

## Звіт подібності

### метадані

Заголовок

2024\_Slivchuk\_126\_master.docx

Автор

Науковий керівник / Експерт

Слівчук Ілля Валентинович

Слівчук Ілля Валентинович

підрозділ

National University of Water and Environmental Engineering

### Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		7
Інтервали		12
Мікропробіли		143
Білі знаки		0
Парафрази (SmartMarks)		173

### Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

13223

Кількість слів

103511

Кількість символів

### Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

#### 10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	КОЛІР ТЕКСТУ
1	<a href="https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/%D0%9E%D0%A5%D0%9E%D0%A0%D0%9E%D0%9D%D0%90_%D0%9F%D0%A0%D0%90%D0%A6%D0%86/R_IT-INDUSTRIA.pdf">https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/%D0%9E%D0%A5%D0%9E%D0%A0%D0%9E%D0%9D%D0%90_%D0%9F%D0%A0%D0%90%D0%A6%D0%86/R_IT-INDUSTRIA.pdf</a>	113	0.85 %
2	<a href="https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf">https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf</a>	64	0.48 %
3	<a href="https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf">https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf</a>	62	0.47 %
4	<a href="http://sites.znu.edu.ua/webprog/lect/1191.ukr.html">http://sites.znu.edu.ua/webprog/lect/1191.ukr.html</a>	60	0.45 %

5	2024_126_mahistr_HradovyiOO_BaranovskyiSV.docx 12/21/2024 National University of Water and Environmental Engineering (National University of Water and Environmental Engineering)	54	0.41 %
6	<a href="https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/%D0%9E%D0%A5%D0%9E%D0%A0%D0%9E%D0%9D%D0%90_%D0%9F%D0%A0%D0%90%D0%A6%D0%86/R_IT-INDUSTRIA.pdf">https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/%D0%9E%D0%A5%D0%9E%D0%A0%D0%9E%D0%9D%D0%90_%D0%9F%D0%A0%D0%90%D0%A6%D0%86/R_IT-INDUSTRIA.pdf</a>	52	0.39 %
7	2024_126_mahistr_HradovyiOO_BaranovskyiSV.docx 12/21/2024 National University of Water and Environmental Engineering (National University of Water and Environmental Engineering)	45	0.34 %
8	<a href="https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf">https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf</a>	41	0.31 %
9	<a href="https://er.knuid.edu.ua/bitstream/123456789/28041/1/Dyplom122_Mnozhynskyi_Astistova.pdf">https://er.knuid.edu.ua/bitstream/123456789/28041/1/Dyplom122_Mnozhynskyi_Astistova.pdf</a>	34	0.26 %
10	<a href="https://essuir.sumdu.edu.ua/bitstream/123456789/90967/1/Viunnik_mag_rob.pdf">https://essuir.sumdu.edu.ua/bitstream/123456789/90967/1/Viunnik_mag_rob.pdf</a>	34	0.26 %

### з бази даних RefBooks (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-----------	--

### з домашньої бази даних (3.93 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	2024_126_mahistr_HradovyiOO_BaranovskyiSV.docx 12/21/2024 <b>National University of Water and Environmental Engineering (National University of Water and Environmental Engineering)</b>	<b>510 (37)</b>	<b>3.86 %</b>
2	2024_126_mahistr_KhmelnycchiyVV_GrycyukPM.docx 12/21/2024 National University of Water and Environmental Engineering (National University of Water and Environmental Engineering)	10 (1)	0.08 %

### з програми обміну базами даних (0.36 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	Дипломна робота_Ціхоцький-14 6/17/2024 Ukrainian Academy of Printing (Кафедра КНІТ)	17 (3)	0.13 %
2	Курсова робота Леган М.С. 5/27/2024 Uzhhorod National University (UzhNU)	13 (2)	0.10 %
3	ICT-23дм_Бас А. В. Пояснювальна записка-compressed.pdf 12/16/2024 East Ukrainian National University named after Volodymyr Dahl (East Ukrainian National University named after Volodymyr Dahl)	11 (1)	0.08 %
4	bitstream_3a038e1f-5151-40f0-8b43-93f7e3cdd7e9 12/8/2024 National Technical University "Kharkiv Polytechnic Institute" students papers (National Technical University "Kharkiv Polytechnic Institute" students papers)	6 (1)	0.05 %

### з Інтернету (12.15 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	<a href="https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/%D0%9E%D0%A5%D0%9E%D0%A0%D0%9E%D0%9D%D0%90_%D0%9F%D0%A0%D0%90%D0%A6%D0%86/R_IT-INDUSTRIA.pdf">https://www.khadi.kharkov.ua/fileadmin/P_vcheniy_secretar/%D0%9E%D0%A5%D0%9E%D0%A0%D0%9E%D0%9D%D0%90_%D0%9F%D0%A0%D0%90%D0%A6%D0%86/R_IT-INDUSTRIA.pdf</a>	344 (16)	2.60 %
2	<a href="https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf">https://ep3.nuwm.edu.ua/30214/1/04-05-84%D0%9C.pdf</a>	218 (7)	1.65 %
3	<a href="https://essuir.sumdu.edu.ua/bitstream/123456789/93067/1/Chevhuз_bak_rob.pdf">https://essuir.sumdu.edu.ua/bitstream/123456789/93067/1/Chevhuз_bak_rob.pdf</a>	153 (9)	1.16 %
4	<a href="https://essuir.sumdu.edu.ua/bitstream/123456789/90967/1/Viunnik_mag_rob.pdf">https://essuir.sumdu.edu.ua/bitstream/123456789/90967/1/Viunnik_mag_rob.pdf</a>	130 (9)	0.98 %
5	<a href="https://rau.ua/novyni/trendi-e-com-2024/">https://rau.ua/novyni/trendi-e-com-2024/</a>	118 (11)	0.89 %
6	<a href="http://elartu.tntu.edu.ua/bitstream/lib/34121/1/Mag_nayk_2020_CTm_Bass_R_V.pdf">http://elartu.tntu.edu.ua/bitstream/lib/34121/1/Mag_nayk_2020_CTm_Bass_R_V.pdf</a>	92 (10)	0.70 %
7	<a href="http://sites.znu.edu.ua/webprog/lect/1191.ukr.html">http://sites.znu.edu.ua/webprog/lect/1191.ukr.html</a>	90 (4)	0.68 %
8	<a href="https://dspace.nuft.edu.ua/bitstreams/4d2c6e38-9406-48fa-abb9-ae8c96751b73/download">https://dspace.nuft.edu.ua/bitstreams/4d2c6e38-9406-48fa-abb9-ae8c96751b73/download</a>	85 (5)	0.64 %
9	<a href="https://elartu.tntu.edu.ua/bitstream/lib/41540/1/%D0%9C%D0%B0%D1%80%D0%BA%D1%83%D1%88%D0%B5%D0%B2%D1%81%D1%8C%D0%BA%D0%B8%D0%B9_%D0%A1%D0%9D%D0%B7-41.pdf">https://elartu.tntu.edu.ua/bitstream/lib/41540/1/%D0%9C%D0%B0%D1%80%D0%BA%D1%83%D1%88%D0%B5%D0%B2%D1%81%D1%8C%D0%BA%D0%B8%D0%B9_%D0%A1%D0%9D%D0%B7-41.pdf</a>	61 (5)	0.46 %
10	<a href="https://er.knutd.edu.ua/bitstream/123456789/23152/3/Dyplom122_K%D0%BEchuk_Astistov_a.pdf">https://er.knutd.edu.ua/bitstream/123456789/23152/3/Dyplom122_K%D0%BEchuk_Astistov_a.pdf</a>	59 (5)	0.45 %
11	<a href="https://elartu.tntu.edu.ua/bitstream/lib/38125/1/%D0%9A%D1%83%D1%87%D0%B5%D1%80_%D0%A1%D0%9D%D0%B7-41_%D0%B1%D0%B0%D0%BA%D0%B0%D0%BB%D0%B0%D0%B2%D1%80_2022.pdf">https://elartu.tntu.edu.ua/bitstream/lib/38125/1/%D0%9A%D1%83%D1%87%D0%B5%D1%80_%D0%A1%D0%9D%D0%B7-41_%D0%B1%D0%B0%D0%BA%D0%B0%D0%BB%D0%B0%D0%B2%D1%80_2022.pdf</a>	50 (3)	0.38 %
12	<a href="https://er.knutd.edu.ua/bitstream/123456789/28041/1/Dyplom122_Mnozhytskyi_Astistova.pdf">https://er.knutd.edu.ua/bitstream/123456789/28041/1/Dyplom122_Mnozhytskyi_Astistova.pdf</a>	49 (2)	0.37 %
13	<a href="https://elartu.tntu.edu.ua/bitstream/lib/41636/1/Dyplom_Mykhailovskiy_O_P_2023.pdf">https://elartu.tntu.edu.ua/bitstream/lib/41636/1/Dyplom_Mykhailovskiy_O_P_2023.pdf</a>	32 (2)	0.24 %
14	<a href="http://gi.edu.ua/en/college-main/subsection/library/repozytorii/file/download/d531aa6f50478fd0a67ea11cf48e60bb">http://gi.edu.ua/en/college-main/subsection/library/repozytorii/file/download/d531aa6f50478fd0a67ea11cf48e60bb</a>	22 (2)	0.17 %
15	<a href="https://epcc.knutd.edu.ua/bitstream/123456789/19448/1/Dyplom122_Yehorov_Astistova.pdf">https://epcc.knutd.edu.ua/bitstream/123456789/19448/1/Dyplom122_Yehorov_Astistova.pdf</a>	18 (2)	0.14 %
16	<a href="https://studfiles.net/preview/5118185/page:10/">https://studfiles.net/preview/5118185/page:10/</a>	16 (2)	0.12 %
17	<a href="https://openarchive.nure.ua/bitstreams/adc0cb1d-3786-47e1-937c-a17df2cf2ef5/download">https://openarchive.nure.ua/bitstreams/adc0cb1d-3786-47e1-937c-a17df2cf2ef5/download</a>	15 (2)	0.11 %
18	<a href="http://ekhsuir.kspu.edu/bitstream/handle/123456789/16158/Andreeva_T_pedfak_2021.pdf?sequence=1">http://ekhsuir.kspu.edu/bitstream/handle/123456789/16158/Andreeva_T_pedfak_2021.pdf?sequence=1</a>	15 (1)	0.11 %
19	<a href="http://eprints.zu.edu.ua/18020/1/%D0%9C%D0%B0%D0%BD%D1%8C%D0%BA%D0%BE%D0%B2%D1%81%D1%8C%D0%BA%D0%B8%D0%B9.pdf">http://eprints.zu.edu.ua/18020/1/%D0%9C%D0%B0%D0%BD%D1%8C%D0%BA%D0%BE%D0%B2%D1%81%D1%8C%D0%BA%D0%B8%D0%B9.pdf</a>	14 (1)	0.11 %
20	<a href="https://dspace.znu.edu.ua/jspui/bitstream/12345/12497/1/Takhtarov_M_S.pdf">https://dspace.znu.edu.ua/jspui/bitstream/12345/12497/1/Takhtarov_M_S.pdf</a>	13 (1)	0.10 %
21	<a href="http://elartu.tntu.edu.ua/bitstream/lib/30726/3/dyplom_Veselovska.pdf">http://elartu.tntu.edu.ua/bitstream/lib/30726/3/dyplom_Veselovska.pdf</a>	12 (2)	0.09 %

## Список принятых фрагментів (немає принятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ Національний університет водного господарства та природокористування  
 Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії Кафедра комп'ютерних технологій та економічної кібернетики Допущено до захисту: Завідувач кафедри комп'ютерних технологій та економічної кібернетики д.е.н., проф. П.М. Грицюк «  
 » 20 р. КВАЛІФІКАЦІЙНА РОБОТА на здобуття ступеня «магістр»



SQL - Мова структурованих запитів

UI - Інтерфейс користувача

UX - Досвід користувача

URL - Уніфікований ресурсний локатор

WYSIWYG - What You See Is What You Get (редактор, що дозволяє редагувати контент без знань програмування)

XML - Мова розмітки розширена

YARN - Менеджер пакетів для JavaScript

Firebase - Платформа для мобільних та веб-додатків, що включає базу даних, хмарне зберігання та автентифікацію

**Tailwind CSS - Утилітарний CSS-фреймворк, який дозволяє швидко створювати адаптивні та стилізовані веб-інтерфейси** за допомогою класів для стилів

Back-end - Програмно-апаратна частина сервісу

Front-end - Клієнтська частина користувацького інтерфейсу у програмно-апаратній частині сервісу

## ЗМІСТ

ВСТУП 8

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРЕДМЕТНОЇ ОБЛАСТІ 10

- 1.1. Визначення та характеристики веб-додатків 10
- 1.2. Технології хмарних обчислень 14
- 1.3. Використання Next.js для створення веб-додатків на базі React 18
- 1.4. Відмінність між веб-сайтом та веб-додатком 21

РОЗДІЛ 2. АНАЛІЗ ТА ПРОЕКТУВАННЯ СТРУКТУРИ ІНТЕРНЕТ-МАГАЗИНУ 24

- 2.1. Дослідження специфіки функціонування та впровадження інтернет-магазинів 24
- 2.2. Визначення функціональних і нефункціональних вимог 30
- 2.3. Проектування архітектури інтернет-магазину 34
- 2.4. Вибір платформи для реалізації проекту 36

РОЗДІЛ 3. РОЗРОБКА ФУНКЦІОНАЛУ ІНТЕРНЕТ-МАГАЗИНУ 38

- 3.1. Вибір інструментів для розробки 38
- 3.2. Розробка серверної частини: бази даних 41
- 3.3. Розробка ключових алгоритмів 47
- 3.4. Створення інтерфейсу 57

РОЗДІЛ 4. ОХОРОНА ПРАЦІ В ІТ-ГАЛУЗІ 62

- 4.1. Удосконалення умов праці програмістів 62
- 4.2. Інформаційна культура в ІТ-управлінні та охороні праці 65
- 4.3. Ергономічні умови праці в офісах і віддаленій роботі 70
- 4.4. Організація заходів з безпеки праці для ІТ-проектів 73

ВИСНОВКИ 76

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ 77

ДОДАТКИ 80

## ВСТУП

Актуальність теми обумовлена швидким розвитком електронної комерції та необхідністю створення ефективних, зручних і надійних інтернет-магазинів, які здатні задовольнити потреби сучасних споживачів. В умовах високої конкуренції та зростаючої кількості онлайн-покупок, важливим є впровадження сучасних веб-технологій, які дозволяють створити ефективні, масштабовані та швидкодіючі платформи для торгівлі. Використання технологій, таких як Next.js, MongoDB, Firebase, Strapi та Tailwind CSS, забезпечує високий рівень продуктивності, безпеки та зручності користування інтернет-магазином.

Метою роботи є проектування та розробка повнофункціонального інтернет-магазину, який буде інтегрувати сучасні веб-технології для забезпечення високої швидкості завантаження, зручності для користувачів і безпечної обробки даних. Інтернет-магазин повинен включати основні функціональні можливості, такі як каталог товарів, система управління замовленнями, персоналізація для користувачів, а також інтеграція з платіжними системами та аналітичними інструментами для ефективного управління бізнесом.

Об'єктом дослідження є процеси створення та функціонування інтернет-магазину з використанням сучасних веб-технологій, **а предметом - технології та методи розробки програмного забезпечення для** створення високоякісних веб-платформ. Особливу увагу приділено використанню Next.js для оптимізації продуктивності, MongoDB для ефективного зберігання даних, Firebase для управління користувачами та автентифікації, Strapi для гнучкого управління контентом, а також Tailwind CSS для адаптивного дизайну.

**Практичне значення роботи полягає у створенні ефективного веб-продукту, який** дозволить власникам інтернет-магазинів зручно керувати асортиментом товарів, обробляти замовлення, забезпечувати безпеку та конфіденційність даних клієнтів, а також здійснювати аналітику для покращення бізнес-процесів. Розробка такого рішення дозволить знизити витрати часу на управлінські завдання та покращити взаємодію з покупцями.

Очікуваними результатами є розробка та впровадження стабільного та масштабованого інтернет-магазину, який відповідатиме всім сучасним вимогам до функціональності, продуктивності та безпеки. Це забезпечить підвищення ефективності бізнес-процесів, оптимізацію роботи інтернет-магазину та покращення користувацького досвіду, що сприятиме досягненню стратегічних цілей бізнесу.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРЕДМЕТНОЇ ОБЛАСТІ

- Визначення та характеристики веб-додатків

Веб-додаток - це розподілений додаток, де клієнтською частиною виступає веб-браузер, а серверною - веб-сервер [1]. Веб-додатки можуть бути написані скриптовими мовами (наприклад, JavaScript, Perl, PHP) або мовами високого рівня (такими як C#, Ruby, Swift), скомпільованими

під відповідну операційну систему. Вони **працюють на стороні веб-сервера і призначені для створення інтерфейсу між користувачем і веб-сайтом.**

**Веб-додаток - це комп'ютерна програма, яка працює в браузері, зберігання та обробка інформації при такій організації обчислень відбувається на віддаленому сервері, а веб-переглядач служить програмою-клієнтом і призначеним для користувача інтерфейсом (рис. 1.1) [2].**

### Рис.1.1. Схема функціонування веб-додатку.

Мобільний додаток - це програма, яка працює безпосередньо на мобільному пристрої, тоді як веб-додаток є програмним забезпеченням, яке функціонує через веб-браузер, виконуючи свої завдання онлайн.

Веб-додатки можуть варіюватися від простих, таких як контактні форми на вебсайтах, до складних, як текстові процесори або багатокористувацькі онлайн-ігри, доступні через браузер. Вони часто зберігають дані в хмарі, що дозволяє користувачам взаємодіяти з ними без необхідності зберігати файли на своїх пристроях. Наприклад, Google Docs - це веб-додаток, що дозволяє зберігати документи в хмарному середовищі та здійснювати редагування, а також надавати можливість завантаження файлів на локальні пристрої. З останніх нововведень, технологія AJAX використовується для створення більш чутливих і швидких веб-додатків. Сучасними прикладами таких додатків є Workspace (раніше G Suite) та Microsoft Office 365. Крім того, мобільні додатки, такі як Facebook, онлайн-банкінг або Dropbox, підключаються до Інтернету та були розроблені для зручного використання мобільних мереж.

Основні властивості **веб-додатків Веб-додаток отримує запит від користувача, обробляє його, виконує необхідні обчислення, генерує веб-сторінку і надсилає її клієнту.** Для передачі даних **через мережу** зазвичай використовується протокол HTTP. Якщо база даних знаходиться на іншому сервері або в іншому веб-додатку, веб-додаток може виступати клієнтом, звертаючись до цієї бази для отримання необхідної інформації.

Розглянемо основні характеристики веб-додатків:

#### 1. Незалежність від операційної системи клієнта

Веб-додаток розробляється один раз і може працювати на різних операційних системах клієнта. Однак різні реалізації HTML, CSS, DOM в різних браузерах можуть створювати труднощі при розробці та подальшій підтримці веб-додатків.

#### 2. Простота використання

Веб-додатки зазвичай орієнтовані на зручний і інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко освоювати їх без складних налаштувань чи інструкцій.

#### 3. Доступність з будь-якого пристрою

Веб-додатки доступні **з будь-якого пристрою, підключеного до Інтернету, що дозволяє користувачам працювати з додатком з різних місць, використовуючи різні пристрої - комп'ютери, смартфони, планшети.**

#### 4. Масштабованість і оновлення

**Веб-додатки можуть бути легко масштабовані**, щоб обробляти більший обсяг користувачів або даних. Оновлення веб-додатка відбуваються на сервері, що дозволяє користувачам автоматично отримувати останні версії без необхідності завантажувати нові оновлення. Технології, такі як Next.js, MongoDB, Firebase, Strapi і Tailwind CSS, пропонують сучасні підходи для створення та розробки веб-додатків, що підвищують ефективність розробки та забезпечують зручність для користувачів.

1. Next.js - фреймворк для створення веб-додатків на основі React, який підтримує серверний рендеринг і статичні сайти. Він дозволяє швидко створювати продуктивні веб-сторінки, де сторінки попередньо рендеряться на сервері, що покращує швидкість завантаження.

2. MongoDB - документно-орієнтована база даних, яка зберігає дані у форматі JSON. Це дозволяє легко масштабувати додатки і зберігати великі обсяги неструктурованих даних, що робить MongoDB ідеальним вибором для веб-додатків, які потребують гнучкої структури даних.

3. Firebase - платформа для швидкої розробки мобільних та веб-додатків з інтегрованими сервісами, такими як автентифікація користувачів, хмарне зберігання даних і повідомлення в реальному часі. Firebase допомагає розробникам швидко будувати і масштабувати додатки без необхідності турбуватися про сервери.

4. Strapi - безголовий CMS, який дозволяє створювати гнучкі API для управління контентом веб-додатків. Strapi ідеально підходить для розробників, які хочуть мати повний контроль над тим, як дані зберігаються і обробляються в додатках, але при цьому мати зручний інтерфейс для управління контентом.

5. Tailwind CSS - CSS-фреймворк, що дозволяє швидко створювати адаптивні інтерфейси за допомогою утилітних класів. Завдяки цьому розробники можуть уникнути написання кастомного CSS, що значно прискорює розробку та дозволяє створювати ефективні, гнучкі дизайни.

Ці технології дозволяють швидко та ефективно будувати масштабовані, швидкі та зручні веб-додатки, оптимізуючи процеси розробки та надаючи розробникам потужні інструменти для роботи з даними, інтерфейсами та сервісами.

Основною характеристикою сучасних Інтернет-додатків є їх надання у вигляді сервісів, що призводить до значних змін у бізнес-моделях компаній, які займаються розробкою програмного забезпечення. Перехід від традиційного софту-артефакту до софту-сервісу вимагає постійної підтримки та оновлення продукту.

Google навчився автоматизувати процеси обслуговування мереж, системне адміністрування **я** **що** **є** ключово**ю** частино**ю** **його** **ціново** **переваги** **перед**

**конкурентами.** Google **оновлює** своє оточення в залежності **від того, що новенького виявить користувач в своєму комп'ютерному оточенні сьогодні".** Відстежування поведінки користувачів в реальному часі дозволяє бачити, які нові властивості використовуються і як вони використовуються - і це це одна ключова складова успіху технології. Веб-розробник одного з розкритих мережевих сервісів **відзначає: "ми додаємо дві-три нових властивості в різні частини сайту щодня, і якщо користувачам вони не подобаються - ми відмовляємося від цих нововведень. Якщо подобаються - упроваджуємо на всьому сайті"** [3].

Веб-додатки мають значну перевагу в прикладному використанні, оскільки існує безліч програм і сервісів, які дозволяють кожному створювати корисні додатки для особистих потреб і розваг, **причому абсолютно безкоштовно.** **Наприклад, популярний Gmail є повноцінним поштовим клієнтом, який працює на вашому комп'ютері, пропонує безліч додаткових функцій і виконує всі завдання, які виконують інші поштові клієнти.** **Bloglines - це веб-додаток для читання новин, який конкурує з традиційними програмами цього типу і навіть має перевагу над ними.** **Ці веб-додатки функціонують на сервері, а їх користувацький інтерфейс (UI) відображається у вигляді веб-сторінок, що доступні через браузер.**

- Технології хмарних обчислень

Хмарні технології є концепцією, яка передбачає зберігання та обробку даних на віддалених серверах [3]. Вперше ця ідея була озвучена в 2008 році. Під терміном «хмара» мається на увазі центр обробки даних (ЦОД), сервер чи мережа серверів, де зберігаються дані та програми, доступ до яких здійснюється через Інтернет. Це дає можливість користувачам отримувати доступ до ресурсів без необхідності зберігати їх на власних пристроях (рис. 1.2).

Рис. 1.2. - Центр обробки даних

Хмарні технології охоплюють велику кількість аспектів, таких як програмне забезпечення, інфраструктура, платформи, дані, робочі місця та інші технологічні компоненти. Основною метою таких технологій є забезпечення користувачів можливістю доступу до ресурсів і обробки даних на відстані. Хостинг - це послуга з розміщення клієнтського обладнання на території провайдера з підключенням до високошвидкісних каналів зв'язку [2]. Хмарні технології дозволяють використовувати програми без їхнього встановлення на комп'ютері користувача. Це надає змогу працювати з особистими файлами з будь-якого комп'ютера або пристрою, який підключений до Інтернету. Окрім того, хмарні технології значно підвищують ефективність обробки, управління та роботи з інформацією завдяки централізації даних, що дозволяє знижувати витрати на локальне зберігання та обробку.

До основних прикладів хмарних технологій можна віднести сервіси, такі як Gmail, Meta (Facebook), Google Drive тощо. Наприклад, для відправлення електронної пошти через Gmail достатньо мати доступ до Інтернету. Для цього не потрібно спеціального програмного забезпечення чи серверів - все знаходиться в хмарі. Користувач може отримати доступ до своїх листів, файлів або іншої інформації з будь-якого пристрою, який має доступ до Інтернету. Хмарні технології надають послуги хостингу через Інтернет, і їхня популярність постійно зростає, оскільки вони дозволяють підприємствам та окремим користувачам отримувати доступ до потужних ресурсів без необхідності купувати та обслуговувати власну інфраструктуру.

Інфраструктура хмарних технологій ґрунтується на наданні комп'ютерних ресурсів через Інтернет у вигляді онлайн-сервісів. Ці сервіси можна поділити на три основні категорії:

1. Програмне забезпечення як сервіс (SaaS) - дає змогу користуватися програмами через Інтернет без їхнього встановлення на локальні пристрої.
2. Платформа як сервіс (PaaS) - надає платформи для розробки і тестування програмного забезпечення, включаючи бази даних, середовища для програмування та інше.
3. Інфраструктура як сервіс (IaaS) - надає доступ до основних ресурсів, таких як віртуальні машини, сервери, мережеві ресурси, на яких можна розгорнути власні додатки або сервіси.

#### Програмне забезпечення як сервіс (SaaS)

Модель SaaS передбачає постачання користувачам апаратної інфраструктури та програмного забезпечення через інтерфейсний портал. Вона охоплює широкий спектр послуг, від електронної пошти до управління запасами та обробки баз даних. Основною перевагою SaaS є доступ до сервісів з будь-якого місця світу. Модель також забезпечує високу гнучкість і масштабованість, оскільки всі оновлення та технічну підтримку виконує провайдер. SaaS допомагає зменшити витрати на придбання і підтримку локального ПЗ та інтегрується з іншими хмарними технологіями, спрощуючи робочі процеси.

#### Платформа як сервіс (PaaS)

PaaS пропонує набір інструментів для розробки та програмних продуктів, розміщених на інфраструктурі провайдера. Розробники мають можливість створювати власні додатки, використовуючи платформу провайдера через Інтернет. Для доступу до послуг PaaS можуть використовуватись шлюзи, портали, API або спеціальне програмне забезпечення, встановлене на пристроях клієнта.

#### Інфраструктура як сервіс (IaaS)

IaaS - це модель, що забезпечує доступ до віртуальних серверів та інших ресурсів через API. Користувачі можуть запускати, зупиняти, налаштовувати віртуальні сервери та системи зберігання даних відповідно до своїх потреб. Вартість залежить від фактично використаних ресурсів. Цю модель іноді називають "комунальні обчислення" [5]. Хмарні сервіси IaaS дозволяють ефективно управляти значними інфраструктурами, обслуговувати велику кількість користувачів і забезпечують повну залежність від провайдера.

#### Переваги хмарних технологій:

1. Користувач сплачує лише за використані послуги;
2. Економія на придбанні, підтримці та оновленні програмного забезпечення й обладнання;
3. Масштабованість, висока відмовостійкість і безпека;
4. Автоматичне виділення ресурсів відповідно до поточних потреб;
5. Технічна підтримка й оновлення програмного забезпечення виконується провайдером.

#### Недоліки хмарних технологій:

1. Залежність збереження даних від надійності провайдера;
2. Необхідність мати стабільний і швидкісний доступ до Інтернету;
3. Ризик втрати даних у разі відсутності резервного копіювання з боку провайдера;
4. Передаючи свої дані провайдеру, користувач втрачає частину контролю над ними.

- Використання Next.js для створення веб-додатків на базі React

Next.js - фреймворк для серверного рендерингу веб-додатків на основі React. Це один із найпопулярніших інструментів, що розширює можливості JavaScript-бібліотеки React. Next.js дає змогу створювати продуктивні, оптимізовані для пошукових систем сайти, які будуть зручними не тільки для пошукових роботів, але й для користувачів. Фреймворк максимально використовує можливості React, підтримує TypeScript, складну маршрутизацію та надає багато корисних функцій «з коробки». Next.js дозволяє як створювати нові веб-додатки будь-якої складності, так і модернізувати існуючі сайти, роблячи їх швидшими та більш стабільними.

React - це JavaScript-бібліотека для розробки інтерфейсів користувача (UI). Вона є однією з найпопулярніших технологій для створення сучасних інтерфейсів, проте не є повноцінним фреймворком. React відповідає за зовнішній вигляд програми, тоді як інші її компоненти розробляються з використанням додаткових бібліотек чи фреймворків, таких як Next.js. Саме Next.js додає підтримку гібридного статичного та серверного рендерингу, а також інші корисні функції, що полегшують процес розробки. Для інших популярних JavaScript-бібліотек також існують фреймворки серверного рендерингу. Наприклад, у Vue.js, основного конкурента React, є Nuxt.js, який виконує аналогічні функції. Основна перевага Next.js - вбудована підтримка серверного рендерингу (SSR), що підвищує продуктивність і покращує SEO. Завдяки SSR HTML-сторінки генеруються на сервері й передаються користувачу в готовому вигляді. Це дозволяє зменшити час завантаження сторінки, покращуючи взаємодію з користувачами. Клієнт, отримуючи готову HTML-сторінку, уникає необхідності завантажувати та рендерити окремі компоненти, що значно підвищує швидкість роботи додатка (рис. 1.3).

Рис. 1.3. - React та Next.js

Інші переваги Next.js:

1. Швидкість завантаження. Завдяки серверному рендерингу сайти на Next.js працюють значно швидше, ніж клієнтські React-додатки;
2. Статична генерація. Фреймворк підтримує функції експорту статичних сайтів;
3. Інтуїтивність. Розробникам, які вже працювали з React, легко адаптуватися до Next.js;
4. Автоматичне розділення коду. Кожна сторінка завантажується лише з необхідними компонентами, що зменшує розмір бандлу;
5. Інтеграція API. Next.js дозволяє створювати внутрішні API через вбудовані маршрути API;
6. Підтримка сучасних технологій. Фреймворк інтегрує CSS, JSX, TypeScript, а також надає легкий спосіб додавання плагінів;
7. Компонентний підхід. Next.js підтримує всі переваги React, включно з модульністю та системою управління станами.

Недоліки Next.js

Єдиним помітним недоліком Next.js є його певна «самодостатність» як фреймворка. Він має встановлені методи й інструменти, які необхідно використовувати для розробки додатків. Проте ці можливості зазвичай відповідають потребам більшості проектів.

Next.js найкраще підходить для створення:

1. оптимізованих лендінгів і домашніх сторінок;
2. сторінок, орієнтованих на органічний пошуковий трафік;
3. високопродуктивних і стабільних веб-додатків, які потребують швидкого рендерингу.

Цей фреймворк забезпечує ідеальний баланс між простотою використання та широкими можливостями для розробки сучасних веб-додатків.

Next.js є потужним інструментом для створення сучасних веб-додатків, який поєднує всі переваги React із додатковими можливостями для серверного рендерингу, оптимізації продуктивності та покращення SEO. Цей фреймворк не лише спрощує розробку, але й допомагає створювати сайти, які задовольняють як користувачів, так і пошукові системи. Завдяки своїй гнучкості та широкому функціоналу, Next.js підходить як для невеликих проєктів, так і для масштабних систем із високими навантаженнями. Використання Next.js дозволяє зекономити час розробки, підвищити якість кінцевого продукту та забезпечити комфортний досвід роботи як для розробників, так і для користувачів. Це ідеальний вибір для тих, хто прагне створити продуктивний, стабільний і сучасний веб-додаток.

- Відмінність між веб-сайтом та веб-додатком

Сайт не завжди є веб-додатком і зазвичай створюється для інформаційних цілей. Він може бути як статичним, так і динамічним, але його головне завдання - надання інформації користувачам [8]. Особливістю, яка відрізняє веб-додаток від веб-сайту, є інтерактивність. Це означає, що користувач стає активним учасником у роботі з сайтом. Наприклад, користувач веб-додатку може шукати необхідну інформацію, заповнювати форми, здійснювати платежі через особистий кабінет тощо.

За складністю реалізації подібні рішення є значно складнішими, ніж інформаційні сайти, а отже, їхня розробка коштує дорожче. Наприклад, App Store. (рис. 1.3).

Рис. 1.4. - Головна сторінка App Store

На домашній сторінці App Store, яка є гібридом, можна побачити багато інформації: описи додатків, їхні рейтинги, відгуки користувачів, скріншоти, відео-презентації та категорії - усе це є прикладами інформаційного наповнення. Однак головною метою користувача є взаємодія. Він може виконати пошук, переглянути популярні або рекомендовані додатки, ознайомитися з деталями конкретного продукту, придбати чи завантажити його, а також керувати підписками або оновленнями.

Усі ці дії є прикладами інтерактивності. Якщо сайт лише представляє інформацію, необхідно знайти баланс між інформаційністю та інтерактивністю. Якщо взаємодії занадто багато, то це вже інтерактивний сайт.

Сайти й веб-додатки суттєво відрізняються за своїм призначенням і функціональністю. Інформаційні сайти орієнтовані на надання статичної або динамічної інформації, такої як новини, статті, довідники чи оголошення. Вони не потребують складної логіки чи взаємодії з користувачем, тому їхня розробка є менш трудомісткою. З іншого боку, веб-додатки забезпечують значно вищий рівень інтерактивності, дозволяючи користувачам виконувати різноманітні дії, наприклад, здійснювати покупки, бронювати послуги або працювати з персоналізованими даними. У сучасному цифровому світі все більше компаній обирають веб-додатки, оскільки вони забезпечують багатofункціональність і зручність для користувачів. Такі рішення часто використовуються в електронній комерції, соціальних мережах і корпоративних системах. Крім того, веб-додатки дозволяють легко масштабувати функціонал у відповідь на змінні потреби бізнесу. Наприклад, електронна комерція потребує інтеграції з платіжними системами, розширених пошукових функцій і персоналізованих рекомендацій.

Додатковою перевагою веб-додатків є можливість роботи з великими обсягами даних у реальному часі. Інформаційні сайти, на відміну від них, переважно обмежуються статичними даними, які не змінюються без участі адміністратора. Наприклад, новинні портали оновлюють інформацію лише після публікації нових матеріалів. Натомість веб-додатки можуть автоматично оновлювати дані, залежно від дій користувачів або інтеграції з іншими системами. Рівень взаємодії також відрізняється. У той час як інформаційні сайти пропонують обмежений набір функцій, веб-додатки орієнтовані на створення інтуїтивного інтерфейсу, який дозволяє користувачам отримати доступ до складних функцій без спеціальних знань.

З розвитком технологій межа між сайтами та веб-додатками стає все більш розмитою. Наприклад, багато сучасних сайтів пропонують

інтеграцію елементів взаємодії, таких як форми зворотного зв'язку або персоналізовані рекомендації. Під час розробки проекту важливо враховувати потреби аудиторії. Якщо користувачам потрібна лише інформація, то створення простого сайту буде достатнім рішенням. У випадках, коли необхідно забезпечити постійний обмін даними між користувачем і системою, варто обирати веб-додаток. Також слід врахувати технічні вимоги. Розробка веб-додатків потребує значно більше ресурсів, ніж створення інформаційного сайту. Це стосується як фінансових витрат, так і часу, необхідного для реалізації проекту. Проте отримані результати часто компенсують ці зусилля, адже інтерактивні рішення мають значно ширші можливості для розвитку бізнесу та покращення користувацького досвіду. Із такими можливостями веб-додатки стають ідеальним вибором для проектів, які потребують гнучкості та динамічності, тоді як сайти залишаються ефективним інструментом для донесення інформації до широкої аудиторії.

Враховуючи цю інформацію, можна зробити висновок, що веб-додатком є певний тип сайту, а саме динамічний сайт, який максимально інтерактивний, а інформативна складова відіграє другорядну роль. Хоча можливі винятки, наприклад розважальні інформаційні сайти, які надають цікаву інформацію та пропонують користувачам інтерактивні можливості для взаємодії з цією інформацією.

## РОЗДІЛ 2. АНАЛІЗ ТА ПРОЕКТУВАННЯ СТРУКТУРИ ІНТЕРНЕТ-МАГАЗИНУ

- Дослідження специфіки функціонування та впровадження інтернет-магазинів

Інтернет-магазини значно спрощують процес пошуку необхідних товарів, ознайомлення з їх характеристиками, оформлення замовлення, отримання відгуків від інших покупців, а також, за потреби, підтримки щодо налаштування та використання придбаних виробів. Значущість цієї теми полягає в тому, що інтернет-магазини наразі є найпоширенішою формою електронної комерції. Вони зазвичай мають структуровані каталоги з докладним описом товарів і зазначенням цін, що сприяє залученню потенційних клієнтів, полегшує процес вибору та сприяє здійсненню покупки. Багато користувачів шукають в Інтернеті точну й актуальну інформацію про компанії, товари чи послуги, що їх цікавлять. У наш час важко уявити велику організацію без офіційного веб-сайту. Наявність якісного онлайн-представництва не лише покращує імідж компанії, а й зміцнює її позиції на ринку, формуючи довіру клієнтів і позитивне враження.

Інтернет-магазин полегшує пошук потрібного товару, перегляд його властивостей, замовлення товару, отримання зворотного зв'язку з іншими споживачами цього товару, а також, якщо необхідно, отримання підтримки щодо налаштування та експлуатації придбаного товару. Актуальність цього питання полягає в тому, що найпопулярнішим видом віртуальної комерції сьогодні є інтернет-магазини. Інтернет-магазини зазвичай містять чіткі та розширені каталоги з достатнім описом і вказівкою цін, що дозволяє викликати інтерес у **потенційного покупця, допомогти йому зробити вибір і в кінцевому підсумку здійснити покупку.** Багато людей шукають в Інтернеті детальну та актуальну інформацію, на якій можна дізнатися про цікаві компанії, **товари та послуги. Зараз важко уявити велику компанію без Інтернет-сайту.** Наявність хорошого представника мережі в компанії не тільки підвищує позитивний імідж компанії, але й сприяє надійності та стійкості компанії, створюючи певний імідж та враження.

На сьогодні **2,71 млрд людей по всьому світу** здійснюють покупки онлайн через спеціалізовані платформи електронної комерції або соціальні мережі, що становить 33% населення планети, що на 2,7% більше порівняно з минулим роком. У 2025 році кількість онлайн-покупців зросте до 2,77 млрд завдяки розвитку електронної комерції, збільшенню проникнення інтернету та зручності покупок. Найбільшу кількість онлайн-покупців мають Китай (**915,1 млн**) і США (**270,1 млн**). Очікується, що до кінця 2024 року 20,1% роздрібних покупок буде здійснено онлайн, а до 2027 року цей показник досягне 22,6%. У 2024 році **загальний обсяг продажів електронної комерції** перевищить **\$ 6,3 трлн, що на 8,76% більше, ніж у 2023 році** [24].

Рис. 2.1. - Статистика покупок

**3 2021 року частка роздрібних онлайн-покупок зростає на 0,32% щорічно, що** вказує на необхідність ритейлерам інвестувати в онлайн, аби зберегти конкурентоспроможність. **Станом на 2024 рік у світі налічується 26,6 млн сайтів електронної комерції, що на 3,83% більше** порівняно з попереднім роком. Майже половина всіх цих сайтів знаходиться в США. Більшість інтернет-магазинів працюють на платформі Shopify (19,07%) та Wix (11,84%).

За перші шість місяців 2024 року загальний обсяг онлайн-продажів у США склав \$579,45 млрд, і до кінця року експерти прогнозують, що цей показник досягне \$1,26 трлн, а до 2027 року - \$1,72 трлн. У той час онлайн-продажі складатимуть близько 22,6% від загального обсягу роздрібних продажів США.

Рис. 2.2. - Обсяг онлайн-продажів

Соціальні мережі, де зареєстровано понад 3,5 млрд користувачів, є ідеальним майданчиком для електронної комерції. У 2024 році риннок соціальної комерції оцінюється в \$1,69 трлн, що на 30,81% більше, ніж у попередньому році. За прогнозами, до 2026 року цей ринок досягне \$2,9 трлн, а до 2030 року він може зрости до \$8,5 трлн. Найпопулярнішими платформами для соціальної комерції є **Facebook, Instagram, Pinterest та TikTok.**

Рис. 2.3. - Статистика соц. мереж для розміщення комерції

Найактивніші покупці через соціальні мережі - люди віком від 18 до 34 років, з яких 73% здійснювали покупки через ці канали. Для порівняння, серед осіб віком від 65 років лише 25% використовували соціальні мережі для покупок.

Рис. 2.4. - Статистика покупців через соц. мережі

Загалом, інтернет-магазини стають невід'ємною частиною сучасної комерції, демонструючи значне зростання обсягів продажів, що підтверджується зростанням числа онлайн-покупців і зростанням частки онлайн-продажів у загальних роздрібних продажах. Особливо це стосується молодшої аудиторії, яка активно користується платформами електронної комерції для здійснення покупок. Відкриття інтернет-магазину дозволяє компанії вийти на новий рівень розвитку бізнесу, зробивши свій асортимент більш доступним для широкої аудиторії клієнтів. Інтернет створює широкі можливості для реалізації товарів через інтернет-магазини, які забезпечують зручність та економію часу для споживачів.

Переваги інтернет-магазинів включають:

1. Цілодобова доступність. Інтернет-магазини працюють 24/7 без вихідних, що дає змогу користувачам здійснювати покупки у будь-який зручний час.
2. Широкий асортимент. Великий вибір товарів забезпечує задоволення різноманітних потреб клієнтів.
3. Лояльність до клієнтів. Надійні інтернет-магазини прагнуть створити позитивний досвід для користувачів, забезпечуючи високий рівень сервісу.
4. Зручна навігація. Інтуїтивний інтерфейс дозволяє легко знаходити потрібні товари та оформляти замовлення.
5. Гнучка доставка. Різні способи доставки дають можливість отримати товар у найкоротші строки.
6. Глобальний доступ. Користувачі можуть відвідувати інтернет-магазини з будь-якого місця, включаючи зарубіжні ресурси.

З кожним роком кількість інтернет-магазинів зростає завдяки їхній вигоді для кінцевого споживача. Інтернет-магазини економлять час та фінансові ресурси, працюють автоматично і не потребують участі продавця. Відсутність необхідності у великому складі дозволяє зменшити витрати на зберігання товарів. На відміну від звичайних магазинів, які обмежені локальною аудиторією, інтернет-магазини охоплюють клієнтів з різних регіонів та країн.

Важливість створення веб-сайту зумовлена такими факторами:

- оперативне поширення інформації серед великої кількості людей;
- зміцнення репутації компанії та підвищення її впізнаваності;
- можливість організації зворотного зв'язку з клієнтами;
- покращення комунікації з філіями та партнерами;
- проведення маркетингових досліджень;
- залучення нових клієнтів через рекламу;
- збільшення відвідуваності ресурсу.

Отже, покупці прагнуть отримати інформацію про товари та здійснюють покупки відповідно до своїх потреб. Завдання власника інтернет-магазину полягає у реалізації продукції. Ефективність вирішення цього завдання залежить від правильної взаємодії з різними групами клієнтів.

- Визначення функціональних і **нефункціональних вимог** **Функціональні та нефункціональні вимоги до програмного забезпечення є ключовими критеріями для** оцінки відповідності нової системи вимогам замовника. Однак наявність чітко сформульованих вимог важлива не тільки для оцінки результатів проєкту, а й має суттєве значення на етапі розробки програмного продукту.

Функціональні вимоги - це вимоги, що визначають, яким чином **продукт або послуга повинні задовольняти потреби клієнтів** [9]. Вони охоплюють **функції та функціональні можливості у варіантах використання, що документують, як користувачі будуть взаємодіяти з продуктом або послугою.**

Зважаючи на цілі системи, у таблиці 1.1 були сформульовані **функціональні вимоги у вигляді user story та user case.**

**User story - це короткий опис** дій користувача, які він здійснюватиме під час використання вебсайту чи програмного забезпечення [10].

Зазвичай історія користувача викладається у такому **форматі: "Я, як користувач, хочу виконати певну дію, щоб досягти певного результату"**.

**Use case** представляє собою **вимогу у вигляді взаємодії користувача з системою** [11]. Варіанти **використання завжди формулюються з урахуванням конкретної мети користувача. Кожен варіант використання має включати користувача та дієслово. Наприклад, "онлайн-покупець" є користувачем, а "дати товар у кошик" - дієсловом.**

Таким чином, користувачі веб-орієнтованої системи продажу іграшок можуть мати такі ролі:

- покупець (customer);
- адміністратор (administrator).

**No Тип вимоги Роль користувача Формулювання користувачької вимоги**

- | No | Тип вимоги | Роль користувача | Формулювання користувачької вимоги   |
|----|------------|------------------|--|
| 1  | User story | customer         | Користувач може створити обліковий запис, зазначивши основні дані: ім'я, електронну пошту, пароль тощо.              |
| 2  | User story | customer         | Користувач може увійти в систему за допомогою електронної пошти та паролю.   |
| 3  | User story | administrator    | Адміністратор має можливість додавати нові товари, змінювати опис, ціни, фотографії тощо.                            |
| 4  | User story | administrator    | Користувач може ознайомитися зі списком поточних акцій, знижок і спеціальних пропозицій, які доступні для товарів.   |
| 5  | User case  | customer         | Користувач має можливість вибрати спосіб доставки та оплати, а також вказати адресу для доставки.                    |
| 6  | User case  | customer         | Користувач може додавати обрані товари до кошику для подальшого оформлення замовлення.                               |
| 7  | User story | customer         | Користувач може переглядати деталі товарів (фотографії, опис, ціна) перед додаванням до кошику.                      |
| 8  | User case  | customer         | Користувач може редагувати кількість товарів у кошику або видаляти їх.   |
| 9  | User case  | customer         | Користувач може перевірити вміст кошику перед завершенням покупки та перейти до оформлення замовлення.               |
| 10 | User case  | customer         | Користувач може завершити покупку, вказавши способи доставки та оплати, після чого отримує підтвердження замовлення. |
| 11 | User story | administrator    |  |

Адміністратор може переглядати всі замовлення та змінювати статус обробки (наприклад, обробка, відправлено).

- |    |            |               |  |
|----|------------|---------------|--|
| 12 | User case  | customer      | Користувач може переглядати історію своїх замовлень через обліковий запис. |
| 13 | User story | administrator |  |

Адміністратор може оновлювати інформацію про користувачів, змінювати їх статуси чи банити за порушення правил.

- |    |            |               |   |
|----|------------|---------------|---|
| 14 | User case  | customer      | Користувач може отримувати повідомлення про стан свого замовлення на електронну пошту чи в системі. |
| 15 | User story | administrator |   |

Адміністратор має можливість видаляти товари з каталогу.

- |    |           |          |   |
|----|-----------|----------|---|
| 16 | User case | customer | Користувач може використовувати фільтри для пошуку товарів за категоріями, розмірами, ціною тощо. |
|----|-----------|----------|---|

- 17 User case customer Користувач може переглядати список рекомендацій товарів на основі своїх попередніх покупок або переглядів.  
18 User story administrator

Адміністратор може додавати, редагувати та видаляти категорії товарів для покращення навігації в магазині.

- 19 User story administrator

Адміністратор може переглядати статистику продажів, популярних товарів і тенденцій.

- 20 User case customer Користувач може переглядати відгуки інших покупців на товари перед покупкою.

Таблиця 1.1 - **Функціональні вимоги користувача до веб-орієнтованої системи продажу** одягу та взуття

Нефункціональні вимоги, відповідно, сприяють ефективній роботі програмного забезпечення [12]. Хоча **ці вимоги не є обов'язковими для системи**, їх виконання зазвичай покращує загальну якість, швидкість та ємність програмного забезпечення. **Нефункціональні вимоги дозволяють користувачам** зручніше використовувати функції програмного забезпечення. **До нефункціональних вимог відносяться [13, 14]:**  
**1. Масштабованість та продуктивність.** **Продуктивність визначає кількість запитів, які система** здатна **обробити за секунду.** **Масштабованість додатку** характеризує максимальні **робочі навантаження, при яких система все ще відповідає вимогам щодо продуктивності.** **2. Сумісність.** **Сумісність системи оцінюється її здатністю працювати в різних середовищах.** **Система повинна бути сумісною з** мобільними додатками для старих версій **операційних систем та підтримувати кросбраузерність.** **3. Безпека.** **Система повинна бути здатною захищатися від шкідливого програмного забезпечення та несанкціонованого доступу.**  
**4. Доступність та надійність.** **Доступність системи** означає її готовність до роботи для користувача в будь-який момент часу. Вимога доступності нерозривно пов'язана з надійністю. Як правило, надійність характеризує період часу, протягом якого система працює **без збоїв.**  
**5. Локалізація.** **Нефункціональна вимога локалізації визначає, наскільки система адаптована до місцевих вимог користувачів.** Оскільки розроблена система орієнтована на користувачів в Україні, вона повинна підтримувати українську мову.

- Проєктування архітектури інтернет-магазину

Проєктування архітектури інтернет-магазину є важливим етапом у розробці сучасних веб-застосунків. Ефективна архітектура дозволяє забезпечити високу продуктивність, масштабованість, безпеку та зручність у використанні. У рамках магістерської роботи буде розглянуто побудову інтернет-магазину з використанням сучасних технологій, таких як Next.js 13, MongoDB, Firebase, Strapi та Tailwind CSS. Сучасний підхід до розробки інтернет-магазину вимагає використання передових інструментів, які забезпечують продуктивність завдяки Next.js 13 із вбудованим рендерингом на стороні сервера (SSR) та статичним рендерингом (SSG), масштабованість завдяки MongoDB як NoSQL базі даних, що дозволяє гнучко управляти даними та легко масштабувати їх, реальновчасний функціонал через Firebase, який пропонує можливості для автентифікації користувачів, зберігання файлів та сповіщень, контент-менеджмент із використанням Strapi для швидкого створення API та управління контентом, а також зручність дизайну завдяки Tailwind CSS, який дозволяє створювати адаптивні інтерфейси з мінімальними зусиллями. Архітектурна модель включає кілька ключових компонентів: клієнтська частина інтернет-магазину реалізована за допомогою Next.js 13, що дозволяє використовувати компонентний підхід для створення UI, оптимізувати продуктивність завдяки SSR і SSG, а також забезпечити інтуїтивний UX із підтримкою динамічних маршрутизацій. Серверна частина базується на Next.js API Routes для обробки запитів, Strapi для управління контентом і MongoDB для зберігання даних. Strapi використовується для налаштування категорій товарів, управління користувачами та контентом, а також для створення API для інтеграції з клієнтською частиною. MongoDB застосовується для зберігання інформації про товари, користувачів і замовлення, підтримуючи документ-орієнтовану структуру для складних даних. Firebase використовується для забезпечення безпечної автентифікації користувачів, дозволяючи реалізувати реєстрацію та вхід за допомогою електронної пошти, Google або інших соціальних мереж, а також захист даних користувачів за допомогою Firebase Authentication. Tailwind CSS спрощує створення адаптивного інтерфейсу користувача завдяки попередньо налаштованим класам стилів, можливості швидкої адаптації під різні екрани та мінімізації необхідності написання власного CSS-коду. Інтеграція компонентів реалізована наступним чином: використання Mongoose для моделювання даних і взаємодії з MongoDB, обробка серверних запитів через API Routes, підключення Firebase SDK для автентифікації та зберігання файлів, наприклад, зображень товарів, використання Strapi API для отримання та оновлення контенту, а також інтеграція Tailwind для стилізації компонентів. Приклад функціональних модулів включає реалізацію авторизації з Firebase Authentication, перегляд, фільтрацію та сортування товарів з використанням MongoDB для зберігання інформації про товари, а також додавання товарів до кошику та оформлення замовлень із підтвердженням через Firebase.

Запропонована архітектура інтернет-магазину забезпечує баланс між продуктивністю, зручністю використання та масштабованістю.

Використання сучасних інструментів, таких як Next.js 13, MongoDB, Firebase, Strapi і Tailwind CSS, дозволяє створювати ефективний і гнучкий веб-застосунок для електронної комерції.

- Вибір платформи для реалізації проєкту

Для успішної реалізації проєкту був обраний комплекс сучасних інструментів і платформ, які забезпечують високу продуктивність, масштабованість, зручність у розробці та підтримці. Нижче описано ключові компоненти обраного стеку технологій і їх переваги в контексті даного проєкту.

#### 1. Next.js

Next.js є сучасним фреймворком для розробки веб-застосунків, який поєднує в собі можливості серверного **рендерингу (SSR) та статичної генерації сторінок (SSG).** **Це** дозволяє значно покращити продуктивність і **швидкість завантаження сторінок, що позитивно впливає на** досвід користувачів. Крім того, підтримка динамічних маршрутів, інтеграція з різними API та потужні інструменти для оптимізації, такі як автоматичний розподіл ресурсів і кодування, роблять Next.js ідеальним вибором для створення сучасного веб-додатка.

#### 2. MongoDB

У ролі бази даних було обрано MongoDB, яка є документно-орієнтованою нереляційною системою управління даними. Її гнучка структура дозволяє легко працювати з даними, що мають різну структуру, а також забезпечує високу продуктивність навіть при великих обсягах інформації. Вбудовані можливості горизонтального масштабування і реплікації забезпечують надійність та ефективність у випадках високого навантаження. MongoDB також має багатий набір інструментів для аналітики та інтеграції, що робить її зручною для використання в складних проєктах.

### 3. Firebase

Firebase був обраний для реалізації автентифікації, управління користувачами, обміну даними в реальному часі та хмарного зберігання. Ця платформа дозволяє легко впровадити безпечний функціонал для входу та реєстрації користувачів за допомогою популярних методів автентифікації, таких як електронна пошта, соціальні мережі або OTP-коди. Крім того, Firebase Realtime Database і Firestore дають змогу забезпечити обмін даними між клієнтами в реальному часі, що є важливим для інтерактивних застосунків.

### 4. Strapi

Strapi був обраний як headless CMS для управління контентом. Цей інструмент надає розробникам потужну платформу для створення API, а також зручний інтерфейс для контент-менеджерів. Завдяки своїй модульній архітектурі, Strapi дозволяє легко адаптуватися до змін вимог проєкту. Він підтримує інтеграцію з різними базами даних і сервісами, що робить його універсальним рішенням для роботи з контентом.

### 5. Tailwind CSS

Для стилізації веб-застосунку було обрано Tailwind CSS - утилітарну CSS-бібліотеку, яка надає широкі можливості для створення адаптивного дизайну. Tailwind дозволяє значно прискорити процес розробки інтерфейсів завдяки своїй системі класів, яка виключає необхідність написання додаткових стилів вручну. Крім того, бібліотека підтримує тему, кастомізацію, що робить її ідеальною для створення унікальних дизайнів. Таким чином, вибір зазначеного стеку технологій забезпечує високу якість реалізації проєкту. Завдяки потужним інструментам та інтеграціям, команда розробників зможе ефективно працювати з даними, створювати сучасний і адаптивний інтерфейс, а також забезпечити стабільність і продуктивність системи, навіть за умов зростання користувачької бази.

## РОЗДІЛ 3. РОЗРОБКА ФУНКЦІОНАЛУ ІНТЕРНЕТ-МАГАЗИНУ

### 1. Вибір інструментів для розробки

**Вибір засобів розробки є одним із найважливіших етапів проєктування програмного забезпечення, оскільки саме інструменти визначають функціональність, продуктивність і адаптивність системи.** Для розробки інтернет-магазину PixelMall було обрано Next.js, MongoDB, Firebase, Strapi та Tailwind CSS. **Кожен із цих засобів забезпечує високий рівень надійності, продуктивності та масштабованості, що робить їх ідеальним вибором для створення сучасних веб-додатків подібного масштабу.**

Next.js був обраний як основний фреймворк для фронтенд-розробки завдяки своїй здатності забезпечувати серверний рендеринг і генерацію статичних сайтів, що значно покращує продуктивність і SEO. Крім того, Next.js має потужну інтеграцію з React, що дозволяє створювати динамічні та інтуїтивно зрозумілі інтерфейси користувача. Його зручність у налаштуванні маршрутизації та підтримка API-роутів роблять його ідеальним вибором для розробки повнофункціональних веб-додатків.

MongoDB була обрана як база даних через її гнучку структуру документів і можливість обробляти великі обсяги даних. Її модель NoSQL забезпечує швидкість і ефективність доступу до даних, що є критично важливим для інтернет-магазину. MongoDB також підтримує масштабування, що дозволяє безперешкодно обробляти зростаючий обсяг інформації про товари, користувачів та замовлення.

Firebase надає надійні сервіси бекенд-інфраструктури, включаючи аутентифікацію, хмарне зберігання та реального часу базу даних. Це забезпечує швидку інтеграцію функціоналу користувачів, таких як реєстрація та авторизація, а також зберігання мультимедійних даних. Firebase є ідеальним вибором для забезпечення стабільності та швидкодії.

Strapi був обраний як головний CMS завдяки своїй відкритості, кастомізації та підтримці API на основі REST або GraphQL. Це дозволяє легко керувати контентом магазину, таким як описи товарів, категорії та зображення, забезпечуючи зручність для адміністраторів.

Tailwind CSS використовується для створення стильного й адаптивного дизайну інтерфейсу. Завдяки утилітарному підходу Tailwind CSS дозволяє швидко й ефективно створювати унікальні стилі, які відповідають сучасним стандартам веб-дизайну.

Додатковою перевагою використання обраних технологій є їх активна спільнота та велика кількість доступної документації, що значно полегшує процес розробки та вирішення можливих проблем. Усі інструменти мають широкі можливості інтеграції з іншими сервісами, що дозволяє безперешкодно впроваджувати нові функції, такі як аналітика, автоматизація маркетингу та персоналізація взаємодії з користувачами.

Важливо також зазначити, що обраний стек технологій сприяє забезпеченню високого рівня безпеки системи. Наприклад, Firebase пропонує потужні механізми захисту аутентифікації, а MongoDB підтримує шифрування даних як під час передачі, так і в стані спокою. Tailwind CSS, у свою чергу, мінімізує ризики, пов'язані з використанням застарілих CSS-бібліотек, забезпечуючи сучасний і безпечний підхід до дизайну.

Розробка PixelMall на основі таких інноваційних і надійних технологій гарантує конкурентоспроможність інтернет-магазину в динамічному середовищі сучасної електронної комерції. Завдяки адаптивності обраного стека, платформа здатна легко впроваджувати нові функції та відповідати вимогам зростаючого бізнесу, забезпечуючи при цьому високий рівень користувацького досвіду та задоволеності клієнтів.

Розробка інтернет-магазину PixelMall є стратегічно обґрунтованим кроком для створення сучасної, продуктивної та масштабованої платформи електронної комерції. Вибір цих технологій забезпечує високу надійність, гнучкість та ефективність системи, що є критично важливими для успішного функціонування онлайн-магазину в умовах швидко змінюваного ринку.

Система безпеки, підтримка масштабованості та можливість легкої інтеграції з іншими сервісами є важливими перевагами цього стека, що дозволяє легко впроваджувати нові функції та відповідати вимогам сучасної електронної комерції. Обраний стек технологій забезпечує не тільки конкурентоспроможність на ринку, але й дозволяє швидко адаптувати платформу до потреб зростаючого бізнесу та вимог клієнтів, що робить PixelMall готовим до викликів майбутнього.

2. Розробка серверної частини: **бази даних** Розробка бази даних **є важливим етапом створення інформаційної системи, адже вона визначає структуру зберігання даних, їх взаємозв'язки та механізми доступу.** Для інтернет-магазину PixelMall була спроектована реляційна база даних, яка забезпечує надійне **збереження, обробку та інтеграцію даних, необхідних для** роботи платформи. **Основна увага приділялася нормалізації даних, оптимізації запитів і забезпеченню цілісності інформації.** Завдяки цьому вдалося створити ефективну та стабільну систему для підтримки бізнес-процесів магазину.

### Рис. 3.1. - Users

Сутність користувачів (Users) є основною **в базі даних, оскільки вона представляє зареєстрованих осіб, які взаємодіють із системою.** **Таблиця користувачів містить унікальний ідентифікатор (id) як первинний ключ, а також основні атрибути:** ім'я (name), логін (login),

пароль (password), імейл (email) та роль (role). Для забезпечення безпеки **паролі зберігаються у зашифрованому вигляді, що** запобігає їх компрометації.

Рис. 3.2. - Addresses

Сутність Addresses **відповідає за збереження інформації про адреси користувачів. Таблиця містить такі атрибути, як унікальний ідентифікатор (id),** ідентифікатор користувача (userID), повне ім'я (fullName), адресу (address), місто (city), країну (country), поштовий індекс (postalCode), дату створення (createdAt) та дату оновлення (updatedAt). Ця структура дозволяє ефективно зберігати та обробляти адресні дані користувачів, що важливо для організації доставки товарів в інтернет-магазині.

Рис. 3.3. - Carts

Сутність Carts відповідає за збереження інформації про товари в кошику користувача. Таблиця містить такі атрибути, як унікальний ідентифікатор (\_id), ідентифікатор користувача (userID), ідентифікатор товару (productID), кількість (quantity), дату створення (createdAt) та дату оновлення (updatedAt). Ця структура дозволяє ефективно зберігати інформацію про товари, які користувачі додали до своїх кошиків, що необхідно для подальшого оформлення замовлення в інтернет-магазині.

Рис. 3.4. - Orders

Сутність Orders відповідає за збереження інформації про замовлення користувачів. Таблиця містить такі атрибути, як унікальний ідентифікатор (id), ідентифікатор користувача (userID), ідентифікатор адреси доставки (addressID), дата створення замовлення (createdAt), статус замовлення (status), сума замовлення (totalAmount), а також дату оновлення (updatedAt). Ця структура дозволяє зберігати всі необхідні дані для обробки та виконання замовлень, а також відслідковувати їхній статус до завершення доставки.

Рис. 3.5. - Products

Сутність Products відповідає за збереження інформації про товари в інтернет-магазині. Таблиця містить такі атрибути, як унікальний ідентифікатор (\_id), назву товару (name), опис товару (description), ціну (price), категорію (category), доступні розміри (sizes), інформацію про доставку (deliveryInfo), статус акції (onSale), знижку (priceDrop), посилання на зображення (imageUrl), дату створення (createdAt) та дату оновлення (updatedAt). Ця структура дозволяє зберігати детальну інформацію про кожен товар, що сприяє ефективній організації каталогу та покращує користувацький досвід під час покупок.

**Загальна структура бази даних інтернет-магазину була спроектована з урахуванням нормалізації, що дозволило уникнути дублювання даних і забезпечити їхню цілісність. Наприклад, інформація про продукти, категорії, замовлення та користувачів зберігається окремо, що дозволяє змінювати дані в кожній категорії або товарі без необхідності оновлення всіх пов'язаних записів. Зв'язки між колекціями реалізовані через зовнішні ключі, що забезпечує підтримку референційної цілісності та зручний доступ до даних,** а також дозволяє взаємодіяти з різними частинами системи без дублювання інформації.

**База даних інтегрується із системою** через API, що дозволяє ефективно взаємодіяти з даними у реальному часі. Використання Firebase для автентифікації та Strapi для управління контентом допомагає синхронізувати дані між клієнтською частиною та сервером, що покращує функціональність системи. Структура бази даних дозволяє працювати з даними на рівні об'єктів, **що значно спрощує розробку та зменшує кількість необхідного коду.**

**Автоматичне створення схем бази даних** та інтеграція з системою дозволяють зберігати **відповідність між моделями даних і їхньою реалізацією, що полегшує впровадження змін і підтримку системи** в умовах зростання обсягів даних та розширення функціоналу магазину.

### 3. Розробка ключових алгоритмів

**Розробка основних алгоритмів є** важливим етапом у створенні інформаційної системи інтернет-магазину, оскільки вони визначають **логіку роботи всіх функціональних модулів. Для** інтернет-магазину було спроектовано та реалізовано алгоритми, які відповідають за ключові операції, такі як додавання нового товару, **перегляд інформації про товар, авторизація користувача та підтвердження замовлення. Для** спрощення **розуміння цих алгоритмів** були створені **блок-схеми, які відображають послідовність дій і основні розгалуження логіки. Такі схеми допомагають наочно продемонструвати складні процеси,** що сприяє легшому аналізу і підтримці системи.

1. Першим розглянуто алгоритм реєстрації та входу користувача, який є ключовим для забезпечення доступу до персонального кабінету. **Блок-схема цього алгоритму демонструє, як отримані дані користувача перевіряються на наявність помилок, і якщо такі знайдено, система повертає користувача на форму** реєстрації або входу з відповідними повідомленнями про помилки (наприклад, некоректний email або занадто короткий пароль). У разі успішної перевірки пароля та email, система шифрує пароль користувача, зберігає його в базі даних та створює токен для автентифікації.

У разі входу користувача, система перевіряє наявність користувача в базі даних та порівнює введений пароль з зашифрованим у базі. Якщо пароль вірний, створюється новий токен для автентифікації. Токен передається на клієнтську сторону для подальших запитів, забезпечуючи доступ до закритих частин сайту. Завершується алгоритм передачею токена користувачу, що дозволяє йому продовжити здійснення покупок або керувати своїм акаунтом. Така послідовність гарантує безпеку користувача, зберігаючи його дані захищеними, а також забезпечує ефективну обробку та перевірку введених даних.

2. Другий розглянутий алгоритм роботи з продуктами, що включає додавання, оновлення та видалення продуктів. Блок-схема цього алгоритму демонструє, як користувач надає дані продукту, зокрема назву, опис, ціну та зображення, і як система перевіряє наявність всіх необхідних даних. Якщо зображення додається, використовується Firebase для зберігання зображення та отримання URL. Після цього дані продукту, разом із збереженим URL, зберігаються в базі даних MongoDB. При оновленні продукту система перевіряє, чи існує продукт у базі даних, і якщо так, оновлює старі дані новими. У разі видалення продукту система перевіряє наявність продукту в базі даних і видаляє його, а також видаляє зображення з Firebase, якщо це необхідно. Такий підхід гарантує правильне управління продуктами, забезпечуючи коректну перевірку даних на

кожному етапі та збереження їх у базі даних, що сприяє безпеці та ефективності роботи системи.

3. Розглянуто алгоритм роботи з кошиком, який включає додавання, оновлення та видалення товарів. Блок-схема цього алгоритму демонструє, як користувач взаємодіє з кошиком, додаючи товар, змінюючи кількість або видаляючи товар.

Користувач додає товар до кошику. Алгоритм перевіряє, чи є цей товар вже в кошику. Якщо товар присутній, кількість одиниць товару збільшується, а якщо товару немає - він додається як новий елемент. Зміни зберігаються в сесії користувача або в базі даних, якщо кошик прив'язаний до акаунта користувача.

Користувач змінює кількість товару в кошику. Алгоритм перевіряє нову кількість товару та відповідно оновлює загальну суму кошику, забезпечуючи коректне обчислення вартості замовлення.

Користувач видаляє товар з кошику. Алгоритм видаляє товар з кошику, який зберігається або в сесії користувача, або в базі даних, якщо кошик прив'язаний до акаунта.

Цей підхід гарантує, що кошик правильно відображає поточний стан покупок користувача, враховуючи всі зміни, що відбуваються з товарами. Всі операції виконуються з дотриманням принципів ефективності та безпеки, забезпечуючи правильність обчислень і актуальність інформації в кошику.

4. Розглянуто алгоритм оформлення замовлення, який включає процес вибору товарів, введення даних для доставки, підтвердження замовлення та перевірку його стану.

Процес оформлення

1. Користувач вибирає продукти в кошику та переходить до оформлення замовлення.

2. Алгоритм перевіряє, чи користувач авторизований. Якщо користувач не авторизований, він автоматично перенаправляється на сторінку входу або реєстрації для підтвердження своєї особи.

3. Користувач надає необхідні дані для доставки, зокрема адресу, спосіб доставки та спосіб оплати.

4. Після підтвердження замовлення алгоритм створює новий запис про замовлення в базі даних. Це замовлення містить усі деталі покупок, статус замовлення та спосіб оплати.

5. Для подальшого відслідковування замовлення створюється запис, який містить інформацію про поточний стан замовлення (наприклад, "в обробці", "відправлено", "доставлено").

Перевірка стану замовлення

1. Алгоритм регулярно перевіряє стан замовлення, зокрема чи оплачено замовлення і чи доставлено воно.

2. Інформація про стан замовлення доступна користувачу через API, що дозволяє йому стежити за його прогресом і бути в курсі актуального стану.

Цей алгоритм забезпечує ефективне управління процесом оформлення замовлення, гарантує безпеку даних користувача і дозволяє відслідковувати стан замовлення на кожному етапі його обробки.

5. Розглянуто алгоритм обробки адміністративних дій, який включає перегляд і керування замовленнями, а також керування продуктами. Це забезпечує адміністраторам контроль над процесами в системі та дозволяє ефективно управляти контентом.

Перегляд і керування замовленнями

1. Адміністратор може переглядати список усіх замовлень, що містить деталі про кожне замовлення, такі як статус, замовлені продукти, спосіб оплати тощо.

2. Алгоритм дозволяє адміністратору редагувати статуси замовлень (наприклад, змінювати статус на "в обробці", "відправлено" або "доставлено").

3. Після зміни статусу замовлення інформація оновлюється в базі даних, і ці зміни відображаються на клієнтській стороні, що дозволяє користувачу бачити актуальний стан свого замовлення.

Перегляд і керування продуктами

1. Адміністратор може переглядати всі продукти, додавати нові, оновлювати існуючі або видаляти продукти з системи.

2. Алгоритм забезпечує доступ до операцій CRUD (Create, Read, Update, Delete) через API, що дозволяє адміністратору керувати товарами, не вимагаючи прямого доступу до бази даних.

Цей алгоритм дозволяє адміністраторам ефективно управляти замовленнями та продуктами, що забезпечує безперебійну роботу онлайн-магазину та своєчасну обробку запитів користувачів.

6. Розглянуто алгоритм авторизації та автентифікації, який забезпечує безпеку доступу до ресурсів системи шляхом перевірки валідності токена при кожному запиті.

- При кожному запиті до серверу алгоритм перевіряє, чи є токен в заголовку запиту.

- Якщо токен присутній, алгоритм перевіряє його коректність і валідність, перевіряючи його підпис, термін дії та інші критерії.

- Якщо токен валідний, користувач отримує доступ до запитуваних ресурсів, і сервер дозволяє виконати операцію або отримати дані.

- Якщо токен не валідний або його немає, система не надає доступ і перенаправляє користувача на сторінку входу або реєстрації, де він може увійти до свого акаунта або створити новий.

Цей алгоритм забезпечує високий рівень безпеки в системі, гарантуючи, що лише авторизовані користувачі можуть доступати чутливу інформацію або виконувати важливі операції.

7. Розглянуто алгоритм взаємодії з Firebase для завантаження зображень на сервер і збереження даних у базі даних MongoDB.

- Користувач вибирає зображення на фронтенді (наприклад, через форму завантаження файлу).

- Алгоритм передає зображення на сервер Firebase Storage для зберігання.

- Після успішного завантаження зображення на сервер, Firebase генерує унікальний URL для цього зображення.

- Алгоритм отримує цей URL після завершення завантаження.
- Алгоритм зберігає URL зображення разом з іншими даними продукту (назва, опис, ціна тощо) в базі даних MongoDB.
- Збережені дані включають URL, який дає доступ до зображення через Firebase Storage, і відповідні дані про продукт, що дозволяє користувачам бачити зображення разом з іншими характеристиками.

Цей підхід забезпечує ефективне зберігання зображень у хмарному сховищі Firebase та надає легкий доступ до них, зберігаючи всі необхідні дані в базі MongoDB для подальшого використання в системі.

8. Розглянуто алгоритм обробки помилок, який дозволяє ефективно управляти різними типами помилок, що можуть виникати під час виконання операцій у системі.

- Алгоритм перевіряє успішність кожної операції з базою даних (наприклад, при додаванні, оновленні або видаленні записів).
- Якщо виникає помилка під час взаємодії з базою даних (наприклад, через відсутність з'єднання або неправильний запит), система генерує відповідне повідомлення про помилку.
- Повідомлення повинне бути зрозумілим і вказувати на можливу причину помилки (наприклад, "Не вдалося з'єднатися з базою даних").
- Алгоритм обробляє помилки, що можуть виникнути під час завантаження файлів на сервер (наприклад, зображень або документів).
- Якщо завантаження файлу не вдалося (через непідтримуваний формат, перевищення розміру файлу або інші помилки), система повідомляє користувача про помилку.
- Повідомлення про помилку повинно чітко пояснювати, що саме стало причиною збою, та пропонувати рішення (наприклад, "Файл занадто великий. Спробуйте завантажити менший файл").
- Алгоритм перевіряє вхідні дані на відповідність необхідним вимогам (наприклад, перевірка правильності формату email, чи ціна не є від'ємною, чи поле не порожнє).
- Якщо дані не проходять валідацію, система повідомляє користувача про помилку з вказівкою на конкретну проблему (наприклад, "Невірний формат email" або "Ціна не може бути від'ємною").
- Повідомлення повинне бути конструктивним і допомагати користувачу швидко виправити помилку.
- Алгоритм перевіряє, чи є у користувача відповідні права для доступу до ресурсу або виконання операції (наприклад, перевірка токена автентифікації).
- Якщо користувач не авторизований або токен недійсний, система перенаправляє користувача на сторінку входу або реєстрації та надає відповідне повідомлення ("Вам необхідно увійти в систему для доступу до цього ресурсу").

Цей підхід дозволяє ефективно обробляти різноманітні помилки, забезпечуючи коректну реакцію на них, підтримку зручного інтерфейсу для користувачів і безперервну стабільність роботи системи.

Розробка таких алгоритмів потребує правильного планування, розподілу обов'язків між фронтендом і бекендом, а також ефективного використання бази даних і зовнішніх сервісів (Firebase, API). Всі ці етапи забезпечують безперебійну та ефективну роботу сайту з інтерактивними елементами.

#### 4. Створення інтерфейсу

Розробка інтерфейсу користувача є критично важливою складовою при створенні інтернет-магазину, **оскільки саме інтерфейс визначає взаємодію між покупцем та функціональними можливостями** платформи. У системі інтернет-магазину PixelMall інтерфейс був спроектований з урахуванням потреб користувачів, забезпечуючи простоту використання, інтуїтивно зрозумілу навігацію та естетичний дизайн, що сприяє створенню позитивного досвіду покупок. Особлива увага була приділена забезпеченню швидкого доступу до основних функцій, таких як перегляд категорій товарів, додавання товарів до кошику, оформлення замовлення, а також доступу до особистого кабінету користувача. Інтерфейс PixelMall забезпечує легкість у навігації завдяки чітко структурованим розділам, зручному пошуку та фільтрам для швидкого знаходження потрібних товарів. Усі елементи сторінок, включаючи кнопки, меню та форми, оптимізовані для комфортного використання на різних пристроях, що робить покупки в інтернет-магазині зручними на будь-якому екрані. Крім того, дизайн інтерфейсу сприяє естетичному задоволенню користувачів завдяки сучасним шрифтам, кольоровій палітрі та стильним іконкам, що підвищує ефективність взаємодії та заохочує до повторних покупок.

Рис. 4.1. - Інтерфейс інтернет-магазину PixelMall

Одним із важливих **елементів інтерфейсу** інтернет-магазину PixelMall **є сторінка перегляду списку товарів, яка надає адміністратору доступ до основної інформації про асортимент магазину. Ця сторінка містить таблицю, в якій відображаються** такі дані, як назва товару, знижки, ціна. Для зручності роботи з великим обсягом товарів реалізовано групування товарів, що дозволяє організувати їх за категоріями та підкатегоріями, спрощуючи навігацію та пошук необхідних товарів.

Рис. 4.2. - Список товарів

**Іншою важливою частиною інтерфейсу є сторінка** кошику, яка дозволяє адміністраторам переглядати деталі замовлень користувачів. **Ця сторінка містить таблицю, в якій відображаються** інформація про додані товари, їх кількість, ціна та загальна сума кошику. Адміністратори можуть перевіряти, які товари найчастіше додаються до кошику, а також коригувати кількість товарів або видаляти їх, якщо це необхідно. Такий інтерфейс дозволяє ефективно управляти асортиментом та зручніше обробляти замовлення.

Рис. 4.3. - Кошик, який заповнений 2 товарами

**Окремою частиною інтерфейсу є модуль** підсумку замовлення в кошику, який дозволяє користувачам **отримувати детальну інформацію про замовлення перед оформленням. На цій сторінці відображаються кількість** товарів, загальна сума кошику, а також можливість застосування знижок або промокодів. Крім того, користувач може вибрати адресу для доставки товару, що додає зручності при оформленні **замовлення. Всі дані представлені у** зручному форматі, що дозволяє швидко перевірити замовлення перед підтвердженням і допомагає користувачам ефективно планувати свої покупки та доставку.

Рис. 4.4. - Підсумок замовлення

**Реалізація зручного та інтуїтивно зрозумілого інтерфейсу забезпечує легкий доступ до всіх функцій системи. Використання сучасних технологій та принципів адаптивного дизайну** гарантує, що система працює ефективно **як на стаціонарних комп'ютерах, так і на мобільних пристроях**. Це дозволяє користувачам взаємодіяти з системою **у будь-який час і з будь-якого пристрою. Інтерфейс** розроблений так, щоб дії користувачів були логічно послідовними, без зайвих елементів, зосереджуючись на основних функціях. Також інтерфейс забезпечує високу інтерактивність, дозволяючи **виконувати дії без необхідності перезавантаження сторінки. Наприклад, зміна кількості записів на сторінці або перемикання між сторінками** здійснюються через асинхронні запити, що значно прискорює роботу системи. Це особливо важливо при великих обсягах даних, де швидкість і зручність взаємодії є основними критеріями.

#### РОЗДІЛ 4. ОХОРОНА ПРАЦІ В ІТ-ГАЛУЗІ

##### 1. Удосконалення умов праці програмістів

Комп'ютери давно стали невід'ємною частиною повсякденного життя, а правила роботи з ними вже міцно закріпились в ІТ-компаніях за кордоном. Однак в Україні цей процес ще триває, хоча й наближається до свого завершення. Це питання залишається **актуальним для таких міст, як** наприклад **Маріуполь, оскільки в столиці та інших великих обласних центрах воно практично вирішене. На перший погляд, робота з комп'ютером здається безпечною**, проте нехтування певними аспектами може призвести до проблем зі здоров'ям. Професія програміста та інших ІТ-спеціалістів супроводжується великим розумовим навантаженням. Розробники настільки захоплені своєю роботою, що навіть під час перерви продовжують думати про проекти. Вони часто вважають відпочинком паралельну діяльність, як-от читання спеціалізованої літератури, створення вебсайтів або вивчення нових мов програмування. Однак **мозок не здатен безкінечно сприймати лише корисну інформацію, яку** розробники прагнуть отримати для професійного та особистісного розвитку. Мозок людини не є машиною, і він не може постійно обробляти інформацію без втрат у продуктивності [15]. Тому багато ІТ-компаній облаштовують свої офіси зонами відпочинку та лаунджами, щоб забезпечити працівникам психофізіологічне розвантаження. Адже вже давно не дивує робочий стіл з ноутбуком. Задля підвищення продуктивності, **міжнародні компанії перетворюють одноманітні офіси на креативні простори, де нові ідеї** з'являються без значних зусиль. Наприклад, у компанії Inventionland співробітники працюють у казкових умовах, серед яких є **гігантський гоночний трек, пряниковий будиночок і дуже реалістичний піратський корабель, на палубі якого розташовані комп'ютерні столи**, що підтримуються винними бочками. Офіс Google у Цюріху нагадує великий вулик, а офіс шведського провайдеру Bahnhof розташований у бомбосховищі часів холодної війни, схожому на підземне укриття після глобальної катастрофи. Щоб співробітники не прагнули йти додому, роботодавці створюють умови для відпочинку без відходу **від робочого місця, обладнуючи басейни, ігрові кімнати та спортзали** [16]. **Хоча можна цілий день працювати за комп'ютером, а ввечері відчувати втому, схожу на наслідки фізичної праці. Проте це хибне відчуття, і спорт допомагає боротися з ним. Активний відпочинок чудово бадьорить**, покращує циркуляцію крові, надає енергії.

Попри комфортні умови, важливо пам'ятати, що тривала робота за комп'ютером може призвести до фізичної втоми, яка не завжди відчувається одразу. Багато людей після кількох годин за екраном відчувають себе виснаженими, що є результатом не тільки психологічного навантаження, але й малорухомого способу життя. Спорт і активний відпочинок є важливими засобами боротьби з таким станом. Фізичні вправи стимулюють вироблення ендорфінів, покращують циркуляцію крові і знімають напругу в м'язах, що дозволяє працівникам швидше відновлюватися і повернутись до роботи з новими силами. Тому інтеграція фізичної активності в робочий процес, навіть через короткі перерви на розтяжку або прогулянки, допомагає підтримувати баланс між інтелектуальним і фізичним навантаженням, що позитивно впливає на загальний рівень продуктивності.

Багато програмістів інтуїтивно потребують активного відпочинку, **вибираючи спорт як хобі для вільного часу. Однак варто пам'ятати, що умови праці програмістів** можуть бути пов'язані з іншими шкідливими виробничими факторами: шумом, тепловим випромінюванням, причому шкідливими можуть бути не тільки високі, але й низькі температури, іонізуючим та неіонізуючим випромінюванням, як **рентгєнівське, інфрачервоне, електромагнітне ВЧ і СВЧ** випромінювання, статичною електрикою, недостатнім освітленням, а також візуальними факторами, як **яскравість, контрастність, мерехтіння зображення, відблиски тощо**. В таких умовах особливого значення набувають заходи охорони праці, спрямовані на **збереження здоров'я та працездатності працівників**, зокрема **правові, соціально-економічні, організаційно-технічні, санітарно-гігієнічні та** лікувально-профілактичні заходи. Адже науково-технічний прогрес буде ефективним лише в тому випадку, якщо він забезпечить безпеку, комфорт і зручність трудової діяльності людини [17]. Це передбачає не тільки удосконалення технічних засобів і процесів, але й створення таких умов, які дозволяють людям працювати без перевантажень, стресу та шкоди для здоров'я. Успішне впровадження новітніх технологій у робочий процес повинно супроводжуватись розробкою ергономічних стандартів, які допомагають адаптувати робочі місця під індивідуальні потреби працівників. Тільки за умови такого підходу можна забезпечити довгострокову продуктивність праці, зберегти фізичне та психологічне здоров'я працівників, а також забезпечити стабільний розвиток організації і суспільства в цілому.

##### 2. Інформаційна культура в ІТ-управлінні та охороні праці

Для покращення ефективності системи управління охороною праці (СУОП) важливим є розвиток інформаційної культури фахівців ІТ-технологій, оскільки це сприяє удосконаленню інформаційного контексту сучасних підприємств. Це дозволяє створювати точні прогнози щодо умов праці, стану здоров'я та працездатності, травматизму на виробництві та професійних захворювань, а також визначати стратегії розвитку організацій через різні підходи до **охорони праці (інноваційні, маркетингові, інвестиційні, фінансові, технологічні, диверсифікаційні)**. Крім інформаційної культури, **важливо використовувати у рамках СУОП «трикутник» її основних складових: правову (8,1 за 10-бальною шкалою), організаційну (8,0), управлінську (7,5)**.

У сфері охорони праці необхідно реалізовувати основні принципи та **теоретико-методологічні підходи інформаційного менеджменту. За стан СУОП повинні відповідати фахівці служби охорони праці сучасного підприємства. Сучасне суспільство** зазвичай описують як постіндустріальне, постеконімічне та інформаційне, оскільки в ньому відбуваються значні **зміни в розвитку цивілізації. Інформаційне суспільство передбачає** глибокі трансформації, де інформація і знання стають основним професійним **і виробничим потенціалом особистості, соціуму та держави. На етапі постіндустріального розвитку суспільства вирішальним фактором стає інформація. Її домінування започатковано науково-** технічною революцією, яка отримала назву інформаційної, оскільки охоплює всі форми інтелектуальної діяльності, від інформаційних образів штучного інтелекту в новітніх технологіях і економіці до інформатизації суспільства в умовах глобалізації науки та освіти. Інформаційні технології вважаються потужним фактором **економічного зростання України. Для цього необхідні великі стратегічні інвестиції в комп'ютерну та комунікаційну інфраструктуру, програми досліджень** та розробок, а також в освітній сектор [18]. Інформаційну культуру розуміють як сукупність елементів НІТ (новітні інформаційні технології), технологічної, правової, психологічної, соціологічної та ергономічної підсистем, що впливають на соціальні процеси в суспільстві та колективі, виховуючи усвідомлене ставлення до

праці та виконання прав і обов'язків [19]. Поняття інформаційної культури з'явилося через зростаючий інтерес до механізмів інформаційного обміну в умовах збільшення ролі інформації у соціокультурних процесах суспільства, яке розглядається як інформаційне суспільство знань, де в центрі знаходяться інформаційні технології. Робота з інформацією та розвиток інформаційної культури стали одними з ключових складових у процесі управління змінами в компаніях. Існує три основні причини, чому інформаційна культура є важливою сьогодні.

По-перше, вона все більше стає частиною загальної організаційної культури компанії. Багато компаній усвідомлюють необхідність змін, орієнтуючись на задоволення потреб споживачів. Для того щоб впливати на майбутнє, потрібно передбачати, яким воно буде. Для цього необхідно працювати з різноманітною інформацією – діловою, професійною, технологічною, соціальною, ринковою та політичною.

По-друге, інформаційні технології дозволяють створювати в компаніях комп'ютерні мережі для комунікації між менеджерами. Однак важливо розуміти, як ці мережі використовуються для передачі інформації. Створення таких мереж із усіма робочими станціями та мультимедійними можливостями не гарантує ефективного використання інформації.

По-третє, для різних функціональних служб і підрозділів сучасних підприємств у сфері охорони праці існують різні рівні інформаційної культури, що передбачає різні методологічні підходи до збору, організації, обробки, поширення та використання інформації. Це підтверджує важливість корпоративної інформаційної культури для розробки стратегій охорони праці та впровадження заходів щодо її вдосконалення. Для деяких галузей, таких як розробка програмного забезпечення, інформаційна культура є критично важливою умовою для підтримки конкурентоспроможності. Це пояснюється тим, що технологічні зміни у цій сфері відбуваються кожні 6-8 місяців, а інвестиції у підготовку персоналу та освоєння нових технологій можуть сягати величезних сум – від 1,5 до 2 млрд. доларів на рік у великих компаніях [20]. Аналіз ситуації показує, що процеси інформатизації та інтеграції комунікаційного простору в Україні сприяють підвищенню інформаційної та професійної компетентності, активізації бізнес-діяльності, стимулюванню конкуренції, розвитку інноваційних підприємств і організацій, створенню нових робочих місць та зниженню витрат на управлінські функції [21].

Однак, поряд з досягненнями, виникають певні проблеми, пов'язані з використанням інформаційних технологій:

**1. Надмірне інформаційне навантаження**, що полягає в перевищенні кількості корисної інформації, яка поступає в мережу, порівняно з можливостями її сприйняття людиною. **2. Велика кількість інформації, яка сприймається, але на даний момент не є корисною для фахівців.**

**3. Інформаційний голод**, який виникає через надлишок інформації, спричинений перенавантаженням.

**4. «Інформоманія»** як психологічне захворювання, що веде до знеособлення людини, її залежності від перебування в інформаційному просторі і роботи з комп'ютером, при цьому уникання «живого» спілкування з людьми.

**5. Поява «кіберспільнот»**, які, за своїми соціокультурними характеристиками, ближчі до представників інших культур у глобальному інформаційному просторі, ніж до своєї етнічної спільноти або населення, не охопленого Інтернетом. **6. Індивідуалізм і дегуманізація способу життя « мешканців» Інтернету**, - відсутність готовності ділитися своїми знаннями. Результати експертної оцінки щодо впливу різних складових на формування та розвиток інформаційної культури майбутніх фахівців ІТ-індустрії наведено в таблиці 1.2.

Таблиця 2.1 - Вплив складових на рівень інформаційної культури ІТ-фахівців (за 10-бальною шкалою) Складові, що впливають на рівень інформаційної культури їх вплив, бали Ранг Соціальна 8,50 1 Інноваційна 8,09 2 Технічна 7,83 3 Економічна 7,64 4 Організаційна 7,43 5 Педагогічна 7,42 6 Духовна 7,00 7 Психологічна 6,91 8 Політична 6,67 9 Екологічна 5,91 10 Корпоративна 5,58 11 Математична 5,30 12 Правова 5,10 13 Гендерна 5,00 14

Справді, комп'ютерні технології, а особливо їх мережі, мають значний вплив на життєдіяльність людини, сприяючи глобалізації та технократизації суспільства. Цей вплив особливо відчутно проявляється на центральній нервовій системі, оскільки вона звикає працювати в режимі багатозадачності, що потребує швидких інтуїтивних реакцій замість тривалих логічних роздумів. Кількість оброблюваної щодня інформації зростає експоненційно, що може негативно позначитися на розвитку когнітивних здібностей людини [22].

Дослідження продуктивності розумової праці в великих ІТ-компаніях показали, що на її результати значно впливає організаційна (корпоративна) культура. Важливими складовими є тимблдинги, заходи та тренінги для розвитку персоналу. Кожен керівник повинен розуміти, що саме мотивує його співробітників і надавати їм завдання, які відповідають їхнім інтересам і сильним сторонам [23]. На таких тренінгах повинна приділятися увага і розвитку інформаційної культури працівників, що включає освоєння, використання та перетворення інформації з допомогою сучасних технологій. Ці вміння особливо важливі в контексті навчання з охорони праці та у професійній діяльності загалом. Вони можуть бути корисними як доповнення до інструктажів з охорони праці або для моніторингу психологічного стану та взаємовідносин у колективі.

**Інформаційна культура** - це не просто набір навичок роботи з комп'ютером. Вона є частиною особистісного розвитку, що включає мотивацію до застосування нових знань та вмінь. Володіння інформаційною культурою допомагає людині зрозуміти своє місце у виробничому колективі і формує нове покоління фахівців інформаційного суспільства. Важливими аспектами є здатність виділяти релевантну інформацію, оцінювати її та ефективно використовувати в межах систем управління охороною праці (СУОП).

### 3. Ергономічні умови праці в офісах і віддаленій роботі

Ергономіка - це наука, що вивчає взаємодію людини з навколишнім середовищем і технічними засобами. Вона має надзвичайно важливе значення для забезпечення комфортних і безпечних умов праці, особливо в умовах сучасних офісів і віддаленої роботи. Від того, наскільки враховуються ергономічні вимоги, залежить не лише ефективність праці, а й здоров'я працівників, їхня працездатність і загальний стан здоров'я. Забезпечення правильних ергономічних умов є важливим для підвищення продуктивності та збереження довгострокового здоров'я працівників. У традиційних офісах, де працівники проводять значну частину свого часу за комп'ютерами, ергономіка є ключовим фактором для зниження ризиків для здоров'я. Найпоширенішими проблемами, з якими стикаються офісні працівники, є болі в спині, шиї, руках і зап'ястках, а також проблеми із зором, викликані тривалим сидінням і неправильними позами за комп'ютером.

Основні принципи ергономіки для офісів:

1. Правильна організація робочого місця. Столик і крісло повинні бути налаштовані так, щоб співробітник міг зручно сидіти, маючи правильну поставу. Висота стільця повинна дозволяти ногам бути в комфортному положенні, а руки - на рівні з клавіатурою. Екран комп'ютера має бути розташований так, щоб верхній край екрану був на рівні очей, а відстань між очима та екраном становила 40-70 см.

2. Ортопедичні меблі. Використання ергономічних стільців із підтримкою для спини, можливістю регулювання висоти та кута нахилу є важливим для збереження правильного положення тіла. Це допомагає знизити навантаження на хребет, особливо в поперековій зоні. Такі стільці повинні мати підтримку для рук, що дозволяє уникнути перенапруги м'язів при роботі з клавіатурою.

3. Освітлення. Правильне освітлення робочого місця важливе для збереження зору та комфорту. Найкраще використовувати природне

освітлення, але при цьому необхідно мати достатнє штучне освітлення, яке не створює відблисків на екрані. Освітлення має бути достатнім, але не надто яскравим.

4. Регулярні перерви. Працівникам слід робити перерви для розслаблення очей, розтягування м'язів шиї та спини. Рекомендується робити перерву хоча б кожні 30 хвилин. Програми, що нагадують про необхідність перерви, можуть допомогти в цьому.

Віддалена робота, що стала популярною під час пандемії COVID-19, має свої специфічні вимоги до організації робочого місця вдома.

Основні проблеми, з якими стикаються працівники під час віддаленої роботи:

1. Неоптимальні умови робочого місця вдома. Багато працівників працюють за звичайними домашніми меблями, такими як кухонні столи чи стільці без належної підтримки. Це може призвести до хронічних болів у спині, шиї та зап'ястках. Важливо мати зручне робоче місце вдома, з можливістю регулювання висоти столу та стільця.
2. Відсутність належного освітлення та вентиляції. Удома працівники часто стикаються з недостатнім освітленням або поганим природним світлом, що може погіршити зір і спричинити втому. Також важливо забезпечити хорошу вентиляцію та доступ до свіжого повітря для комфортної роботи.
3. Труднощі з концентрацією. Відсутність чіткої межі між робочим і домашнім середовищем може знижувати концентрацію та спричиняти стрес. Щоб зберегти продуктивність, необхідно створити чіткий графік роботи та включати перерви для фізичних вправ.
4. Психологічні аспекти. Відсутність соціальних взаємодій з колегами та відчуття ізоляції можуть призвести до стресу, вигорання та зниження мотивації. Важливо створювати можливості для віртуальних зустрічей і командних тренінгів, а також підтримувати психологічний клімат у колективі.

Ергономічні умови праці мають величезне значення для забезпечення здоров'я та продуктивності працівників. Як в офісах, так і при віддаленій роботі, важливо правильно організувати робочі місця, забезпечити належне освітлення та перерви, а також створити умови для підтримки здорового психологічного клімату. Забезпечення комфортних умов роботи знижує ймовірність виникнення хронічних захворювань, таких як болі в спині, порушення зору чи карпальний тунельний синдром, що можуть виникнути через неправильну позу або перенапругу м'язів. Правильна організація робочого середовища також сприяє підвищенню мотивації працівників, їхньому задоволенню від роботи та загальному емоційному стану. У разі віддаленої роботи, важливо, щоб працівники мали доступ до необхідних технічних засобів і могли комфортно працювати з дому, маючи можливість підтримувати зв'язок з колегами та керівництвом. Водночас, важливим аспектом є забезпечення збалансованого робочого графіка, що дозволяє не тільки досягати високих результатів, але й зберігати здоров'я та емоційне благополуччя. Врахування цих факторів не лише покращує фізичний стан працівників, а й сприяє їхній високій продуктивності, що, в свою чергу, позитивно впливає на ефективність організації в цілому.

#### 4. Організація заходів з безпеки праці для IT-проектів

У сучасному світі IT-проекти займають важливе місце в економіці, і забезпечення безпеки праці в цій сфері є важливим аспектом для ефективної та стабільної роботи команд. Оскільки більшість IT-спеціалістів працюють з комп'ютерними системами, програмним забезпеченням та великими масивами даних, необхідно враховувати не лише фізичну безпеку, але й психологічний комфорт, захист інформації та збереження здоров'я працівників.

Одним із перших кроків у забезпеченні безпеки праці є проведення детальної оцінки потенційних ризиків. Це включає як аналіз технічних аспектів (наприклад, ризик помилок програмного забезпечення, збоїв у роботі серверів), так і організаційних питань (наприклад, несанкціонований доступ до даних або порушення процесів управління проектами). Для кожного виявленого ризику повинні бути визначені методи його зменшення або усунення, а також відповідальні особи.

Безпека праці в IT-сфері неможлива без постійного навчання персоналу. Це стосується як технічних аспектів (правила безпечного користування програмним забезпеченням, дотримання стандартів кодування), так і ергономічних рекомендацій (правильне розташування робочого місця, профілактика перевтоми та стресу). Важливо, щоб усі працівники розуміли ризики, пов'язані з їхньою діяльністю, і знали методи їх уникнення. Проводити тренінги з безпеки, розробляти інструкції для новачків, організувати регулярні перевірки знань працівників.

IT-спеціалісти проводять більшу частину робочого часу за комп'ютерами, тому особлива увага повинна приділятися організації робочих місць. Вибір правильного ергономічного обладнання (крісло, стіл, монітор), правильна організація робочого простору (освітлення, розташування пристроїв) можуть значно зменшити ризик виникнення захворювань, таких як болі в спині або перенапруження очей. Важливо також надавати працівникам час для відпочинку, що є складовою частиною профілактики професійних захворювань.

Одним із ключових аспектів безпеки в IT-проектах є захист інформації. Враховуючи значну кількість чутливої інформації, що обробляється в проектах, необхідно впроваджувати сучасні системи захисту даних. Це включає:

- Використання сучасних методів криптографії для захисту даних при передачі та зберіганні.
- Впровадження антивірусного програмного забезпечення та фаєрволів для запобігання кібератакам.
- Регулярне оновлення ПЗ та забезпечення безпеки серверних систем.

Для забезпечення безпеки важливо впроваджувати чітко визначену систему доступу до інформації. Використання багатофакторної аутентифікації дозволяє знизити ризики несанкціонованого доступу до критичних даних. Доступ до чутливої інформації має бути обмежений лише тим співробітникам, які мають відповідні права. Розробка та впровадження політик доступу, а також ведення журналів доступу є важливою частиною захисту інформаційних систем.

IT-проекти часто супроводжуються високими навантаженнями та стресовими ситуаціями, що може негативно впливати на продуктивність і здоров'я працівників. Створення підтримувального робочого середовища, організація консультацій з психологами, розвиток програм стрес-менеджменту та управління емоційним станом є необхідними для збереження психологічного здоров'я команди. Важливо також стимулювати здорові комунікації в колективі та забезпечувати механізми для розв'язання конфліктів.

Наявність належного моніторингу та регулярний аудит безпеки є необхідними для підтримки високого рівня захисту на всіх етапах реалізації IT-проекту. Це включає в себе контроль за роботою інформаційних систем, перевірки наявності уразливостей, аудит виконаних завдань і перевірку відповідності вимогам безпеки. Результати аудиту повинні бути використані для коригування політик безпеки та покращення заходів із захисту працівників.

Забезпечення безпеки праці в IT-проектах є багатограним процесом, що вимагає комплексного підходу. Важливими аспектами є як фізична безпека працівників, так і захист їхніх даних, психологічний комфорт та належні умови для продуктивної роботи. Впровадження таких заходів дозволяє не лише мінімізувати ризики, але й створити здорову робочу атмосферу, що є важливим для успіху IT-проекту.

## ВИСНОВКИ

**У результаті виконаної роботи** був розроблений повнофункціональний інтернет-магазин, який забезпечує **автоматизацію ключових бізнес-процесів, включаючи управління каталогом товарів, обробку замовлень, управління користувачами**, інтеграцію з платіжними системами та аналіз фінансових показників. **Досягнуто мети дослідження шляхом створення системи, що відповідає сучасним вимогам надійності, безпеки, продуктивності та зручності користування. Використання сучасних інструментів і технологій, таких як Next.js, MongoDB, Firebase, Strapi та Tailwind CSS, дозволило реалізувати ефективне, масштабоване та інтуїтивно зрозуміле рішення. Розроблена система успішно пройшла тестування, що підтвердило її функціональну відповідність технічним вимогам. Усі критичні сценарії роботи були перевірені, і система продемонструвала високу стабільність, продуктивність та коректність виконання основних завдань, таких як обробка замовлень, аутентифікація користувачів та забезпечення безпеки даних. Це забезпечує надійну основу для впровадження інтернет-магазину в реальну комерційну діяльність.**

**Практичне значення розробки полягає у зменшенні витрат часу на управлінські завдання, підвищенні точності обробки замовлень та покращенні взаємодії з клієнтами. Система надає адміністраторам зручний інструмент для управління асортиментом товарів, контролю замовлень і аналізу фінансових показників, що сприяє оптимізації операційних процесів та підвищенню конкурентоспроможності бізнесу.**

**Таким чином, результати роботи підтверджують доцільність автоматизації бізнес-процесів у сфері електронної комерції та демонструють можливості ефективного використання сучасних веб-технологій для досягнення стратегічних цілей. Розроблена система є готовим до впровадження продуктом, який може бути використаний для створення високопродуктивних і масштабованих інтернет-магазинів, що забезпечать успішну діяльність в умовах сучасного цифрового бізнес-середовища.**

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Астістова Т. І. **SEO - оптимізація в системі моніторингу веб-ресурсів** // Т.І. Астістова // **Технології та інжиніринг «Вісник КНУТД. Серія Технічні науки», Київський національний університет технологій та дизайну, Україна, No1(12), 2023 р., С. 9-17. Scopus Author ID: 6506601603. Режим доступу: <https://vistnik.knutd.edu.ua/wp-content/uploads/sites/2/2023/05/1-1-2023.pdf> (дата звернення: 18.12.2024).**
2. Астістова Т. І. **Розробка системи електронного поселення** / Т.І. Астістова, Д.Д. Ляховська // **Тези V Міжнародної науково-практичної конференції «Мехатронні системи: інновації та інжиніринг - MSIE-2021», Київський національний університет технологій та дизайну, 4 листопада 2021 р., С. 152 (дата звернення: 18.12.2024).**
3. Astistova Tetyana. **Development of software for accounting for the presence of student attendance** / Т.І. Астістова, Д.Д. Ляховська, М.В. Нікітюк, Н.В. Сморжевський // **Збірник наукових праць VI Міжнародної науково-практичної конференції «Science, innovations and education: problems and prospects», 13-15 січня 2022 р., Токіо, Японія, С. 58-67, сертифікат (дата звернення: 18.12.2024).**
4. Астістова Т. І., Єгоров Д.С. **Розробка QR- коду для веб-додатку «Система ідентифікації студентів»** / Т.І. Астістова, Д.С. Єгоров // **Інформаційні технології в науці, виробництві та підприємстві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій, Київ: Освіта України, 2021, С. 214-217. Режим доступу: <https://er.knutd.edu.ua/handle/123456789/23652> (дата звернення: 18.12.2024).**
5. Астістова Т. І., Глембоцький В.С. **Програмне забезпечення для системи ідентифікації студентів** / Т.І. Астістова, В.С. Глембоцький // **Інформаційні технології в науці, виробництві та підприємстві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій, Київ: Освіта України, 2021, С. 208-211. Режим доступу: <https://er.knutd.edu.ua/handle/123456789/236881> (дата звернення: 18.12.2024).**
6. Що таке Next.js та для чого він потрібен // PX Studio. Режим доступу: <https://pxstudio.pw/blog/chto-takoe-next-js-i-dlya-chego-on-nuzhen> (дата звернення: 19.12.2024).
7. Brander. **Технології: Next.js [Електронний ресурс]. - Режим доступу: <https://brander.ua/technologies/nextjs> (дата звернення: 19.12.2024).**
8. 8. The World Bank. **World Development Report 2016: Digital Dividends. 2016, р. 330 (дата звернення: 19.12.2024).**
9. **Capturing functional requirements with use cases and user stories, 2022.** Режим доступу: <https://www.cybermedian.com/ru/capturing-functional-requirements-withuse-cases-and-user-stories> (дата звернення: 20.12.2024).
10. **Reshkova A. The difference and relationship between Use case and User story, 2018.** Режим доступу: <https://medium.com/@a.reskova/the-difference-and-relationship-between-use-case-and-user-story-25e24df777a3> (дата звернення: 20.12.2024).
11. **What is a Use Case? Definition & Examples [Електронний ресурс]. - Режим доступу: <https://study.com/academy/lesson/what-is-a-use-case-definition-examples.html> (дата звернення: 20.12.2024).**
12. **Козак О.Л. Опорний конспект лекцій з курсу «Аналіз вимог до програмного забезпечення» для студентів напряму підготовки «Програмна інженерія». - Тернопіль, 2011. - 56 с. - Режим доступу: [http://dspace.wunu.edu.ua/bitstream/316497/7281/1/FCIT\\_kKN\\_sPZS\\_dAVPZ\\_%20LEC.pdf](http://dspace.wunu.edu.ua/bitstream/316497/7281/1/FCIT_kKN_sPZS_dAVPZ_%20LEC.pdf) (дата звернення: 20.12.2024).**
13. **Рекомендації з управління нефункціональними вимогами [Електронний ресурс]. - Режим доступу: <https://e5.ua/en/blogpost-2-en/rekomendatsiyi-z-upravlinnya-nefunktsionalnimi-vimogami/> (дата звернення: 20.12.2024).**
14. **Rumarenko B. Underestimated Power Of Non-Functional Requirements [Електронний ресурс]. - Режим доступу: <https://www.uptech.team/blog/non-functional-requirements> (дата звернення: 20.12.2024).**
15. **Психологія безпеки праці / Укладач Кальянов А.В. - Донецька обласна рада профспілки, 2008. - 32 с.**
16. **Сьогодні UA [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://www.segodnya.ua/lifestyle/fun/pochti-kak-u-google-chem-udivlyayut-ofisy-ukrainskih-it-kompaniy--764025.html> (дата звернення: 21.12.2024).**
17. **Конспект лекцій з курсу «Охорона праці в галузі» / Викладачі: Яскілка В.Я., Олійник М.З. - Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. - 56 с.**
18. **Шконда В.В., Кальянов А.В. Культурологічні засади професійно-особистісного становлення майбутніх фахівців: Монографія. - Донецьк, 2012. - 262 с.**
19. **Шконда В.В., Кальянов А.В., Давидов П.Г. Феномен синергетики: наука - суспільство - освіта: Монографія / Ред. Шконда В.В. - Донецьк: Норд-Прес, 2009. - 156 с.**
20. **Інформаційна культура підприємств, види інформаційної культури, інформаційна поведінка [ Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://lektsii.com/1-78900.html> (дата звернення: 21.12.2024).**
21. **Пивоваров М.Г., Медко Д.А. Перспективи створення та розвитку інформаційно-комунікаційної системи України // **Економіка: проблеми теорії та практики: зб. наук. праць. - Вип. 49. - Дніпропетровськ: Дніпропетр. Нац. Ун-т, 2000. - С. 56-61.****
22. **Землянова Л.М. Мережеве суспільство, інформаціоналізм і віртуальна культура / Л.М. Землянова // Вісн. Моск. ун-ту. - 1999. - Сер. 10, No 2. - С. 58-56.**

23. **Левон М.С. Система** факторів, які впливають на продуктивність праці в ІТ-компаніях // Економіка, Статистика та Інформатика. - 2015. - No 3. - С. 127-128.

24. **Глобальна електронна комерція: ключові цифри та тренди e-commerce 2024** [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://rau.ua/novyni/trendi-e-com-2024/> (дата звернення: 22.12.2024).

ДОДАТКИ

```
"use client";
```

```
import ComponentLevelLoader from "@components/Loader/componentlevel";
import { GlobalContext } from "@context";
import { getAllOrdersForAllUsers, updateStatusOfOrder } from "@services/order";
import { useContext, useEffect } from "react";
import { PulseLoader } from "react-spinners";
```

```
export default function AdminView() const allOrdersForAllUsers setAllOrdersForAllUsers user pageLevelLoader setPageLevelLoader
componentLevelLoader setComponentLevelLoader = useContext(GlobalContext) async function extractAllOrdersForAllUsers
setPageLevelLoader(true) const res = await getAllOrdersForAllUsers console.log(res) if (res.success) setPageLevelLoader(false)
setAllOrdersForAllUsers res.data && res.data.length ? res.data.filter((item) => item.user._id !== user._id) : else
setPageLevelLoader(false) useEffect if (user !== null) extractAllOrdersForAllUsers, [user] console.log(allOrdersForAllUsers) async function
handleUpdateOrderStatus(getItem) setComponentLevelLoader({ loading: true, id: getItem._id) const res = await updateStatusOfOrder ...getItem
isProcessing: false) if (res.success) setComponentLevelLoader({ loading: false, id: getItem._id) extractAllOrdersForAllUsers) else setComponentLevelLoader({
loading: true, id: if (pageLevelLoader) return <div className="w-full min-h-screen flex justify-center items-center"><PulseLoader color="#000000"
loading={pageLevelLoader} size={30} data-testid="loader" </div> return <section <div className="mx-auto px-4 sm:px-6 lg:px-8"><div <div
className="px-4 py-6 sm:px-8 sm:py-10"><div className="flow-root"> {allOrdersForAllUsers && allOrdersForAllUsers.length} <ul
className="flex flex-col gap-4"> {allOrdersForAllUsers.map((item) && li) key={item._id} className="bg-gray-200 shadow p-5 flex flex-col space-y-3 py-6
text-left" <div className="flex" <h1 className="font-bold text-lg mb-3 flex-1" #order: {item._id} </h1> <div className="flex flex-col gap-2"
<div className="flex items-center" <p className="mr-3 text-sm font-medium text-gray-900" Username} </p> <p className="text-sm;font-
semibold text-gray-900" {item?.user?.name} </p> <div className="flex items-center" <p className="mr-3 text-sm font-medium text-gray-
900" User email} </p> <p className="text-sm;font-semibold text-gray-900" {item?.user?.email} </p> </div> <div className="flex items-center"
<p className="mr-3 text-sm font-medium text-gray-900" Total Paid Amount} </p> <p className="text-sm;font-semibold text-gray-900"
${item?.totalPrice} </p> </div> </div> </div> <div className="flex gap-2"> {item.orderItems.map((orderItem, index) <div key={index}
className="shrink-0" <img alt="Order Item" className="h-24 w-24 max-w-full rounded-lg object-cover" src={orderItem.orderItem.product
orderItem.product.imageUrl} </div> </div> <div className="flex gap-5"> <button className="disabled:opacity-50 mt-5 mr-5" inline-block bg-black
text-white px-5 py-3 text-xs font-medium uppercase tracking-wide" {item.isProcessing} ? "Order is processing" : "Order is delivered" </button>
<button onClick={() => handleUpdateOrderStatus(item) disabled={!item.isProcessing} className="disabled:opacity-50 mt-5 mr-5" inline-block bg-
black text-white px-5 py-3 text-xs font-medium uppercase tracking-wide" {componentLevelLoader} componentLevelLoader.loading}
componentLevelLoader.id === item._id <ComponentLevelLoader text="Updating Order Status" color="#ffffff" loading} componentLevelLoader
componentLevelLoader.loading} "Update Order Status" </button> </div> </li> </ul> ) : null </div> </div> </div> </div> </section>
);
}
```