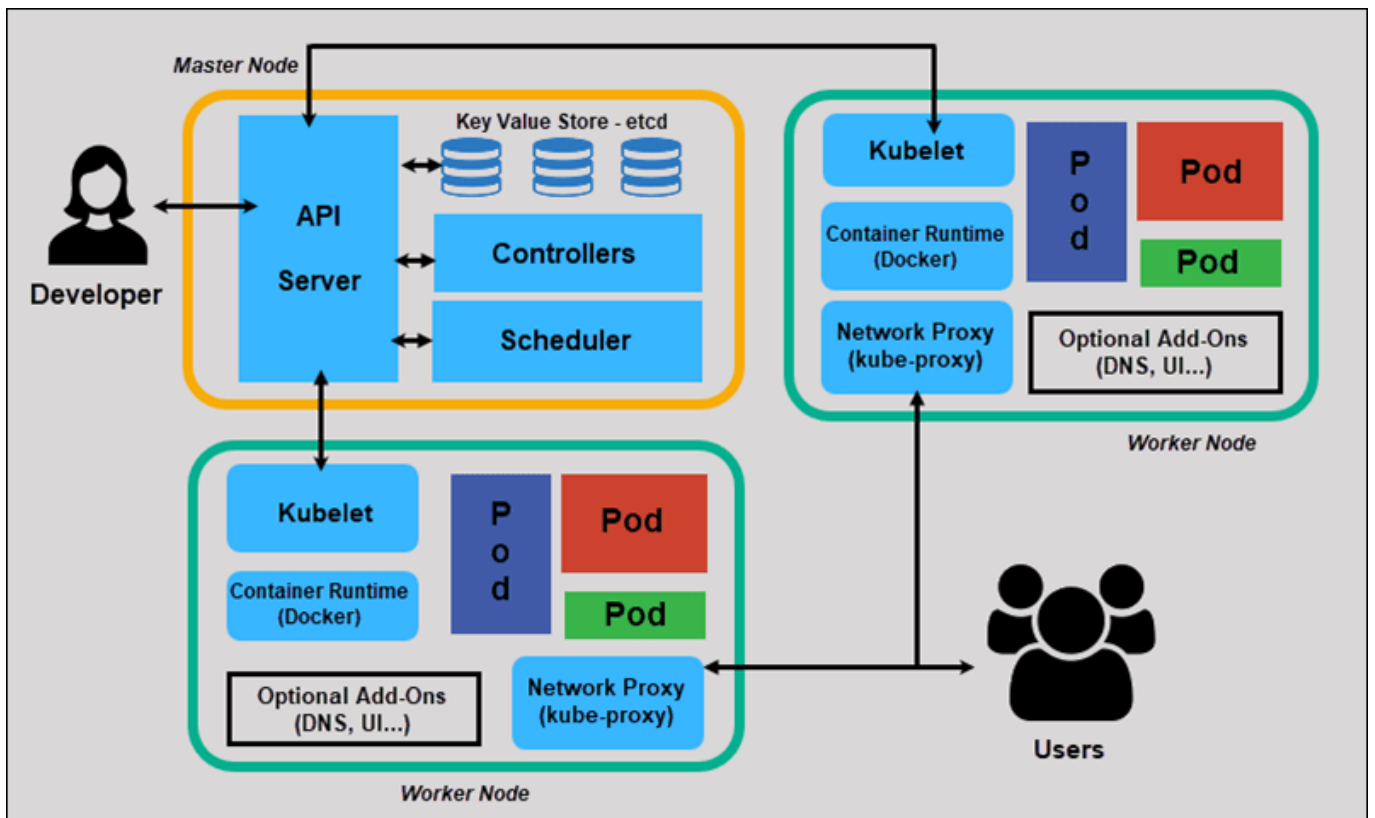


Рис. 1.8. Спрощене ілюстрація архітектури Kubernetes

## 1.5. Психологічний аналіз користувачів інтернет магазинів

Аналіз статистики свідчить про те, що зростає кількість осіб, які використовують Інтернет-технології, і це робить розуміння психологічної поведінки користувачів у сфері інтернет-торгівлі надзвичайно важливим.

Зазвичай клієнти, які витрачають більше часу в соціальних мережах, схильні до необдуманих та імпульсивних покупок. Такі користувачі володіють високим рівнем споживчої активності та збільшеною покупною здатністю.

Підпалювана постійними акціями та розпродажами, імпульсивність людей є важливим фактором. Підвищення загальної вартості кошика покупця збільшує ймовірність нових імпульсивних придбань.



Рис. 1.9. Яскрава вивіска одразу привертає увагу

Великий вплив на покупну здатність та необдумані покупки має дизайн веб-сайту. Чим естетичніше, яскравіше та простіше виглядає товар, тим більше ймовірностей для клієнтів зробити покупку.

Маркетингові прийоми, такі як фокусування уваги клієнта на товарі та вдалі акційні пропозиції, також відіграють важливу роль.

Зазвичай середній клієнт приймає рішення про покупку за 90 секунд. Отже, швидкодія сайту є ключовим фактором, оскільки якщо сторінка завантажується понад три секунди, 57% користувачів перейдуть на сайт конкурента, а 80% з них може не повернутися.

Додатковий аспект, на який слід звернути увагу, це вплив оцінок та відгуків користувачів на рішення про покупку. Сучасні покупці активно вивчають відгуки і рейтинги товарів перед тим, як зробити вибір. Таким чином, для інтернет-магазинів важливо створювати сприятливий механізм для збору та відображення відгуків клієнтів.

Окрім цього, персоналізація контенту на веб-сайті грає значущу роль у залученні користувачів. Використання алгоритмів рекомендацій та персоналізованих пропозицій може значно підвищити ефективність конверсії та задоволення покупців.

Необхідно також враховувати мобільність та адаптивність веб-сайту, оскільки все більше користувачів здійснюють покупки через мобільні пристрої. Забезпечення оптимального відображення та зручного користування на різних пристроях є ключовим аспектом для збільшення конверсії.



Таким чином, враховуючи ці додаткові фактори, інтернет-магазини можуть створити сприятливі умови для залучення та утримання клієнтів, забезпечуючи при цьому позитивний досвід покупок та підвищуючи конкурентоспроможність на ринку.

Для поліпшення обсягів продажів важливо уважно дотримуватися наступних технічних рекомендацій. Ці вказівки залишать у покупця приємне враження, що незаперечно призведе до повторного відвідування даного веб-сайту:

### **user-friendly**

Оптимізація користувацького досвіду включає в себе не лише простоту використання, але й дотримання принципів ефективно організації контенту. Наприклад, важливо розташовувати ключову інформацію на легкодоступних місцях та забезпечувати зрозумілу структуру меню. Додавання інтуїтивно зрозумілих підказок та ілюстрацій також може сприяти зручності взаємодії з сайтом.

Крім того, важливо враховувати функціональність адаптивного дизайну для оптимального відображення на різних пристроях. Забезпечення швидкої реакції інтерфейсу на різних платформах допоможе підтримувати позитивний враження користувачів.

Необхідно також слідкувати за актуальністю та своєчасністю інформації на сайті, щоб користувачі завжди мали доступ до актуальних даних. Це важливо для збереження довіри та стимулювання повторних візитів.

Таким чином, вдосконалення користувацького інтерфейсу — це комплексний підхід, який охоплює якість взаємодії, доступність і структуру сайту, сприяючи задоволенню користувачів та підвищенню ймовірності їхнього повторного відвідування.

### **Ефективність веб-сайту**

Згідно з аналітичними оцінками, видно, що швидкість доставки інформації до кінцевого користувача є ключовим фактором, який визначає 80% успішних транзакцій та залучення користувача на постійній основі.

Крім швидкості та психологічних хитрощів, значущою роллю в успіху веб-сайту є його адаптованість під різні пристрої. Врахування мобільних та планшетних платформ стає необхідністю, оскільки більшість користувачів переглядає веб-сайти

на різних пристроях. Оптимізація сайту під різні екрани забезпечить комфортне взаємодію з контентом та збільшить ймовірність здійснення покупки.

Додатково, важливо вдосконалювати процес оформлення замовлення та спростувати його кроки. Мінімізація форм, яка включає в себе лише необхідні поля для заповнення, може значно полегшити процес для користувачів, збільшуючи ймовірність завершення покупки.

Також важливо не забувати про психологічні методи, які можна впровадити на веб-сайті для підвищення ефективності продажу продукції:

### **Необхідність товару**

Інформація про те, що товар, який переглядається, перебуває в "останніх екземплярах" - це типовий сценарій. Також можна вживати вислови типу "залишилося лише дві (або три) одиниці товару". Як не поспішити з прийняттям рішення в такій ситуації? Очевидно, обраний товар є дуже популярним, тому купівля варта уваги!

Удосконалення тактик маркетингу може включати в себе інші методи створення враження нагальності придбання товару. Наприклад, використання термінів "ексклюзивний товар", "обмежена серія" чи "спеціальна пропозиція на короткий термін" може сприяти формуванню у покупців враження того, що їм слід прийняти рішення швидше, оскільки пропозиція обмежена часовим фреймом або доступна лише обмеженій кількості осіб.

### **Терміновий Аспект**

Інтернет-магазини можуть поспішати покупця, використовуючи більш настирливий метод - встановлення таймера. Таймер зворотного відліку підсвічує потенційному покупцеві, що строк дії знижки скоро закінчиться. У цей момент можуть виникнути перед покупкою лише непереможні обставини.

Ще однією ефективною стратегією для створення враження терміновості є використання фраз типу "близько до завершення акції" чи "останній шанс отримати знижку". Це може викликати у покупця відчуття невідкладності та посилити його

мотивацію прийняти рішення щодо покупки. Такі техніки акцентують увагу на обмеженості пропозиції та сприяють створенню емоційного зв'язку з покупцем.

### **Сортування**

Жоден із зазначених магазинів ніколи за замовчуванням не впорядковував асортимент товарів за ціною – ані в порядку зростання, ані в порядку спадання. Зокрема, деякі з них настільки тщюно ховають опцію сортування, що знайти її стає завданням для людей з міцними нервами. Замість цього покупцям пропонується перегляд варіанту сортування за принципом "спочатку – найпопулярніші". Однак вивчення "популярних товарів" породжує думку, що саме вони – найбажаніші і найдорожчі, які інтернет-магазин старається продати перш за все.

Під час вивчення різних інтернет-магазинів можна відзначити, що деякі з них не встановлюють сортування товарів за ціною за замовчуванням – ні за зростанням, ні за спаданням. Навіть більше, деякі магазини так пильно приховують опцію сортування, що вона стає справжнім випробуванням для нервів покупців. Замість цього, користувачам пропонується переглядати товари в порядку "спочатку – найпопулярніші". Проте вивчення категорії "популярні товари" може створювати враження, що саме вони є найбажанішими та найдорожчими товарами, які магазин намагається продати в першу чергу.

### **Топ-товари**

У рамках цієї ж серії ми можемо зазначити список "елітних товарів" або "популярних товарів місяця", які, як вважається, користувачі купують найчастіше. Проте важливо розуміти, що цей перелік може не відображати реальну ситуацію. Так само, як і при сортуванні "за популярністю", у цей список можуть потрапити товари з надмірно завищеною ціною або ті, які потрібно швидше продати. У кращому випадку це буде змішаний список, де насправді популярні товари перемішані з недостатньо автентичними лідерами.

При підборі "елітних товарів" або "популярних товарів місяця" важливо пам'ятати, що цей перелік може бути використаний для стимулювання попиту на певні товари. Проте його створення може бути також стратегічним ходом магазину для підняття продажів і виносу окремих товарів у центр уваги покупців. Такий підхід

може мати свої переваги, але важливо зберігати об'єктивність і не сліпо довіряти лише перелікам "топ-товарів". Покупцям також важливо мати можливість самостійно досліджувати асортимент магазину і обирати товари відповідно до своїх потреб і переваг.

### **Доставка в подарунок**

"Доставка в подарунок" – за статистикою, 70% усіх наповнених корзин залишаються без оплати. При цьому дві третини клієнтів вказують високу вартість доставки, яка виявляється на етапі оформлення замовлення, як головну причину відмови від покупки. Та не дивно, оскільки "безкоштовна доставка" може бути лише ілюзією, спрямованою на відібрання грошей у покупців.

Про те, що дана торгова точка пропонує безкоштовну доставку, вам повідомлять на найвидатнішому місці. Однак під час покупки виявиться, що ця привілея не поширюється на обрані товари. Наприклад, безкоштовна доставка можлива лише до відділення "Нової пошти" або вимагає накопичення замовлення на суму п'ять тисяч гривень.

На шляху вдосконалення відправлення замовлень, важливо розглядати можливості оптимізації вартості доставки. Вивчення альтернативних варіантів, таких як партнерства з логістичними компаніями чи встановлення програм лояльності для умовно безкоштовної доставки при певних умовах, може виявитися стратегічно вигідним. Такий підхід сприяє якісному обслуговуванню клієнтів та створенню привабливих пропозицій, що не лише збільшують конверсію, а й зміцнюють лояльність покупців.

### **1.6. Завдання дослідження**

Результатом завершення дипломного проекту повинен стати інтернет-ресурс для реалізації товарів для дому. Для успішного виконання вимог проекту необхідно пройти через наступні етапи:

- Провести аналіз сучасних технологій, які можуть бути використані для створення інтернет-магазину.

- Здійснити огляд методів розробки проектів та різних способів подання інформації.

- Визначити оптимальну структуру для інтернет-ресурсу.

- Розробити ефективний інтернет-магазин товарів для дому.

### **1.7. Висновок до розділу 1**

Під час розробки першого розділу виконано читання та аналіз літератури, а також розглянуті всі етапи створення інструментів та платформи для ведення електронної комерції. Виконано оцінку доцільності створення інтернет-магазину як у сучасний період, так і в перспективі.

В дослідженні детально проаналізовано актуальні технології, що використовуються для створення веб-магазинів. Розглянуті різні методи розробки проекту та варіанти подання інформації. На основі отриманих даних визначено оптимальну структуру для веб-сайту, що сприятиме ефективному продажу побутових товарів. Проект включає створення інтернет-магазину, орієнтованого на забезпечення зручного та ефективного онлайн-процесу покупок для користувачів.

## РОЗДІЛ 2

# АНАЛІЗ СЕРЕДОВИЩ РОЗРОБКИ ТА КЛЮЧОВІ ЕТАПИ СТВОРЕННЯ WEB-САЙТА

### 2.1. Підбір середовища розробки

Для створення програмного забезпечення виникло значне число текстових редакторів, які сприяють поліпшенню та прискоренню процесів розробки та розуміння проекту за допомогою зручних інструментів, вбудованих у редактор. Наприклад, вони включають синтаксичне підсвічування коду, швидке переміщення між файлами вихідного коду, а також зручні розширення, що можна налаштовувати під власний стиль написання коду та інші корисні функції. З часом текстові редактори для розробників перетворилися в інтегровані середовища розробки, надаючи ще більше функціоналу та зручностей.

Із зростанням складності розробки програмного забезпечення, інструменти для редагування коду також еволюціонували. Нині інтегровані середовища розробки (ІСР) стали стандартом для багатьох програмістів. Ці ІСР поєднують у собі не лише текстові редактори, але й додаткові інструменти, такі як відлагоджувачі, системи контролю версій, менеджери залежностей та інші, що полегшують розробку та забезпечують її більш ефективним процесом. Такий підхід дозволяє розробникам працювати в єдиному середовищі, максимально оптимізованому для їх потреб.

Порівняємо найпопулярніші середовища розробки:

#### **Visual Studio Code**

Крос-платформенність: Visual Studio Code – це редактор, що діє на різних операційних системах, а саме: Windows, Linux та MacOS. Він є вільним та має

Кафедра КІТ (47)				НАУ 23 01 01 000 ПЗ			
Виконав	Березін А.В.			АНАЛІЗ СЕРЕДОВИЩ РОЗРОБКИ ТА КЛЮЧОВІ ЕТАПИ СТВОРЕННЯ WEB-САЙТА	Літера	аркуш	аркуші
Керівник	Моденов Ю.Б.					38	17
Консульт.					<b>УС-211М      122</b>		
Н-контрол.	Райчев І.Е.						

відкритий вихідний код, що означає, що ви можете користуватися ним на будь-якій платформі, яка вам зручна, незалежно від операційної системи вашого пристрою.

**Гнучкість:** Розробники можуть працювати на тій операційній системі, яка їм найбільше відповідає або яку вони вже використовують. Вони не обмежені певною платформою.

**Сумісність команд:** Якщо команда розробників складається з користувачів різних операційних систем, крос-платформенні інструменти дозволяють уникнути проблем, пов'язаних з несумісністю.

**Вартість:** Використання крос-платформенних інструментів може сприяти економії коштів, оскільки не потрібно відділяти ресурси на розробку та підтримку окремих версій для кожної операційної системи.

**Розподілені проекти:** Крос-платформенність зручна для проектів, які мають розподілену команду або розробляються на різних платформах.

Visual Studio Code вражає своєю високою універсальністю, адже він дійсно підтримує широкий спектр мов програмування. У своєму арсеналі цей редактор включає:

## 2. Кросс-language

1. Python: Як одна з найпопулярніших мов програмування, Python отримує повноцінну підтримку у Visual Studio Code.

2. JavaScript та TypeScript: Для фронтенд- та бекенд-розробки ви можете зручно користуватися JavaScript і його сучасним супутником TypeScript.

3. HTML і CSS: Редагуйте веб-сторінки та стилізуйте їх, використовуючи HTML і CSS, отримуючи всі переваги автодоповнення та перевірки помилок.

4. C++ і C#: Розроблюйте швидкі та ефективні десктопні застосунки з підтримкою для мов C++ та C#.

5. Java: Ваш Java-код стає доступним для редагування та управління через Visual Studio Code.

6. PHP і SQL: Розробка веб-застосунків з використанням PHP та обробка баз даних SQL теж є частиною можливостей цього редактора.

7. Ruby і Objective-C: Visual Studio Code дає можливість працювати з розширеними мовами, такими як Ruby та Objective-C.

Це лише кілька мов програмування, які підтримуються у Visual Studio Code. Незалежно від того, чи ви розробляєте веб-додатки, мобільні застосунки чи десктопні програми, цей редактор стане потужним інструментом для вашої продуктивності.

### 3. Динамічна зміна мови файлу

Ця функція дає вам можливість налаштувати мовний режим для конкретного файлу в Visual Studio Code. Ви можете змінювати мову для відкритих файлів в залежності від їхнього вмісту або ваших власних уподобань. Ось кілька ключових аспектів цієї функції:

1. Автоматичне визначення мови за промовчанням: Редактор може автоматично визначити мову для файлу на основі його розширення або вмісту. Це спрощує роботу з файлами різних мов програмування.

2. Зміна мови за бажанням: Ви можете змінити мову вручну, натискаючи на індикатор мови внизу справа у рядку стану. Після цього вам доступний випадаючий список для вибору альтернативної мови.

3. Індивідуальна настройка для кожного файлу: Ви маєте можливість встановлювати різні мови для кожного відкритого файлу, щоб легше працювати з документами різного призначення.

Ця функція особливо корисна, коли ви маєте проекти, які включають файли різних мов, і ви хочете отримати максимальний комфорт при роботі з кожним з них.

### 4. Документація

Visual Studio Code (VS Code) має докладну документацію для різних мов програмування. Документація мов програмування у VS Code зазвичай включає в себе наступне:

1. Синтаксичне підсвічування: VS Code надає можливість виділяти синтаксичні елементи коду кольорами, що полегшує читання та редагування коду.

2. Підтримка мовних розширень: Документація описує, як встановлювати та налаштувати розширення для підтримки конкретної мови програмування.



3. Налаштування оточення розробки: Інструкції з конфігурації VS Code для зручної роботи з конкретною мовою. Це може включати в себе встановлення лінтерів, відладчиків, плагінів та інших інструментів.

4. Використання важливих функцій: Документація пояснює, як використовувати основні функції VS Code для роботи з кодом, такі як автодоповнення, відладка, інтеграція з системами контролю версій тощо.

5. Приклади та підказки: Документація може містити приклади коду, які демонструють правильне використання особливостей мови в VS Code, а також практичні поради щодо розробки.

Для зручності користувачів Visual Studio Code, Microsoft також надає велику кількість розширень для підтримки різних мов програмування, які можна легко встановити з магазину розширень (Extension Marketplace).

#### 5. Налагодження

VSCode має вбудований інструмент для налагодження, що є однією з його основних можливостей. Це сприяє прискоренню циклу редагування, компіляції та налагодження будь-якого розробника. Проте за замовчуванням він оснащений лише відладчиком, який підтримує NodeJS і здатний налагоджувати все, що транслюється в JavaScript. Зазначте, що ви можете розширити його функціональність за допомогою додаткових відладчиків для інших середовищ виконання.

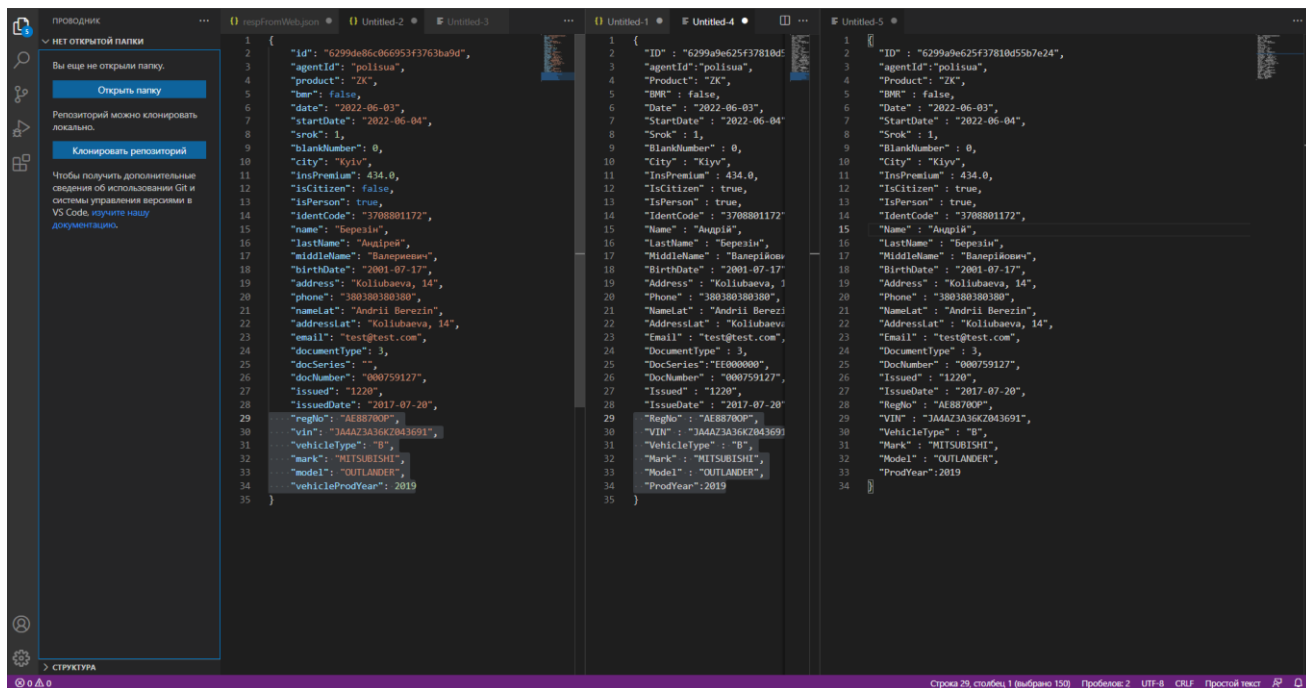


Рис. 2.1. Интерфейс Visual Studio Code

## Sublime Text

Цей редактор є універсальним, простим у використанні та легким для обчислювальної системи. Він підтримує більшість мов програмування та надає синтаксичне підсвічування коду для кожної з цих мов. Ось перелік мов, які може підтримувати Sublime Text: C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, Rust, SQL, TCL та XML. Також серед його можливостей: використання гарячих клавіш, регулярний пошук, що полегшує навігацію по проекту, локальна історія документу, що дозволяє відстежувати локальні зміни у документі - корисна функція при роботі над проектом. Крім того, редактор має такі можливості:

## Package Control

У Sublime Text використовується менеджер пакетів під назвою "Package Control". Package Control дозволяє легко встановлювати, оновлювати та видаляти різноманітні розширення і плагіни для розширення функціональності редактора. Користувачі можуть встановлювати пакети, які підтримують різні мови програмування, забезпечують розширені функції редагування, автодоповнення, синтаксичне підсвічування і багато іншого.

Для використання Package Control користувач повинен встановити його спочатку, а потім може шукати, встановлювати та керувати пакетами за допомогою простого інтерфейсу. Це дозволяє швидко розширювати можливості редактора з урахуванням індивідуальних потреб програміста чи розробника.

### **SublimeCodeIntel**

SublimeCodeIntel - це плагін для Sublime Text, який надає розширені можливості інтелектуального автодоповнення для різних мов програмування. Основна функціональність SublimeCodeIntel включає в себе:

1. Автодоповнення коду: Під час роботи з редактором Sublime Text і введення коду, SublimeCodeIntel пропонує варіанти автодоповнення для класів, методів, функцій та інших елементів мови програмування.

2. Система визначення типів інтелектуального коду: Плагін використовує інтелектуальний аналіз коду для визначення типів та відносин між елементами програмування.

3. Підтримка багатьох мов програмування: SublimeCodeIntel працює для різних мов програмування, включаючи, але не обмежуючись, Python, JavaScript, PHP, HTML, CSS, Ruby, та інші.

4. Перегляд документації: Плагін може надавати доступ до документації для різних функцій та методів під час роботи з кодом.

5. Підтримка різних фреймворків: SublimeCodeIntel може надавати автодоповнення для популярних фреймворків і бібліотек, полегшуючи розробку відповідного коду.

Додавання SublimeCodeIntel до Sublime Text може значно поліпшити продуктивність розробників та забезпечити зручний інструментарій для роботи з різними мовами програмування.

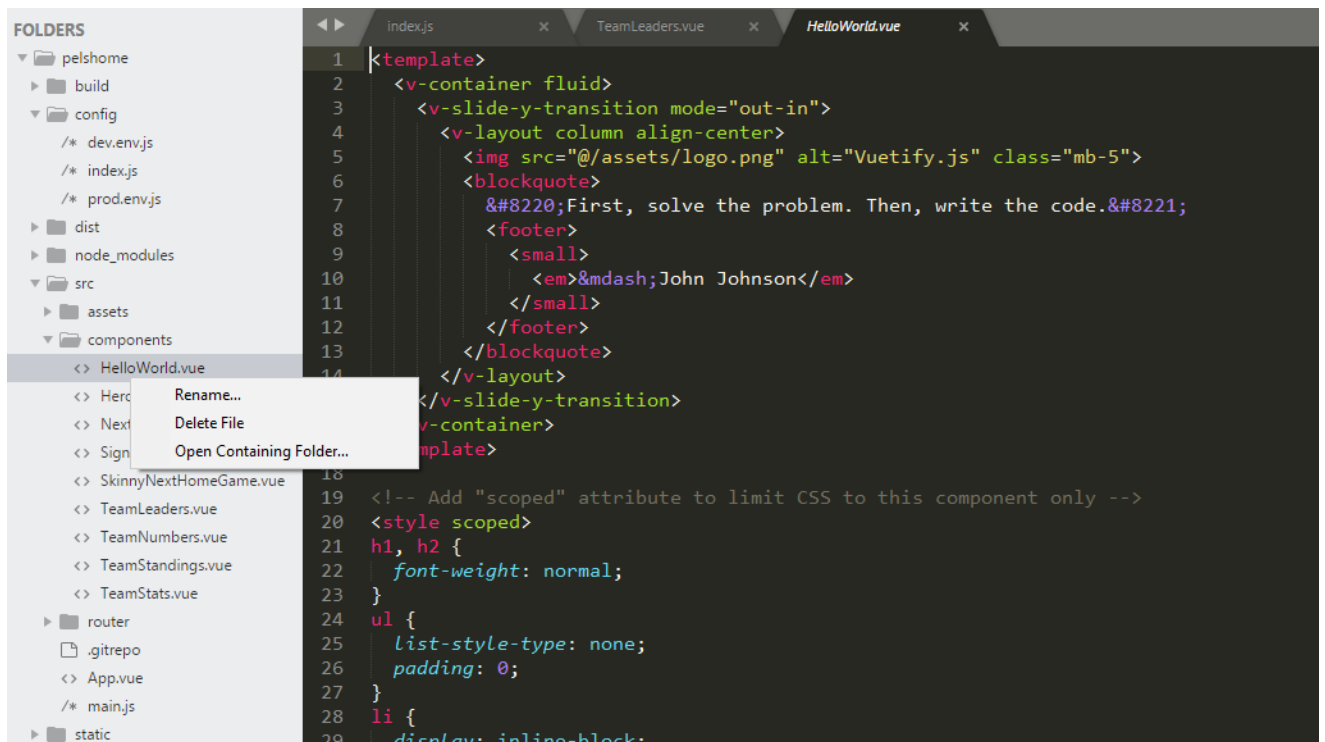


Рис 2.2. Інтерфейс Sublime Text 3

## Sidebar

У Sublime Text, Sidebar (бічна панель) — це інтерфейс, який дозволяє вам оглядати та керувати файлами та папками у вашому проєкті. Вона розташована по боку головного вікна редактора і надає розгорнутий погляд на структуру вашого проєкту.

Основні функції Sidebar в Sublime Text:

1. Дерево файлів та папок: Sidebar відображає структуру вашого проєкту в деревоподібному вигляді, що дозволяє легко переглядати інформацію про файли та папки.
2. Можливість відкривати та закривати файли: Ви можете відкривати файли просто натисканням на них у бічній панелі. Також є опція закривати файли, які вам вже не потрібні, просто видаляючи їх із списку.
3. Маніпулювання папками та файлами: Ви можете створювати нові папки, копіювати, переміщати та видаляти файли безпосередньо з бічної панелі.

4. Швидкий доступ до функцій: Sidebar надає швидкий доступ до різних функцій, таких як пошук файлів, відновлення закритих файлів, додавання нових файлів тощо.

5. Робота з проектами: Ви можете легко відкривати та зберігати проекти через Sidebar, дозволяючи швидко перемикатися між різними робочими середовищами.

Sidebar у Sublime Text — це потужний інструмент, який спрощує навігацію та управління файлами та папками під час роботи над вашим проектом.

### **Notepad++**

Notepad++ – це безкоштовний текстовий редактор з відкритим вихідним кодом, доступний тільки на операційній системі Windows, і, ймовірно, використовується багатьма розробниками. Однією з його дуже корисних функцій є можливість згортання блоків, що зробиє його вельми зручним для роботи з різними мовами програмування. Також існує можливість редагування кількох документів в одному вікні, що істотно прискорює процес розробки. Здатність редагувати один і той самий документ в різних частинах впливає на ефективність розробки.

### **Розуміння синтаксису**

У Notepad++, розуміння синтаксису є важливою функцією, яка допомагає розробникам працювати з різними мовами програмування. Ця функція включає в себе автоматичне виділення синтаксичних елементів коду, забезпечуючи кольорову розмітку, що полегшує читання та редагування коду. Крім того, Notepad++ підтримує згортання блоків коду, що дозволяє розробникам зручно взаємодіяти з великими і складними проектами.

Ця функціональність робить процес написання коду більш приємним і продуктивним, допомагаючи розробникам уникати синтаксичних помилок і зробити їхню роботу більш ефективною.

### **Згортання блоків коду**

Згортання блоків коду у Notepad++ - це функціональність, яка дозволяє скривати частини коду для полегшення навігації та поліпшення оглядовості проектів. Коли ви використовуєте цю функцію, ви можете згорнути (закривати) окремі блоки

коду, такі як функції, умови або цикли, щоб зробити код більш компактним та зручним для перегляду.

Під час роботи з великими файлами або складними проектами ця функція може значно полегшити орієнтацію у коді. Згортання блоків коду дозволяє зосередитися лише на тій частині коду, яку ви розглядаєте, і при необхідності розгортати блоки для подальшого аналізу.

Використання згортання блоків коду в Notepad++ робить редагування та роботу з кодом більш зручною та ефективною.

### **Паралельне редагування файлів**

В Notepad++ є зручна можливість редагування одного або декількох файлів паралельно, що полегшує роботу з проектами, що складаються з багатьох файлів. Ця функція дозволяє вам відкривати та редагувати кілька файлів одночасно та переключатися між ними зручно.

Основні кроки для редагування файлів паралельно в Notepad++:

#### 1. Відкриття декількох файлів:

- Можна відкрити кілька файлів, використовуючи меню "Файл" та опцію "Відкрити" або використовуючи комбінацію клавіш Ctrl+O.

#### 2. Вкладки файлів:

- Кожен відкритий файл відображається в окремій вкладці в верхній частині вікна Notepad++. Ви можете зручно переключатися між вкладками для редагування різних файлів.

#### 3. Розташування вікна:

- Ви можете розташовувати вікна файлів паралельно одне поруч з іншим або на різних частинах екрану для одночасного перегляду та редагування.

#### 4. Синхронізація прокрутки:

- Якщо ви відкриваєте два вікна різних файлів, то вони можуть бути синхронізовані так, що прокрутка в одному вікні також відобразиться в іншому. Це полегшує порівняння та аналіз коду.

Редагування файлів паралельно в Notepad++ дозволяє розглядати та змінювати кілька частин коду одночасно, що є зручною функцією для великих проектів чи великої кількості вихідних файлів.

## **IntelliJ IDEA**

IntelliJ IDEA - це інтегроване середовище розробки (IDE), призначене для створення комп'ютерного програмного забезпечення, написаного на мовах програмування, таких як Java, Kotlin, Groovy та інших, що базуються на JAR. Розроблене компанією JetBrains (попередньо відомою як IntelliJ), це доступне як ліцензована громадська версія Apache 2, а також як пропрієтарна комерційна версія. Обидві опції можна використовувати для комерційного розроблення.

Оригінальна версія IntelliJ IDEA була випущена у січні 2001 року і стала однією з перших Java IDE з розширеними можливостями навігації за кодом та виконанням рефакторингу коду.

### **Допомога у кодуванні**

IDE забезпечує ряд корисних функцій, таких як автоматичне завершення коду за допомогою аналізу контексту, швидка навігація за кодом для безпосереднього переходу до класу або оголошення коду, проведення рефакторингу, відлагодження коду, використання лінтингу та можливість виправлення невідповідностей за допомогою пропозицій.

### **Вбудовані інструменти та інтеграція**

IntelliJ IDEA має вбудовану підтримку інтеграції з різними інструментами для автоматизації рутинних задач та управління залежностями в проектах Java. Ось декілька важливих аспектів щодо вбудованих інструментів та інтеграції:

#### **1. Система збірки Maven та Gradle:**

IntelliJ IDEA підтримує системи збірки Maven і Gradle, що дозволяє легко взаємодіяти з цими інструментами прямо з інтерфейсу розробки. Ви можете створювати нові проекти, додавати та оновлювати залежності, запускати збірку проектів та багато іншого безпосередньо з IntelliJ IDEA.

#### **2. Управління залежностями:**

Редактор має зручний інтерфейс для управління залежностями проекту. Ви можете швидко додавати, оновлювати та видаляти бібліотеки, а також використовувати автоматичне завантаження бібліотек.

### 3. Підтримка тестування:

IntelliJ IDEA інтегрується з різними інструментами для тестування, включаючи JUnit, TestNG тощо. Ви можете легко виконувати тести та переглядати їхні результати прямо в середовищі розробки.

### 4. Автоматична перевірка коду:

Інтегрована підтримка плагінів як FindBugs, PMD, Checkstyle дозволяє автоматично перевіряти код на відповідність стандартам, виявляти можливі помилки та надавати рекомендації з покращення.

### 5. Інтеграція з базами даних:

Є вбудовані інструменти для роботи з базами даних, такі як виконання SQL-запитів, відображення структури бази даних та інші.

### 6. Керування версіями:

IntelliJ IDEA підтримує інтеграцію з різними системами контролю версій, зокрема Git, що дозволяє ефективно працювати з відстеженням та об'єднанням змін у коді.

Ці функції забезпечують розробникам зручний інструментарій для автоматизації та оптимізації рутинних завдань, що дозволяє їм швидше та ефективніше розробляти програмне забезпечення.

Порівнявши чотири редактори, для використання у дипломному проекті було обрано Visual Studio Code. Це обумовлено його оптимальною взаємодією з апаратною частиною, що призводить до ефективної продуктивності редактора. Відзначається зручним інтерфейсом та інтеграцією з системою контролю версій Git, а також наявністю безкоштовної версії.



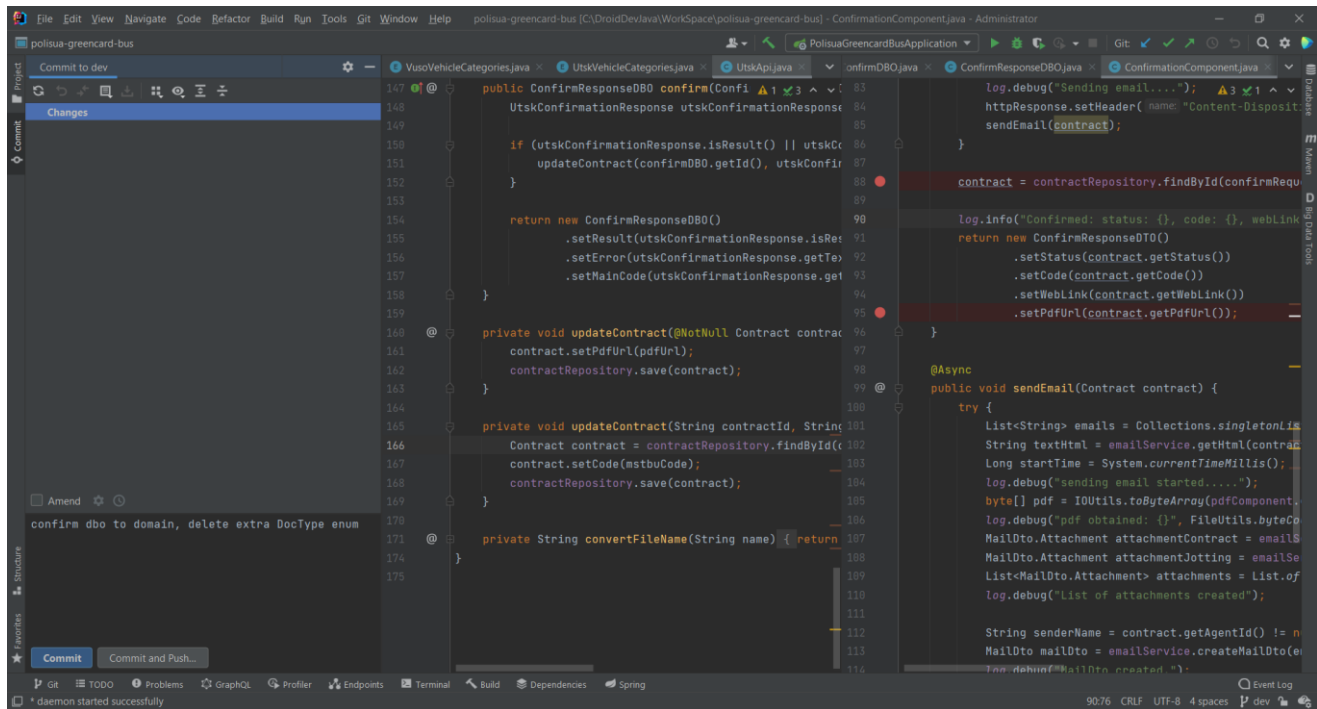


Рис 2.3. Інтерфейс IntelliJ IDEA

## 2.2. Ключові пункти створення веб-сайту

Кожний софт пройде кілька фаз, які відомі як життєвий цикл програмного забезпечення.

### Фаза ініціалізації програмного забезпечення

Фаза ініціалізації програмного забезпечення – це початковий етап в життєвому циклі розробки ПО, який передбачає визначення потреб і визначення основних параметрів проекту. На цьому етапі визначається обсяг робіт, збирається команда розробників, ініціюється процес планування та встановлюються початкові вимоги до функціоналу програми.

Фаза ініціалізації програмного забезпечення є ключовою, оскільки вона визначає фундаментальні параметри проекту. Основні етапи включають в себе визначення бізнес-проблеми, яку програмне забезпечення має вирішити, визначення цілей та обсягу проекту, а також ідентифікацію ключових учасників і зацікавлених сторін. На цьому етапі також може визначитися технічна та бізнес-фізибільність

проекту, враховуючи обмеження та ризики. Усе це допомагає створити чітку основу для подальших етапів розробки програмного забезпечення.

### **Фаза планування**

Фаза планування програмного забезпечення є важливим етапом у життєвому циклі розробки. Під час цієї фази розробники визначають стратегію і план робіт, необхідних для успішної реалізації проекту. Основні аспекти фази планування включають в себе:

1. Визначення завдань і обсягу робіт: Чітке визначення завдань, які потрібно виконати, і обсягу робіт, що охоплює весь проект.

2. Встановлення термінів та графіку: Створення реалістичного графіка проекту та визначення термінів виконання кожного етапу.

3. Розподіл завдань і ресурсів: Визначення, які розробники або команди будуть відповідальні за конкретні завдання, і розподіл ресурсів, таких як технічний персонал і обладнання.

4. Визначення бюджету: Встановлення фінансового обсягу проекту та розподіл коштів між різними аспектами, такими як розробка, тестування та впровадження.

5. Оцінка ризиків і розробка стратегій управління ризиками: Визначення можливих ризиків і розробка планів їх управління для забезпечення успішності проекту навіть у випадках непередбачених обставин.

6. Створення системи контролю: Розробка системи моніторингу та контролю за ходом виконання проекту, щоб вчасно виявляти та вирішувати проблеми.

7. Залучення учасників проекту: Залучення всіх учасників проекту, включаючи замовника, розробників та інших зацікавлених сторін, для визначення їх очікувань та вимог.

Фаза планування є фундаментом для подальших етапів розробки програмного забезпечення і дозволяє забезпечити успішне завершення проекту.

### **Фаза реалізації**

Фаза реалізації програмного забезпечення є однією з ключових етапів у життєвому циклі розробки. Під час цієї фази розробники переходять від

концепційного проектування до фактичної реалізації програмного продукту. Основні аспекти фази реалізації включають в себе:

1. Програмування: На цьому етапі розробники пишуть код, використовуючи вибрану мову програмування. Основна мета - втілити проектні концепції в робочий програмний код.

2. Тестування: Розроблений код піддається тестуванню для виявлення помилок, багів і недоліків. Це може включати юніт-тести, інтеграційні тести та системні тести, спрямовані на забезпечення вірності та ефективності програми.

3. Оптимізація: Процес покращення продуктивності, швидкодії та оптимізації ресурсів. Розробники можуть вносити зміни в код для поліпшення його ефективності.

4. Документація: Створення документації для розуміння та підтримки коду. Це може включати коментарі до коду, технічні описи та іншу документацію.

5. Версіонування: Використання систем контролю версій для відстеження змін у коді, легшого управління версіями та спільної роботи розробників.

6. Вивчення впровадження: Розгляд аспектів впровадження програмного продукту, таких як підготовка користувачів, оновлення та міграція даних.

7. Управління якістю: Вживання заходів для забезпечення високої якості коду та виконання вимог проекту.

8. виправлення помилок: Виявлення і виправлення будь-яких помилок чи недоліків, що можуть виникнути під час реалізації.

Фаза реалізації є вирішальним моментом у створенні програмного продукту та визначає його функціональність та продуктивність.

### **Фаза тестування**

Фаза тестування програмного забезпечення (ПО) є критично важливим етапом у життєвому циклі розробки, спрямованим на впевнення в якості, стабільності та відповідності програмного продукту вимогам. Основні аспекти фази тестування включають в себе:

1. Юніт-тестування: На цьому етапі окремі компоненти програми (юніти) тестуються ізольовано для перевірки їх правильності та взаємодії. Це може включати використання автоматизованих тестів або тестів, розроблених розробниками.

2. Інтеграційне тестування: Тестування взаємодії між різними компонентами або модулями програми для переконання, що вони працюють разом правильно.

3. Функціональне тестування: Перевірка відповідності програмного продукту функціональним вимогам, визначеним у специфікації вимог.

4. Верифікація та валідація: Перевірка відповідності розробленого ПО специфікаціям та переконання в його вірності та ефективності.

5. виправлення помилок: виявлення та виправлення будь-яких помилок, що можуть виникнути під час тестування.

6. Стрес-тестування: випробування програмного продукту під екстремальними умовами для визначення його стійкості та представлення непередбачених ситуацій.

7. Автоматизоване тестування: Використання автоматизованих інструментів для виконання тестів та збірки звітностей про їх результати.

8. Тестування відмовостійкості: випробування програмного продукту на відмовостійкість та можливість відновлення після відмов.

9. Системне тестування: Перевірка роботи програмного продукту в цілому, у відповідності до вимог клієнта.

10. Тестування безпеки: виявлення та усунення можливих слабких місць, які можуть бути використані для порушення безпеки системи.

Фаза тестування дозволяє виявити та виправити помилки та недоліки в програмному продукті перед його впровадженням, що є важливим для забезпечення надійності та високої якості ПО.

### **Аналіз та порівняння технологій контейнеризації**

**Docker** є однією з найпопулярніших технологій контейнеризації. Він забезпечує стандартизований спосіб упаковки, доставки та виконання додатків у контейнерах.

Переваги:

- Простота використання та широке сприйняття у спільноті.
- Велика кількість готових образів у Docker Hub.

- Можливість використання у різних середовищах, від локального розробника до хмарних платформ.

**Kubernetes** це оркестратор контейнерів, який дозволяє керувати та автоматизувати роботу контейнерів в розподіленому середовищі.

Переваги:

- Автоматизація розгортання, масштабування та керування контейнерами.
- Самовідновлення та висока доступність.
- Гнучкість та розширюваність для роботи в різних конфігураціях.

- **Podman** є альтернативою Docker, спрямованою на власних облікових записів та бездіяльність демона.

Переваги:

- Відсутність потреби в демоні, що робить його більш безпечним та легшим.
- Підтримка багатокористувацькості та можливість використовувати ізольовані контейнери.

**OpenShift** є платформою, яка базується на Kubernetes і додає додатковий функціонал для підтримки управління життєвим циклом додатків.

- Інтеграція з іншими сервісами та інструментами для розробки та управління додатками.

- Управління безпекою, доступністю та моніторингом.

Containerd:

- Containerd є компонентом, який використовується Kubernetes та іншими оркестраторами для управління контейнерами.

Переваги:

- Мінімалістичний та спрощений функціонал, що робить його легким та стабільним.

- Додаткові можливості через плагіни.

Порівняня:

- Легкість використання: Docker та Podman славляться простотою використання, в той час як Kubernetes має більш високий поріг входу.

- Функціональність: Kubernetes виграє за функціональністю, оскільки він оркеструє та автоматизує контейнери, а Docker та Podman забезпечують базовий функціонал.

- Застосування: Docker широко використовується для розробки та локальних тестів, Kubernetes — для розгортання та управління виробничими системами, а Podman — для бездіяльного використання та ізольованих контейнерів.

- Залежність від демона: Docker вимагає працездатного демона, тоді як Podman працює без демона, що робить його безпечнішим та легшим.

- Екосистема: Docker має велику екосистему готових образів, а Kubernetes є стандартом для оркестрації та розгортання.

### **2.3. Висновок до розділу 2**

Під час реалізації другої фази, проведено детальний розгляд та визначено ключові етапи створення інтернет-магазину та вибрано середовище для його розробки.

# РОЗДІЛ 3

## РОЗРОБКА ФУНКЦІОНАЛЬНОЇ ЧАСТИНИ ВЕБ-САЙТУ З МЕТОЮ ЗДІЙСНЕННЯ ПРОДАЖУ

### 3.1. Концепція сайту

Цільова мета аудиторії для онлайн-магазину - користувачі, які виражають бажання придбати побутову техніку. Асортимент великого розмаїття, і впроваджений програмний продукт дозволяє швидко додавати нові категорії та вилучати неактуальні.

Веб-сайт розроблено з метою розширення охоплення користувачів, спрямоване на залучення клієнтів та збільшення обсягів продажів товарів.

Основні функціональні можливості інтернет-магазину включають:

- оперативна підтримка від оператора;
- статистичний аналіз, заснований на обробці даних, які зберігаються в базах даних, таких як кількість щоденних відвідувачів, обсяги продажів тощо;
- просте управління, включаючи додавання нових акцій та знижок;
- зручна комунікація та інтеграція з іншими партнерами у бізнесі;
- можливість представлення функціоналу через розроблене API.

### 3.2. Структура веб-сторінки

Однією з ключових рис веб-сайту є його організація, оскільки саме структура визначає його наповнення - як розміщені розділи, гіперпосилання, текстовий контент і взаємозв'язок між ними.

Кафедра КІТ				НАУ 23 01 01 000 ПЗ			
Виконав	Березін А.В.			РОЗРОБКА ФУНКЦІОНАЛЬНОЇ ЧАСТИНИ ВЕБ- САЙТУ З МЕТОЮ ЗДІЙСНЕННЯ ПРОДАЖУ	Літера	аркуш	аркушів
Керівник	Моденов Ю.Б.					55	11
Консульт.					<b>УС-412Б      122</b>		
Н-контрол.	Райчев						

Під час розробки та формування структури важливо враховувати, який функціонал нам необхідний для нашого веб-сайту, наявність каталогів, система навігації та наповнення.

Під час створення структури веб-сторінки також важливо враховувати аспекти візуальної привабливості та зручності для користувача. Ефективне використання графічних елементів, правильне розташування важливих елементів і інтуїтивно зрозуміла навігація можуть покращити загальний враження від відвідування веб-сайту. Також, враховуючи сучасні тенденції, мобільна адаптабельність грає важливу роль, оскільки багато користувачів використовують різноманітні пристрої для доступу до інтернету.

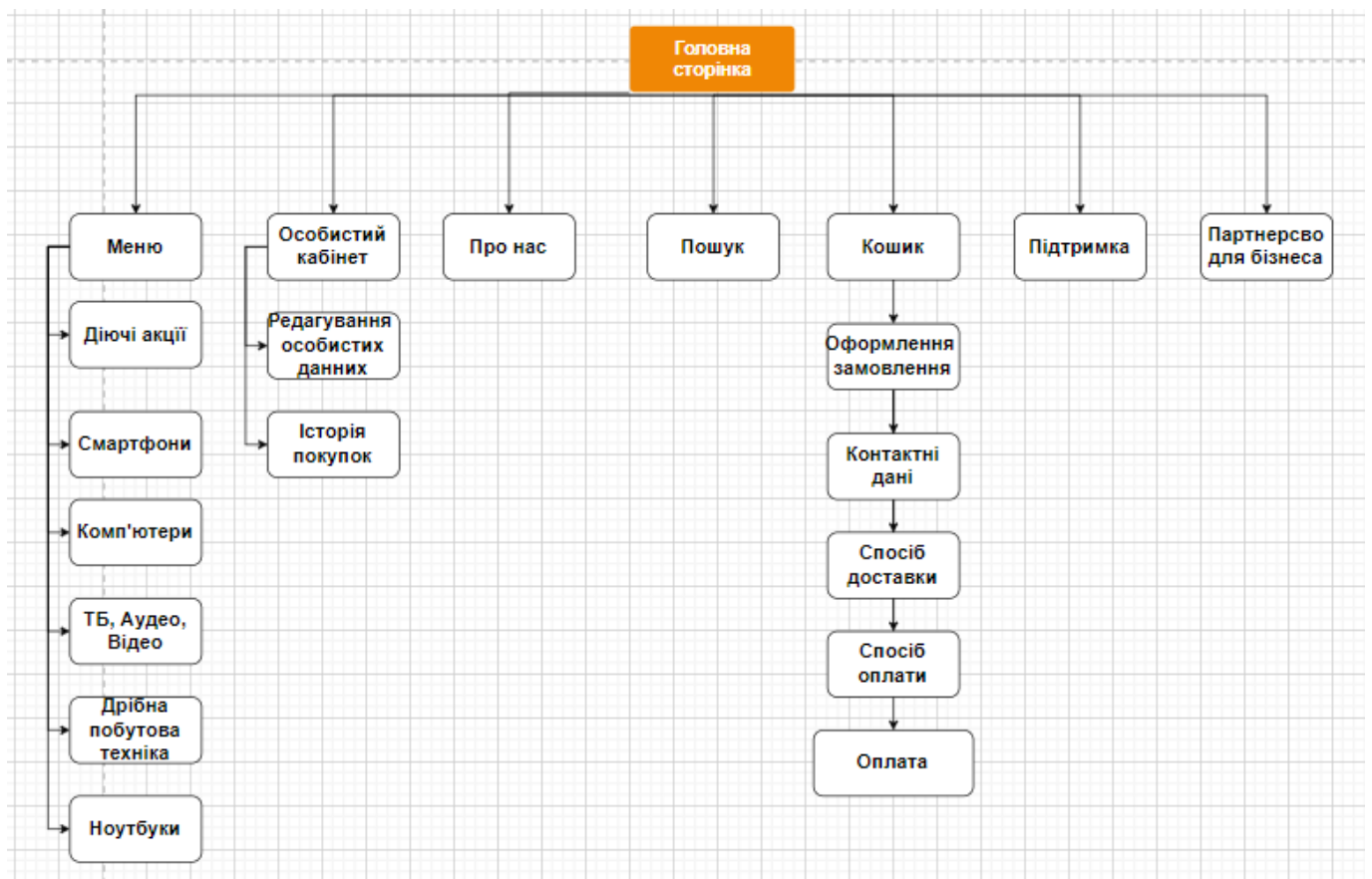


Рис. 3.1. Складові структури веб-сайту «GO HOME»



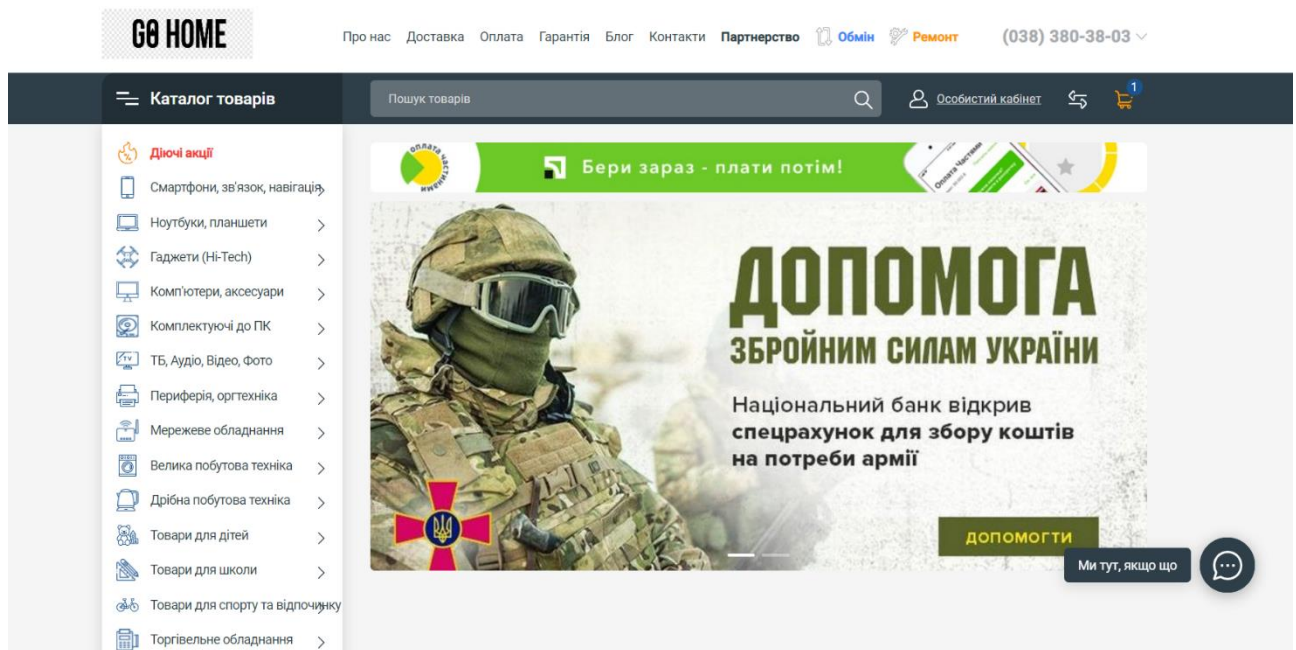


Рис. 3.2. Головна сторінка веб-сайту «GO HOME»

Однією з фундаментальних структур є структура, схожа на дерево, що виявляється дуже зручною, легкою для розуміння та універсальною для організації веб-сторінки. Головна сторінка виступає в якості точки виходу в подорожі по структурному дереву та має можливість виконати необхідні дії або знайти інформацію, що цікавить користувача, оскільки все систематизовано за категоріями. Користувач обирає потрібну категорію, що дозволяє йому продовжити подорож вздовж гілки дерева.

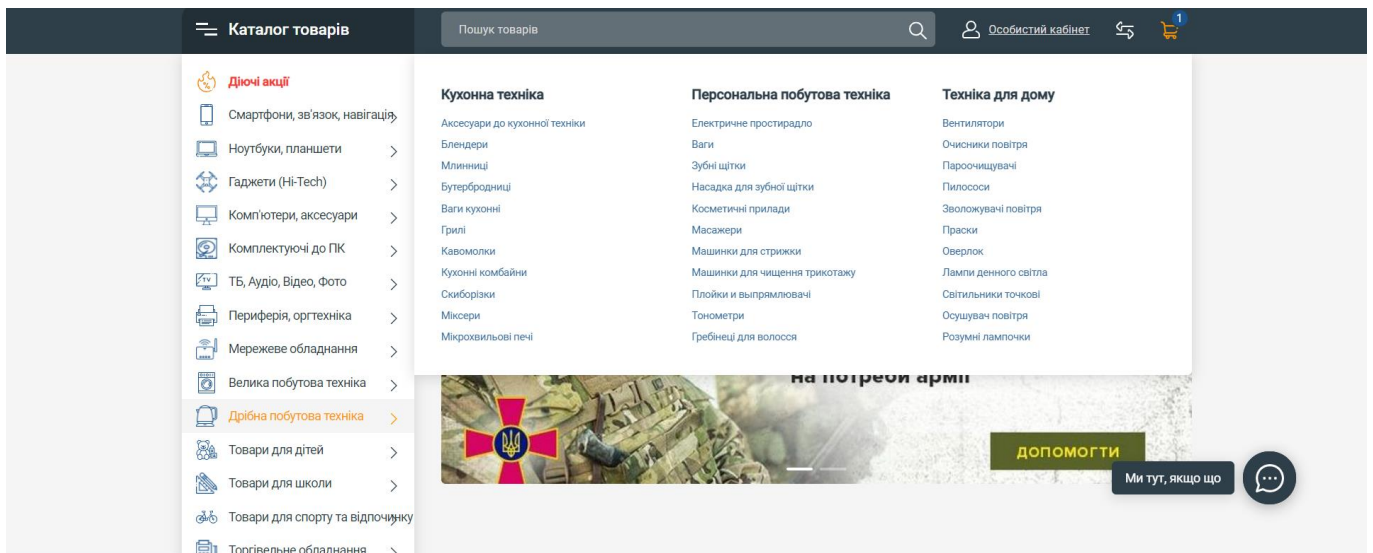


Рис. 3.3. Головна сторінка веб-сайту «GO HOME»

Переходячи до категорії "Велика побутова техніка", відкривається багато субкатегорій, які прямо відносяться до батьківської категорії. Серед всіх субкатегорій вибираємо "Холодильники" і спускаємося по дереву на один рівень нижче. На цьому етапі користувач може переглядати конкретні товари вибраної тематики.

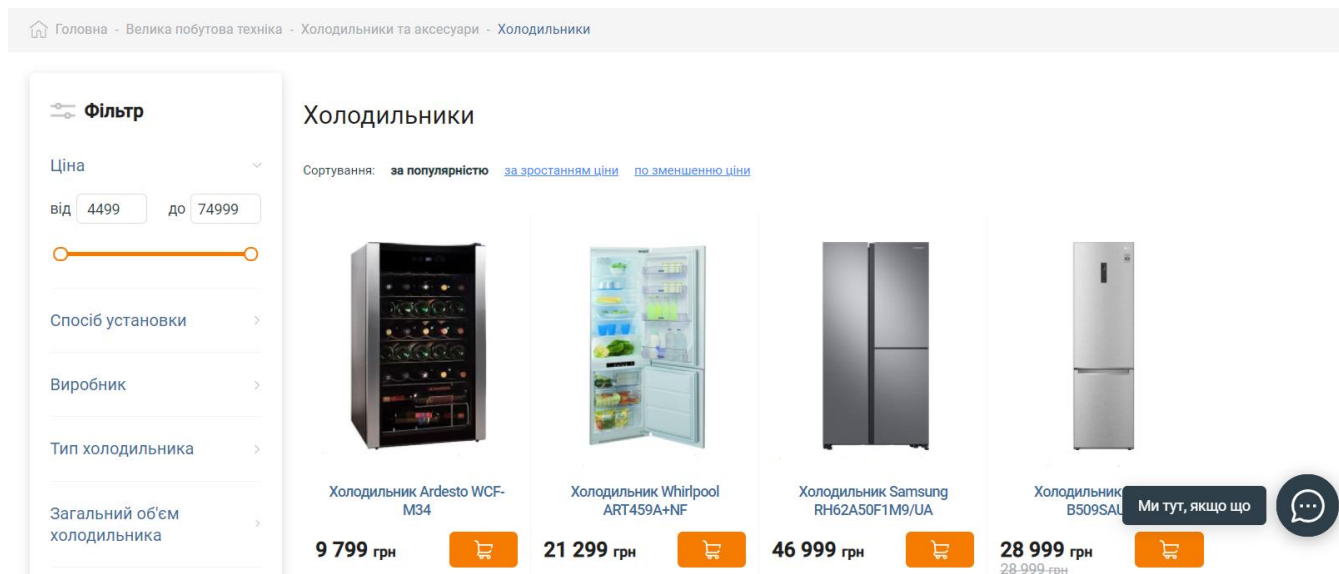


Рис. 3.4. Товари у категорії «Велика побутова техніка», та підкатегорії «Холодильники»

## **Особистий кабінет**

Особистий обліковий запис є зручною функцією, де клієнт може редагувати, вводити та відстежувати інформацію, яка стосується виключно його. В особистих даних користувача включені його ім'я, прізвище, номер телефону, поштове відділення тощо. Усі ці дані зберігаються в базі даних і, при оформленні замовлення, ефективно зчитуються серверною частиною, передаючи їх клієнтській для автоматичного заповнення, що надзвичайно зручно. Для постійних клієнтів система запам'ятовує їхні дані, щоб вони не мусили вводити їх при кожному новому замовленні. Однак важливо надавати можливість клієнту редагувати автоматично заповнені дані. Увесь цей бізнес-процес повинен бути ретельно прокерований до дрібниць, щоб уникнути будь-яких непорозумінь для клієнта та уникнути постійного перегляду логіки проекту, що може ускладнити внесення змін у майбутньому. Створення облікового запису та збір інформації про клієнта відбувається шляхом процедури реєстрації через відповідну форму або ж за допомогою інформації, яку клієнт вказує під час першої покупки, яка ініціює неявну реєстрацію без створення пароля. Унікальним ідентифікатором юзера стає логін, який він вводить при першій покупці. Якщо цей логін використовується для майбутніх замовлень, дані з бази даних автоматично переносяться на клієнтську частину для автоматичного заповнення полів.

Додатково можна враховувати, що в особистому кабінеті користувач також може встановлювати власні налаштування, такі як вподобання щодо отримання сповіщень, вибір мови інтерфейсу чи персоналізація дизайну. Забезпечення можливості самостійно налаштовувати свій обліковий запис додає до зручності та індивідуального підходу до кожного користувача. Також важливо розробляти систему безпеки для забезпечення конфіденційності особистих даних і запобігання несанкціонованому доступу.

# Створити новий обліковий запис клієнта


## Персональна інформація

---

Ім'я \*

Прізвище \*

Підписатися на новини

Дозволити дистанційну допомогу в покупках 

## Інформація для входу

---

Електронна пошта \*

Пароль \*

Рис. 3.5. Форма реєстрації у клієнтську базу інтернет магазину «GO HOME»

### Фільтр товарів

Однією з найзручніших функцій інтернет-магазину є швидкість та точність пошуку інформації, що робить користувача надзвичайно продуктивним у обробці об'ємної інформації про товари. Цю зручність в інтернет-магазині забезпечує використання фільтрів товарів. Сортування проводиться миттєво за параметрами пошуку. Наприклад, при сортуванні товарів в розділі "Ноутбуки" в секції фільтрів відображаються можливі параметри сортування для товарів даної категорії.

Як видно на Рис. 3.6, доступні параметри для сортування включають ціну, виробника та країну виробника. Процес сортування працює за таким ланцюгом:

1) клієнт вказує значення параметрів;

- 2) клієнтська частина генерує запит для фільтрації та надсилає його на сервер до потрібного ендпоінта;
- 3) сервер приймає запит, створює запит до БД на основі переданих параметрів від клієнта;
- 4) БД формує список товарів, що відповідають вказаним параметрам, і повертає ці дані серверу;
- 5) сервер отримує "сирі дані" від БД і перетворює їх у формат, зрозумілий для клієнтської частини, формуючи відповідь;
- 6) сервер надсилає відповідь на клієнтську частину;
- 7) клієнтська частина отримує очікувану відповідь від сервера та на основі цих даних, у нашому випадку список ноутбуків, що відповідають обраним користувачем параметрам пошуку, відображає їх у вигляді, заданому стилями клієнтської частини.

### **Покупка**

Основна мета електронного магазину – реалізація товарів. Саме тому для клієнта важливо, щоб шлях від головної сторінки до оплати товару був якнайбільш простим. На структурі сайту, що нагадує дерево (див. рис. 3.1), можна спостерігати, що для завершення покупки, клієнту потрібно здійснити лише 6 кроків:

- 1) Кошик – користувач переглядає обрані товари, що дозволяє зручно перевірити їх будь-де на сайті;
- 2) Після перевірки кошика і задоволеності вибором, клієнт натискає "Оформити замовлення";
- 3) Під час процесу оформлення замовлення клієнт заповнює особисті дані або, у разі попередніх покупок, вводить свій електронний адрес, за допомогою якого система "витягує" його дані з бази даних;
- 4) Далі - вибір методу доставки;
- 5) Вибір методу оплати;
- 6) У випадку, якщо клієнт обирає "Оплата карткою на сайті", його перенаправляють на платіжну систему LiqPay, де здійснюється переказ коштів.

## **Налаштування розсилки про наявність нового товару по підписці**

Перейдемо до реалізації backend частини, а саме до автотичного налаштування серверної частини, так щоб при будь-якому навантаженні на наші сервери, автоматично додавались ресурси як апаратні так і софтверні.

Основна задача така:

У системі існує база даних підписок від користувачів, які бажають отримувати повідомлення про надходження конкретних товарів. Щодня, в контексті мікросервісу товарів, всі підписки обробляються. Якщо товар надійшов, формується лист, який має бути відправлений через брокер повідомлень RabbitMQ, а потім розподілений у мікросервісі підправки повідомлень. Однак, в ситуації, коли кількість надійшлих товарів є великою, може виникнути проблема із завантаженням на мікросервіс підправки повідомлень. Тому використовуючи Docker та Kubernetes, потрібно автоматично створювати нові поди мікросервісу підправки повідомлень, якщо існуючі не впораються з навантаженням. Крім того, брокер повідомлень RabbitMQ повинен розподіляти навантаження порівну між усіма подами мікросервісу підправки повідомлень, забезпечуючи оптимальне використання ресурсів та високу доступність системи.

### **1. Створення Бази Даних Підписок:**

- Використовуйте підходящу систему управління базами даних (наприклад, MySQL, PostgreSQL) для зберігання підписок користувачів.
- Реалізуйте схему бази даних, яка містить інформацію про користувачів та їхні підписки на товари.

### **2. Мікросервіс Товарів:**

- Створіть мікросервіс для обробки товарів, який буде щоденно перевіряти надходження товарів та обробляти підписки користувачів.
- Якщо товар надійшов, створіть лист для відправлення через брокер повідомлень RabbitMQ.

### **3. Брокер Повідомлень RabbitMQ:**

- Налаштуйте RabbitMQ для прийому та розподілу повідомлень між мікросервісом товарів та мікросервісом підправки повідомлень.

- Використовуйте обмінників та черги для забезпечення високої доступності та розподілення навантаження.

#### 4. Мікросервіс Підправки Повідомлень:

- Створіть мікросервіс для підправки повідомлень, який отримує листи від брокера RabbitMQ та розсилає їх користувачам.

- Використовуйте Docker для контейнеризації мікросервісу та Kubernetes для автоматичного масштабування.

#### 5. Автоматичне Масштабування:

- Налаштуйте Kubernetes для моніторингу роботи мікросервісу підправки повідомлень.

- Встановіть правила автоматичного масштабування, щоб створювати нові поди, якщо існуючі не впораються з навантаженням.

#### 6. Розподілення Навантаження:

- Забезпечте налаштування RabbitMQ для розподілу повідомлень порівну між усіма подами мікросервісу підправки повідомлень.

- Використовуйте можливості RabbitMQ для обробки повідомлень в режимі розподілення, забезпечуючи оптимальне використання ресурсів.

#### 7. Моніторинг та Логування:

- Використовуйте інструменти моніторингу та логування Kubernetes для відстеження працездатності та виявлення проблем в системі.

#### 8. Тестування та Оптимізація:

- Проводьте тестування системи для визначення її продуктивності та надійності.

- Виправляйте помилки та оптимізуйте систему для забезпечення її ефективності в умовах реального середовища.

Реалізація цієї системи забезпечить ефективне управління повідомленнями про надходження товарів, з масштабованою та високодоступною інфраструктурою.

### **3.3. Висновок до розділу 3**

На третьому етапі розділу представлено висновки розробки, яка призвела до створення електронного магазину.

У ході розвитку електронного магазину особливу увагу слід звернути на вдосконалення користувацького інтерфейсу та оптимізацію процесів оформлення замовлення. Оптимізація цих аспектів сприятиме покращенню загального враження клієнтів та збільшенню конверсії відвідувачів в покупців. Також, важливим є впровадження маркетингових інструментів, таких як персоналізовані рекомендації товарів та програми лояльності, що сприятимуть залученню та утриманню клієнтів



## ВИСНОВКИ

В результаті виконання дипломного проекту був створений електронний ресурс, призначений для торгівлі побутовою технікою. Задачі, поставлені перед проектом, успішно виконані, включаючи:

- аналіз технологічних рішень для розробки програмного забезпечення;
- оцінку методів розробки комерційних проектів;
- визначення структури та організації веб-сайту;
- розроблення інтернет-магазину побутової техніки.

Було проведено ретельний аналіз та розробка веб-сайту для покупки побутових товарів з використанням сучасних технологій. Розглянуті та впроваджені новітні підходи та інструменти веб-розробки, такі як мультиплатформенність, адаптивний дизайн, штучний інтелект для персоналізованих рекомендацій, а також зручний інтерфейс користувача.

Створений веб-сайт дозволяє користувачам ефективно шукати, порівнювати та придбавати побутові товари в онлайн-режимі. Застосування механізмів фільтрації, швидка обробка замовлень та зручна система оплати сприяють покращенню користувацького досвіду та забезпечують ефективний обіг інформації на веб-сайті.

Цей диплом підкреслює важливість використання новітніх технологій у веб-розробці для покращення електронних торговельних платформ та забезпечення задоволення потреб сучасного споживача.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розширювальна мова розмітки [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/XML/> (дата звернення 30.05.2022). – Назва з екрану.
2. Етапи створення веб сайтів [Електронний ресурс]. Режим доступу: [http://school.xvatit.com/index.php?title=Етапи\\_створення\\_веб-сайтів](http://school.xvatit.com/index.php?title=Етапи_створення_веб-сайтів) (дата звернення 30.05.2022). – Назва з екрану.
3. Мова гіпертексту HTML [Електронний ресурс]. Режим доступу: <http://www.znannya.org/?view=html> (дата звернення 29.05.2022). – Назва з екрану.
4. Мова web-стилів CSS [Електронний ресурс]. Режим доступу: [https://css.in.ua/article/shcho-take-html\\_10](https://css.in.ua/article/shcho-take-html_10) (дата звернення 29.05.2022). – Назва з екрану.
5. Spring [Електронний ресурс]. Режим доступу: <https://uchika.in.ua/referat-na-temu--spring-framework.html> (дата звернення 29.05.2022). – Назва з екрану.
6. VS CODE [Електронний ресурс]. Режим доступу: <https://open.zeba.academy/pochemu-vscode-populyaren/> (дата звернення 29.05.2022). – Назва з екрану.