

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ**  
**ТЕХНОЛОГІЙ**

# **КВАЛІФІКАЦІЙНА РОБОТА**

першого (бакалаврського) рівня вищої освіти

на тему: «Розробка інтернет-магазину для продажу комп'ютерної  
техніки»

Виконав: студент 4 курсу групи Іт-41сп

Спеціальності 126 «Інформаційні системи  
та технології»

(шифр і назва)

Павловський Антон Олександрович

(Прізвище і ініціали)

Керівник: к. е. н., доц. Шувар Б. І.

(Прізвище і ініціали)

Рецензент: к. т. н., доц. Шолудько Я. В.

(Прізвище і ініціали)

**ДУБЛЯНИ-2024**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ПРИРОДОКОРИСТУВАННЯ

ФАКУЛЬТЕТ МЕХАТРОНИКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти  
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 202\_ р.

**ЗАВДАННЯ**

на кваліфікаційну роботу студенту

Павловського Антона Олександровича

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка інтернет-магазину для продажу комп'ютерної техніки»
2. Керівник роботи к.е.н., доцент, Шувар Богдан Іванович  
(наук.ступінь, вч. звання, прізвище, ініціали)  
затверджені наказом Львівського НУП № 641 /к-с з 27.11.23р.  
Строк подання студентом роботи 17.06.2024р.
3. Вихідні дані до роботи: CMS Payload, SCSS, HTML, TypeScript, NextJS, MongoDB
4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)  
Вступ
  1. Постановка завдання
  2. Проектування інтернет-магазину
  3. Програмна реалізація проекту
  4. Охорона праціВисновки  
Бібліографічний список
5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових схем та моделей): огляд аналогів інтернет-магазину; способи технології реалізації веб-сайту; оцінка основних характеристик вибору технологій розробки веб-сайту; проектування моделі бази даних; структурна схема проєктованої інформаційної системи; етапи розробки інтернет-магазину за вибраним технологією; результати ручного тестування розробленого веб-сайту; опитування користувачів пройденого ручного тестування; загальний вигляд розробленої сторінки та її функціоналу.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	<i>Шувар Б. І., к.е.н., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І. М., доцент кафедри управління проектами та безпеки виробництва</i>		

7. Дата видачі завдання \_\_\_\_\_ .2024 р.

Календарний план

№ з/п	Назва етапів дипломного проекту	Терміни виконання етапів роботи	Примітка
1.	<i>Написання першого розділу</i>	01.02.2024 – 01.03.2024	
2.	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	02.03.2024 – 02.04.2024	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	03.04.2024 – 25.04.2024	
4.	<i>Написання розділу «Охорона праці»</i>	26.04.2024 – 28.04.2024	
5.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	29.04.2024 – 29.05.2024	
6.	<i>Завершення роботи цілому</i>	30.05.2024 – 07.06.2024	

Студент \_\_\_\_\_, Павловський А.О.  
(підпис) (Прізвище, ініціали)

Керівник роботи \_\_\_\_\_, Шувар Б. І.  
(підпис) (Прізвище, ініціали)

УДК: 004.738.5:339]:004.38

Розробка інтернет-магазину для продажу комп'ютерної техніки.

Павловський А.О. Кафедра ІТ. - Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 72 ст. текст. част., 25 рис., 5 табл., 20 літ. джерел.

Зроблено аналіз предметної області. Наведено фактори українського ринку в інтернет-магазині. Проаналізовано основні аналоги, переваги та недоліки інтернет-магазинів. Розглянуто етапи створення інтернет-магазину.

Проаналізовано різні методи розробки веб-сайту інтернет-магазину Computer Store. Обрано один із наведених методів розробки веб-сайту. Досліджено концепції корисного навантаження – Payload CMS та способи оплати платежів інтернет-магазину.

Проектовано функціональне моделювання інтернет-магазину, модель нереляційної бази даних MongoDB. Побудовано на основі UML-діаграми – діаграма послідовності. Розроблено та описано коди сторінки над створення авторизації ОЗ, персонального користувача, авторизації ОЗ, оформлення замовлення товару. Також додатково, розроблено сторінку каталогу товарів, головна сторінка, сторінка опису кожних товарів з вартістю та можливістю додаванням в кошик та сторінка кошику.

Запропоновано рекомендації з охорони праці під час роботи над технікою.

Ключові слова: інтернет-магазин, веб-сайт, комп'ютерні техніки, Payload CMS, MongoDB, платіжна система Stripe.

Key word: online store, website, computer technics, Payload CMS, MongoDB, payment system Stripe.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНИХ ТЕРМІНІВ .....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1. Опис предметної області .....	9
1.2. Огляд аналогів інтернет-магазинів .....	11
1.3. Функціональність. Переваги та недоліки інтернет-магазину .....	13
1.4. Етапи створення інтернет-магазину.....	15
РОЗДІЛ 2. ПОСТАНОВКА ЗАВДАННЯ ТА ВІДБІР АРХІТЕКТУРНИХ ІНСТРУМЕНТІВ.....	17
2.1. Мета та постановка завдання проєкту .....	17
2.2. Концепції Корисного навантаження (Payload CMS).....	22
2.3. Аналіз корисного навантаження CMS для створення веб-сайту.....	19
2.4. Способи оплати інтернет-магазину.....	23
РОЗДІЛ 3. ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ.....	27
3.1. Функціональне моделювання діяльності інтернет-магазину для продажу комп'ютерної техніки.....	27
3.1.1. Проєктування моделі бази даних із MongoDB .....	30
3.2. Побудова UML-діаграми інтернет-магазину.....	33
3.3. Розробка інтернет-магазину для продажу комп'ютерної техніки .....	35
3.3.1. Розробка та опис коду над створенням персональної інформації користувача .....	39
3.3.2. Розробка та опис коду створення авторизації облікованого запису .....	42
3.4. Результат ручного тестування розробленого інтернет-магазину .....	48
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	52

	6
4.1. Поняття й завдання техніки безпеки .....	52
4.2. Комп'ютерна техніка та її вплив на організм людини.....	54
4.3. Шляхи оптимізації технічних, середовищних та ергономічних факторів ....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	61
ДОДАТКИ .....	63

## **ПЕРЕЛІК СКОРОЧЕНИХ ТЕРМІНІВ**

TS – TypeScript

TSX – TypeScript XML

SCSS – Sassy Cascading Style Sheets

HTML – HyperText Markup Language

Mongo DB – Mongo Data Base, Inc

CMS – Content Management System

IT – інформаційні технології

ОЗ – Обліковий запис

## ВСТУП

У сучасну цифрову епоху технологічний розвиток дозволив нам отримати доступ до широкого спектру товарів і послуг безпосередньо з наших зручних пристроїв. Інтернет-магазини є важливим і незамінним аспектом цього процесу, пропонуючи на широкий асортимент товарів від електроніки до одягу та зручні умови оплати і доставки. У цьому контексті створення інтернет-магазину з продажу комп'ютерної техніки видається зручною та перспективною ідеєю.

Метою кваліфікаційної роботи є дослідження інформаційних технологій у процес створення та ефективного функціонування інтернет-магазину з продажу комп'ютерної техніки. Основними напрямками дослідження є розробка веб-сайту, його оптимізація для мобільних пристроїв, використання баз даних для зберігання товарів, користувачів та замовлень.

Процес дослідження включав аналіз інформаційних потреб користувачів та вивчення практик у сфері дизайну інтерфейсу та користувацького досвіду.

Особливий інтерес викликає використання інформаційних технологій для розробки інноваційних стратегій маркетингу та просування в цифровому середовищі. Це аналіз даних клієнтів та історії покупок для персоналізованих пропозицій.

Також будуть розглянуті питання щодо кібербезпеки та захисту даних, які стають все більш важливими в міру зростання обсягів і значення онлайн-комерції.



## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Опис предметної області

Електронна комерція є найважливішим складовим елементом електронного бізнесу. Найвідомішим прикладом здійснення електронної комерції є інтернет-магазин, який являє собою веб-ресурс з каталогом продукції та можливістю замовлення і оплати товарів, які сподобались покупцю. Інтернет-магазин є однією найпоширеніших комерційних моделей електронної торгівлі у сфері B2C [11].

Інтернет-магазин (e-shop) являє собою спеціалізований веб-сайт, який належить фірмі-товаровиробнику, торговій фірмі тощо та призначений для просування споживчих товарів на ринку, збільшення обсягів продажу, залучення покупців.

Українська електронна комерція поки що тільки на початку шляху. Однак останні два кризових роки підкреслити значення інтернет-магазинів як ефективного каналу просування товарів і послуг – онлайн-продажі впали менше, ніж традиційні роздрібні. За рахунок цього онлайн-майданчиком навіть вдалося трохи збільшити свою частку в загальній структурі роздрібних продажів [11].

Основними факторами, що гальмують розвиток українського ринку електронної комерції, є [11]:

- ❖ низький рівень проникнення Інтернету (особливо в регіонах);
- ❖ відсутність законодавчої бази, що регулює процес купівлі/продажу онлайн, здійснення електронних платежів за оплачені товари / послуги та встановлює прозорі гри на ринку;
- ❖ слаборозвинена національна система електронних платежів;
- ❖ низька ефективність (з погляду як ведення бізнесу, так і цінності, що надається кінцевим споживачам) більшості українських торговельних майданчиків.

Електронна комерція бізнес-до-користувача (B2C), або торгівля між компаніями й споживачами, передбачає збирання інформації клієнтами; купівлю фізичних речей чи інформаційних/електронних товарів; і, для інформаційних товарів, одержання товару на електронній мережі. Це друга за величиною й найбільш рання форма електронної комерції. Один з найвідоміших прикладів B2C – це [www.amazon.com](http://www.amazon.com), американський книготорговий сайт (на рисунок 1.1), послугами якого користуються понад 50 млн. клієнтів у всьому світі. Саме він зумів підірвати звичайний букіністичний ринок США протягом декількох років.

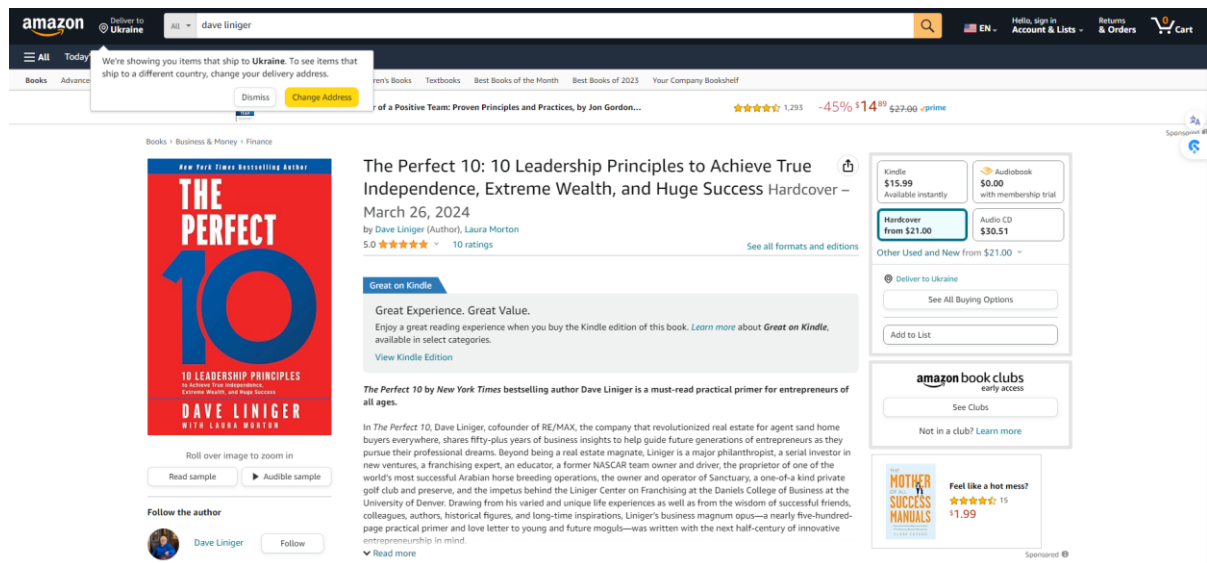


Рисунок 1.1 – Американський веб-сайт Amazon

B2C електронна комерція зменшує ціну угод, збільшуючи доступ споживачів до інформації й допомагаючи споживачам знайти найконкурентноспроможнішу ціну за товар або послугу. B2C електронна комерція також зменшує ринкові бар'єри входу, тому що вартість створення й розкручування сайту набагато менша, ніж побудова структури фірми [11].

Один з найпопулярніших B2C інструментів у електронній комерції – інтернет-магазин.

## 1.2. Огляд аналогів інтернет-магазинів

Розглянемо перший аналоговий інтернет-магазин від бренду ComrX «ComrX. Інтернет-магазин комп'ютерної техніки» (на рис. 1.2):

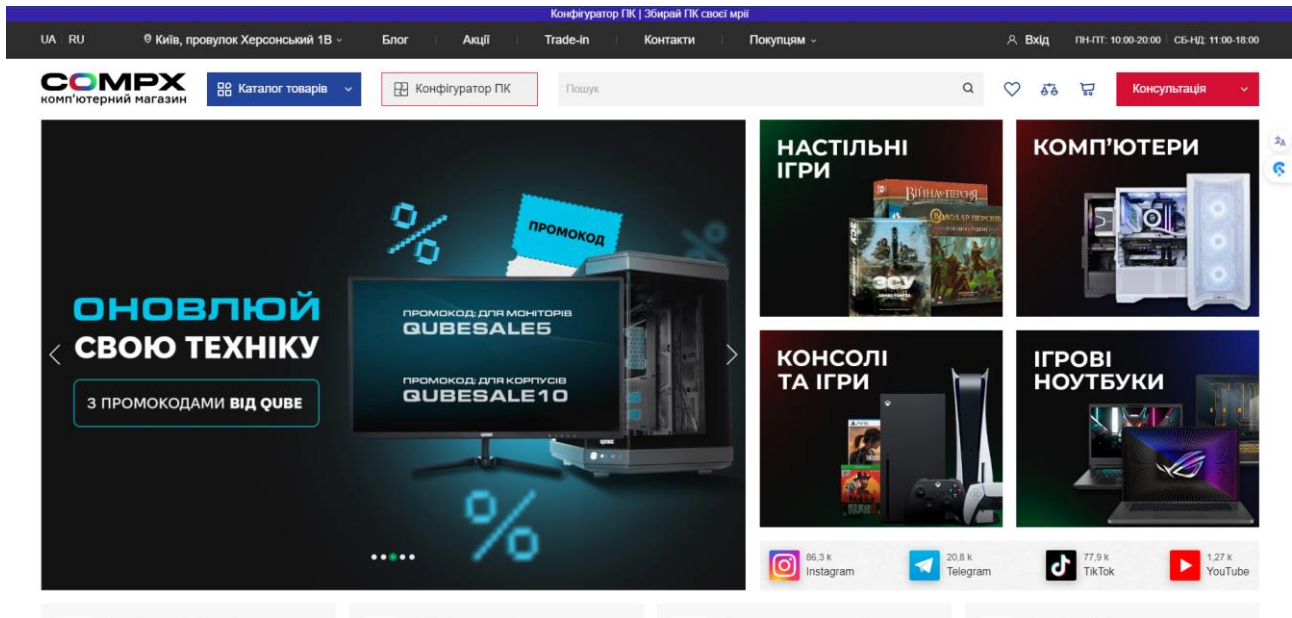


Рисунок 1.2 – Перший аналоговий сайт COMRX

В даному сайті, можна побачити великий середній та великий вибір комп'ютерної техніки, стаціонарні ПК, монітори, периферійні пристрої (миші, клавіатури, принтери тощо), а також комплектуючі (процесори, відеокарти, оперативна пам'ять) та інші товари пов'язані з ІТ. По інтерфейсу даного сайту дизайну, стилізований кнопковими блоками, на заголовку зазначено: значками, такі як: «вподобання», «порівнятор» та «кошик»; логотипами, пошук, каталог товарів.

На головному сайті, розділяється на декілька групи блоків:

- ❖ Блок 1. Акційні оголошення, чотири категорії у вигляді кнопкових блоках та посилання соцмережах;
- ❖ Блок 2. Хіт продажів – нові популярні комп'ютерні товари (відеокарта, материнська плата і так ін.);
- ❖ Блок 3. Підкатегорії комплектуючих для комп'ютера - блокові кнопки (процесори, материнська плата, відеокарти, оперативна пам'ять, SSD накопичувачі, HDD жорсткі диски, підібрати комплектуючі);
- ❖ Блок 4. Замовлення з адресами спеціалізованих магазинів.

В другому аналогічному сайті – «Інтернет-магазин електроніки та комп'ютерної техніки» від бренду ITbox, можна сказати, що по інтерфейсу мінімізований, але наповнений з товарами й категоріями, як аналог сайту Rozetka (на рис. 1.3):

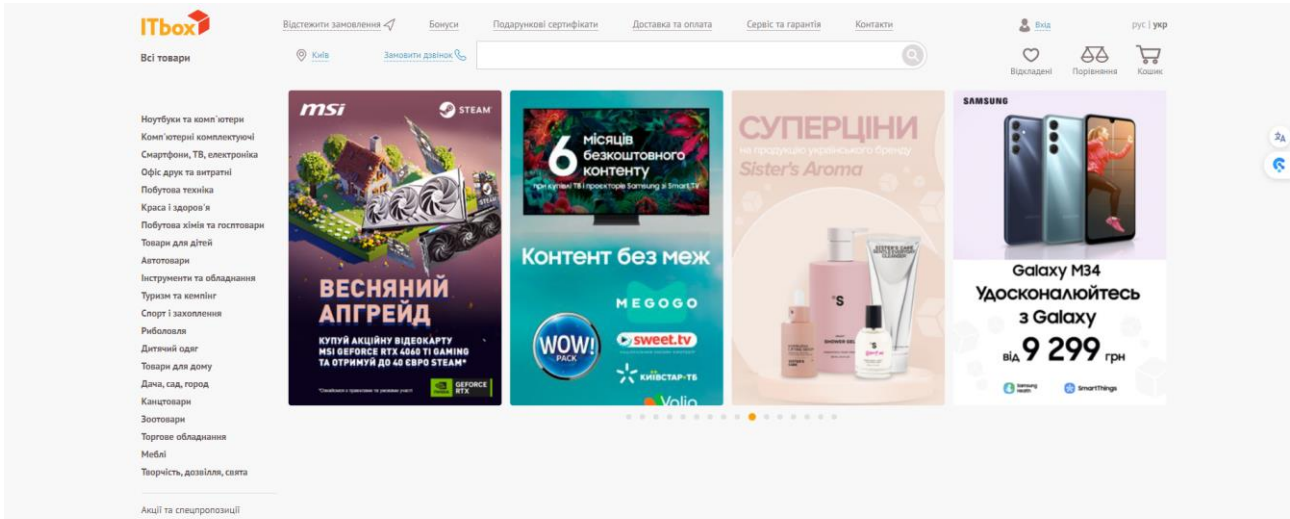


Рисунок 1.3 – Другий аналоговий сайт ITbox

В даному сайті, розташовано блоки акційних оголошень, різні категорії, такі як:

- ❖ Популярні товари – це товари, які більш привабливими клієнтами;
- ❖ Ноутбуки, Планшети та ПК – товари, які знаходяться в даному розділі категорії;
- ❖ ТОП комплектуючих для ПК – товари, які необхідні для внутрішнього системного блоку (материнська плата, накопичувач SSD, блок живлення, процесор, оперативна пам'ять та ін.);
- ❖ Аксесуари до комп'ютера – товари, які необхідні для зовнішнього системного блоку та користування користувачам (мишка, маршрутизатор, навушники, комутатор, мережева плата, принтер та ін.);
- ❖ Кращі товари для дому – пилосос, електрочайник, мікрохвильова піч, ваги підлогові, бойлер, холодильник та ін.;
- ❖ Інструменти та автотовари – канистра автомобільна, моторна олива, антифриз та ін.;

- ❖ Ноутбуки та комп'ютери – це категорії ноутбуки, комп'ютери, монітори, відеокарта блок живлення і та ін.;
- ❖ Популярні бренди – це товари із бренду Asus, Samsung, Lenovo, Vinga, HP, DELL, Xiaomi, Canon, Intel, Kingston, Acer, AMD, Philips, Apple.

Також й нові/популярні блоги та новинок.

І на останок аналоговий сайт від бренду Brain – «Роздрібний інтернет-магазин комп'ютерної техніки та електроніки» (на рис. 1.4):



Рисунок 1.4 – Третій аналоговий сайт Brain

В головному сайті, у бічній зоні розташовано «Каталог товарів», з декількома категоріями. На основному зоні, розташовані: акційні банери, знакові визначені категорії, топ товари, комп'ютерні комплектуючі, аксесуари до комп'ютера, побутові техніки, авто та інструменти, дрони та комплектуючі, та Brain-блог.

### 1.3. Функціональність. Переваги та недоліки інтернет-магазину

Інтернет магазин – це по суті, цифрові платформи, де продавці та покупці можуть взаємодіяти. Основними характеристиками таких магазинів є:

- ❖ Каталог товарів: містить детальні описи товарів, фотографії та ціни;
- ❖ Кошик: дозволяє покупцям зберігати вподобані товари перед оформленням замовлення;

- ❖ Платіжна система: інтегрований платіжний шлюз полегшує обробку платежів;
- ❖ Пошукові системи та фільтри: дозволяють покупцям швидко знаходити відповідні товари;
- ❖ Реєстрація користувача/кабінет покупця: дозволяє користувачам керувати історією покупок та особистими даними.

Функціональність інтернет-магазину визначається його здатністю ефективно виконувати різні задачі, які включають [18]:

- ❖ Легкість Навігації: інтуїтивно зрозумілий інтерфейс, що сприяє поліпшенню користувацького досвіду;
- ❖ Мобільна Адаптація: оптимізація для різних пристроїв, особливо для смартфонів;
- ❖ Безпека: захист даних користувачів та транзакцій є критично важливим;
- ❖ Швидкість Завантаження: швидке завантаження сторінок для підвищення задоволеності користувачів;
- ❖ SEO Оптимізація: для залучення трафіку через пошукові системи;
- ❖ Аналітика та Звітність: інструменти для моніторингу продажів, поведінки користувачів, та інші ключові метрики [18].

Розробка інтернет-магазину – це складний процес, який вимагає глибокого розуміння цих характеристик і функціональності. Важливо не лише створити привабливий веб-сайт, а й забезпечити його ефективність та зручність для кінцевих користувачів. Таким чином, інтернет магазин стає не просто цифровою вітриною, а потужним інструментом для розвитку бізнесу [18].

Основні переваги [18]:

- ❖ Відсутність Географічних Обмежень: можливість продавати товари та послуги клієнтам у будь-якій точці світу;
- ❖ Зниження Оперативних Витрат: онлайн магазини зазвичай вимагають менших капіталовкладень та операційних витрат у порівнянні з фізичними магазинами;

- ❖ Індивідуальній Підхід до Клієнта: можливість персоналізації пропозицій та маркетингу згідно з перевагами та поведінкою користувачів;
- ❖ Підтримка 24/7: можливість обслуговувати клієнтів цілодобово без додаткових витрат.

Основні недоліки [13]:

- ❖ Віртуальність: клієнт інтернет-магазину не може доторкнутися до товару, приміряти його, оцінити реальний зовнішній вигляд. Фотографії продукції можуть відрізнятися від оригіналу навмисно (клієнт викладає зображення кращих екземплярів, ніж відправляє покупцеві) або мимоволі (спотворення розміру або кольору);
- ❖ Очікування: будь-який товар, замовлений в інтернет-магазині, доведеться чекати. Причому тут є багато чинників, через які час очікування може істотно варіюватися. Термін доставки залежить від відстані пересилання, роботи кур'єрської служби і порядності продавця;
- ❖ Шахрайство: онлайн-шахраї можуть обдурити клієнтів різними способами, наприклад за допомогою підроблених веб-сайтів, контрафактної продукції та шахрайських платіжних схем.

#### 1.4. Етапи створення інтернет-магазину

Створення власного інтернет-магазину – це крок до зростання компанії та продажу продукції не лише локально у місті чи регіоні, а й всьому світі. За даними Forbes, у 2023 року п'ята частина всіх роздрібних покупок буде здійснюватися протягом наступних чотирьох років із середньорічним темпом зростання понад 1%. Тому при правильній стратегії розвитку і просування інтернет-магазинів цей сектор бізнесу, безсумнівно, повинен стати прибутковим [12].

Над створення власного проєкту інтернет-магазину буде включати такі етапи:

- ❖ **Планування** – визначення обсягу проєкту, складання графіку робіт, визначення термінів та ресурсів;

- ❖ **Дизайн інтерфейсу** – розробка дизайну інтерфейсу користувача (UI) та взаємодії користувача (UX) з використанням певних інструментів;
- ❖ **Розробка функціональності** – реалізація основних функцій магазину: каталог товарів, кошик товарів, обробка з замовленнями, оплата тощо;
- ❖ **Інтеграція платіжних систем** – підключення системи оплати для обробки платежів;
- ❖ **Оптимізація для пошукових систем (SEO)** – впровадити оптимізації сайту для пошукових систем, налаштування мета-тегів, і так інші;
- ❖ **Тестування** – відзначати всіх аспектів веб-сайту для виявлення помилок;
- ❖ **Запуск проєкту** – старт готового продукту на основному сервері;
- ❖ **Підтримка продукту** – підтримка веб-сайту для вирішення проблем, оновлення контенту та функціональності.

Отже, зроблено аналіз предметної області, а саме поняття та основні фактори інтернет магазину, досліджено три аналогових інтернет-магазинів за певною темою, визначено переваги та недоліки інтернет магазину, а також функції його.



## РОЗДІЛ 2. ПОСТАНОВКА ЗАВДАННЯ ТА ВІДБІР АРХІТЕКТУРНИХ ІНСТРУМЕНТІВ

### 2.1. Мета та постановка завдання проєкту

У сучасному світі інтернет-торгівля стала невід'ємною складовою ринку споживчої техніки, зокрема комп'ютерної техніки. Розвиток технологій, зростання популярності онлайн-покупок та зручності їх виконання є основними причинами для створення інтернет-магазину для продажу комп'ютерної техніки.

Щодо розробки проєкту створення інтернет-магазину застосовують багато різних технологій, такі як:

- ❖ WordPress CMS – популярна система керування з відкритим кодом;
- ❖ MERN Stack – набір технологій, які створюють надійні та масштабні веб-програми за допомогою JavaScript. Він побудований на «М» - MongoDB, «Е» - Express, «R» - React, «N» - Node JS, які відіграють певну роль у процесі розробки;
- ❖ PERN Stack – складається з PostgreSQL, Express, React і Node JS;
- ❖ Payload CMS (на базі Headless CMS) – керована система управління контентом з додатками фреймворками.

Таблиця 2.1 – Основні характеристика якими керувалися для вибору одної технологій

	<b>Payload CMS</b>	<b>WordPress CMS</b>	<b>MERN Stack</b>	<b>PERN Stack</b>
<b>Зручність використання</b>	2	3	3	3
<b>Швидкість розробки</b>	3	2	3	3
<b>Гнучкість та розширюваність</b>	2	3	3	3

Продовження таблиці 2.1

<b>Безпека</b>	2	2	3	3
<b>Вартість розробки</b>	1	2	2	2

З аналізу із основної характеристики, можна зазначити, що:

- ❖ Payload CMS – дана система має середній рівень зручності використання та швидкості, але менш гнучка та безпечна, що є недоліком для конкретних проєктів. Щодо вартості розробки є нижчою через безкоштовну або доступну ціну CMS.
- ❖ WordPress CMS – є відомим простотою використання та швидкістю розробки, що робить популярним вибором для проєкту. Також має високий рівень гнучкості та розширюваності завдяки кількості плагінів та тем. Але безпека, є проблемою через використання та часті оновлення.
- ❖ MERN Stack – відзначає високу гнучкість, швидкість та безпекою. Даний стек, які часто використовують для створення веб-додатків. Проте вартість розробки – вищою, а особливо й якщо потрібні знання або розширення.
- ❖ PERN Stack – подібний до стеку MERN, але використовує базу даних PostgreSQL замість MongoDB. Він має високу гнучкість, швидкість та безпеку, але вартість розробки аналогічна як у MERN Stack.

Метою проєкту є розробка інтернет-магазину з продажу комп'ютерної техніки, використовуючи технологій, який мають забезпечити зручний та механізм покупок.

Payload CMS. Так як, він може бути кращим для проєктів, в плані гнучкості, продуктивності та безпеки.

Щодо реалізації даного проєкту в ході кваліфікаційною роботою, були застосовані такі завдання:

- ❖ Дослідити основні поняття інтернет-магазину, аналогів веб-сайтів інтернет-магазинів щодо даної теми;
- ❖ Обрати необхідний інструмент реалізації проєкту, звернувши увагу над сучасними технологіями;
- ❖ Візуалізувати модель бази даних для зберігання інформації про товари, категорій, замовлення, користувачів;
- ❖ Проектувати інтерфейсу користувача: основна сторінка, сторінка продукту, сторінка авторизації та створення облікованого запису, і так інші;
- ❖ Реалізувати функціонал для додавання товарів до кошика, створення та авторизації облікованого запису, оформлення замовлення і оплати;
- ❖ Забезпечити оплату платежів для користувачів.

### **2.3. Аналіз корисного навантаження CMS для створення веб-сайту**

В даній розділі, розглянемо сучасні технології, над системою керування вмісту – Payload CMS для створення веб-сайту інтернет-магазину.

**Основні поняття Payload CMS.** Payload CMS – це сучасна система керування вмістом, яка побудована на базі Headless CMS (Без заголовкова система керування вмістом) з структурою додатків. Він призначений для забезпечення величезних поштовх процесу розробки.

Headless CMS – сховище вмісту, яке дозволяє доставляти вміст у будь-який інтерфейс або інтерфейс користувача. Він зосереджений виключно на наданні адміністраторам ефективного способу створення та підтримки вмісту, але не контролює [8, 9].

З порівнюючи від традиційної CMS включає в себе шари адміністрування контенту і відображення, тобто передня і задня частина тісно інтегровані і представлені в одній системі.

З іншого боку, Headless CMS пропонує матеріал через API, який може бути використаний будь-якою кількістю фронтенд-додатків, таких як веб-сайти, мобільні додатки, розумні годинники і так далі (на рис. 2.1).

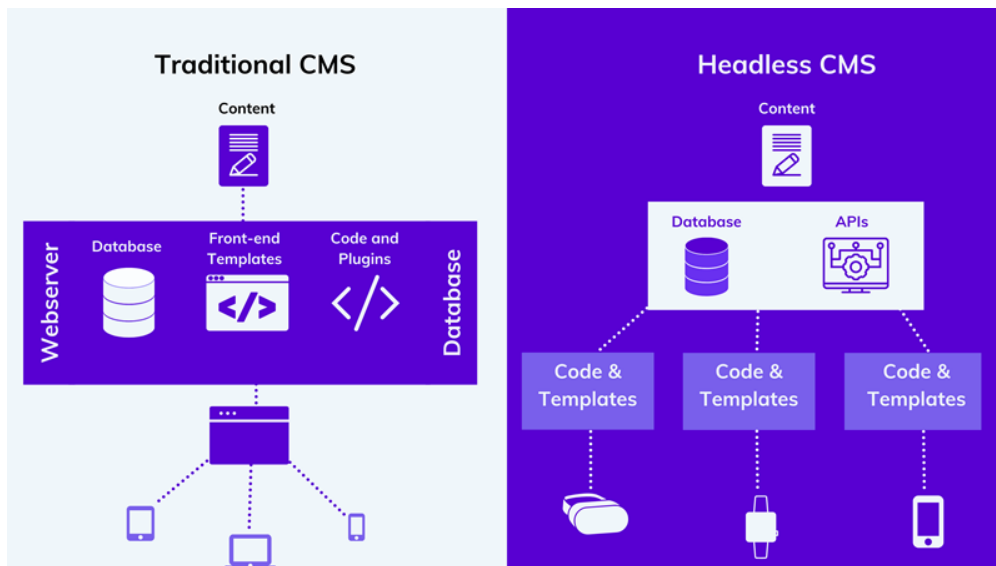


Рисунок 2.1 – Схема будови Traditional та Headless CMS

Таким чином, CMS може зробити процес редагування контенту дуже складним і ефективним, уникаючи при цьому творчих обмежень для дизайнерів і команд розробників. На противагу цьому, традиційні системи управління контентом прив'язують доставку контенту до його зберігання, серйозно обмежуючи творчість, розвиток і зручність використання контенту, яким вони керують [9].

Ця концепція зараз широко обговорюється в Інтернеті, і небезпідставно. Інтернет став складнішим, а зі складністю з'явилася потреба розробників краще структурувати свій код: поява інтерфейсних бібліотек, таких як React і Vue, стала фактичним для створення сучасних додатків [6].

**Панель адміністратора.** Payload динамічно створює красиву, повністю функціональну панель адміністратора React для керування даними. Він потужний і можна налаштувати або розширити шляхом заміни на власних компонентів React.

Панель адміністратора корисного навантаження може бути включена до офіційної підтримки Vite і webpack бандлери, або інтегрувати інший бандлер, дотримуючись підходу шаблону адаптера.

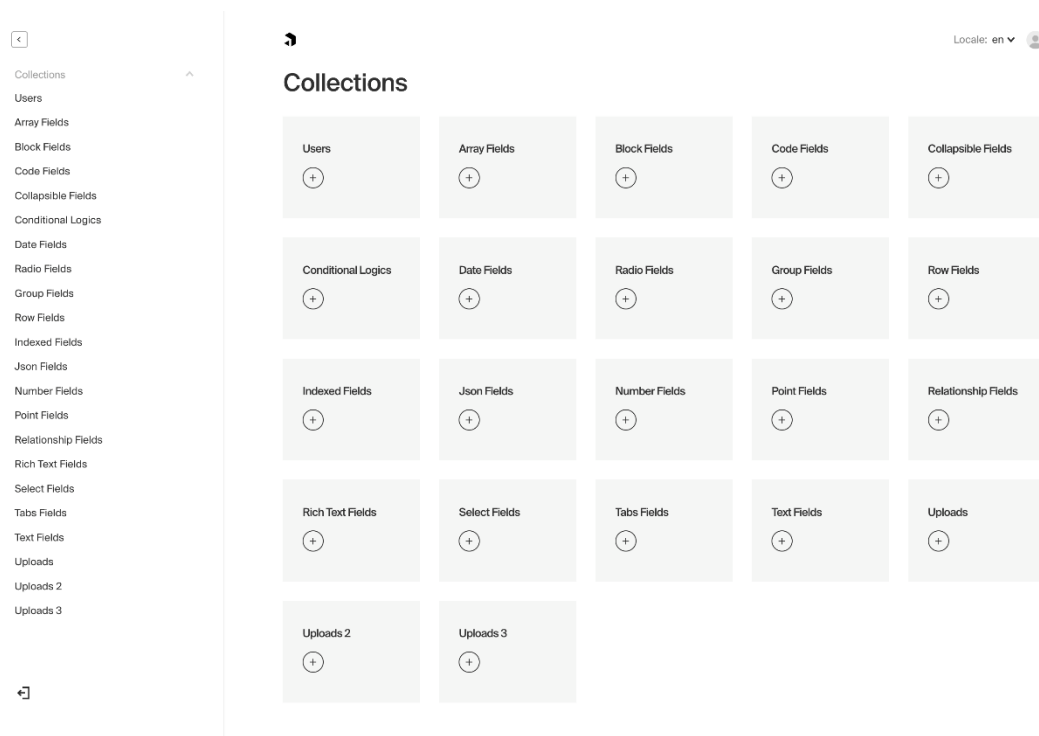


Рисунок 2.2 – Інтерфейс панелі адміністратора Payload CMS

Таблиця 2.2 – Основні параметри панелі адміністратора визначені у базовому конфігураційному файлі Payload.

Опції	Опис
<b>bundler</b>	Збірник, який використовується для об'єднання панелі адміністратора.
<b>user</b>	slug-колекції, які використовують для входу на інформаційну панель адміністратора.
<b>buildPath</b>	Абсолютний шлях для зберігання вбудованої панелі адміністратора
<b>meta</b>	Базові метадані для використання в панелі адміністратора.
<b>indexHTML</b>	За бажанням повністю замінити index.html файл.
<b>css</b>	Абсолютний шлях до таблиці стилів, яку можна використовувати для заміни або налаштувати стилю.

scss	Абсолютний шлях до таблиці стилів змінних/міксинів sass, призначений для заміни стилів корисного навантаження.
dateFormat	Глобальний формат дати, який використовуватиметься для всіх дат.
avatar	Встановлення зображення профілю облікованого запису.
autoLogin	Використовується для автоматизації входу адміністратора для зручності розробників і демонстрації.
logoutRoute	Маршрут для logout (вийти) сторінки.
inactivityRoute	Маршрут для logout (вийти) сторінки не активності.

## 2.2. Концепції Корисного навантаження (Payload CMS)

Payload базується на невеликому та інтуїтивно зрозумілому наборі понять. Для ознайомлення для розробки з Payload (Корисного навантаження), можна виділити значення:

- ❖ Конфігурація (Config) – у конфігурації Payload, налаштовуєте все, що робить Payload. За замовчуванням конфігурація Payload знаходиться в кореневій папці вашого коду та має назву «payload.config.js» (payload.config.ts, якщо це TypeScript). Можливість писати повні функції та навіть повні компоненти React прямо у вашій конфігурації [5];
- ❖ Колекції (Collections) – представляє тип вмісту, який буде зберігати Payload і може містити багато документів. Колекції визначають форму ваших даних, а також усі функції, пов'язані з цими даними. Вони міститимуть один або багато «документів», усі з яких відповідають тим самим полям і функціям. Вони можуть представляти все, що можуть зберігати в базі даних, наприклад сторінки, публікації, користувачів, люди, замовлення, категорії, події, клієнтів, транзакції та все інше, що потрібно програмі [5];

- ❖ Глобальні (Globals) – це «одноразовий» вміст, який ідеально підходить для збереження навігаційних структур, теми, метадані верхнього рівня тощо [5];
- ❖ Поля (Fields) – поля є будівельними блоками Payload. І колекції, і глобали використовують поля для визначення форми даних, які вони зберігають. Payload йде в комплекті багато різних типів полів, які дають вам масу гнучкості під час розробки вашого API. Кожен тип поля має власні потенційні властивості, які дозволяють налаштувати їх роботу [5];
- ❖ Хуки (Hooks) – це місце, де можна «прив'язати» до існуючих дій корисного навантаження, щоб виконати власну додаткову логіку або взагалі змінити спосіб роботи Payload. Хуки є надзвичайно потужною концепцією, яка є центральною для розширення та налаштування вашої програми. Payload надає широкий вибір хуків. [5];
- ❖ Управління доступом (Access Control) – контроль доступу відноситься до системи корисного навантаження, яка визначає, хто і що може робити з вашим API [5];
- ❖ Глибина (Depth) – дає змогу контролювати, на скільки рівнів нижче пов'язані документи мають бути автоматично заповнюється під час отримання [5].

## 2.4. Способи оплати інтернет-магазину

Для оплати на замовлення інтернет-магазину, потребують платіжні системи. Існують такі платіжні системи такі як: Stripe, PaySera, PayPal.

**Stripe** – це зручний і надійний провайдер онлайн-платежів. За замовчуванням у Stripe доступна оплата банківськими картками. Він також дозволяє підключити додаткові способи оплати, як-от Apple Pay, Google Wallet. Це залежить локації та валюти, яку використовує магазин. Її використовують такі імениті компанії, як Amazon, Google, GitHub, Samsung, Ford та інші (як показано на рис. 2.1) [19].

Система Stripe виступає посередником між покупцем та продавцем, який проводить безпечний, з погляду захисту, інформації платіж. Сервіс працює з приватними особами та компаніями різного розміру [19].

За допомогою цього плагіна, можна інтегрувати Stripe у Payload CMS. Це дає змогу легко синхронізувати дані вперед і назад, а також проксі Stripe REST API через контроль доступу Payload. Використовувати даний плагін, щоб повністю розвантажити платіжну систему для Stripe і зберегти повний контроль над даними програми [19].

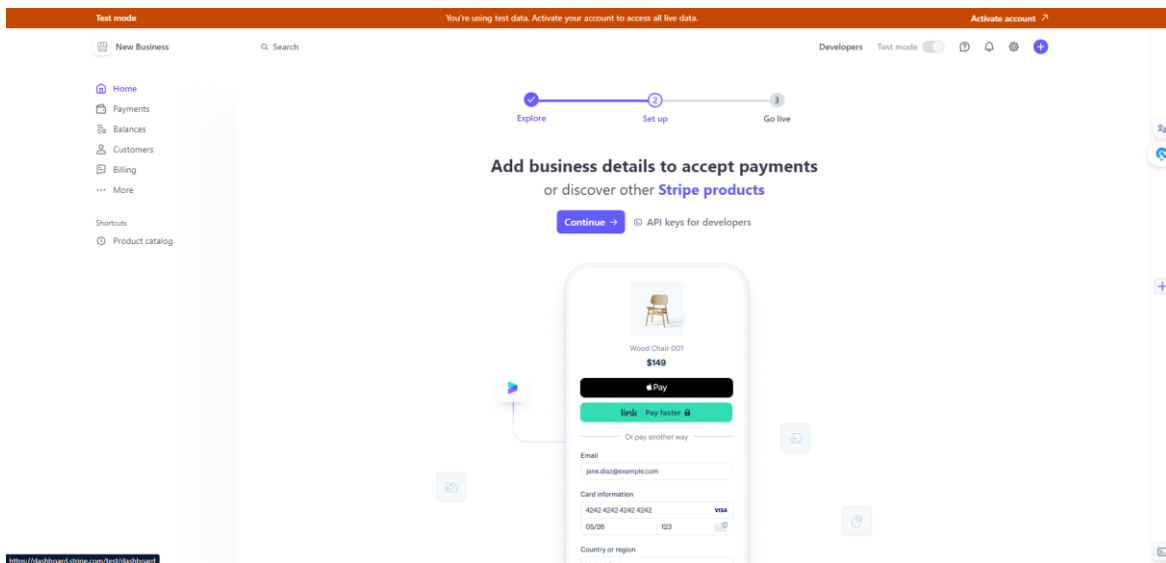


Рисунок 2.1 – Адміністративний панель Stripe

**Paysera** – це литовська фінансово-технологічна компанія та платіжна система, яка пропонує широкий спектр платіжних послуг. Вони створюють систему платежів без кордонів, доступну кожному, забезпечуючи швидкі та прості транзакції за вигідними цінами. Серед послуг Paysera – обслуговування як приватних, так і бізнес-клієнтів по всьому світу [20].

Дана система також послуги зі збору платежів для інтернет-магазинів і пропонує ряд способів оплати, включаючи картки Visa, які можна прив'язати до Apple Pay, Google Pay і Samsung Pay. Безпека є головним пріоритетом для Paysera, і вона використовує найвищі стандарти технологій та систем управління (на рис. 2.2).



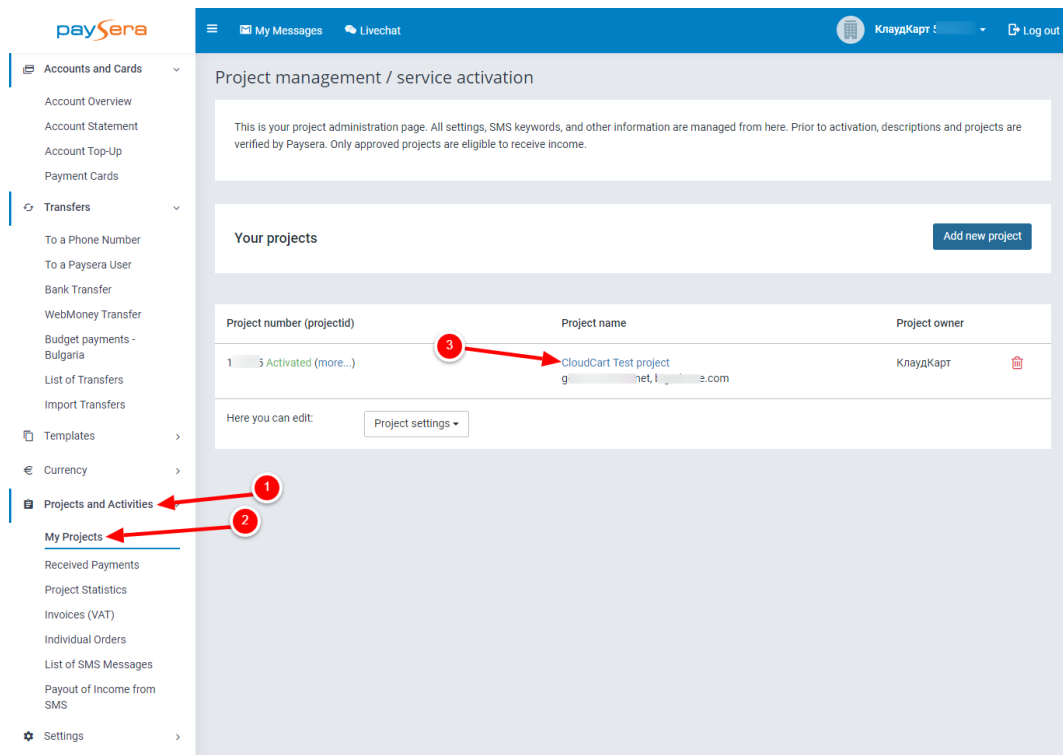


Рисунок 2.2 – Адміністративний панель PaySera

**PayPal** – це найпопулярніша платіжна система у світі, її використовують для оплати покупок, рахунків, грошових переказів, приймання платежів з банківських карток та рахунків (на рис. 2.3).

У різних країнах світу доступний різний перелік послуг:

- ❖ **Send only** – PayPal можна використовувати тільки для оплати товарів та послуг та переказу грошей;
- ❖ **Zero** – користувач може приймати платежі, оплачувати товари та послуги, але не може тримати гроші на рахунку;
- ❖ **SRW Send-Receive-Withdrawal** – зарахування, введення та виведення грошей, можливість тримати гроші на рахунку в доларах/євро та перераховувати їх на картку;
- ❖ **Local Currency** – все тещо в пункті вище плюс можливість вести операції в національній валюті [6].

Переваги PayPal:

- ❖ Простота використання;
- ❖ Швидкість виконання платіжних операцій;

- ❖ Для оплати або переказу грошей достатньо знати адресу електронної пошти одержувача, на яку зареєстровано рахунок;
- ❖ PayPal пропонує своїм клієнтам додатковий захист від шахраїв;
- ❖ Якщо умови угоди вимагають повернення товару, то відшкодовує вартість доставки;
- ❖ Оплата покупок на Ібей та інтернет-магазинах здійснюється без комісії.

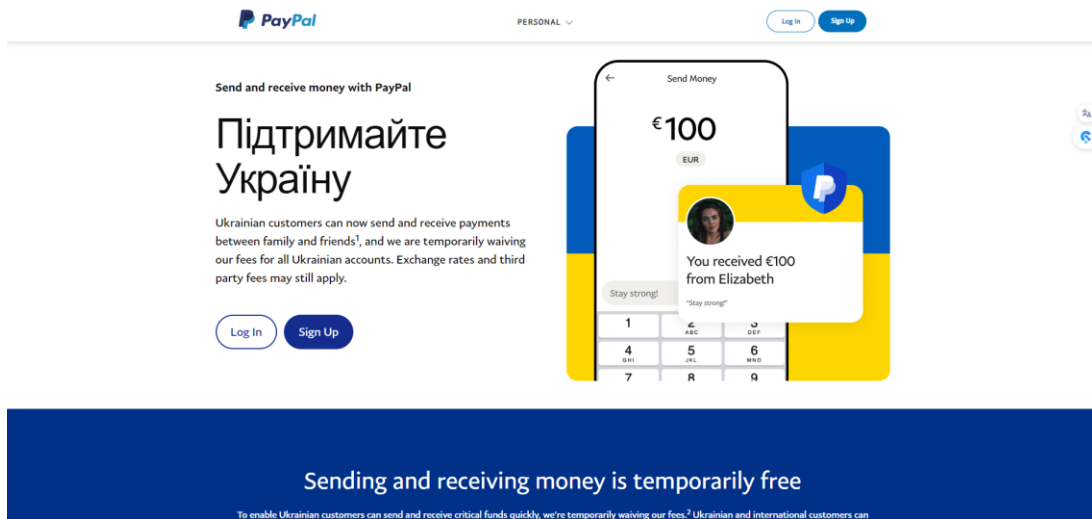


Рисунок 2.3 – Офіційний сайт PayPal для України

Отже, розглянуто аналіз відкритого коду над контентом вмісту управління Корисного навантаження (Payload CMS), яка складається з таких фреймворків, мов програмування та каскадних стилів як: React, Next JS, TypeScript та SCSS. Проаналізовано основні Payload CMS концепції, досліджено та аргументовано способи оплати інтернет-магазину за допомогою сучасних платіжних систем.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ

### 3.1. Функціональне моделювання діяльності інтернет-магазину для продажу комп'ютерної техніки

IDEF0 – це функціональний інструмент моделювання підприємства, який був розроблений у рамках програми Інтегрованого автоматизованого виробництва (ICAM) ВПС США. IDEF означає визначення ICAM. IDEF0 використовується для представлення функціональної (тобто орієнтованої на діяльність або процесу) [7].

Функціональне моделювання діяльності інтернет-магазину включає опис процесів та операцій, які відбуваються в системі з моменту відвідування користувачем сайту до отримання замовленню товару.

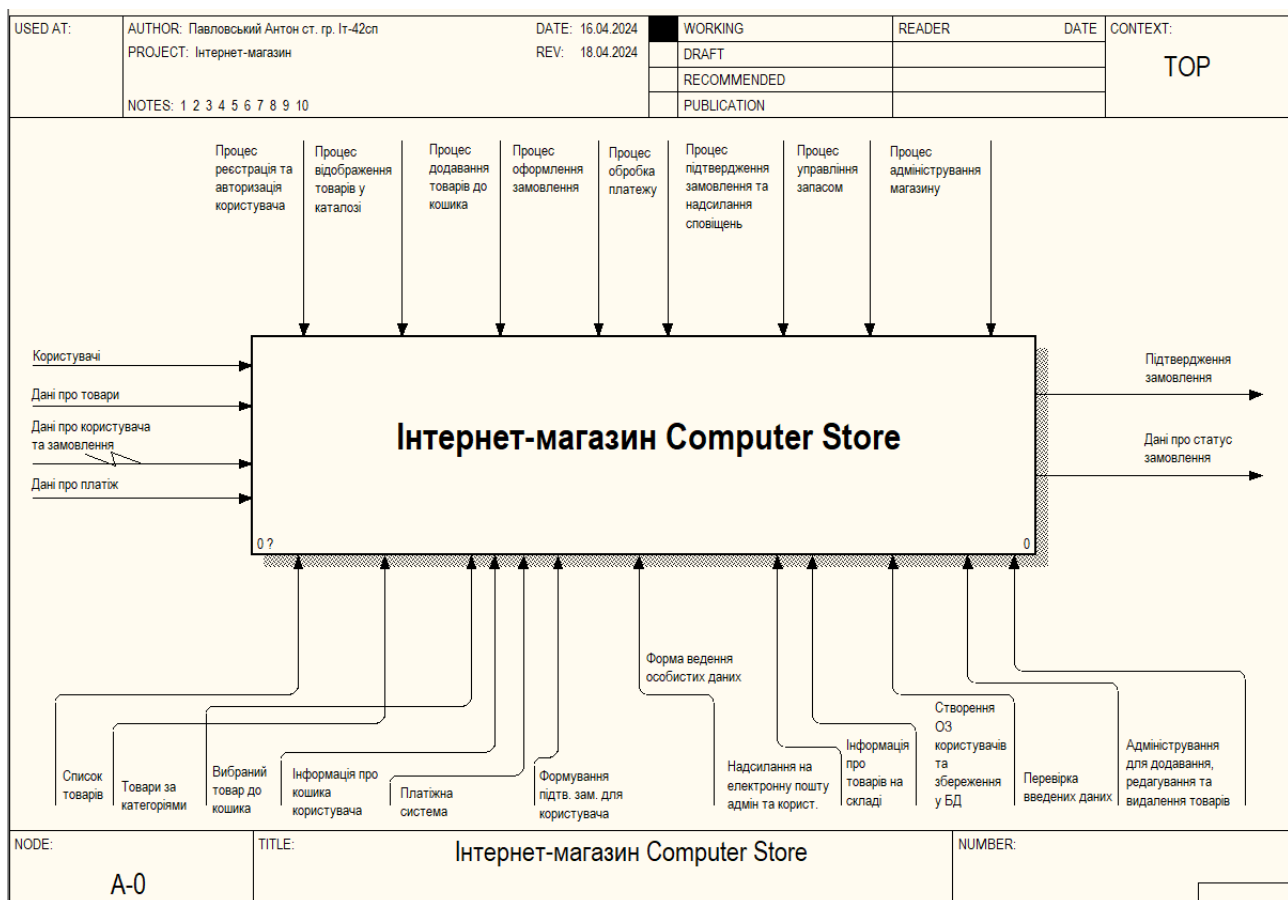


Рисунок 3.1 – Інформаційна система інтернет-магазину для продажу комп'ютерної техніки

Основними функціями інтернет-магазину є:

❖ **Перегляд товарів з каталогу:**

- Вхідні дані: користувач переглядає товар із каталогу на сайті;
- Операції: 1. Система відображає список доступних товарів. 2. Користувач застосовує фільтри для пошуку товару;
- Вихідні дані: користувач переглядає каталог товарів та вибирає певний товар для оформлення замовлення.

❖ **Додавання товару до кошика:**

- Вхідні дані: користувач обирає товар, переглядаючи на характеристику про даного товару, додає товар до кошика;
- Операції: 1. Система додає вибраний товар до кошика користувача. 2. Відображає повідомлення про успішне додавання товару до кошика;
- Вихідні дані: користувач має товар в кошику та переходить до оформлення замовлення.

❖ **Оформлення замовлення:**

- Вхідні дані: користувач перейшов до сторінки оформлення замовлення з кошика;
- Операції: 1. Система відображає форму для введення даних для оплати та доставки. 2. Користувач вводить необхідні дані та обирає спосіб оплати та доставки;
- Вихідні дані: система отримує дані для обробки замовлення.

❖ **Обробка платежу:**

- Вхідні дані: користувач вибрав спосіб оплати та ввів необхідні дані;
- Операції: 1. Система передає дані для обробки платежу до платіжної системи. 2. Платіжна система проводить операцію оплати;
- Вихідні дані: система отримує підтвердження про успішну оплату.

❖ **Підтвердження замовлення та надсилання сповіщень:**

- Вхідні дані: система отримує підтвердження про здійснену оплату;

- Операції: система формує замовлення та надсилає на електронну пошту користувача;
- Вихідні дані: користувач отримує підтвердження про замовлення, а власник сайту отримує сповіщення обробки замовлення.

❖ **Управління запасами:**

- Вхідні дані: система отримує інформацію про нове замовлення та здійснену оплату;
- Операції: система оновлює кількість товарів на складі, видаляючи одиниці замовлень;
- Вихідні дані: система має поточну інформацію про кількість товарів на складі.

Додатковими функціями інтернет-магазину є:

❖ **Авторизація та реєстрація користувачів:**

- Вхідні дані: користувач вводить необхідні дані із форми авторизації або реєстрації;
- Операції: система приховує паролів користувача та перевіряє введення форм на достовірність користувача;
- Вихідні дані: система після перевірки, пускає користувача на даний сайт.

❖ **Адміністрування магазину:**

- Вхідні дані: власник керує та контролює роботу над інтернет-магазином;
- Операції: власник додає, видаляє, редагує товари. Система моніторить (наповняє) замовлень та їх статусів. Власник запускає промоакцій та знижок на товари;
- Вихідні дані: система оновлює дані на сайті продажу товару.

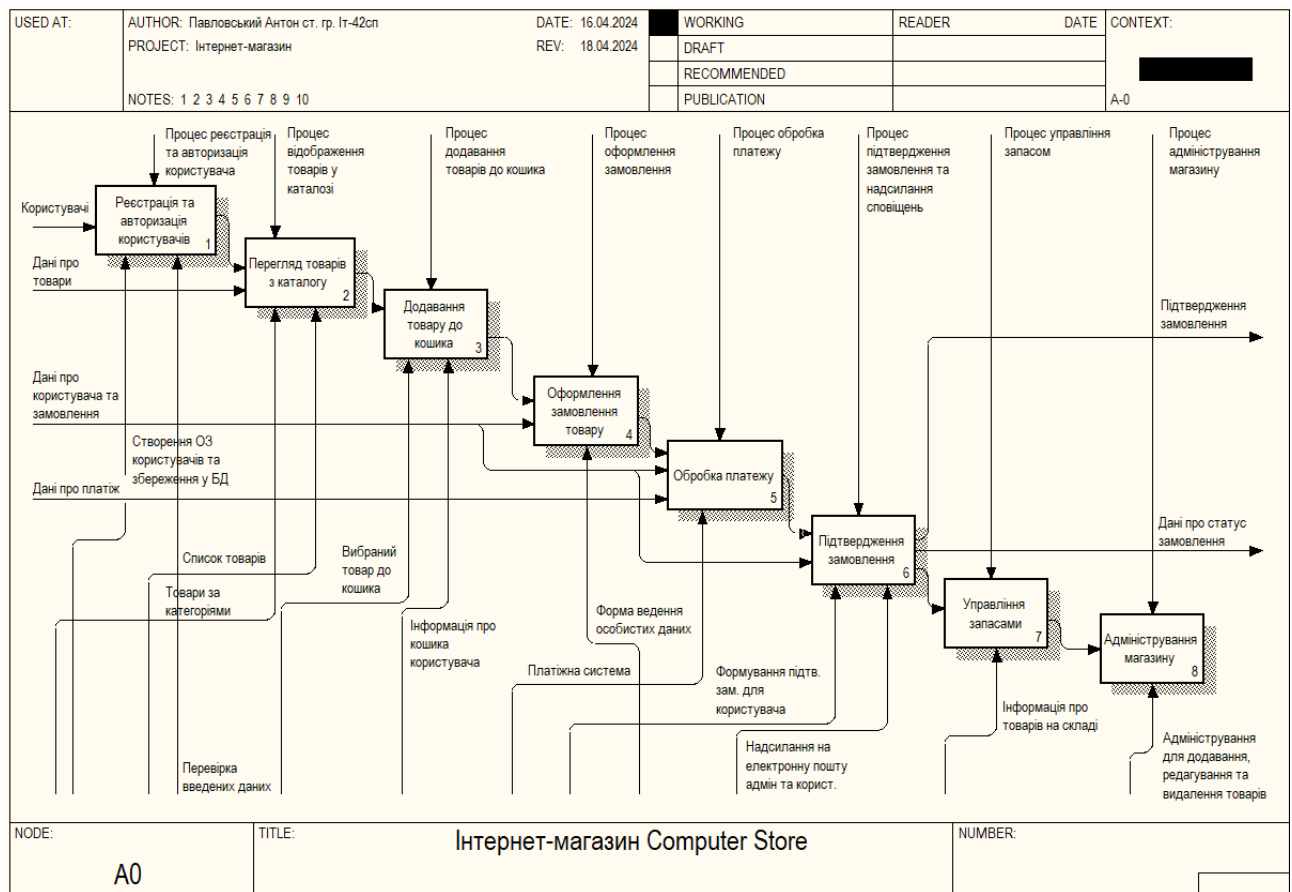


Рисунок 3.2 – Діаграма IDEF0 інтернет-магазину для продажу комп'ютерної техніки

### 3.1.1. Проектування моделі бази даних із MongoDB

У цьому пункті, розглядається процес проектування моделі бази даних для інтернет-магазину, побудованого на базі Payload CMS з використанням MongoDB. Дана модель включає такі колекції даних, які стандартно надаються Payload CMS, а саме: `_page_versions`, `_product_versions`, `categories`, `media`, `orders`, `pages`, `payload-migrations`, `payload-preferences`, `products`, `redirects`, `users`.

MongoDB є типовим представником баз даних NoSQL. Завдяки архітектурі він має безліч переваги перед традиційними рішеннями SQL. Основні моменти є гнучка схема, масштабованість і простота використання [10].

Перш за все, визначені до бази даних на основі інтернет-магазину. З основних вимог зазначає:

- ❖ Зберігання інформації про товари – назва, опис, ціни, наявність тощо;

- ❖ Організація товарів за категоріями для полегшення користувачів;
- ❖ Зберігання даних про замовлення користувачів – замовленні товари, вартість, статус тощо;
- ❖ Авторизація та створення облікових записів користувачів – особисті дані, історія покупок й замовлень, змінення даних;
- ❖ Зберігання медіафайлів – зображення категорій, товарів;

Для відповідності вимогам і структурування інформації було розроблено основні наступні моделі баз даних (на рис. 3.3):

- ❖ Таблиця users:
  - id – унікальний ідентифікатор користувача;
  - name – ім'я користувача;
  - roles – масив ролей користувача, наприклад, адміністратор, модератор, звичайний користувач;
  - cart – об'єкт, який містить інформацію про кошика користувача;
  - hash – хешування пароллю;
  - loginAttempts – кількість спроб входу;
  - purchases – масив замовлень, які зробив користувач.
- ❖ Таблиця orders:
  - id – унікальний ідентифікатор замовлення;
  - orderedBy – ідентифікатор користувача, який зробив користувач;
  - total – загальна сума замовлення;
  - items – об'єкт, що містить дані про елементи замовлення, наприклад, інформацію про кожен продукт, кількість тощо.
- ❖ Таблиця media:
  - id – унікальний ідентифікатор медіафайлу;
  - alt – альтернативний текст для медіафайлу;
  - caption – підпис до медіафайлу;
  - filename – назва файлу;
  - filesize – розмір файлу в байтах;
  - width – ширина медіафайлу;

- height – висота медіафайлу.
- ❖ Таблиця categories:
  - id – унікальний ідентифікатор замовлення;
  - title – назва категорій;
  - media – посилання на медіафайл.
- ❖ Таблиця products:
  - id – унікальний ідентифікатор продукту;
  - title – назва продукту;
  - publishedOn – дата публікації продукту;
  - status – статус продукту;
  - categories – ідентифікатори категорій, до яких належить продукт;
  - relativeProducts – ідентифікатор пов'язаних продуктів.

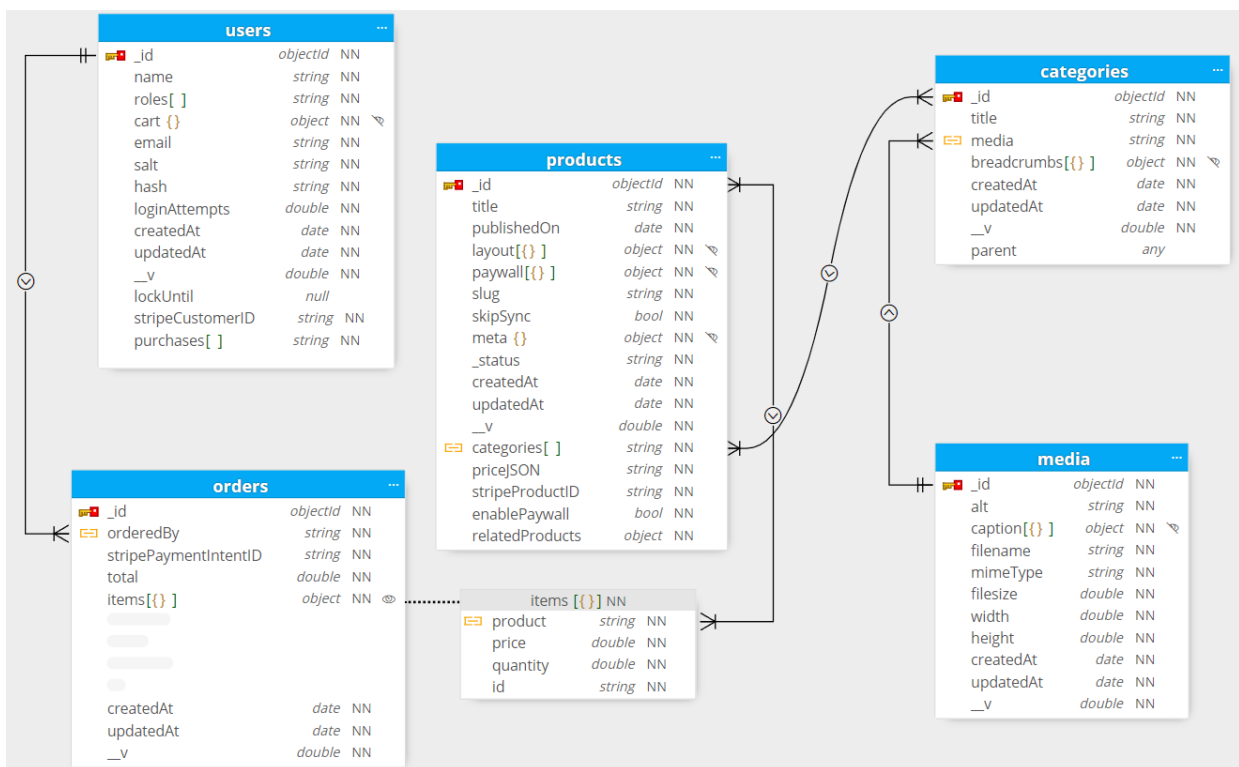


Рисунок 3.3 – Модель бази даних на основі Payload CMS

Зв'язки між таблицями в базі даних інтернет-магазину:

- ❖ Зв'язок «один-до-багатьох» між таблицями «users» та «orders». Кожен користувач може здійснюватиме декілька замовлень, але кожне



замовлення належить лише одному користувачеві лише одному користувачеві;

- ❖ Зв'язок «багато-до-багатьох» між таблицями «orders» та «products». У кожному замовленні може бути декілька продуктів, а кожен продукт може частиною декількох замовлень;
- ❖ Зв'язок «багато-до-багатьох» між таблицями «products» та «categories». Продукти розміщені у різних категоріях, і кожна категорія може містити декілька продуктів;
- ❖ Зв'язок «один-до-багатьох» між таблицями «categories» та «media». Кожен категорій має декілька зображень, але кожне зображення належить одному категорій.

### 3.2. Побудова UML-діаграми інтернет-магазину

Діаграма системної послідовності – це створений артефакт, який ілюструє події введення та виведення, пов'язані з обговорюваними системами. UML містить позначення у формі діаграми послідовності для ілюстрації подій від зовнішніх акторів до системи. UML включає діаграми послідовності як нотацію, яка може ілюструвати взаємодію акторів та операції, ініційовані ними [4].

Варіанти використання описують, зовнішні актори взаємодіють із програмною системою. Під час цієї взаємодії актор генерує події для системи, зазвичай запитуючи певну операцію у відповідь [4].

Для проєктування UML-діаграми даного інтернет-магазину були визначені такі основні вимоги:

- ❖ Показ взаємодії між користувачами, системою магазину та системою оплати;
- ❖ Відображення процесу замовлення та оплати товарів;
- ❖ Врахування інтерфейсів користувача для перегляду продуктів, додавання їх до кошика та оформлення замовлення.

Для відображення взаємодії компонентів системи було розроблено Діаграму послідовностей для оформлення товару інтернет-магазину (показано на рис. 3.4).

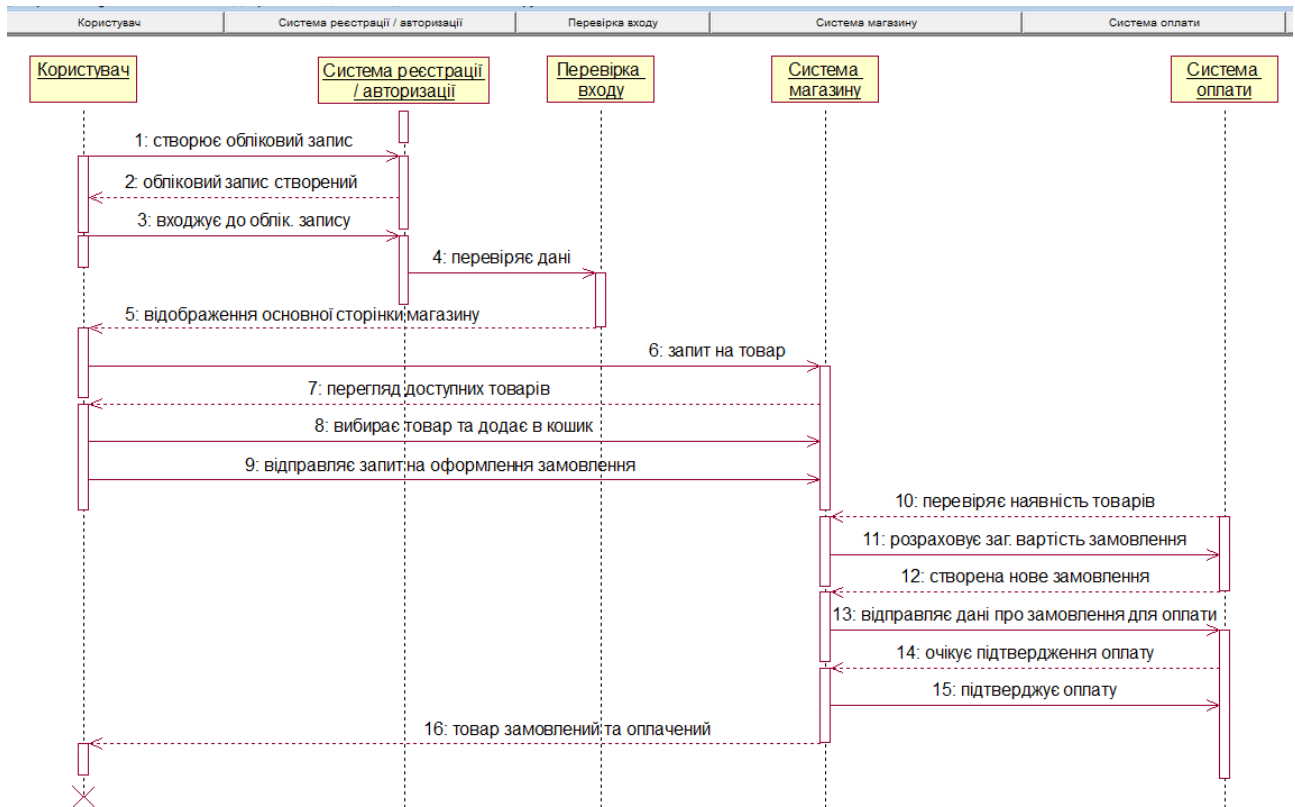


Рисунок 3.4 – Діаграма послідовностей інтернет-магазину для замовлення товару

В даній діаграмі показує послідовність взаємодії між користувачами та системою під час процесу замовлення товарів. Тобто, включає кроки вибору продуктів, додавання їх до кошика, оформлення замовлення та оплати.

Послідовність в даній діаграмі полягає в тому, що:

- Користувач створює обліковий запис;
- Система реєстрації/авторизації створює обліковий запис та повідомляє про це користувачеві;
- Користувач входить до облікованого запису;
- Система реєстрації/авторизації перевіряє дані користувача та повідомляє про результат перевірки системі магазину;
- Система відображає користувачеві основну сторінку магазину;
- Користувач запитує товар;

- Система магазину переглядає доступні товари та повідомляє про них користувачеві;
- Користувач вибирає товар та додає його в кошик;
- Користувач відправляє запит на оформлення замовлення;
- Система магазину перевіряє наявність товарів у кошику та повідомляє про неї користувачеві;
- Система магазину створює нове замовлення та повідомляє про неї користувачеві;
- Система магазину відправляє дані про замовлення системі оплати
- Користувач очікує підтвердження оплати;
- Система оплати підтверджує оплату та повідомляє про це системі магазину;
- Система магазину підтверджує факт оплати та повідомляє про це користувачеві.

### **3.3. Розробка інтернет-магазину для продажу комп'ютерної техніки**

Для розробки інтернет-магазину було обрано технологію – Payload CMS (Корисне навантаження) та базу даних MongoDB, так як вона гнучка.

Першим етапом, було встановлення даної системи Payload CMS. Також необхідно створити базу даних (MongoDB або PostgreSQL). На даному часі існують два способи для встановлення системи та створення бази даних:

- 1) За допомогою команди «`npx create-payload-app@latest`» у терміналі в програмі WebStorm або VS Code, створивши базу даних MongoDB на офіційному сайті для введення ключа бази даних під час встановлення проекту;
- 2) За допомогою офіційного сайту Payload CMS для створення інтернет-магазину (e-commerce), крім того й веб-сайт на сайті «<https://payloadcms.com/new>».

Якщо припускати, до першого способу, то під час та в результаті розробки проєкту інтернет-магазину, весь функціонал запуститься тільки через локальний сервер.

Якщо до другого способу, то набагато простіше й цікавіше, так як під час створення проєкту, потребує приєднання/підключення до власного GitHub для створення файлу проєкту. А база даних від MongoDB по замовчуванню буде створена разом із проєктом (на рис. 3.5).

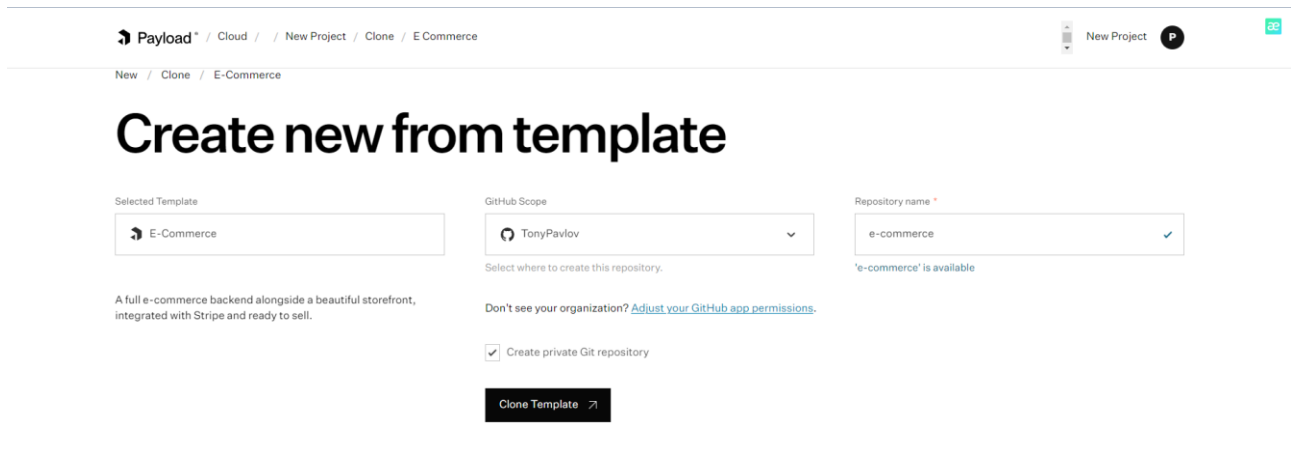


Рисунок 3.5 – Створення проєкту інтернет-магазину Payload CMS

Різниця між двома способами, у другому способі є можливість захостити за допомогою плагіну Payload Cloud та GitHub, де знаходяться файли створеного проєкту. Але даний сервіс є платним, тобто спочатку після створення проєкту дають на 7 днів пробного періоду, на далі, перейде на тариф Standard, який за ціною буде – 35\$. А у першому – за встановлення проєкту від Payload CMS та окремо створення бази даних MongoDB є безкоштовним, але з певними обмеженнями, тобто наприклад: під час запуску розробленого проєкту дають – 60 год/міс.

Другим етапом, так як вибрав другий спосіб, створивши проєкт, витягнув файли із панель керування Payload CMS, де знаходяться у власному GitHub, помістивши шаблонні файли в проєкт у програмі WebStorm (на рис. 3.6).

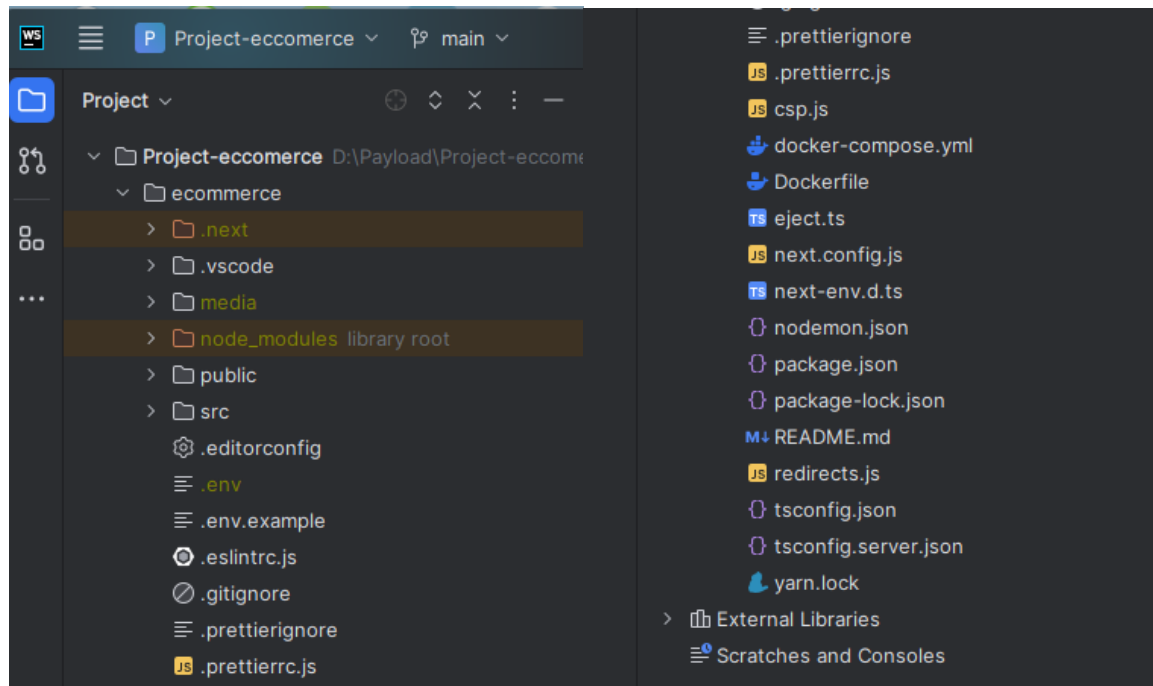


Рисунок 3.6 – Файли створеного проекту інтернет-магазину Computer Store

В даному файлі, по замовчуванню додані початкові й шаблонні файли для проекту e-commerce. Назва даного проекту – ecommerce. Також проект побудовано на основі мови програмування TypeScript, Next JS та SCSS.

TypeScript – мова з відкритим кодом, розроблена Microsoft, і є надмножиною JavaScript, тобто, функціонує як JavaScript із необов'язковими статичним типом. На практиці трапляються рідко, оскільки більшість програм, написаних на JavaScript, використовують пакети та «зломи» [2].

Sassy Cascading Style Sheets (SCSS) – мова препроцесора, дозволяє використовувати змінні та цикли в CSS. Завдяки цьому дизайн є узгодженим для всіх веб-сторінок і зменшує обслуговування завдяки централізації визначених змінних [3].

Next JS – це фреймворк, розроблений для подолання обмежень React у відображенні на стороні сервера [1].

З основних директорій каталогів та файлів у папці «src/app» поділяються на:

- ❖ (pages) – каталог, який містить сторінки сайту: «[slug]» – головна, «account» – персональна інформація користувача, «cart» – кошик, «checkout» - оформлення замовлень та оплати, «login» - авторизація користувача, logout – вихід з ОЗ, orders – перелік замовлень» тощо;

- ❖ `_ari` – каталог, містить у різних файлах для налаштування API веб-сайту;
- ❖ `_components` – каталог, які знаходяться компоненти, використовуючи на сторінках веб-сайту;
- ❖ `_css` – каталог містить файли стилів CSS для веб-сайту;
- ❖ `_hero` – каталог містить файли з компонентами веб-сайту;
- ❖ Файл `layout.tsx` – макет веб-сайту, тобто містить загальну структуру сторінки: шапка, навігаційне меню тощо.

Третім етапом, під час роботи над даним проєктом було реалізовано та від стилізовано ключові функції інтернет-магазину:

- ❖ Авторизація входу (Login) та реєстрація користувача (Create account);
- ❖ Відновлення паролю користувача (Recover password);
- ❖ Каталог товарів з фільтром (Products);
- ❖ Кошик товарів (Cart), оформлення замовлень та оплати (Checkout);
- ❖ Персональна інформація про користувача (Account);
- ❖ Мої замовлення (Order) та мої покупки (Purchases).

Для стилізації даного сайту, потребують файл з розширенням `scss`. В більшості, різні стилізовані файли розміщені у директорії «`app/(pages)`» та «`app/_components`» (на рис. 3.7).

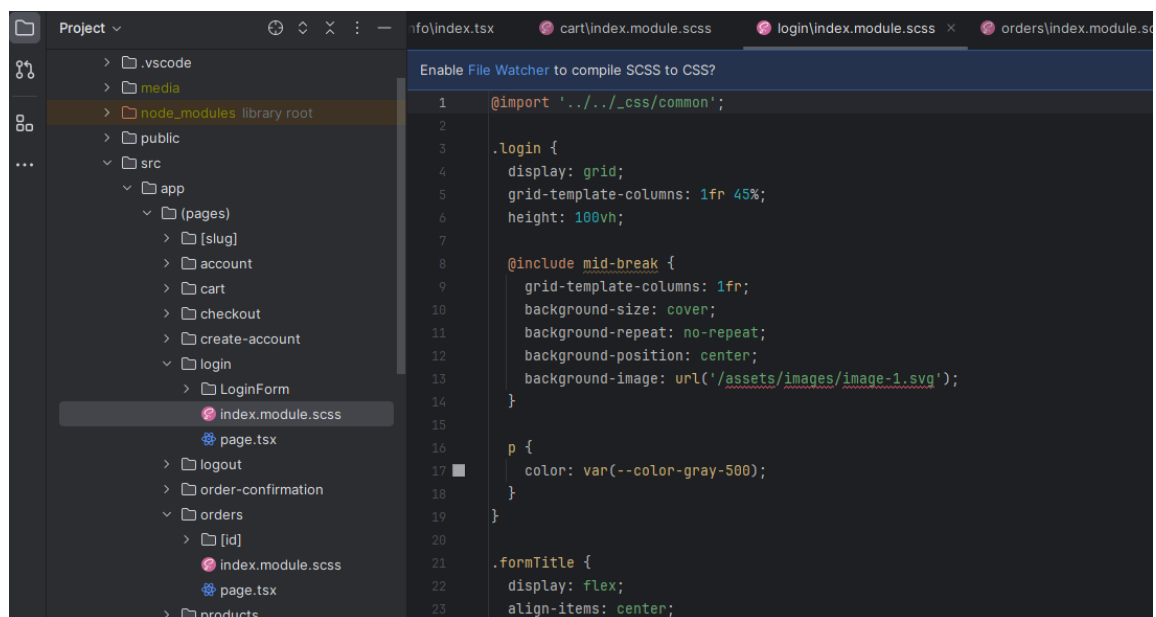
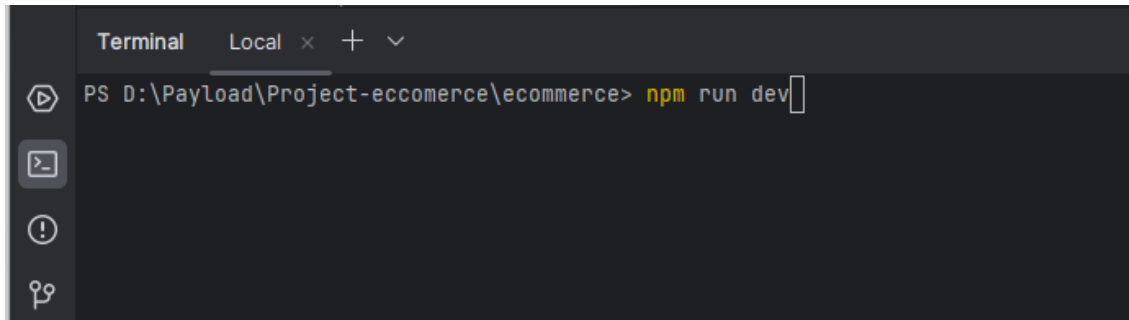


Рисунок 3.7 – Загальний вигляд стилізації сторінки авторизація користувача

Для запуску та тестування даного проєкту на локальний сервер, слід у терміналі у кореневому каталозі «ecommerce» прописати таку команду (як показано на рис. 3.8):



```
Terminal Local x + v
PS D:\Payload\Project-ecommerce\ecommerce> npm run dev
```

Рисунок 3.8 – Команда, для запуску проєкту інтернет-магазину на локальний сервер

### 3.3.1. Розробка та опис коду над створенням персональної інформації користувача

Для створення розробки персональної інформації користувача, використовував мову програмування TypeScript з фреймворком Next JS на основі React.

На початковому сторінці інтерфейсу створеного проєкту персональної інформації користувача по замовчуванню буде показано на рис. 3.9:

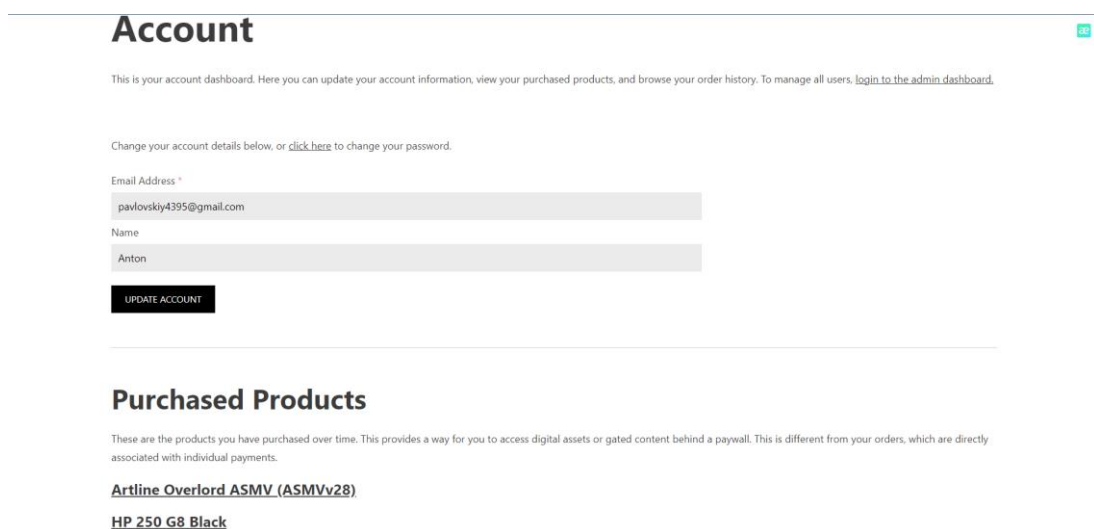


Рисунок 3.9 – Інформаційний панель облікованого запису

В даному сайті персоналізованої інформації користувача поділяються на такі блоки:

- ❖ Придбані товари (Purchased Products) – це продукти, які придбали з часом, який дає можливість отримати до цифрових активів або закритого вмісту за платним доступом.
- ❖ Замовлення (Order) – це замовлення, які користувачі розмістили протягом певного часу. Кожне замовлення пов'язане з наміром оплати.
- ❖ Можливість оновити інформацію та пароля свого облікованого запису.

Ця сторінка, обслуговується у директорії каталогу «src/app/(pages)/account», там знаходяться файл функціоналу сторінки, файл форми персональної сторінки та файли його стилів.

Тепер від стилізуємо та змінимо самий код, для того щоб змінити самий вигляд інтерфейсу та функціонал логіки процесу. Для змінення коду у програмі WebStorm, були добавлені у директорії «account» в підкатегорії: «purchases», «UserInfo» та «orders» [Додаток А]. З основних елементів компоненту та функцій, можна навести:

- ❖ Форми облікованого запису (AccountForm):
  - Форма для зміни даних облікованого запису;
  - Повідомлення про помилку та успіх – відображає повідомлення про помилку та успіх у разі виникнення під час зміни даних ОЗ;
  - Умовний рендерінг для зміни пароля – оператор на відображення одного блоку, якщо користувач не вибрав зміну пароля, та інший блок, якщо користувач вибрав зміну пароля;
  - Поля для введення електронної пошти та ім'я – компонент «<Input/>» з певними параметрами, для введення електронної пошти та імені користувача для зміни даних ОЗ;
  - Поля для зміни пароля – компонент «<Input />» з певними параметрами, для введення нового пароля та підтвердження пароля;
  - Кнопка для збереження змін – компонент «<Button />», який відображає кнопку для збереження змін у ОЗ.



❖ Перелік замовлень користувача (orders):

- Список замовлень користувача – містить такі елементи та компоненти:
  - «<h5>» заголовок, який нижче буде відображення список замовлень;
  - Оператори, який перевіряє, чи існують замовлення;
  - «<RenderParams />» – відображення додаткових параметрів замовлень;
  - «<ul>» – відображення списку замовлення користувача.
- Інформація про кожне замовлення – в них містять такі елементи та компоненти:
  - «<li>» – кожне замовлення відображає елемент списку;
  - «<Link>» – посилання на деталі замовлення;
  - «<h6>» – заголовок з номером замовлення;
  - «<p>» – відображає дату та час створення замовлення.
- Кнопка для перегляду замовлення – компонент «<Button />», дозволяє користувачеві переглянути деталі замовлення;

❖ Придбані товари (purchases):

- Список придбаних товарів – в них містять елементи та компоненти:
  - «<h5>» – заголовок, що нижче відображений список придбаних товарів;
  - Оператор, який перевіряє, чи існують придбані товари;
  - «<ul>» – відображення списку придбаних товарів користувача;
  - Ітерація по кожному придбаному товару у масиві «purchases» користувача.
- Інформація про кожен придбаний товар – в них містять такі елементи та компонентів:
  - «<li>» – кожен придбаний товар;
  - Оператори, які перевіряють тип купленого товару;

- «<Link>» – посилання на сторінку з деталями придбаного товару;
  - «<h6>» – заголовок відображення назви придбаного товару;
- Повідомлення про відсутність придбаних товарів – це повідомлення, яке відображає, якщо користувач не має придбаних товарів.

Результат оновленого сайту сторінки інформаційної панелі користувача показано на рис. 3.10:

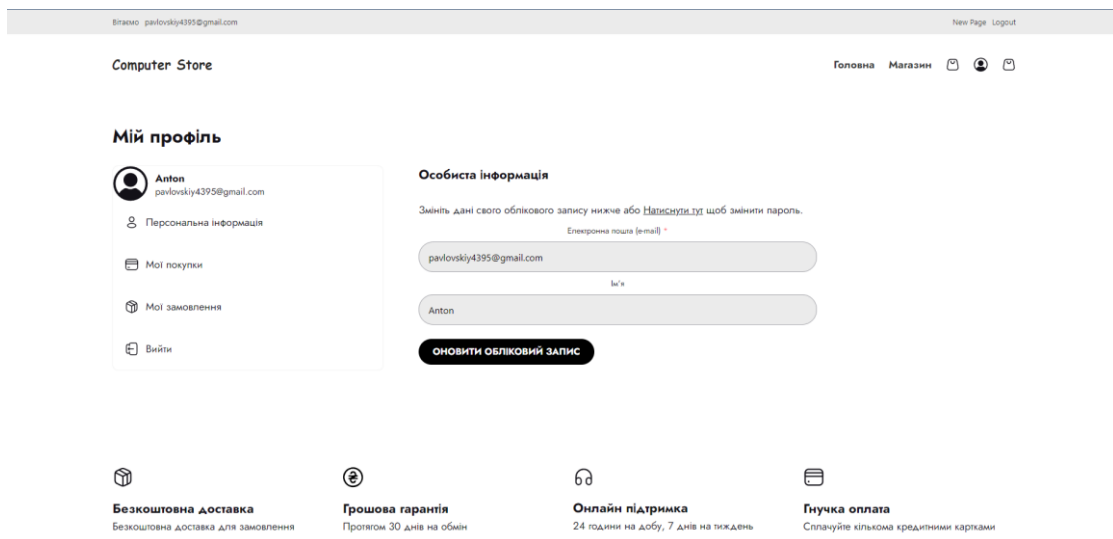


Рисунок 3.10 – Оновлений сайт персональної інформації користувача

### 3.3.2. Розробка та опис коду створення авторизації облікованого запису

Розглянемо, створення авторизації облікованого запису. Дана сторінка призначена для входження до головної сторінки, крім того при відсутності паролю або створення ОЗ є можливість реєструвати ОЗ та відновлення паролю. По замовчуванню після створеного проєкту буде виглядати ось так:

## Log in

This is where your customers will login to manage their account, review their order history, and more. To manage all users, [login to the admin dashboard](#).

Email Address \*

Password \*

LOGIN

[Create an account](#)

[Recover your password](#)

### Рисунок 3.11 – Авторизація облікованого запису

Дана сторінка знаходиться у директорії «login» з підкатегоріями «LoginForm» з файлами сторінки «page.tsx», логіки функції index.tsx та файли стилів. Для змінення функціоналу та стилів основної сторінки та форми авторизації користувача в кодї, потрібно застосовувати певні React-функції та стилів. З основних елементів з коду у [Дотаток Б] можна зазначити:

- ❖ Основна сторінка авторизації (login):
  - Розділ сторінки для входу – визначає розділ сторінки для входу користувача;
  - Логотип – компонент «Link», який посилає на головну сторінку сайту, використовуючи компоненту «Image»;
  - Форма входу – компонент «<LoginForm />» містить поля для введення необхідної інформації, такі як: електронна пошта та пароль;
  - Додаткові елементи – компонент «<RenderParam />», відображає додаткові компоненти, наприклад додаткова інформація про користувача;
  - Зображення та текстовий блок з привітанням – містить зображення руки та текстовий блок з привітанням користувача, містить

компонент «<Image>» для зображення руки, а «<h3>» - для заголовка тексту.

❖ **Форми авторизації (LoginForm):**

- Форма для входу – визначає форму для введення електронної пошти та пароля користувача для входу на сайт;
- Повідомлення про помилку – відображає повідомлення про помилку, у разі виникнення під час спроби входу;
- Поля для введення електронної пошти та пароля – компонент «<Input />» з певними параметрами, який відображає поля вводу для електронної пошти та паролю;
- Кнопка для відправлення форми – компонент «<Button />», який відображає кнопку для відправки форми;
- Посилання для створення облікового запису та відновлення паролю – компоненти «<Link />», які відображають гіперпосилання на сторінки «Створення облікового запису» та «Відновлення паролю».

Результат оновленого функціоналу та стилів авторизації користувача показано на рис. 3.12:

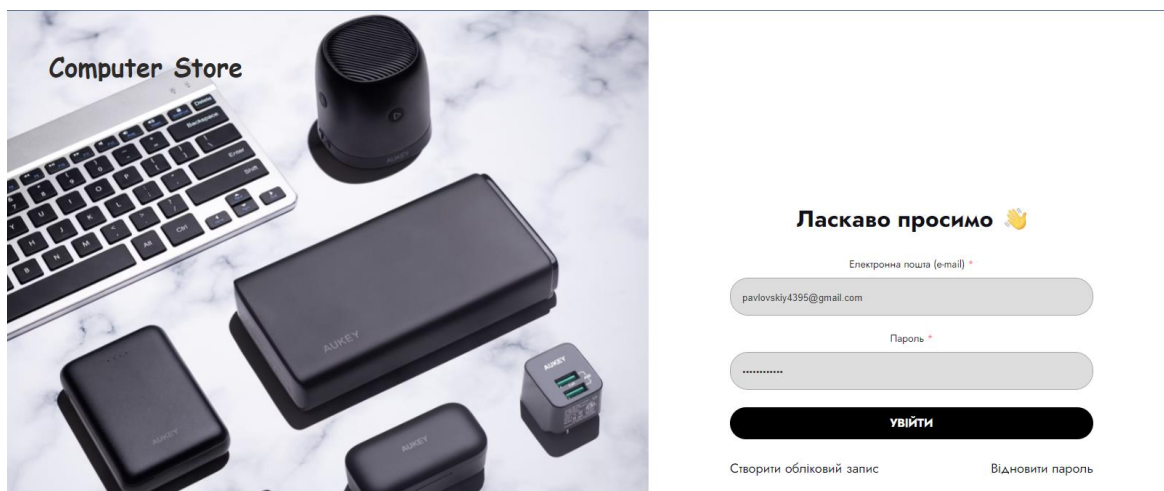


Рисунок 3.12 – Оновлена сторінка авторизації користувача

Також, передбачено, сам власник-адміністратора може обрати своїх помічників-модераторів із зареєстрованими користувачами, тобто у ролі «admin» для додавання, редагування, видалення товарів. А всі решта користувачі, які

користуються тим інтернет-магазину у ролі «custom». Для обрання модераторів, потрібно у інформаційному панелі власника, змінити визначену користувачу у вкладці «Users» (як показано на рис. 3.13):

The screenshot shows a user management interface. On the left is a sidebar menu with items: Media, Categories, > Users, Redirects, Глобальні, Settings, Header, and Footer. The main area shows a form for editing a user. The 'Name' field contains 'Миськіє Павло Андрійович'. The 'Roles' dropdown menu is open, showing 'customer' as the current role and 'admin' as the selected role. Below the dropdown is a 'Вибрати значення' (Select value) button with a plus sign.

Рисунок 3.13. – Обрання користувача-модератора від власника роль «admin»

### 3.3.3. Розробка та опис коду створення замовлення товару

Оформлення замовлень – це етап, коли зареєстровані користувачі завершують свої покупки, вказуючи свої дані доставки, обираючи методи оплати та переглядаючи своє замовлення перед остаточним підтвердженням.

Над реалізацією створення оформлення замовлень буде включати відповідних форм «CheckoutForm», списків «CheckoutItem» товарів та основної сторінки оформлення замовлень «CheckoutPage».

Початкова сторінка оформлення замовлень по замовчуванню виглядає на рис. 3.14. Тут вказано, опис вибраних товарів та загальна вартість всіх вибраних товарів:

The screenshot shows a checkout page titled 'Checkout'. It includes a disclaimer: 'This is a self-hosted, secure checkout using Stripe's Payment Element component. To create a mock purchase, use a test credit card like 4242 4242 4242 with any future date and CVC. An order will be generated in Stripe and will appear in your account. In production, this checkout form will require a real card with sufficient funds.' Below the disclaimer is a list of items. The first item is 'acer-aspire-5' with a placeholder image. To the right of the image, the text reads: 'Acer Aspire 5 A515-57-58MF Grey' and '\$25,999.00'.

Рисунок 3.14 – Оформлення замовлень з визначеними товарами

Для змінення функціоналу та стилів основної сторінки оформлення замовлень в кодї, потрібно застосовувати React-функції компонентів, конструкції та стилів. З основних та ключових елементів з коду у [Додаток В] можна зазначити:



- ❖ Основна сторінка оформлення замовлень (CheckoutPage):
  - Умовні конструкції для відображення відповідного вмісту в залежності від стану кошика та процесу оплати – перевірка на порожність кошику, процес оплати та відображення контенту для кожного стану;
  - Відображення списку товарів у кошику – завдяки циклу «map», який проходить по кожному товару у кошику користувача і відображає контент для кожного товару;
  - Форма оплати – при наявності клієнтського секрету відображає форму введення платіжних даних користувача за допомогою React-компоненту «<CheckoutForm/>»;
  - Відображення повідомлень про помилки – у випадку виникнення помилок під час отримання клієнтського секрету;
  - Посилання на продовження покупок – у випадку порожнього кошика.
- ❖ Форма оформлення замовлення (CheckoutForm):
  - Форма для введення платіжних даних – визначає форму для введення платіжних даних;
  - Відображення повідомлення про помилку – наявність помилок, відображає повідомлення про помилку за допомогою компоненту «<Message />»;
  - Поле для введення платіжних даних користувача – номер кредитної картки, терміну дії та коду CVV за допомогою компоненту «<PaymentElement />»;
  - Кнопки дій – кнопки для повернення користувача кошика та для оформлення замовлення.

- ❖ Список товарів у оформлення замовлення (CheckoutItem):
  - Посилання на товар – посилання на сторінку деталей товару;
  - Відображення зображення товару – в ній є дві конструкції, один з них, що якщо зображення відсутнє – виводить текст «No image». А другий – якщо зображення існує – відображає зображення товару за допомогою компоненту «<Media />»;
  - Відображення деталей товару – в ній є відображення назви товару, ціну товару за допомогою компоненту «<Price />» та кількості одиниць товару у кошику;
  - Відображення загальної вартості для кожного товару – відображає суму для товару, враховуючи на кількість.

Передбачено у оформлення замовлень, платіжна система Stripe дозволяє оплачувати банківськими картками за певний товар. Ціни поставлені за визначені товари працює завдяки платіжній системі Stripe.

Результат оновленого інтерфейсу та функціонал логіки сторінки показано на рис. 3.14:

The screenshot shows the checkout page for 'Computer Store'. At the top, there are navigation links for 'Головна' (Home), 'Магазин' (Store), and user icons. Below is a table of items in the cart:

Товари	Кількість	Підсумок
 Acer Aspire 5 A515-57-58MF Grey 25 999,00 грн	x1	25 999,00 грн
 Lenovo IdeaPad 3 15ITL6 Sand 14 999,00 грн	x1	14 999,00 грн
<b>Всього замовлень</b>		<b>40 998,00 грн</b>

Below the cart is the 'Платіжні реквізити' (Payment details) section. It features a 'Secure, 1-click checkout with Link' indicator. The form includes fields for 'Card number' (1234 1234 1234 1234), 'Expiration' (MM / YY), and 'CVC'. The 'Country' is set to 'Ukraine'. At the bottom, there are two buttons: 'НАЗАД ДО КОШИКУ' (Back to cart) and 'ОФОРМИТИ' (Checkout).

Рисунок 3.14 – Оновлена сторінка оформлення замовлень разом з платіжною системою Stripe

### 3.4. Результат ручного тестування розробленого інтернет-магазину

У ході даного роботи, були залучені 5 осіб для ручного тестування сайту інтернет-магазину «Computer Store». Для ручного тестування, поділені на 2-ох етапах:

- 1) Запуск локального сервера проєктного сайту через інший пристрій користувача за допомогою терміналу (консоль);
- 2) Спеціальне опитування користувачів для результату ручного тестування мого сайту у формі Google Form.

Для запуску локального сервера проєктного мого сайту слід передбачити, чи встановлена програма Node JS. На командному рядку, слід прописати таке посилання папки проєкту за допомогою команди – `cd`, аби в результаті з'явилася таке посилання, наприклад: «`D:\Payload\Project-ecommerce\ecommerce`». Далі, прописати таку команду «`npm run dev`» (див. на рис. 3.7), який дозволяє запускати локальний сервер. Після запуску даного сервера, написати у браузері – `localhost:3000`.

За допомогою сервісу Google Form, дане опитування тестування мого сайту було поділено на 4 блоки з 2-ох запитань по 5-ти бальній шкалі:

❖ Функціональність:

- Чи працюють всі функції сайту правильно?;
- Чи відбувається додавання товарів до кошика, оформлення замовлень та обробка платежів без проблем?;

❖ Сумісність з браузерами:

- Чи відображається сайт коректно на різних веб-переглядачах?;
- Чи є які-небудь проблеми з відображенням сторінок або функціоналом в певних браузерах?;

❖ Адаптивність для мобільних пристроїв:

- Чи адаптується сайт до різних розмірів екранів на мобільних телефонах та планшетах?;
- Чи зберігається зручність використання на малих екранах?;



❖ Дизайн інтерфейсу:

- Чи є дизайн сайту зручним для користувачів?;
- Чи легко знаходити необхідну інформацію та взаємодіяти з функціями сайту?;

З результату опитувань пройдених користувачів ручного тестування сайту, можна зазначити (див. на рис. 3.15):

- 1) У блоці функціональність, 83,35% користувачів оцінили мій інтернет-магазин сайту на 5 балів. Це означає, що даний сайт функціонує добре, а самі функції працюють належним чином. А 16,65% користувачів, оцінили на 4 балів. Це означає, всі функції правильно працюють, але є незначні проблеми;
- 2) У блоці сумісність з браузерами, 83,35% користувачів оцінили на 5 балів. Це означає, що сайт оптимізований добре для браузерів і працює без проблем. А 16,65% користувачів – є деякі малі проблеми з певними браузерами, але в основному все працює;
- 3) У блоці Адаптивність для мобільних пристроїв, 83,35% користувачів оцінили сайту на 4 балів. Це означає, сайт добрий, але має певні мінорні проблеми. А 16,65% користувачів оцінили на 5 балів, це означає сайт пристосовує до всіх розмірів екранів;
- 4) У блоці Дизайн інтерфейсу, 100% користувачів оцінили інтерфейс сайту на 5 балів. Це означає, інтерфейс сайту подобається користувачам.

Таблиця 3.1 – Результат пройдених користувачів по блоках

Оцінка	Функціональність	Сумісність з браузерами	Адаптивність для мобільних пристроїв	Дизайн інтерфейсу
5	83,35%	83,35%	16,65%	83,35%
4	16,65%	16,65%	83,35%	16,65%
3	0,00%	0,00%	0,00%	0,00%
2	0,00%	0,00%	0,00%	0,00%

1	0,00%	0,00%	0,00%	0,00%
---	-------	-------	-------	-------

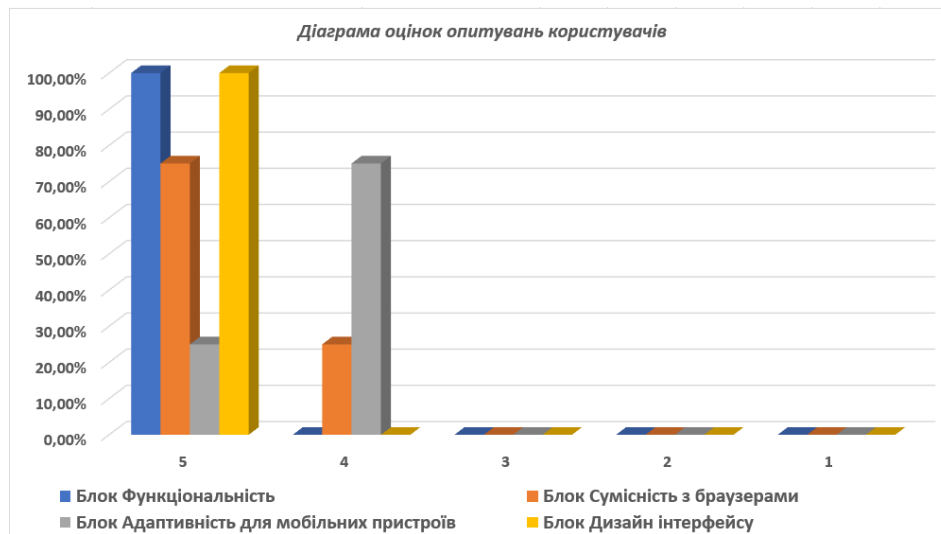


Рис. 3.15 – Діаграма результату оцінок опитувань пройдених користувачів

Недоліком мінорних проблем пройдених користувачів є:

- Присутні два кошики після авторизації;
- Під час мобільної адаптації, текст заходить на картинку;
- Можливість додати рейтинг та відгуків про даний товар.

Переваги даного сайту є:

- Зручність та зрозумілий інтернет-магазин;
- В основному даний інтернет-магазин, коректно та правильно працює;
- Працює на тестовому режимі платіжних банківської картки.

З порівнюючи з наведених аналогів інтернет-магазинів (CompX, ITbox, Brain) та мого розробленого інтернет-магазину Computer Store, мій даний сайт кращий, тому що:

- 1) Мінімалістичний дизайн, що легко сприймається користувачам;
- 2) Можливість швидкого фільтру, при наведенні категорій товарів та брендів;
- 3) Вказані ціни та наявності товарів на складі, яка дозволяє користувачам ефективно приймати рішення на покупку вибраного товару;

- 4) Зв'язок з підтримкою через соціальні мережі, електронної пошти, телефон;
- 5) Конкурентоспроможні ціни з регулярними знижками;
- 6) Спеціальні пропозиції для постійних користувачів.

Під час оплати перед покупкою, платіжна система Stripe краща, так як це популярна платіжна система у світі. Вона має високу репутацію у бізнесі, тобто через надійності, безпеки та легкості інтеграції з сайтами.

Отже, проєктовано функціональне моделювання інтернет-магазину, діаграму послідовності для замовлення товару. Розроблено проєкт інтернет-магазину на основі Payload CMS. Створено сторінки основного сайту, авторизації, персональної інформації користувача, оформлення замовлень та інші, крім того описані код з сторінок сайтів. Проведено ручне тестування для користувачів на зручність, над виявлення дефектів сайту та усунення помилок.

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

### 4.1. Поняття й завдання техніки безпеки

Небезпека – це явища, процеси, об'єкти, що за певних умов можуть завдавати шкоди здоров'ю чи життю людини як одразу, так і в майбутньому, тобто викликати небажані наслідки. Джерелами небезпеки є знаряддя праці (інструмент, спеціальні пристрої, машини), сам предмет праці або виробниче середовище. Оточуюче виробниче середовище, зокрема соціальне оточення, можуть стати джерелом психічної травми [14].

На виробництві, спеціально виділяються роботи з підвищеною небезпекою (із піднімальними кранами, балонами великого тиску, із електромережами високої напруги тощо). Більшість об'єктів підвищеної небезпеки зосереджена в атомній енергетиці, у нафтогазовому, хімічному та нафтохімічному комплексах [14].

У небезпечних зонах діють або періодично виникають фактори, небезпечні для життя та здоров'я людини. При цьому стан умов праці, за якого виключена дія на працівників небезпечних і шкідливих виробничих факторів, називається безпекою праці [14].

Об'єктами вивчення техніки безпеки є [14]:

- ❖ Технологічний і трудовий процеси;
- ❖ Особливості обладнання, інструментів і пристроїв із точки зору безпеки праці;
- ❖ Виробниче середовище в цілому, а також його складники (технічні, організаційні, соціальні), які можуть бути причиною виробничих травм чи сприяти їх виникненню й посиленню їхньою дією.

Виділяють наступні завдання техніки безпеки [14]:

- ❖ Виявлення причин травматизму, профзахворювань і потенційних небезпек;
- ❖ Визначення заходів і технічних засобів, що підтримують безпеку обладнання, а також технологічного і трудового процесу;

- ❖ Підготовку й обґрунтування матеріалів для законодавства з техніки безпеки, правил і норм технічних умов, інструкцій із дотримання безпеки експлуатації будівель, споруд, обладнання;
- ❖ Проведення повного обліку виробничих травм та аналіз причин їхнього виникнення;
- ❖ Вивчення й дослідження наявних технологічних процесів і впровадження нових, більш досконаліх, які підтримують безпеку, а також механізацію тяжких і шкідливих робіт;
- ❖ Розробку матеріалів та організацію роботи з інструктажу та навчання працівників безпечним прийомом праці.

До необхідності постійно підвищувати рівень безпеки праці керівників підприємств (власників), керівників виробничих підрозділів спонукають три причин [14]:

- ❖ Природний людський обов'язок стосовно працівників;
- ❖ Мотиви економічного характеру;
- ❖ Вимоги відповідних правових норм.

Крім того, нещасні випадки завдають чималих матеріальних збитків, негативно позначаються на добробуті окремої людини, знижують ефективність економіки країни. Від нещасного випадку потерпають люди й виробничий капітал не лише безпосередньо на місці аварії. Аварії та нещасні випадки тягнуть за собою безліч витрат, пов'язаних із відновленням ділянки виробництва, що вийшла з ладу [14].

Для підприємств, які усвідомили значення зусиль із дотримання правил техніки безпеки, спрямованих на гарантування безпечності праці, крім усього іншого, ці зусилля обертаються доволі відчутною економічною вигодою. Багато підприємств підрахували, що витрати на утримання штатних спеціалістів із безпеки праці, навіть фінансово вигідніше, адже все це сприяє помітному зменшенню аварій, нещасних випадків і пов'язаних із ними витрат [14].

## 4.2. Комп'ютерна техніка та її вплив на організм людини

Комп'ютерна техніка широко використовується в усіх галузях людської діяльності. Людина, яка працює з комп'ютером, постійно перебуває під впливом небезпечних і шкідливих виробничих факторів: електромагнітних полів (діапазон радіочастот: ЗЧ, УВЧ, СВЧ), інфрачервоного й іонізуючого випромінювань, шуму й вібрації, статистичної електрики тощо. Крім цього, оператор піддається значному розумову і психоемоційному навантаженню, високій напрузі зорової та м'язової (робота з клавіатурою ЕОМ) діяльності [15].

Аналіз скарг операторів комп'ютерного набору, редакторів і коректорів, які під час роботи використовують комп'ютер, наведені в табл. 4.1 [15].

Таблиця 4.1 – Характеристика скарг операторів комп'ютерного набору

Симптоми впливу комп'ютера	Кількість працівників, що повідомили про симптоми, від загальної кількості опитаних, %		
	Стаж роботи		
	До 1 року	1 – 3 роки	3 – 5 років
Біль і різь в очах	58,8	67,5	88,7
Біль у голові	17,6	23,3	42,5
Біль в області спини та шиї	18,5	21,8	32,2
Загальна втома	29,4	25,7	42,6
Утома пам'ять рук	15,1	22,3	38,7
Підвищена роздратованість	11,7	21,6	35,3
Порушення нічного сну	8,3	15,5	20,6
Порушення пам'яті	7,2	12,3	17,1

На функціональний стан користувача комп'ютера впливають: виробниче середовище, трудовий процес, внутрішні та зовнішні засоби діяльності, а також соціально-психологічні фактори. Тому для зменшення ризику захворювань необхідно провадити комплекс медико-гігієнічних, адміністративно-технічних та ергономічних заходів, яких передусім, мають входити [15]:

- Контроль за конструкцією, хорошим станом і функціонуванням комп'ютера;
- Відповідальність місця роботи рекомендацією ергономіки та гігієни;
- Створення оптимальних умов праці у виробничому приміщенні (мікроклімат, освітлення, захист від опромінення комп'ютера, іонізація повітря, вентиляція, кондиціонування повітря);
- Раціональний режим праці;
- Підвищувати опір організму користувачів комп'ютерів до дії несприятливих факторів;
- Особиста участь працівників у догляді за своїм здоров'ям.

На превеликий жаль, у багатьох випадках здійснення названих заходів фірмами й індивідуальними користувачами ігноруються, а тому важливо, аби й адміністрація (роботодавець) і користувачі комп'ютерів зрозуміли важливість і необхідність докладання зусиль кожною сторінкою для створення умов, що гарантують працівникам фізичний і духовний комфорт, високу розумову і творчу працездатність, збереження та зміцнення здоров'я [15].

#### **4.3. Шляхи оптимізації технічних, середовищних та ергономічних факторів**

Відеодистанційний термінал (ВДТ) – пристрій для візуальної подачі інформації, що зберігається електронним способом. Він складається з дисплейного екрана з виведеними на екран інформацією та клавіатурою [16].

Класифікація ВДТ стосовно впливу на здоров'я базується переважно на конструкторських особливостях та окремих параметрах самого дисплея. Залежно від призначення та сфери застосування ВДТ поділяють на групи [16]:

- Група А – кольорові монітори для демонстрації цілей у навчальному процесі, використання у гральних апаратах, тренажерах, пультах тощо.
- Група Б – кольорові монітори для персональної роботи користувачів у навчальному процесі та господарської діяльності, у якій не потрібна постійна обробка тексту, що містить 80 і більше символів у рядку.

- Група В – кольорові монітори для професійної роботи з текстовими документами й насиченим графічним зображенням у вищих і спеціальних навчальних закладах, ПТУ, спецшколах, на підприємствах, в офісах тощо.
- Група Г – монохромні монітори для забезпечення шкільних комплексів навчальної обчислювальної техніки (старші класи), професійної текстової обробки тощо.

Вимоги до освітлення приміщень, у яких встановленні комп'ютери, насамперед, яким має бути колір стін і підлоги [16].

Вікна зорієнтовані на південь: стіни зеленувато-голубого кольору; підлога – зелена [16].

Вікна зорієнтовані на північ: стіни світло-помаранчевого або помаранчево-жовтого кольору; підлога – червонувато-помаранчева.

Вікна зорієнтовані на захід: стіни жовтувато-зеленого або голубувато-зеленого кольору; підлога зеленого або червонувато-помаранчевого.

У приміщенні з комп'ютерами необхідно забезпечити наступні величини коефіцієнта відбиття:

- ❖ Для стелі – 0,7 – 0,8;
- ❖ Для стін – 0,5 – 0,6;
- ❖ Для підлоги – 0,3 – 0,5;
- ❖ Для інших поверхонь – 0,4 – 0,5.

Доцільно для освітлення використовувати лампи денного (білого) світла разом з лампами теплого білого світла (жовтого, рожевого), що разом імітує колірну гаму, яка відповідає спектральному складу природного світла в сонячний день.

За потреби робочий стіл облаштовують настільною лампою, весь світловий потік якої спрямований на робоче місце, а створене освітлення може бути потік мінімальним для виконуваної праці. Щоб уникнути потрапляння яскравих променів відбитого світла на екран або в очі користувача, рекомендують, аби кольори апаратури, меблів, одягу персоналу не були світлих тонів (білого, світло-



жовтого, світло рожевого). Відстань від екрану до шкіри обличчя й рук користувача має бути не меншою 60 см.

Особливо велику небезпеку для здоров'я людей становить підвищена концентрація озону, який вважається не лише подразнюючою, а й канцерогенною речовиною. Відповідно до ДСТУ 12.1 005-88 уміст озону в повітрі робочої зони не повинен перевищувати  $0,1 \text{ мг/м}^3$ ; уміст окисів азоту —  $5 \text{ мг/м}^3$ , уміст пилу —  $4 \text{ мг/м}^3$ .

Рівень шуму на робочому місці математиків-програмістів та операторів відеоматеріалів не має перевищувати 50 дБА, а в залах обробки інформації на обчислювальних машинах – 65 дБА.

Рівень вібрації у приміщеннях обчислювальних центрів можна знизити, установивши обладнання на спеціальні фундаменти та віброізолятори.

Максимальна напруженість електричної складової електромагнітного поля на конусі дисплея. Допустимі значення параметрів неіонізованих електромагнітних випромінювань від монітора комп'ютера наведені в табл. 4.2.

Таблиця 4.2 – Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

Назва параметра	Допустимі норми
Напруженість електромагнітного поля за електричним складником на відстані 60 см від поверхні відеомонітора	10 кВ/м
Напруженість електромагнітного поля за електромагнітним складником на відстані 50 см від поверхні відеомонітора	0,3 А/м
Напруженість електростатичного поля не має перевищувати для дорослих користувачів. Для дітей дошкільних установ учнів і студентів	20 кВ/м 15 кВ/м

ВДТ – джерело електростатичних зарядів. Тривале перебування в електричному полі, створеному цими зарядами, негативно впливає на здоров'я

працюючих: бронхо-легеневі захворювання, порушення серцево-судинної та нервової систем, ураження шкіри тощо.

Відповідно до ДНАОП 0.00-1.31-99 поверхневий електростатистичний потенціал відеотерміналу не має перевищувати 500 В.

Щоб уникнути значного напруження поля та захистити від статичної електрики рекомендують:

1. Встановити нейтралізатори статичної електрики;
2. Підтримувати у приміщенні з ВДТ відносну вологість повітря не нижче 45 – 50% (чим сухіше повітря, тим більше електростатичних зарядів); можна для цього використовувати навіть побутове зволоження;
3. Застелити підлогу у приміщенні з ВДТ антистатичних лінолеумом і проводити щоденне вологе прибирання;
4. Протирати екран і робоче місце антистатичною серветкою або зволоженою тканиною;
5. Користувачам бажано носити одяг, особливо першого шару, із натуральних матеріалів;
6. Для зняття статичного заряду бажано кілька разів на день мити руки й обличчя, або від часу торкатися, наприклад, батареї центрального опалення.

Перед початком роботи за комп'ютером необхідно пройти обстеження в лікаря-окуліста і, якщо є потреба, скерувати рефракцію (окуляри, контактні лінзи).

Медичне обстеження зору необхідно періодично повторювати, що не допустити розвиток зниження зору.

Під час роботи слід дотримуватися нижченаведених правил:

1. Коліна мають бути на достатній відстані одне від одного;
2. Необхідно, щоб ступні добре спиралися на підлогу чи підніжку;
3. Сидіння не повинні стикати стегон;
4. Під час сидіння слід випростатися, витягнутися вгору, перевірити позу;

5. Голову слід тримати прямо й нахилено трохи вниз; верхня лінія екрана має бути трохи нижче рівня очей; екран не повинен блищати;
6. Ліктьовий суглоб має бути на тій же висоті, що і клавіатура: лікті мають щільно прилягати до тулубу (або підтримуватися підлокітниками крісла) при цьому зап'ястя випрямлене;
7. Рекомендується під час роботи згадувати про позу й корегувати її: при відчутті м'язового напруження роблять вправи 5 – 10 разів;
8. Кожні 50 хвилин роботи (доцільно використовувати будильник) улаштовувати 10-хвилинну перерву, під час якої також відповідними вправами знімають хоча б частину м'язової втоми (тіла й очей), покращуючи самопочуття за допомогою спеціального самомасажу та фізичних вправ;
9. За дві години напруженої безперервної роботи відпочинок слід продовжити до 15 хв.; після 4-годинної роботи – до 1 год.; роботу на комп'ютері бажано чергувати з іншими видами діяльності;
10. У після робочий час необхідно, крім загальних фізичних вправ, проводити специфічні вправи для очей, ефективність яких незаперечна.

Велике значення для збереження здоров'я користувачів комп'ютерів має дотримання медичних рекомендацій щодо раціонального харчування й додаткової вітамінізації організму. Із метою розширення функціональних можливостей м'язів ока необхідно застосовувати спеціальні вправи для очей, самомасаж «життєвих» точок, а також ті загальні фізичні вправи і спортивні ігри, які особливо рекомендовані для підвищення функціональних можливостей очей [16].

В цьому розділі, досліджено поняття та завдання техніки безпеки. Проаналізовано симптоми та характеристику скарг операторів комп'ютерного набору. Наведено класифікацію ВДТ (Відеодистанційний термінал), що поділяють на особливостей та параметрами екрана. Зазначено аспекти забезпечення умови для роботи над комп'ютером. Робота з відеодистанційним терміналом пов'язана з ризиком виникнення електростатичних зарядів, що негативно впливають на здоров'я працюючих.

## ВИСНОВКИ

Метою кваліфікаційної роботи є дослідження різних альтернативних розробок інтернет-магазину на даному часі, створення функціоналу роботи та вибір ефективного розробки програмних технологій.

Проведено детальний огляд аналогів інтернет-магазину, розглянуто три альтернативних інтернет-магазинів. Описано функціональність сайту, переваги й недоліки інтернет-магазину та етапи створення інтернет-магазину.

Порівняв над створенням різних технологій для розробки даного сайту та основну характеристику, вибравши одну з перелічених, порівняних технологій. Проаналізовано та детально, описано про дану технологію Payload CMS (Корисне навантаження). Досліджено методи способів оплати інтернет-магазину.

Розроблено детальну функціональне моделювання, включає базу даних, кроки процесу оформлення замовлень та можливості. Проєктовано моделі бази даних на основі MongoDB. Побудовано діаграму послідовностей UML-діаграми для оформлення замовлень.

На основі вимог та специфікацій реалізовано функції сайту включаючи реєстрацію та аутентифікацію користувачів, каталог товарів, кошик покупок, оформлення замовлення тощо.

Зроблено розробку інтерфейсу користувача, який буде мати зручну й привабливу взаємодію із сайтом для користувачів.

Проведено ручне тестування сайту для виявлення помилок та суперечностей, долучаючи 5-ох користувачів. Після тестування внесені корективи та вдосконалення для підвищення якості та надійності системи. Метою кваліфікаційної роботи є дослідження різних альтернативних розробок інтернет-магазину на даному часі, створення функціоналу роботи та вибір ефективного розробки програмних технологій.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hanafi R., Haq A., Agustin N. Comparison of Web Page Rendering Methods Based on Next.js Framework Using Page Loading Time Test. *Teknika*, 13(1): 2024. 102-108 с.
2. Holmberg O. Migrating from JavaScript to TypeScript and its advantages. 2023.
3. Kumke, H., Brunnengräber, M. Cartography Playground: Interactive, web-based applications to explain basic cartographic issues : *Abstracts of the ICA*, 6: 2023. 129 с.
4. Larman C. Use-case model: Drawing system sequence diagrams : *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition)* : 2021. 118 с.
5. Payload Concepts | Documentation | Payload CMS [Електронний ресурс]. Режим доступу: <https://payloadcms.com/docs/getting-started/what-is-payload> (дата звернення: 21.04.2024).
6. PayPal: інструкція для користувачів [Електронний ресурс]. Режим доступу: <https://dniproillc.com/uk/paypal-tse/> (дата звернення: 06.06.2024).
7. Presley A, Liles D.H. The use of IDEF0 for the design and specification of methodologies : *Proceedings of the 4th industrial engineering research conference*. 1995, May. 442-448 с.
8. What is Headless CMS? – Headless CMS Architecture Explained [Електронний ресурс]. Режим доступу: [https://aws.amazon.com/what-is/headless-cms/?nc1=h\\_ls](https://aws.amazon.com/what-is/headless-cms/?nc1=h_ls) (дата звернення: 21.04.2024).
9. What is Payload? | Documentation | Payload CMS [Електронний ресурс]. Режим доступу: <https://payloadcms.com/docs/getting-started/what-is-payload> (дата звернення: 21.04.2024).
10. Афонін А., Зважій Д. Особливості використання документо-орієнтованих баз даних на прикладі MongoDB. 2015.

11. Василюк А. Інтелектуальна система «інтернет-магазин музичного обладнання» : *вісник Національного університету «Львівська політехніка» Комп'ютерні науки та інформаційні технології №751*: 2013. 373-381 с.
12. Етапи створення індивідуального інтернет-магазину [Електронний ресурс]. Режим доступу: <https://wezom.com.ua/ua/blog/etapy-razrabotki-internet-magazina> (дата звернення: 13.05.2024).
13. Онлайн-купівлі: головні переваги та недоліки [Електронний ресурс]. Режим доступу: <https://fractus.com.ua/uk/blog/korysni-statti/prodazhi/onlajn-kupivli-golovni-perevagi-ta-nedoliki/> (дата звернення: 21.04.2024).
14. Пашков В. І., Жовтяк Ж. Г., Бодня З.К. Поняття й завдання техніки безпеки : *конспект лекцій з дисципліни "Основи охорони праці"* : Харків: ХНАМГ, 2012. 62 с.
15. Пашков В.І., Жовтяк Ж. Г., Бодня З.К. Комп'ютерна техніка та її вплив на організм людини : *конспект лекцій з дисципліни "Основи охорони праці"* : Харків: ХНАМГ, 2012. 52 с.
16. Пашков В.І., Жовтяк Ж. Г., Бодня З.К. Шляхи оптимізації технічних, середовищних та ергономічних факторів : *конспект лекцій з дисципліни "Основи охорони праці"* : Харків: ХНАМГ, 2012. 53 с.
17. Плагін Stripe [Електронний ресурс]. Режим доступу: <https://payloadcms.com/docs/plugins/stripe#stripe-plugin> (дата звернення 21.04.2024).
18. Розробка інтернет магазину – основні характеристики та функціональність [Електронний ресурс]. Режим доступу: <https://web24.pro/rozrobka-sajtiv-blog/rozrobka-internet-magazyn-osnovni-harakterystyky-ta-funkczionalnist/> (дата звернення 05.06.2024).
19. Технічні можливості Stripe [Електронний ресурс]. Режим доступу: <https://newline.tech/texnichni-mozhlyvosti-stripe-intervyu/> (дата звернення: 21.04.2024).
20. Що таке Paysera? [Електронний ресурс]. Режим доступу: <https://www.paysera.com/v2/uk/pro-nas> (дата звернення: 21.05.2024).

/

## **ДОДАТКИ**

**Додаток А. Код компонентів функціоналу директорії «account» підкатегорії «AccountForm», «orders», «purchases», «UserInfo» з файлами index.tsx та page.tsx**

```
// src/app/(pages)/account/AccountForm
return (
  <form onSubmit={handleSubmit(onSubmit)} className={classes.form}>
    <Message error={error} success={success} className={classes.message} />
    {!changePassword ? (
      <Fragment>
        <p>
          {'Змініть дані свого облікового запису нижче або '}
          <button
            type="button"
            className={classes.changePassword}
            onClick={() => setChangePassword(!changePassword)}
          >
            Натиснути тут
          </button>
          {' щоб змінити пароль.'}
        </p>
        <Input
          name="email"
          label="Електронна пошта (e-mail)"
          required
          register={register}
          error={errors.email}
          type="email"
        />
        <Input name="name" label="Ім'я" register={register} error={errors.name}
      />
    </Fragment>
  ) : (
    <Fragment>
      <p>
        {'Змініть дані свого облікового запису нижче або '}
        <button
          type="button"
          className={classes.changePassword}
          onClick={() => setChangePassword(!changePassword)}
        >
          cancel
        </button>
      </p>
      <Input
        name="password"
        type="password"
        label="Пароль"
        required
        register={register}
        error={errors.password}
      />
      <Input
        name="passwordConfirm"
        type="password"
        label="Підтвердіть пароль"
        required
        register={register}
        validate={value => value === password.current || 'Паролі не
```



```

збігаються'}
      error={errors.passwordConfirm}
    />
  </Fragment>
)}
<Button
  type="submit"
  label={isLoading ? 'Обробка' : changePassword ? 'Змінити пароль' :
'Оновити обліковий запис'}
  disabled={isLoading}
  appearance="primary"
  className={classes.submit}
  />
</form>
)

// src/app/(pages)/account/UserInfo
return (
  <div className={classes.profile}>
    <Image src="/assets/icons/profile.svg" alt="profile" width={60} height={60} />

    <div className={classes.profileInfo}>
      <p className={classes.name}>{user?.name}</p>
      <p className={classes.email}>{user?.email}</p>
    </div>
  </div>
)

// src/app/(pages)/account/orders
return (
  <div>
    <h5>Мої замовлення</h5>
    {( !orders || !Array.isArray(orders) || orders?.length === 0 ) && (
      <p className={classes.noOrders}>У вас немає замовлень.</p>
    )}
    <RenderParams />
    {orders && orders.length > 0 && (
      <ul className={classes.orders}>
        {orders?.map(order => (
          <li key={order.id} className={classes.order}>
            <Link className={classes.item} href={`\account/orders/${order.id}`}>
              <div className={classes.itemContent}>
                <h6 className={classes.itemTitle}>{`Замовлення ${order.id}`}</h6>
                <div className={classes.itemMeta}>
                  <p>
                    {`Всього: `}
                    {new Intl.NumberFormat('ua-UA', {
                      style: 'currency',
                      currency: 'uah',
                    }).format(order.total / 100)}
                  </p>
                  <p className={classes.orderDate}>{`Замовлено на:
${formatDateTime(
                    order.createdAt,
                  )}`}</p>
                </div>
              </div>
            </li>
          </div>
        )}
      </ul>
      <Button
        appearance="default"
        label="Переглянути замовлення"
        className={classes.button}
        el="link"
        href={`\account/orders/${order.id}`}
      />
    </Link>
  )}
  </div>
)

```

```
        </li>  
    )})  
</ul>  
    )}  
</div>  
)
```

## Додаток Б. Код компонентів функціоналу директорії «login» з підкатегоріями «LoginForm» з файлами page.tsx та index.tsx

```

// src/app/(pages)/login/LoginForm
return (
  <form onSubmit={handleSubmit(onSubmit)} className={classes.form}>
    <Message error={error} className={classes.message} />
    <Input
      name="email"
      label="Електронна пошта (e-mail)"
      required
      register={register}
      error={errors.email}
      type="email"
    />
    <Input
      name="password"
      type="password"
      label="Пароль"
      required
      register={register}
      error={errors.password}
    />
    <Button
      type="submit"
      appearance="primary"
      label={isLoading ? 'Обробка' : 'Увійти'}
      disabled={isLoading}
      className={classes.submit}
    />
    <div className={classes.links}>
      <Link href={` /create-account${allParams}`}>Створити обліковий запис</Link>
      <br />
      <Link href={` /recover-password${allParams}`}>Відновити пароль</Link>
    </div>
  </form>
)
// src/app/(pages)/login
return (
  <section className={classes.login}>
    <div className={classes.heroImg}>
      <Link href="/">
        <Image src="/Logo-title-black.svg" alt="logo" width={250} height={23}
className={classes.logo}/>
      </Link>
    </div>

    <div className={classes.formWrapper}>
      <div className={classes.formContainer}>
        <RenderParams className={classes.params} />

        <div className={classes.formTitle}>
          <h3>Ласкаво просимо</h3>
          <Image src="/assets/icons/hand.png" alt="hand" width={30} height={30} />
        </div>
        <LoginForm />
      </div>
    </div>
  </section>
)

```

**Дотаток В. Код компонентів функціоналу директорії «checkout» з підкатегоріями «CheckoutForm», «CheckoutItem», «CheckoutPage» з файлами index.tsx**

```

// src/app/(pages)/checkout/CheckoutForm
return (
  <form onSubmit={handleSubmit} className={classes.form}>
    {error && <Message error={error} />}
    <PaymentElement />
    <div className={classes.actions}>
      <Button label="Назад до кошику" href="/cart" appearance="secondary" />
      <Button
        label={isLoading ? 'Обробка...' : 'Оформити'}
        type="submit"
        appearance="primary"
        disabled={!stripe || isLoading}
      />
    </div>
  </form>
)

// src/app/(pages)/checkout/CheckoutItem
return (
  <li className={classes.item} key={index}>
    <Link href={`\products/${product.slug}`} className={classes.mediaWrapper}>
      {!metaImage && <span>No image</span>}
      {metaImage && typeof metaImage !== 'string' && (
        <Media className={classes.media} imgClassName={classes.image}
resource={metaImage} fill />
      )}
    </Link>

    <div className={classes.itemDetails}>
      <div className={classes.titleWrapper}>
        <h6>{title}</h6>
        <Price product={product} button={false} />
      </div>
      <p className={classes.quantity}>x{quantity}</p>
    </div>

    <div className={classes.subtotal}>
      <Price product={product} button={false} quantity={quantity} />
    </div>
  </li>
)

// src/app/(pages)/checkout/CheckoutPage
return (
  <Fragment>
    {cartIsEmpty && (
      <div>
        {'Your '}
        <Link href="/cart">cart</Link>
        {' is empty.'}
        {typeof productsPage === 'object' && productsPage?.slug && (
          <Fragment>
            {' '}
            <Link href={`\${productsPage.slug}`}>Continue shopping?</Link>
          </Fragment>
        )}
      </div>
    )}
  </Fragment>
)

```

```

</div>
))
{!cartIsEmpty && (
  <div className={classes.items}>
    <div className={classes.header}>
      <p>Товари</p>
      <div className={classes.headerItemDetails}>
        <p></p>
        <p className={classes.quantity}>Кількість</p>
      </div>
      <p className={classes.subtotal}>Підсумок</p>
    </div>

    <ul>
      {cart?.items?.map((item, index) => {
        if (typeof item.product === 'object') {
          const {
            quantity,
            product,
            product: { title, meta },
          } = item

          if (!quantity) return null

          const metaImage = meta?.image

          return (
            <Fragment key={index}>
              <CheckoutItem
                product={product}
                title={title}
                metaImage={metaImage}
                quantity={quantity}
                index={index}
              />
            </Fragment>
          )
        }
        return null
      })}
      <div className={classes.orderTotal}>
        <p>Всього замовлень</p>
        <p>{cartTotal.formatted}</p>
      </div>
    </ul>
  </div>
))
{!clientSecret && !error && (
  <div className={classes.loading}>
    <LoadingShimmer number={2} />
  </div>
)}
{!clientSecret && error && (
  <div className={classes.error}>
    <p>{`Error: ${error}`}</p>
    <Button label="Назад до кошику" href="/cart" appearance="secondary" />
  </div>
)}
{clientSecret && (
  <Fragment>
    <h3 className={classes.payment}>Платіжні реквізити</h3>
    {error && <p>{`Error: ${error}`}</p>}
    <Elements
      stripe={stripe}
      options={{

```

```

    clientSecret,
    appearance: {
      theme: 'stripe',
      variables: {

        colorText:
          theme === 'dark' ? cssVariables.colors.base0 :
cssVariables.colors.base1000,
        fontSizeBase: '15px',
        fontWeightNormal: '500',
        fontWeightBold: '600',
        colorBackground:
          theme === 'dark' ? cssVariables.colors.base850 :
cssVariables.colors.base0,
        fontFamily: 'Inter, sans-serif',
        colorTextPlaceholder: cssVariables.colors.base500,
        colorIcon:
          theme === 'dark' ? cssVariables.colors.base0 :
cssVariables.colors.base1000,
        borderRadius: '20px',
        colorDanger: cssVariables.colors.error500,
        colorDangerText: cssVariables.colors.error500,
      },
    },
  }}
  >
  <CheckoutForm />
</Elements>
</Fragment>
)}
</Fragment>
)

```

## Додаток Г. Результати опитувань користувачів пройденого вручного тестування даного інтернет-магазину по блоках

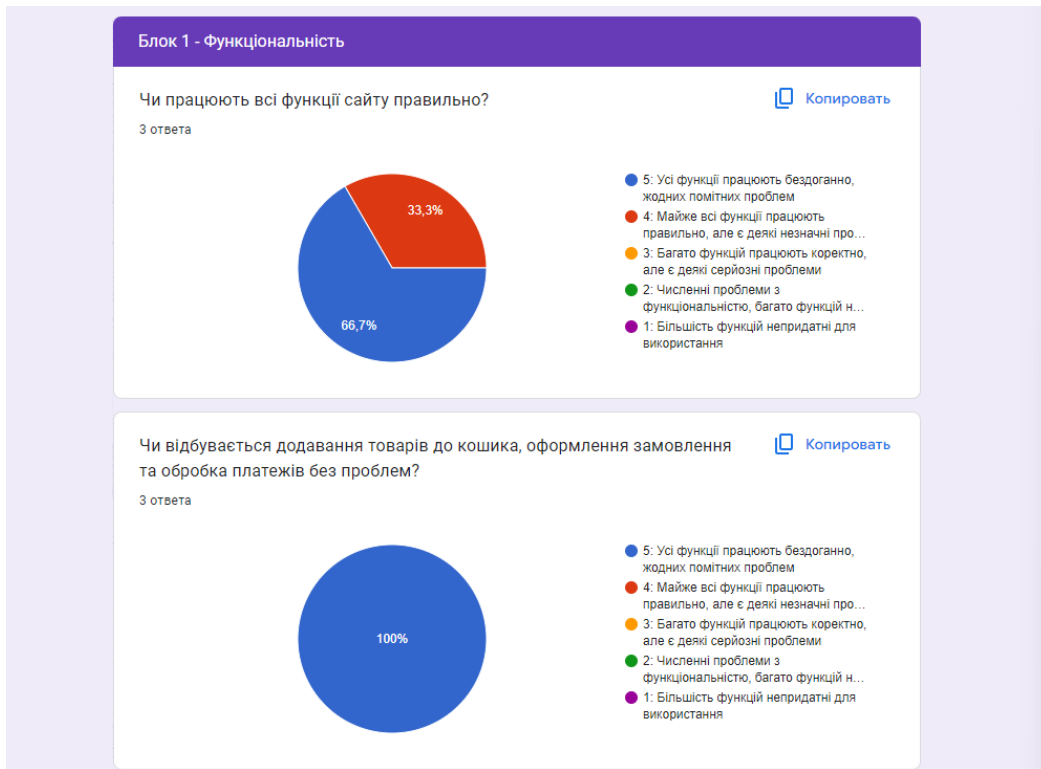


Рисунок Г.1 – Результат оцінювань блоку функціональності

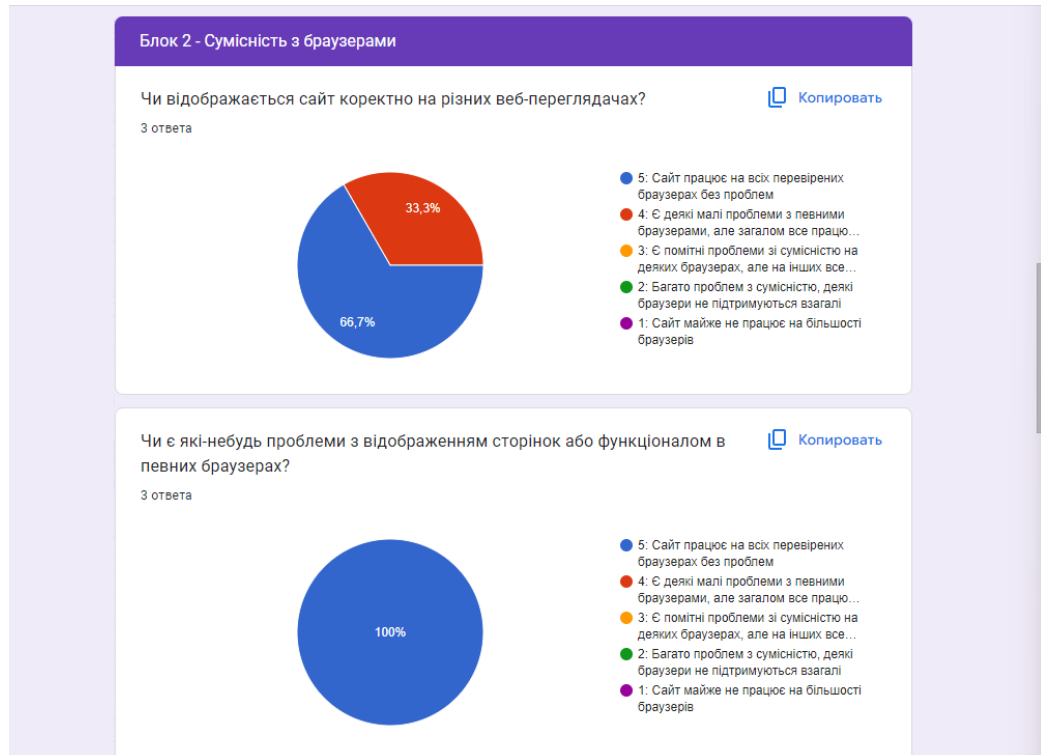


Рисунок Г.2 – Результат оцінювань блоку сумісності з браузерами

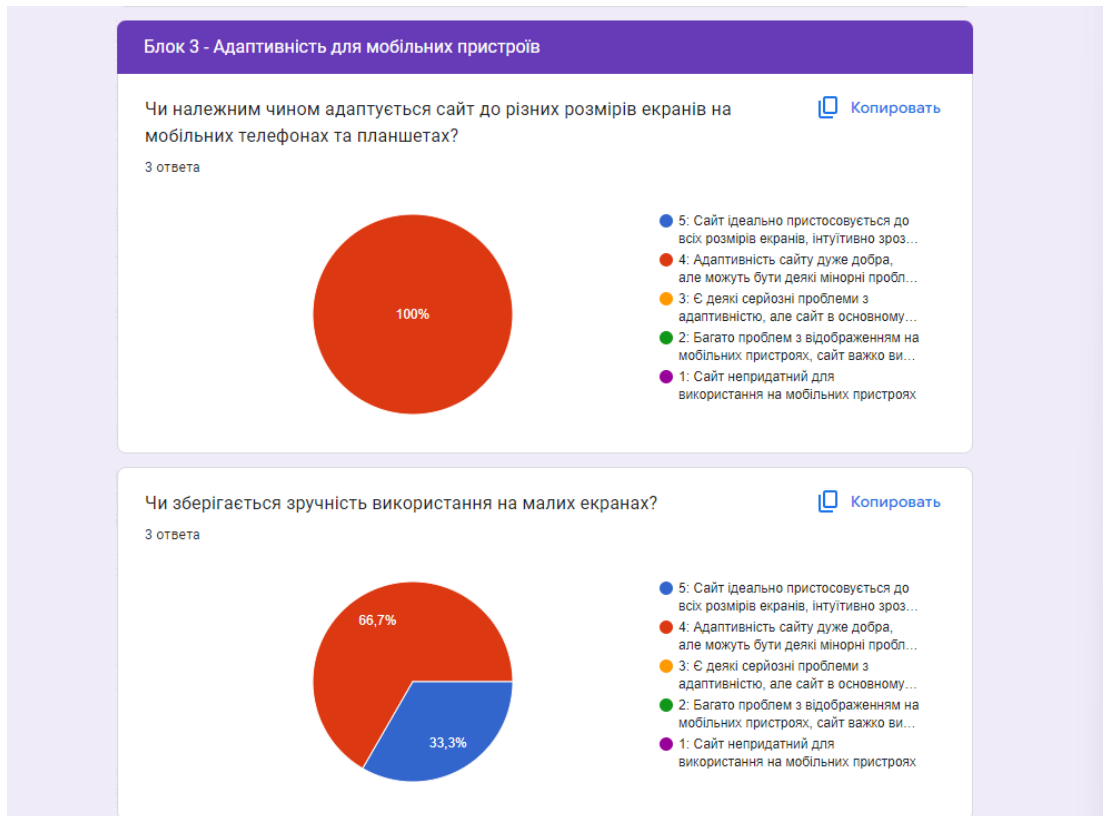


Рисунок Г.3 – Результат оцінювань блоку адаптивності для мобільних пристроїв

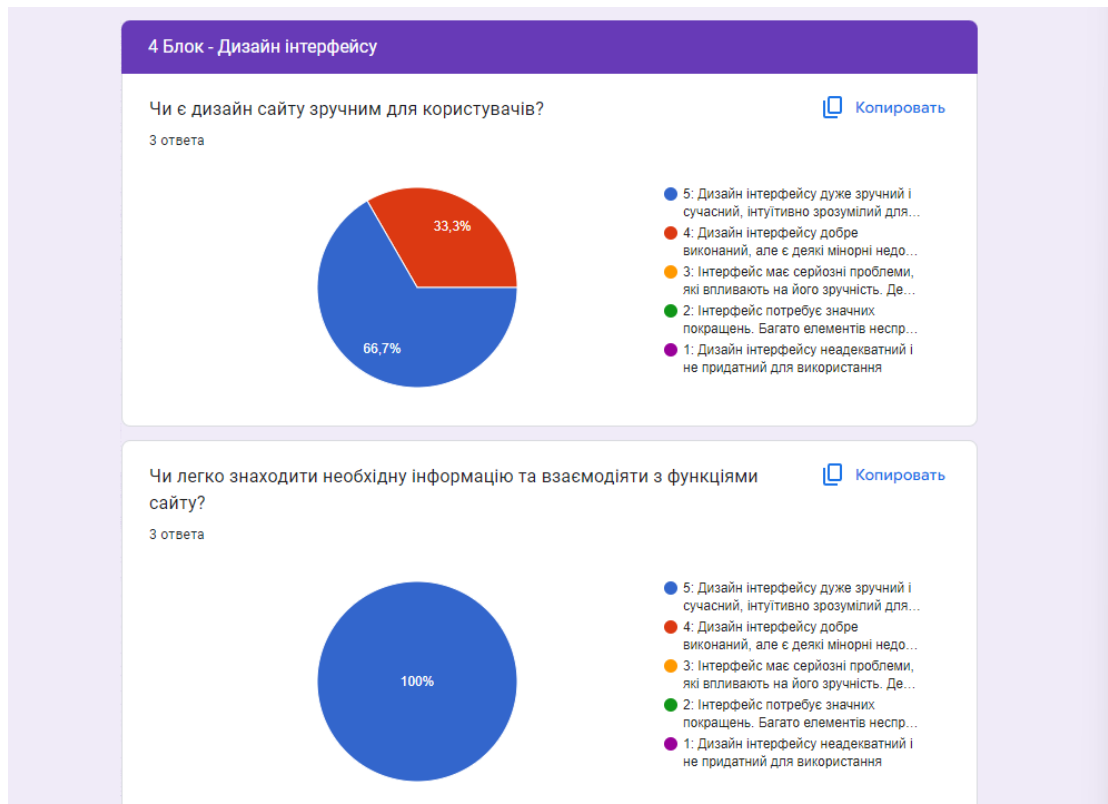


Рисунок Г.4 – Результат оцінювань блоку дизайн інтерфейсу