

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО

«__»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”

Тема: «Інтернет-магазин з продажу електротоварів на основі технології React»

Виконавець: Кузнецов Андрій Андрійович

Керівник: к.т.н., доцент Сергій ВОДОП'ЯНОВ

Нормоконтролер: к.т.н., доцент Вікторія СИДОРЕНКО

КИЇВ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютерних інформаційних технологій

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:

завідувач кафедри КІТ

Аліна САВЧЕНКО

(підпис)

« _____ » _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Кузнєцова Андрія Андрійовича

(ПІБ випускника)

1. Тема роботи: «Інтернет-магазин з продажу електротоварів на основі технології React» затверджена наказом ректора від «05» квітня 2024р. № 517/ст.
2. Термін виконання роботи: з 6 травня 2024 року по 9 червня 2024 року.
3. Вихідні дані до роботи: web-сервіс з продажу електротоварів, в основі якого буде використано технологію React.
4. Зміст пояснювальної записки: 1. Огляд інформаційних технологій для підтримки комерційних процесів. 2. Вимоги до проєктованого інтернет-магазину з продажу електротоварів. 3. Розробка та дизайн web-сервісу інтернет-магазину з продажу електротоварів
5. Перелік обов'язкового ілюстративного матеріалу: 1. UML-діаграма прецедентів. 2. Діаграма IDEF0. 3. Файлова структура фронтенд- та бекенд-проєкту. 4. Структурна схема БД.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Огляд інформаційних технологій для підтримки комерційних процесів. Написання 1 розділу, представлення керівнику.	06.05.2024- 10.05.2024	
2.	Вимоги до проєктованого інтернет-магазину з продажу електротоварів. Написання 2 розділу, представлення керівнику.	11.05.2024- 17.05.2024	
3.	Розробка та дизайн web-сервісу інтернет-магазину з продажу електротоварів. Написання 3 розділу, представлення керівнику.	17.05.2024- 27.05.2024	
4.	Загальне редагування та друк пояснювальної записки.	28.05.2024- 30.05.2024	
5.	Проходження нормоконтролю, перепліт пояснювальної записки.	31.05.2024- 04.06.2024	
6.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	05.06.2024- 10.06.2024	

7. Дата видачі завдання «06» травня 2024р.

Керівник кваліфікаційної роботи

(підпис керівника)

Сергій ВОДОП'ЯНОВ

Завдання прийняв до виконання

(підпис випускника)

Андрій КУЗНЄЦОВ

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Інтернет-магазин з продажу електротоварів на основі технології React» містить: 59 сторінок, 18 рисунків, 32 інформаційних джерел, 7 додатків.

Об'єктом дослідження є програмна реалізація систем доступу до даних та інформаційних систем.

Предметом дослідження є проєкт інформаційної системи комерційного ресурсу на основі технології React.

Метою кваліфікаційної роботи – є створення web-ресурсу для продажу електротоварів з використанням бібліотеки React.js.

В роботі використано різноманітні **методи дослідження**, зокрема: аналіз наявної методичної та наукової літератури, використання індукції та дедукції, порівняння різних підходів, програмне проєктування та встановлення причинно-наслідкових зв'язків.

Наукова новизна роботи полягає у використанні оригінального проєктного матеріалу за напрямком проведеного дослідження, включаючи вперше використаний комбінований метод проєктування систем доступу до інформації на основі технології React.

Результати кваліфікаційної роботи можуть бути використані для застосування на підприємстві в умовах виробничої діяльності.

Обрано: мову розмітки гіпертексту HTML; Scss – скриптову метамову, яка інтерпретується в каскадні таблиці стилів (CSS); React.js – фреймворк для JavaScript – бібліотека допомагає створювати web-програми – сайти, які завантажуються лише один раз; для розробки проєктованого ресурсу web-сервісу з продажу електротоварів нами обрано MySQL – це вільна реляційна система яка керує базами даних. Вона є найбільш широко використовуваною базою даних з відкритим вихідним кодом.

Ключові слова: ВЕБСАЙТ, ІНФОРМАЦІЙНА СИСТЕМА, ОБРОБКА ДАНИХ, ПРОДАЖ, ТОВАРИ, JS, HTML, SCSS, REACT.JS, CSS, MYSQL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП.....	8
РОЗДІЛ 1. ОГЛЯД ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ПІДТРИМКИ КОМЕРЦІЙНИХ ПРОЦЕСІВ	11
1.1. Аналіз розробки проєкту web-сервісу з продажу електротоварів та його можливостей	11
1.2. Вибір сучасних інструментів та аналіз методологій для розробки комерційного web-сервісу та бази даних	15
ВИСНОВКИ ДО РОЗДІЛУ 1	21
РОЗДІЛ 2. ВИМОГИ ДО ПРОЄКТОВАНОГО ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ЕЛЕКТРОТОВАРІВ.....	22
2.1. Аналіз цільової аудиторії програмного продукту	22
2.2. Вибір методів моделювання та проєктування процесу розробки web-сервісу	23
ВИСНОВКИ ДО РОЗДІЛУ 2	26
РОЗДІЛ 3. РОЗРОБКА ТА ДИЗАЙН WEB-СЕРВІСУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ЕЛЕКТРОТОВАРІВ	27
3.1. Сценарії взаємодії.....	27
3.2. Архітектура інформаційної системи	28
3.3. Розробка програмних модулів	30
3.4. Розробка дизайну.....	35
3.5. Розгортання програмного забезпечення проєктованої системи	39
3.6. Тестування інформаційного продукту інтернет-магазину.....	41
ВИСНОВКИ ДО РОЗДІЛУ 3	46
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТКИ	53
Додаток А	53

Додаток Б.....	54
Додаток В.....	55
Додаток Д.....	56
Додаток Е.....	58
Додаток Ж.....	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

SMART	–	Specific, Measurable, Achievable, Relevant, Time bound.
CSS	–	Cascading Style Sheets - це мова стилів, яка використовується для оформлення веб-сторінок.
IS	–	Інформаційна система.
HTML	–	Hypertext Markup Language - це стандартна мова розмітки для створення та представлення веб-сторінок.
SPA	–	Single Page Application.
DOM	–	Document Object Model.
Scss	–	скриптова метамова, яка інтерпретується в каскадні таблиці стилів (CSS).
CMS	–	Content Management System, система керування вмістом — це програмне забезпечення, яке дозволяє користувачам створювати, редагувати, керувати та публікувати контент при веб-розробці.
UML	–	Unified Modeling Language.

ВСТУП

З початку 21-го століття кожна галузь, що має амбіції розвивати свій бізнес або діяльність, повинна мати свій власний web-сайт або профіль в соціальних мережах. Молодь, яка є найпрогресивнішою частиною суспільства та найбільш активними учасниками Інтернету, використовує всі його ресурси з більшою енергією: для купівлі товарів і послуг, навчання, роботи на відстані та спілкування з однолітками.

Актуальність теми дослідження виокремлюється у світлі постійно зростаючих загроз кібербезпеці у цифровому віці. Спільність інтернет-користувачів стикається зі зростаючим рівнем кіберзлочинності, що ставить під загрозу конфіденційність, цілісність та доступність їхніх даних та онлайн-ресурсів. У цьому контексті важливим стає розробка ефективних заходів захисту для web-ресурсів, що є необхідним кроком для створення безпеки користувачів та надійності бізнесу в онлайн-середовищі.

Популярність та розповсюдженість інтернет-технологій призводять до зростання попиту на web-ресурси у всіх сферах життя: від електронної комерції до онлайн-економічної діяльності. Убезпечення цих ресурсів стає пріоритетом для їхніх власників та адміністраторів. До того ж, з появою нових загроз та атак кіберзлочинців, захист web-ресурсів вимагає постійного удосконалення та оновлення.

У вигляді розробленого web-ресурсу, що має високий рівень захисту, стає можливим ефективно забезпечити конфіденційність та цілісність даних користувачів, а також уникнути потенційних фінансових втрат та репутаційної шкоди.

Такий проєкт роботи може виявитися кроком у напрямку забезпечення довіри користувачів до web-платформи та сприяти її подальшому розвитку та успіху в конкурентному середовищі. Враховуючи постійне зростання кіберзагроз та важливість захисту web-ресурсів для користувачів та бізнесу, даний напрямок дослідження має великий потенціал для вивчення та розвитку.

Метою кваліфікаційної роботи – є створення web-ресурсу для продажу електротоварів з використанням бібліотеки React.js.

Задля досягнення поставленої мети необхідно виконати наступні задачі:

- Визначити актуальність створення web-сервісу для продажу електротоварів.
- Виконати вибір сучасних інструментів та аналіз методологій для розробки web-сервісів та бази даних для них.
- Провести дослідження та аналіз цільової аудиторії продукту.
- Проаналізувати існуючі аналоги на ринку.
- Виконати моделювання та проєктування процесу розробки web-сервісу.
- Розробити базу даних ресурсу.
- Провести реалізацію web-сервісу та його дизайну.

Об'єктом дослідження є програмна реалізація систем доступу до даних та інформаційних систем.

Предметом дослідження є проєкт інформаційної системи комерційного ресурсу на основі технології React.

В роботі використано різноманітні **методи дослідження**, зокрема: аналіз наявної методичної та наукової літератури, використання індукції та дедукції, порівняння різних підходів, програмне проєктування та встановлення причинно-наслідкових зв'язків. Джерельна база дослідження охоплює аналіз науково-методичної літератури, методичних посібників, наукових статей, періодичних видань та попередніх розробок у галузі систем доступу до інформації.

Наукова новизна роботи полягає у використанні оригінального проєктного матеріалу за напрямком проведеного дослідження, включаючи вперше використаний комбінований метод проєктування систем доступу до інформації на основі технології React.

Теоретична та практична цінність роботи виявляється у наявності теоретичного матеріалу, який був відібраний з-поміж іншого в процесі пошуку інформації з теми, а також у систематизації матеріалу з обраного напрямку дослідження. Проведене дослідження відрізняється більш глибоким аналізом напрямку дослідження, опираючись на попередні дослідження учених, аспірантів та дослідників у відповідній галузі.

Структура роботи. Робота складається з 59 листів друкованого тексту, 18 рисунків, 5 таблиць та 7 додатків і налічує 32 джерел використаної літератури.

РОЗДІЛ 1
ОГЛЯД ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ПІДТРИМКИ
КОМЕРЦІЙНИХ ПРОЦЕСІВ

1.1. Аналіз розробки проєкту web-сервісу з продажу електротоварів та його можливостей

Для успішної розробки комерційного web-сервісу, насамперед, необхідно приділяти увагу потребам користувачів, зокрема:

1. Розробка зрозумілого дизайну.
2. Відмова від використання газетних шрифтів, оскільки це вже застаріла практика.
3. Мінімізація кількості тексту для уникнення відволікання від головного контенту.
4. Забезпечення зрозумілості інформації, яка подається на веб-сервісі.
5. Надання можливості зворотного зв'язку для користувачів.

У 2024 році для ефективної роботи необхідно мати зручну та інформативну інформаційну систему. Ми ставили перед собою завдання автоматизувати процес запису на певні завдання та забезпечити гнучкий доступ до інформації на порталі. Ці потреби та інші фактори визначили актуальність нашої роботи.

Для забезпечення ефективної автоматизації процесів електронної торгівлі, зокрема у сфері продажу електротоварів, було обрано стратегію впровадження індивідуальної інформаційної системи. Такий підхід дозволяє забезпечити гнучкість у налаштуванні системи та її масштабування відповідно до специфічних потреб бізнесу.

Кафедра				НАУ 24 21 87 000 ПЗ			
Виконав	Кузнєцов А.А.			ОГЛЯД ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ПІДТРИМКИ КОМЕРЦІЙНИХ ПРОЦЕСІВ	Літера	Аркуш	Аркушів
Керівник	Водоп'янов С.В.					11	59
					ТП-415Б - 122		
Н-контр.	Сидоренко В.М.						

Централізація програмної логіки на серверах та доступність користувацького інтерфейсу через інтернет сприяють підвищенню оперативності та зручності використання.

Розробка web-сервісу для комерційного порталу ведеться за методом SMART, який допомагає чітко визначити і досягти поставлених цілей, що включають:

1. Конкретність: Створення веб-web-сервісу як основної платформи продажу.
2. Вимірюваність: Моніторинг відгуків користувачів для оцінки ефективності web-сервісу.
3. Досяжність: Використання сучасних веб-технологій (HTML, CSS, PHP, JavaScript) для розробки web-сервісу.
4. Реалістичність: Використання доступних технічних та програмних ресурсів.
5. Обмеженість за часом: Встановлення чіткого терміну завершення проєкту.

Задля планування структури робіт за проєктом використовується Work Breakdown Structure (WBS) — графічний метод, що дозволяє візуально представити всі компоненти проєкту в логічному і послідовному порядку. WBS може бути також представлена як діаграма Ганта, яка організує задачі у вигляді табличного списку.

Структура декомпозиції робіт (WBS) є невід'ємною частиною проєктного менеджменту, що сприяє детальному розподілу проєкту на керовані сегменти. Кожен сегмент проєкту може мати призначеного керівника, який відповідатиме за виконання конкретного пакету робіт.

Організаційна структура проєкту (OBS) дозволяє управляти проєктом на різних рівнях, розподіляючи робочі пакети між командами та виконавцями. OBS відіграє ключову роль у розподілі відповідальності та обов'язків між учасниками проєкту, відображаючи відносини підпорядкованості.

Крім того, у процесі розробки веб-сервісу для продажу електротоварів особлива увага приділяється управлінню ризиками. Проведення кількісної та якісної оцінки ризиків допомагає ідентифікувати потенційні загрози та розробити стратегії їх мінімізації. Планування реагування на ризики забезпечує підготовку до можливих змін у проєкті, що знижує негативний вплив ризиків на загальні результати.

Такий комплексний підхід у процесі розробки та впровадження веб-сервісу з продажу електротоварів сприяє створенню ефективного, гнучкого та відповідного до потреб користувачів комерційного порталу.

Цей проєкт має на меті не лише створення веб-сервісу, а й забезпечення його стійкості та адаптивності до змінюваних умов ринку. З цією метою, розробка системи передбачає інтеграцію з передовими аналітичними інструментами, що дозволить збирати дані про поведінку користувачів, аналізувати тенденції покупок та прогнозувати майбутні потреби клієнтів. Вбудовування таких функцій допоможе оптимізувати запаси, управління асортиментом та маркетингові кампанії, підвищуючи тим самим загальну ефективність бізнесу.

Ключовим аспектом є також забезпечення високої безпеки транзакцій та захисту даних користувачів. Система буде оснащена сучасними методами шифрування даних та протоколами безпеки, які гарантуватимуть захист інформації від несанкціонованого доступу та інших кіберзагроз.

Розробка і впровадження цього веб-сервісу відбуватиметься за етапами, з чітким дотриманням графіку і бюджету, що було ретельно плановано за допомогою WBS та OBS. Кожен етап буде супроводжуватися тестуванням та оцінкою, забезпечуючи постійне вдосконалення системи та своєчасне виявлення та виправлення помилок.

Завершення проєкту планується з демонстрацією готової продукції ключовим стейкхолдерам та замовникам, а також проведенням навчальних сесій для персоналу, що буде взаємодіяти з новою системою. Запуск веб-сервісу стане значущим кроком у розвитку бізнесу, відкриваючи нові можливості для

зростання та інновацій у сфері електронної комерції. Кількісні та якісні підходи використовуються для оцінювання ризиків у проєктах (табл. 1.1.)

Таблиця 1.1.

Ймовірність виникнення і величина ризику

Ризик	Ймовірність виникнення	Величина втрат
Поломка ПК	4	0
Не підтримка потрібного софту	1	0
Неправильний розподіл часу	5	2
Непередбачувані витрати	2	1
Некоректне складення ТЗ	1	1
Зміна в ТЗ	4	2
Непередбачувані ситуації	2	3

Рішення приймаються на основі аналізу невизначеностей, що виникають у різних елементах системи. Також розглядається залежність між ризиками та складовими проєкту, оцінюється вплив цих ризиків на загальний процес. Важливу роль у цьому процесі грає досвід та компетенція керівника проєкту.

Ризики оцінюються за наступними показниками:

Ймовірність виникнення:

- 1 Слабо ймовірно
- 2 Мало ймовірно
- 3 Ймовірно
- 4 Вельми ймовірно
- 5 Майже можливо

Ступінь втрат:

- 1 Мінімальна
- 2 Низька
- 3 Середня
- 4 Висока
- 5 Максимальна

На рисунку 1.1. представлення матриця ймовірності втрат

Ймовірність виникнення	5				4	
	4			1, 6, 7		2
	3				5	
	2			3		
	1			8		
		1	2	3	4	5

Рисунок 1.1. Матриця «Ймовірність-Втрати»

Класифікація за рівнем ризику:

недопустимі ризики;

виправданні ризики;

прийнятні ризики.

Класифікація за ступенем впливу:

помірні ($9 \leq R \leq 11$);

незначні ($5 \leq R \leq 8$);

критичні ($20 \leq R \leq 25$);

ігноровані ($1 \leq R \leq 4$);

вагомі ($12 \leq R \leq 19$).

1.2. Вибір сучасних інструментів та аналіз методологій для розробки комерційного web-сервісу та бази даних

В рамках розробки комерційного веб-сервісу та бази даних, важливим етапом є вибір сучасних технологій та вивчення наявних методологій.

1. Одним з ключових інструментів у цьому процесі є HTML – мова гіпертекстової розмітки, яка служить для створення структури веб-вмісту. HTML дозволяє розміщувати на сторінці такі об'єкти як інтерактивні форми, фотографії, списки, таблиці та абзаци, використовуючи різноманітні елементи. Ці елементи дозволяють стилізувати текст, створюючи гіперпосилання, виділяючи слова курсивом, регулюючи розмір шрифту, та виконувати інші дії, що впливають на вигляд і поведінку вмісту на сторінці. Важливо зазначити, що хоча браузері й не відображають теги HTML, вони використовують ці теги для коректного відображення та інтерпретації сторінок веб-сервісу.

За допомогою HTML можна ефективно управляти структурною і семантичною організацією документів, що забезпечує не тільки коректне відображення контенту, але й його доступність та зручність використання для кінцевого користувача. Це робить HTML незамінним інструментом у кожному етапі проектування сучасних веб-сервісів.

2. При розробці комерційного веб-сервісу особливу увагу приділяємо не тільки структурі веб-сторінок за допомогою HTML, але й їхньому візуальному

стилю. Для цього ми вибрали Scss, скриптову метамову, яка спрощує та покращує процес створення каскадних таблиць стилів (CSS). SCSS значно розширює можливості звичайного CSS, забезпечуючи розробникам велику гнучкість та швидкість у написанні коду.

Використання SCSS дозволяє нам ефективно управляти стилями, використовуючи можливості, які виходять за рамки базового CSS. Ось декілька переваг SCSS:

- Швидкість розробки. SCSS забезпечує значну економію часу при написанні стилів завдяки коротшим та зрозумілішим конструкціям.
- Програмні можливості. SCSS дозволяє використовувати змінні, умовні оператори, цикли та інші особливості, схожі на звичайні мови програмування, що робить стилі більш модульними та легкими для підтримки.
- Сумісність. CSS сумісний з усіма версіями CSS, що дозволяє використовувати існуючі CSS-бібліотеки без конфліктів.
- Розширені функції. Синтаксис вкладеності, маніпуляції з кольорами, математичні операції та інші допоміжні функції роблять SCSS потужним інструментом для управління стилями.

Ці переваги роблять SCSS ідеальним вибором для проектування складних та динамічно змінюваних інтерфейсів, забезпечуючи не тільки красу, але й функціональність майбутнього веб-сервісу.

3. При створенні веб-сервісу для продажу електротоварів ми обрали React.js — сучасний фреймворк для JavaScript, який ідеально підходить для розробки односторінкових веб-додатків (SPA). SPA є важливим елементом у створенні ефективних та зручних інтернет-магазинів, оскільки вони забезпечують швидке завантаження та плавну інтеракцію з користувачем без потреби перезавантажувати сторінку.

Ряди переваг пропонує React.js для роботи:

- Велике співтовариство. Наявність широкого співтовариства розробників означає, що React.js постійно оновлюється та підтримується, що є критично важливим для сучасних технологій.

- Декларативний підхід. Цей стиль кодування спрощує розробку та тестування додатків, оскільки дозволяє описати, що має бути зроблено, а не як саме досягти результату. Це полегшує підтримку та масштабування проєкту.

- Висока продуктивність розробки. Завдяки модульності React.js дозволяє розбити інтерфейси на незалежні компоненти, які можуть бути легко повторно використані та тестовані окремо.

- Модульність. За допомогою React.js можна створювати ізольовані частини додатка, які можуть бути взаємозамінними або використовуватися повторно в різних частинах системи. Наприклад, модуль для входу через Facebook може бути інтегрований як на сторінці входу, так і в процесі оформлення замовлення.

- Наявність готових рішень. Широкий вибір готових модулів та бібліотек, таких як ui-router для роутингу або ui-grid для таблиць, значно прискорює розробку та забезпечує високу якість кінцевого продукту.

- Двостороннє зв'язування даних. React.js автоматично оновлює інтерфейс користувача відповідно до змін у даних, що значно полегшує розробку динамічних інтерфейсів та підтримку взаємодії користувача.

Завдяки цим перевагам, React.js стає відмінним вибором для розробки інтерактивного веб-сервісу, який забезпечує неперервну взаємодію з користувачами, динамічне оновлення вмісту без перезавантаження сторінки, забезпечуючи швидкий та зручний досвід покупок.

4. Однією з найпопулярніших реляційних систем управління базами даних є MySQL з відкритим вихідним кодом. Ця система широко застосовується для створення масштабованих і високопродуктивних додатків баз даних, що використовуються як у невеликих, так і в глобальних проєктах.

Переваги MySQL включають високий рівень масштабованості та надійності, що робить її ідеальним вибором для компаній, які прагнуть рости та розширювати свою діяльність без втрати продуктивності. Безпека та стабільність роботи MySQL також є ключовими чинниками, які забезпечують безперервну роботу системи, навіть у високонавантажених умовах.

Широкий спектр розширених функцій та інструментів для адміністрування користувачам надає MySQL. Включення підтримки від експертів, доступна при виникненні будь-яких питань, є важливим ресурсом для вирішення проблем та оптимізації продуктивності бази даних.

Ці особливості роблять MySQL незамінним інструментом для розробки надійних веб-сервісів у сфері продажу електротоварів, де потрібна швидка обробка великих обсягів даних та забезпечення високого рівня доступності. MySQL дозволяє ефективно керувати транзакціями, інвентаризацією та даними клієнтів, забезпечуючи плавне та ефективне ведення бізнесу.

Було прийнято рішення для порівняння двох систем управління базами даних таких як MySQL і PostgreSQL (табл. 1.2.)

Таблиця 1.2.

Порівняльна таблиця MySQL і PostgreSQL

№	Критерії для оцінки	PostgreSQL	MySQL
1	Відкрите джерело	Відкрите джерело	Відкритий код, який належить компанії Oracle
2	Можливість зберігати інформацію тільки в пам'яті	Ні	Так
3	Підтримка NoSQL / JSON	Тільки JSON	Підтримка тільки деяких функцій
4	Декларативне секціонування	Підтримується	Підтримується
5	Відповідність вимогам ACID	Повністю відповідає	Частково відповідає
6	Віконні функції	Підтримується	Підтримується

Використання відомих технологічних рішень призводить до значних переваг у процесі розробки та підтримки проєктів. Широко використовувані системи мають великі спільноти, що сприяє швидкому знаходженню інформації та рішень для типових проблем, що можуть виникнути під час використання. На противагу, менш популярні технології часто мають обмежені ресурси для підтримки, що може призвести до затримок і збільшення витрат на вирішення проблем.

Опитування, проведене серед понад 65,000 професійних розробників у 2020 році, демонструє цю тенденцію, виділяючи які саме технологічні

інструменти є найбільш популярними і часто використовуються у професійному середовищі. Ці дані дозволяють розуміти, на які інструменти варто орієнтуватися при плануванні технологічної стратегії проєкту, щоб забезпечити його високу підтримку та відповідність сучасним вимогам.

Таким чином, при виборі технологій для розробки веб-сервісу, особлива увага приділяється популярним та широко підтримуваним рішенням. Це не тільки знижує потенційні ризики та проблеми зі стабільністю та масштабованістю, але й сприяє більш ефективній роботі розробників, що позитивно впливає на загальну продуктивність та успіх проєкту (рис. 1.2.)

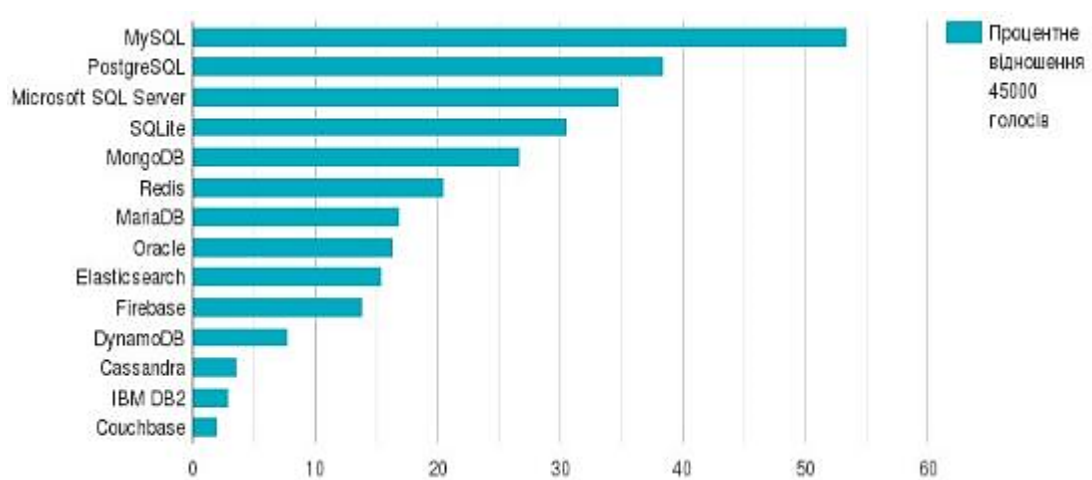


Рисунок 1.2. Використовування систем для баз даних професійними розробниками

Отже, для підтримки комерційного порталу web-сервісу з продажу електротоварів ми вибрали MySQL як основу для розробки нашого проєктованого ресурсу. Вибір MySQL зумовлений численними перевагами цієї системи управління базами даних, які ідеально підходять для наших потреб:

- Доступність. MySQL є безкоштовною та з відкритим кодом, що знижує загальні витрати на розробку та підтримку проєкту.
- Легкість установки. Швидке та просте встановлення MySQL забезпечує невеликий час запуску проєкту.
- Інтуїтивно зрозумілий інтерфейс. MySQL пропонує зрозумілий інтерфейс, який спрощує роботу розробників та адміністраторів баз даних.

- Постійні оновлення. Регулярні оновлення MySQL підтримують високий рівень безпеки та ефективність системи.
- Функціональність. Великий набір інструментів для керування базою даних дозволяє виконувати складні запити та аналізувати великі обсяги даних, що є критично важливим для ефективної організації навчальних процесів.
- Вбудовані заходи безпеки. MySQL має потужні засоби захисту даних, що забезпечують надійне зберігання конфіденційної інформації студентів та викладачів.
- Довіра на ринку. Використання MySQL світовими гігантами, такими як Google, GitHub та NASA, свідчить про її надійність та масштабованість.

Завдяки цим факторам, MySQL стає оптимальним вибором для розробки надійної та ефективної системи, яка підтримує організаційні процеси комерційного порталу, сприяє покращенню доступу до ресурсів та забезпечує високу продуктивність управління даними. Використання такого потужного інструменту як MySQL відкриває широкі можливості для розширення функціоналу і масштабування проєкту в майбутньому, забезпечуючи стійку платформу для постійного розвитку і вдосконалення економічних процесів.

[2,13,15,22,31-32]

ВИСНОВКИ ДО РОЗДІЛУ 1

Отже, для реалізації цього проекту ми провели ретельний вибір технологій та аналіз методологій, що дозволяють створювати сучасні веб-сервіси. Вибрані інструменти включають:

- HTML. Мова гіпертекстової розмітки, що формує основу структури веб-контенту.
- SCSS. Скриптова метамова, яка компілюється у CSS і сприяє ефективному управлінню стилями на веб-сайтах.
- React.js. Потужний фреймворк для JavaScript, який підтримує створення динамічних односторінкових веб-додатків, забезпечуючи високу швидкість і зручність користування.
- MySQL. Вільна реляційна база даних, яка є однією з найбільш популярних на ринку завдяки своїй надійності, масштабованості та широкій підтримці.

Цей вибір технологій забезпечує надійну основу для створення ефективного веб-сервісу, що може адаптуватися до зростаючих потреб бізнесу. Комбінація HTML, SCSS, React.js та MySQL дозволяє розробити стабільну та функціональну платформу, яка може впоратися з великими обсягами даних, гарантуючи безпеку та легкість в управлінні. Така інтеграція передових технологій сприятиме підвищенню якості економічної діяльності в Україні, роблячи її більш доступною та ефективною для всіх учасників економічного процесу.

РОЗДІЛ 2

ВИМОГИ ДО ПРОЄКТОВАНОГО ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ЕЛЕКТРОТОВАРІВ

2.1. Аналіз цільової аудиторії програмного продукту

Перед початком розробки проєкту веб-сервісу для продажу електротоварів, ми мусимо чітко визначити основну мету дослідження та визначити, які інформаційні потреби та послуги необхідні нашим користувачам. Проєкт спрямований на створення інформаційної системи (ІС), яка буде слугувати як навчальний ресурс, забезпечуючи користувачів швидким доступом до важливих ресурсів та можливістю легкої реєстрації та інтеракції на платформі.

Основні характеристики і вимоги до цієї ІС включають:

1. Адаптивність до різних пристроїв. Платформа повинна бути доступною та зручною для користувачів на будь-якому пристрої, незалежно від розміру його екрану.
2. Легкий доступ до інформації. Головна сторінка повинна надавати легкий доступ до актуальної інформації про завдання, нові та виконані заявки.
3. Детальна інформація про сервіс. Повна прозорість інформації про послуги та можливості, які надає веб-сервіс.
4. Інтерактивність. Можливість користувачів створювати онлайн-записи на конкретні завдання.
5. Організація завдань. Зручна та інтуїтивна система управління завданнями.

Кафедра КІТ				НАУ 24 21 87 000 ПЗ			
Виконав	Кузнєцов А.А.			ВИМОГИ ДО ПРОЄКТОВАНОГО ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ЕЛЕКТРОТОВАРІВ	Літера	Аркуш	Аркушів
Керівник	Водоп'янов С.В.					22	59
					ТП-415Б - 122		
Н-контр.	Сидоренко В.М.						

Фінальним продуктом є веб-сайт комерційного порталу з продажу електротоварів, що відіграватиме важливу роль у підтримці роботи користувачів з усіх регіонів України, спрощуючи їхню щоденну діяльність та сприяючи ефективнішій роботі.

Створення цього веб-сервісу вимагає інтеграції передових підходів до розробки ІС, щоб забезпечити безперебійний доступ до ресурсів та зручність користування.

Інтерфейс платформи повинен бути інтуїтивно зрозумілим, включаючи взаємопов'язані розділи з чітко окресленими функціями, такими як інформаційні сторінки про портал, контактні дані, онлайн-записи, та детальний список послуг.

Аналіз цільової аудиторії відіграє ключову роль в оптимізації маркетингових стратегій і досягненні максимальної вигоди від просування та рекламних кампаній, гарантуючи, що послуги відповідають потребам і вимогам користувачів. Основою успіху проекту є знання та розуміння потреб цієї аудиторії, яка складається як з існуючих, так і з потенційних клієнтів.

2.2. Вибір методів моделювання та проєктування процесу розробки web-сервісу

Основною метою кваліфікаційної роботи є створення інформаційної системи (ІС), що надаватиме користувачам зручний доступ до повної інформації про електротовари. Веб-сайт, що розробляється, повинен мати простий та інтуїтивно зрозумілий інтерфейс, складатись з взаємопов'язаних розділів з чітко визначеними функціями.

Для досягнення цієї мети були визначені наступні ключові задачі:

- Аналіз проблеми та ринку. Дослідження актуальності проблеми, аналіз існуючих аналогів та вибір наукових методів та інструментів для розробки ІС.
- Планування розробки. Розробка детального плану роботи та технічного завдання на створення ІС.

- Тестування системи. Проведення тестування для забезпечення функціональності та ефективності розробленої системи.

Додатково, веб-сайт забезпечить користувачам можливість легко отримувати інформацію про актуальні та виконані завдання, а також про нові завдання. Оповіщення про результати запитів будуть автоматично надсилатись на електронну адресу користувача.

Методи, вибрані для реалізації цілей дипломної роботи, включають:

- Методи аналізу. Для дослідження предметної області та вибору інструментів.

- Методи моделювання. Для проєктування архітектури ІС.

- Метод синтезу. Для інтеграції та реалізації ІС.

- У процесі розробки web-сервісу з продажу електротоварів можуть застосовуватися різні методики, зокрема:

- Методи ручного написання. Це включає роботу з HTML, CSS, і можливо JavaScript, які дозволяють детально контролювати весь процес створення web-сервісу.

- Методи автоматизованого написання. Застосування WYSIWYG-редакторів, як Dream Weaver або Front Page, дозволяє швидко створювати веб-сторінки, відразу бачачи кінцевий результат.

Обрання методу розробки залежить від специфіки проєкту та вимог до якості та безпеки продукту. У випадку з економічними ресурсами, особливу увагу слід звернути на захищеність, доступність та інтуїтивність використання web-сервісу.

При використанні «ручних» методів розробки web-сервісу дизайн web-сервісу (графічне оформлення) також створюється вручну. Для цих цілей застосовуються будь-які графічні редактори за бажанням. Вручну можна відредагувати й уже готові шаблони дизайну, як платні так і безкоштовні.

Статичний сайт – сайт, що складається з статичних html (htm, dhtml, xhtml) сторінок, що складають єдине ціле. Зазвичай статичний web-сервіс являє собою набір файлів HTML, розміщених на web-сервері.

Для створення ж динамічного web-сервісу не обійтися без серверних скриптів, таких як PHP, ASP. NET і т.д.

Динамічний сайт – сайт, що складається з динамічних сторінок – шаблонів, контенту, скриптів та іншого, у вигляді окремих файлів. Як правило для відображення будь-якої кількості однотипних сторінок використовується одна сторінка-шаблон, в яку завантажується відповідний вміст, це дозволяє коригувати зовнішній вигляд web-сервісу (безліч всіх його сторінок), редагуючи всього лише один шаблон.

Друга група методів розробки web-сервісів містить у собі методи автоматизованого створення сайтів: за допомогою спеціальних конструкторів сайтів або ж систем керування контентом (CMS). Методи розробки сайтів з використанням CMS – одні з найбільш популярних на сьогоднішній день. CMS, виражаючись умовно, являє собою певну готову візуальну й програмну оболонку, що користувач може заповнити необхідним контентом, а також за своїм бажанням змінити та налаштувати. Широкі можливості для розробки web-сервісів будь-якої складності надають CMS. Саме цей метод розробки сайтів по праву вважається одним з найбільш зручних і практичних. Гнучка система налаштувань, можливість редагування самої CMS або ж окремих її елементів, легкість додавання й зміни контенту – все це зробило розробку сайтів на базі CMS по-справжньому ефективною.[14,2]

ВИСНОВКИ ДО РОЗДІЛУ 2

Отже, основна сфера проєктування – це створення інформаційної системи для веб-сервісу з продажу електротоварів, використовуючи інтеграцію технології React. Мета цієї системи полягає у спрощенні та оптимізації комерційних процесів для користувачів глобального інтернет-порталу, дозволяючи їм легко знаходити, виконувати та переглядати історію завдань. Користувачі будуть мати доступ до коротких описів завдань з посиланнями на детальнішу інформацію.

Для досягнення цих цілей, в рамках розробки системи, планується створення наступних компонентів:

Код інформаційної системи. Він буде формувати основу всієї системи, забезпечуючи необхідну функціональність та взаємодію з користувачами.

- База даних. Це центральне сховище для всіх даних, включно з деталями користувачів, історією завдань, а також параметрами і характеристиками товарів.

- Фронтенд та бекенд проєкти. Забезпечення інтерфейсу користувача та адміністрування засобами програмного забезпечення, що дозволяє керувати вмістом та взаємодіями на сайті.

Інформаційна модель проєкту web-сервісу з продажу електротоварів повинна чітко відображати всі вхідні та вихідні потоки даних, включаючи вхідні дані про користувачів та їх діяльність в системі, а також результати виконаних завдань і операцій. Це забезпечить ефективну організацію даних та їхню легку доступність для аналізу та звітності.

Таким чином, створюючи цю інформаційну систему комерційного порталу, ми прагнемо не тільки оптимізувати комерційні процеси, але й забезпечити користувачам максимально зручний та інтуїтивно зрозумілий інтерфейс для ефективної взаємодії з веб-сервісом.

РОЗДІЛ 3

РОЗРОБКА ТА ДИЗАЙН WEB-СЕРВІСУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ЕЛЕКТРОТОВАРІВ

3.1. Сценарії взаємодії

Інформаційна система є відкритим ресурсом, який може використовуватися будь-якою людиною з доступом до мережі Інтернет. При введених адреси користувач переходить на головну сторінку. Через неї відбувається доступ до функціоналу ІС. Потрапивши на головну сторінку, зверху користувач може:

1. Побачити детальну інформацію про портал.
2. Побачити слайдер з фото порталу.
3. Види послуг.
4. Інформація про товари.
5. Замовлення товару.
6. Футер з посиланнями на соціальні мережі, та швидкий макет сайту.

Нами була розроблена UML-діаграма прецедентів(рис. 3.1.), яка відображає відносини між користувачами і прецедентами і є складовою частиною моделі прецедентів, що дозволяє описати систему на концептуальному рівні.

Основними можливостями користувача на сайті є отримання необхідної інформації та підтвердження закріпленої послуги при її фіксації за певним користувачем чи групою користувачів та надсилання звіту на пошту. Для обслуговування web-сервісу необхідні адміністратор web-сервісу (оновлення та редагування web-сервісу).

Кафедра КІТ				НАУ 24 21 87 000 ПЗ			
Виконав	Кузнєцов А.А.			РОЗРОБКА ТА ДИЗАЙН WEB- СЕРВІСУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ЕЛЕКТРОТОВАРІВ	Літера	Аркуш	Аркушів
Керівник	Водоп'янов С.В.					27	59
					ТП-415Б - 122		
Н-контр.	Сидоренко В.М.						

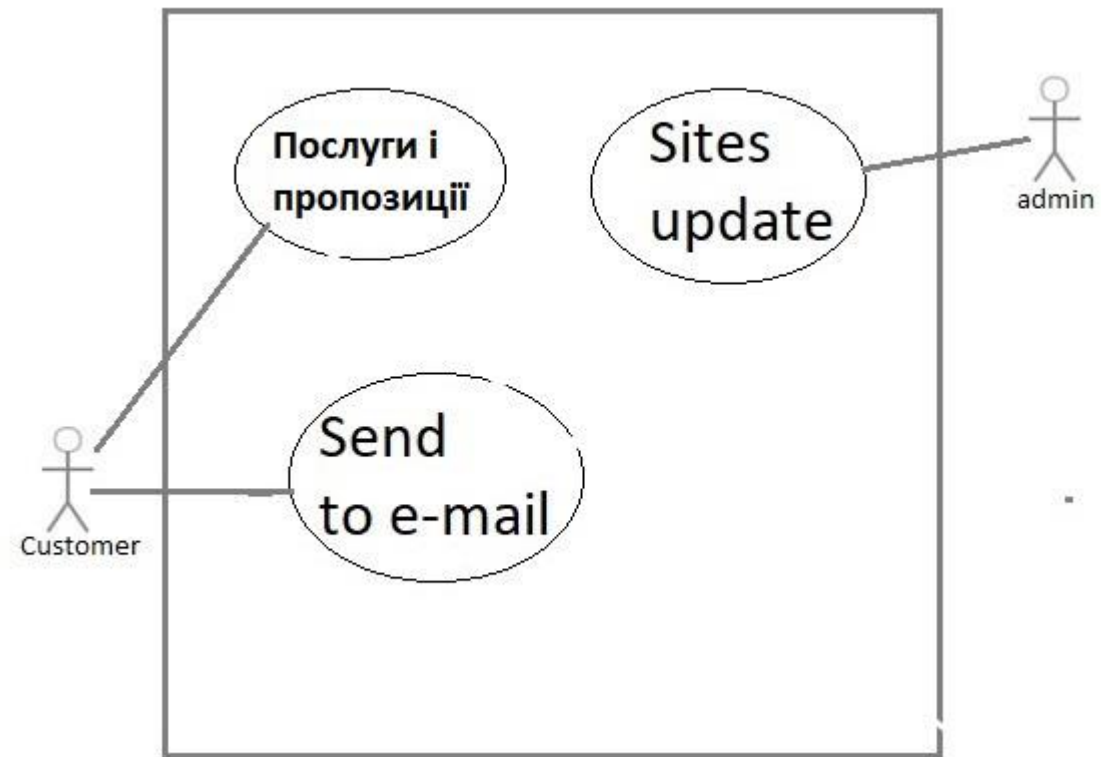


Рисунок 3.1. UML-діаграма прецедентів

Варіанти використання ІС у:

1. Send to e-mail – можливість відправки інформації на пошту.
2. Sites update – додавання інформації на сайт.
3. Послуги і пропозиції – можливість перегляду видів послуг.

3.2. Архітектура інформаційної системи

За основу архітектури інформаційної системи було взято клієнт-серверну технологію. Архітектура клієнт-сервер – одна з найвідоміших концепцій взаємодії користувача з системою, а також швидким обміном даних.

Модель клієнт-серверної взаємодії дозволяє багатьом клієнтам звертатися до серверу одночасно і це не відобразиться на роботі серверу. Виділимо операції, які виконуватимуться:

- інтерфейс користувача (відповідає за представлення даних та введення керуючих команд від користувача);
- прикладний рівень (обробка інформації та виведення результатів роботи).

- управління даними (зберігання та доступ до даних);

Згідно поставлених задач на рисунку 3.2. зображена загальна архітектура системи.

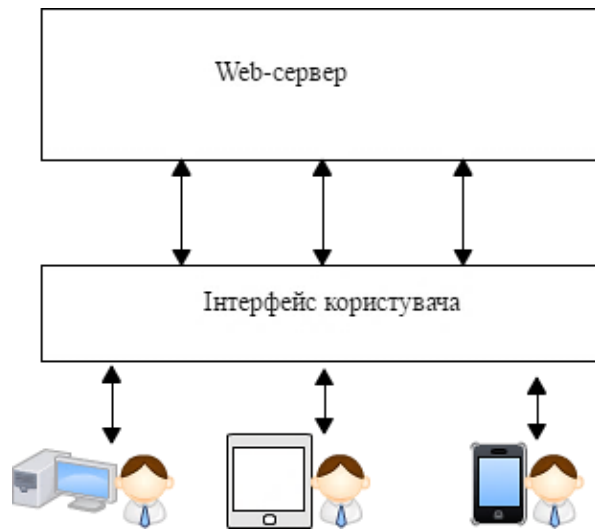


Рисунок 3.2. Архітектура інформаційної системи

Для формалізації процесів була розроблена діаграма IDEF0. Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. В IDEF0 розглядаються логічні зв'язки між роботами. Так само відображаються всі сигнали управління.

Діаграма IDEF0 представлена на рисунку 3.3. На вхід системи поступають запити користувача, який на виході отримує підтвердження замовлення товару.

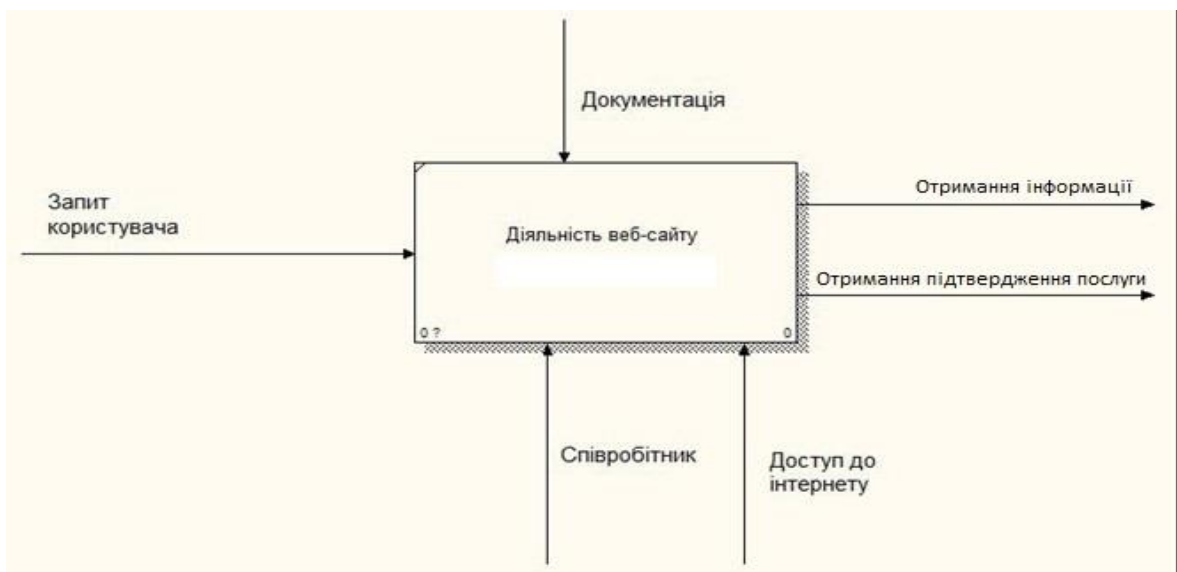


Рисунок 3.3. Діаграма IDEF0

Така модель є однією з найпрогресивніших моделей і використовується в організації бізнес проєктів і проєктів, що базуються на моделюванні всіх процесів як адміністративних, так і організаційних.

3.3. Розробка програмних модулів

Для прив'язки розробленої архітектури до середовища зберігання використовується модель даних фізичного рівня (для стислості часто називається фізичною моделлю). Ця модель визначає використовувані запам'ятовувальні пристрої, способи фізичної організації даних у середовищі зберігання. Модель фізичного рівня також будується з урахуванням можливостей, що представляються базою даних. Опис фізичної структури бази даних називається схемою зберігання. Відповідний етап проєктування бази даних називається фізичним проєктуванням. Основним завданням цього етапу проєктування є визначення відповідності типів логічної моделі даних до типів даних, що використовуються у потрібній базі даних.

Метою створення фізичної моделі є забезпечення адміністратора відповідною інформацією для переносу логічної моделі даних у базу даних.

Erwin підтримує автоматичну генерацію фізичної моделі даних для конкретної бази даних. При цьому логічна модель трансформується у фізичну за наступним принципом: сутності стають таблицями, атрибути стають стовпцями, а ключі стають індексами.

Для даного проєкту файлова структура фронтенд- та бекенд-проєкту зображена на рисунку 3.4.

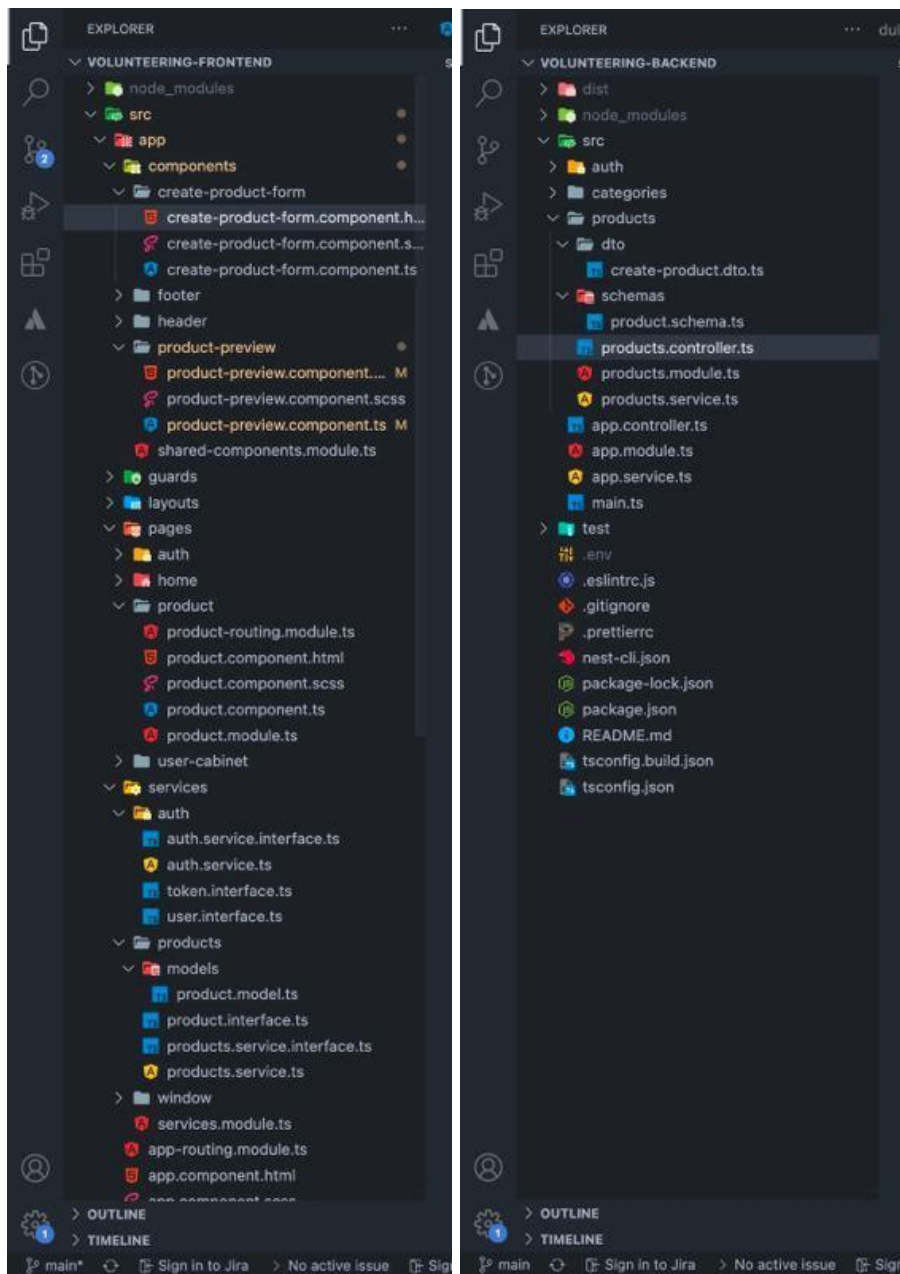
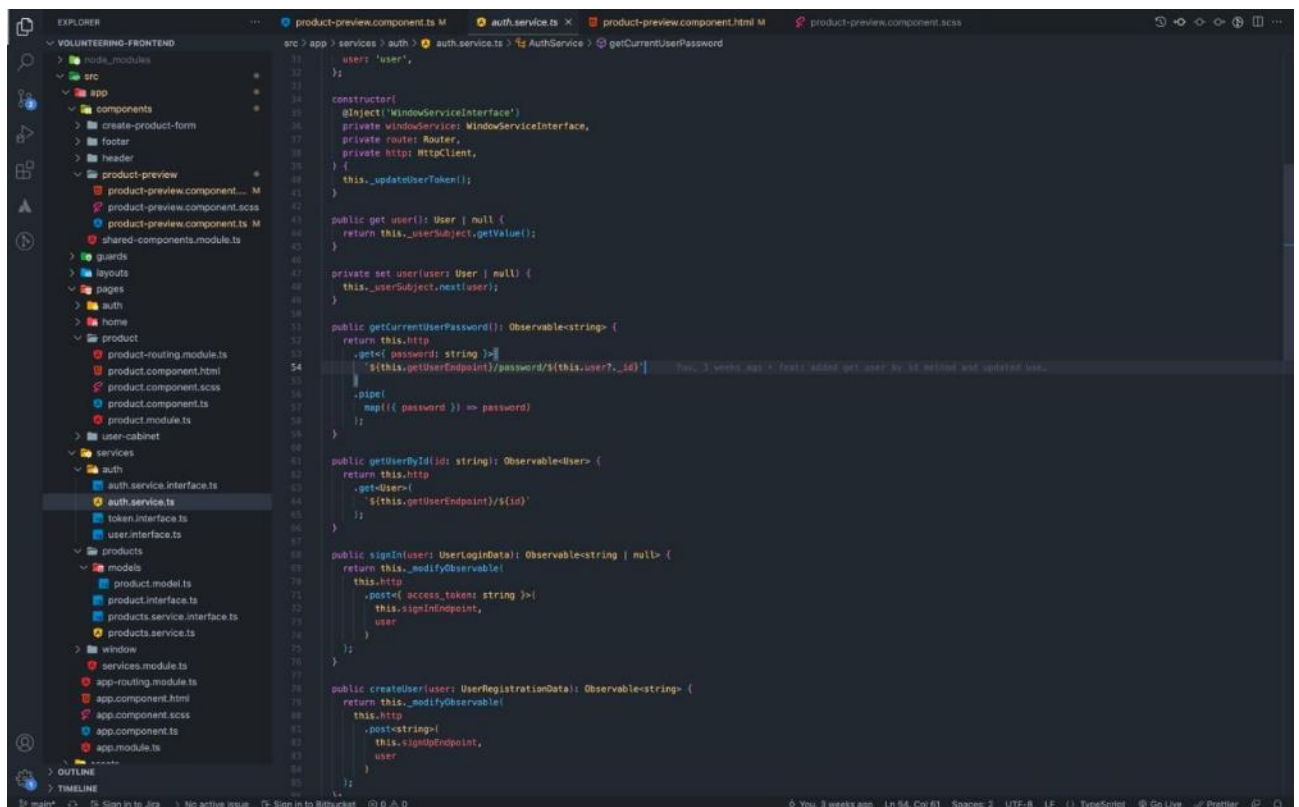


Рисунок 3.4. Файлова структура фронтенд- та бекенд-проекту

Для зберігання графічних об'єктів використовується база даних. Також деякі логотипи зберігаються в окремій папці. Бізнес логіка розділена на дві частини: клієнтська та адміністраторська взаємодія з програмою. Клієнт може реєструватись, логінитись, шукати послугу за частиною його назви, переглядати інформацію про послугу, його параметри і наявність та обрати послугу для виконання. Адміністратор може редагувати наступні таблиці: завдань, користувачів інформаційної системи web-сервісу з продажу електротоварів та їх облікових даних, всі інші лише переглядати.

Створення облікової анкети на сайті (рис. 3.5). Облікова анкета є частиною системи, яка відповідає за облік користувачів ресурсу web-сервісу з продажу електротоварів, і не дає можливості помилкового редагування картки товару. Користувач при попаданні на сайт може реєструватися для того, щоб мати повний доступ до всіх послуг, які надає проєктований сервіс. Облік також дає можливість адміністраторам інформаційної системи та замовникам інформаційної системи контролювати картки товару та облікові дані користувачів.



```
src > app > services > auth > auth.service.ts | AuthService > getCurrentUserPassword
21 user: 'user',
22 };
23
24 constructor(
25   @Inject('WindowServiceInterface')
26   private windowService: WindowServiceInterface,
27   private router: Router,
28   private http: HttpClient,
29 ) {
30   this._updateUserToken();
31 }
32
33 public getUser(): User | null {
34   return this._userService.getUser();
35 }
36
37 private set user(user: User | null) {
38   this._userService.setUser(user);
39 }
40
41 public getCurrentUserPassword(): Observable<string> {
42   return this.http
43     .get<password: string>(>
44       `${this.getUserEndpoint}/${this.user?.id}`
45     )
46     .pipe(
47       map((password) => password)
48     );
49 }
50
51 public getUserById(id: string): Observable<User> {
52   return this.http
53     .get<User>(>
54       `${this.getUserEndpoint}/${id}`
55     );
56 }
57
58 public signIn(user: UserLoginData): Observable<string | null> {
59   return this._modifyObservable(
60     this.http
61       .post<access token: string>(>
62         this.signInEndpoint,
63         user
64       )
65   );
66 }
67
68 public createUser(user: UserRegistrationData): Observable<string> {
69   return this._modifyObservable(
70     this.http
71       .post<string>(>
72         this.signInEndpoint,
73         user
74       )
75   );
76 }
77
78 }
```

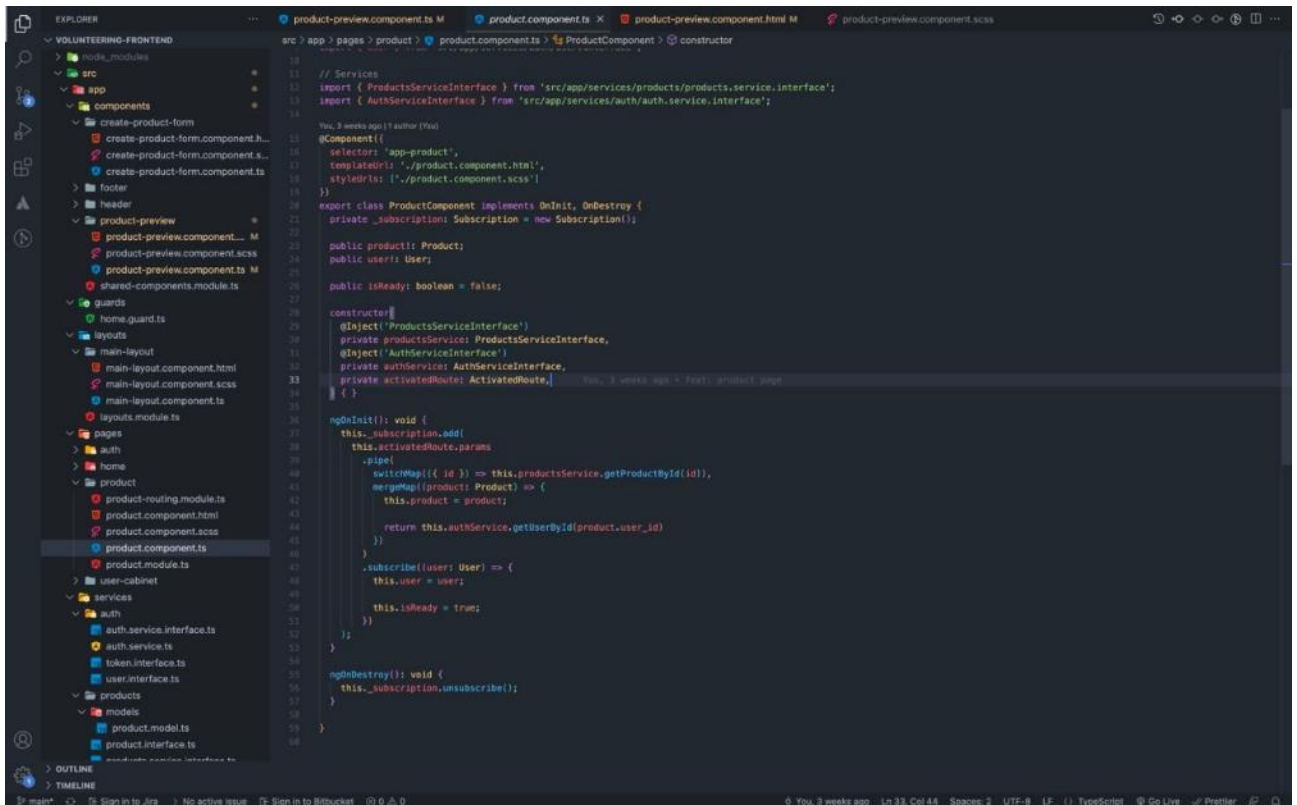
Рисунок 3.5. Процес авторизації користувача у фронтенд-частині проєкту

Вихідними даними підсистеми є: імена користувачів, дата народження користувача, та його соціальні профілі, що є важливим. На виході підсистема видає оброблену коротку характеристику користувача, і дає можливість редагувати ці дані, в разі невірного введення із початку реєстрації.

Проєктована система має можливість існувати без інформаційної системи, але тільки як для збирання інформації про користувачів системи, але ця інформація нікуди не пересилається та не зберігається як база даних.

Підсистема «Створення облікового запису» виконує наступні функції:

- Ввід даних та редагування на сайті (рис. 3.6.)
- Зберігання інформації та отримання від системи контролю за обліком.
- Формування корзини для завдань.
- Перевірки правильності введених даних.



```
12 // Services
13 import { ProductServiceInterface } from 'src/app/services/products/products.service.interface';
14 import { AuthServiceInterface } from 'src/app/services/auth/auth.service.interface';
15
16 @Component({
17   selector: 'app-product',
18   templateUrl: './product.component.html',
19   styleUrls: ['./product.component.scss']
20 })
21 export class ProductComponent implements OnInit, OnDestroy {
22   private _subscription: Subscription = new Subscription();
23
24   public product!: Product;
25   public user!: User;
26
27   public isReady: boolean = false;
28
29   constructor(
30     @Inject('ProductsServiceInterface')
31     private productService: ProductServiceInterface,
32     @Inject('AuthServiceInterface')
33     private authService: AuthServiceInterface,
34     private activatedRoute: ActivatedRoute
35   ) {}
36
37   ngOnInit(): void {
38     this._subscription.add(
39       this.activatedRoute.params
40         .pipe(
41           switchMap(({ id }) => this.productService.getProductById(id)),
42           mergeMap((product: Product) => {
43             this.product = product;
44           })
45         )
46       .return(this.authService.getUserById(product.user_id))
47     );
48     .subscribe((user: User) => {
49       this.user = user;
50       this.isReady = true;
51     });
52   }
53
54   ngOnDestroy(): void {
55     this._subscription.unsubscribe();
56   }
57 }
```

Рисунок 3.6. Процес редагування картки товару у фронтенд-частині проєкту

Підсистема «Формування корзини для завдань» формує так звану «корзину» для зареєстрованих користувачів інформаційної системи web-сервісу з продажу електротоварів, куди є можливість зберігати нові (рис. 3.7) та активні картки товарів для публікації на головній сторінці інформаційної системи. В корзині зберігається: код картки товару, номер користувача, ціна товару, коротка характеристика необхідного товару, та можливість порівняння декількох вибраних раніше послуг в системі інформаційної системи.

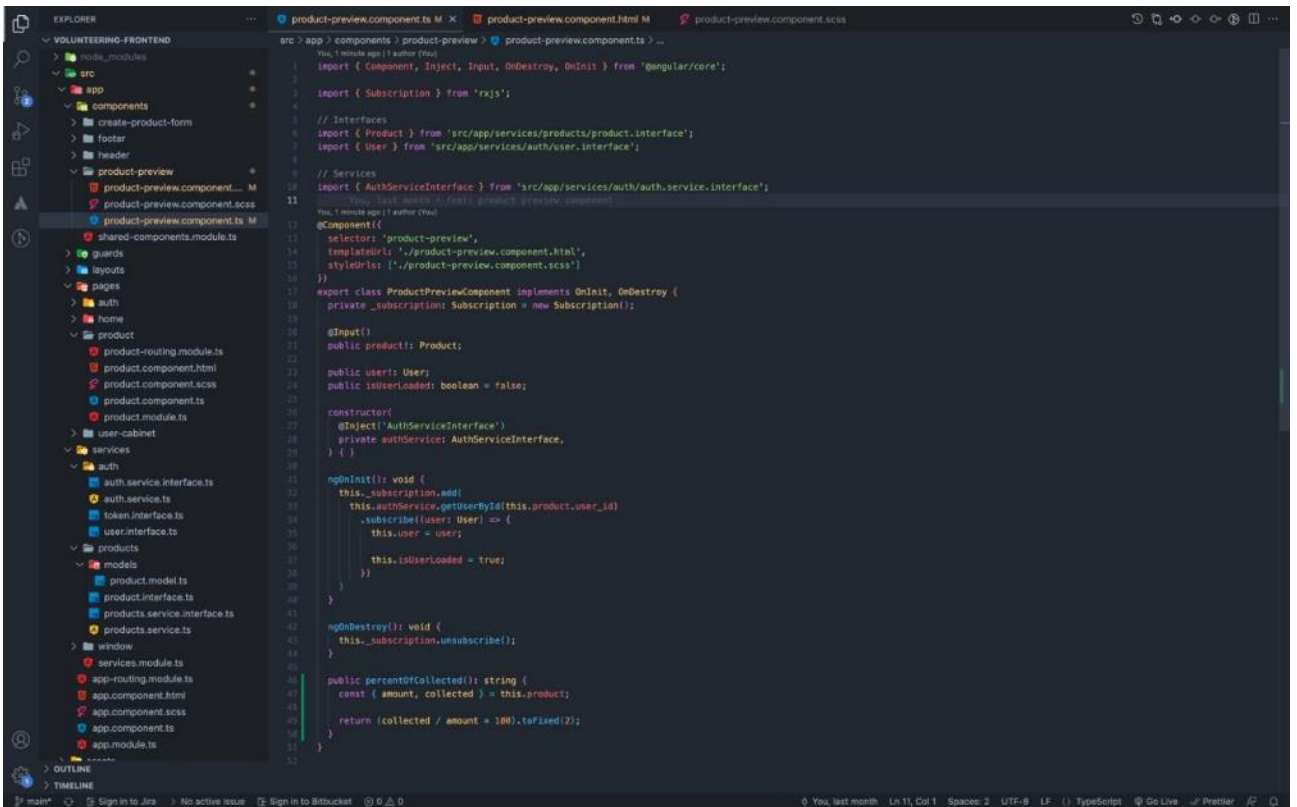


Рисунок 3.7. Створення нового компонента

Перевірки правильності введених даних проводиться автоматичними налаштуваннями інформаційної системи для того, щоб визначити та видалити застарілі та не вірно зареєстровані обліки. Це дає змогу вивільнити місця в Базі даних інформаційної системи та не займати місце, а також збільшує наглядність бази даних.

Це дуже зручно для адміністратора, так зменшується обсяг Базі даних інформаційної системи. Ці бази можна зберігати в архіві, для це більш зручного користування, а архів поділити за місяцем або роком (рис. 3.8.)

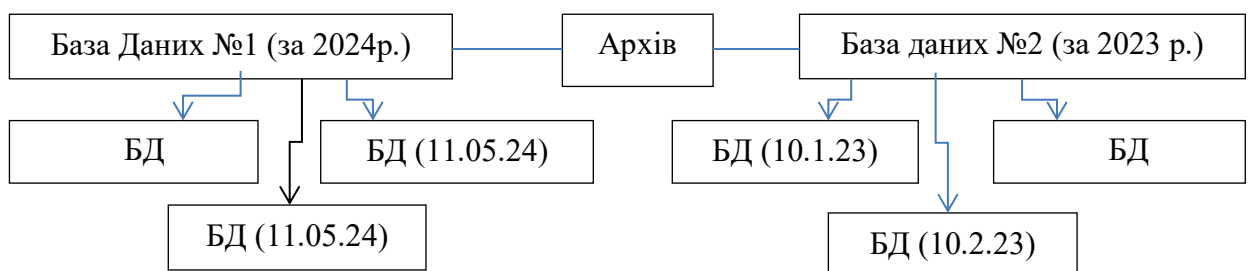


Рисунок 3.8. Структурна схема підсистеми «Архів»

Визначимо структуру створення картки товару на сайті web-сервісу з продажу електротоварів(рис. 3.9.):

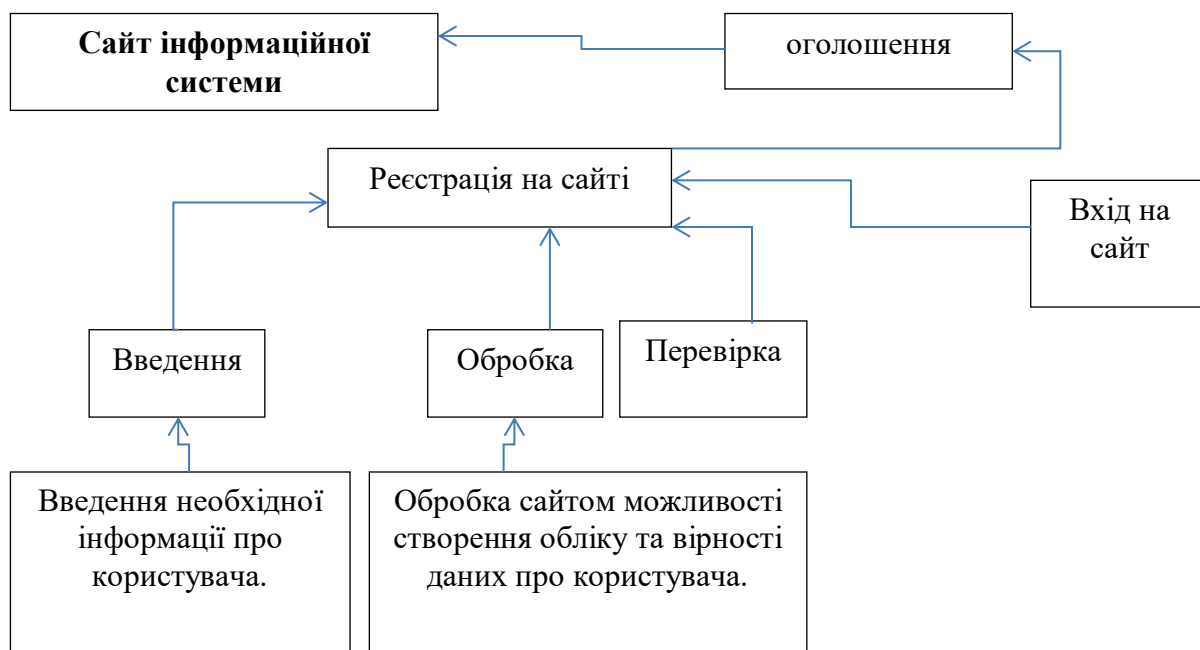


Рисунок 3.9. Структура створення картки товару на проектованому сервісі

Після проходження всіх етапів створення картки товарів, вона вже буде створена в системі у вигляді віртуальної картки та занесена автоматично до бази даних інформаційної системи, а згодом і до стартової сторінки для відображення цільовій аудиторії.

3.4. Розробка дизайну

Розробка макету дизайну – важливий етап у розробці web-сервісу. Він дозволяє уявити програмісту загальне положення елементів майбутнього проекту. Макет являє собою зображення виконане або графічними редактором, або розробкою за допомогою HTML та CSS. Розроблювана ІС повинна забезпечувати наступний функціонал:

- Достовірність та повнота інформації. Постійне оновлення web-сервісу та додавання нових елементів.

- Простий інтерфейс. Для зручного користування необхідно створити інтуїтивно зрозумілий інтерфейс.

- Швидкість роботи та відображення інформації. Необхідно використовувати нові технології для швидкої роботи системи.

Вирішено створити ІС, як web-орієнтований додаток для браузерів. В плані доступності web-технології найбільш зручні, так як надають швидший доступ до даних та не вимагають встановлення додаткового програмного забезпечення на персональний комп'ютер.

Головними дизайнерськими ознаками сучасної ІС web-сервісу з продажу електротоварів є:

- логотип;
- фірмовий стиль;
- зрозумілий інтерфейс.

Фірмовий стиль передбачає візуальний образ компанії або її продукту. Це набір графічних зображень і смислових форм, об'єднаних однією спільною ідеєю.

За основу було вирішено використовувати відтінки теплого кольору. Для збільшення функціональності використовуємо React.js для роботи з прихованими блоками. Як результат, поєднання HTML5, Scss та React.js дозволили створити яскраву та якісну інформаційну систему, яка не залишить нікого байдужим. На рисунках 3.10.–3.13. представлено макет web-сервісу.

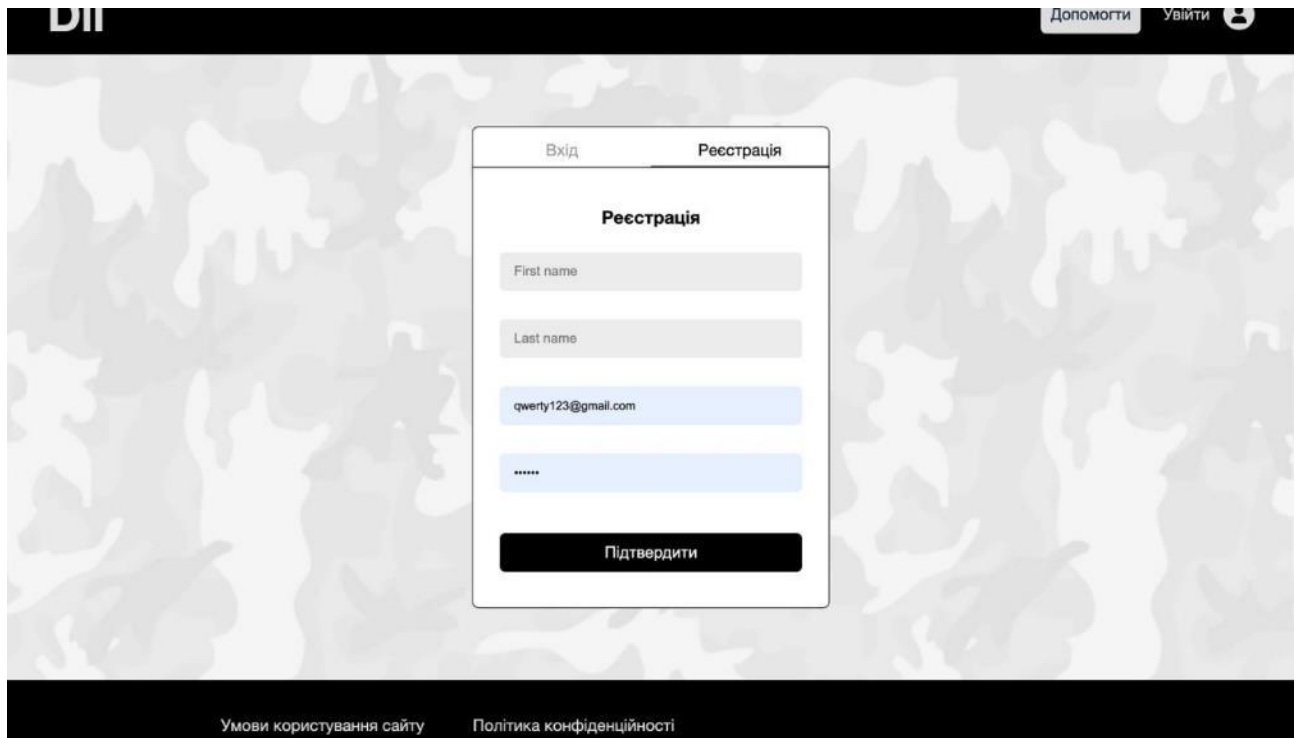


Рисунок 3.10. Форма реєстрації нового користувача

Після реєстрації відбувається зворотна автентифікація користувача шляхом перевірки правильності електронної пошти надсиланням листа з посиланням для підтвердження реєстрації.

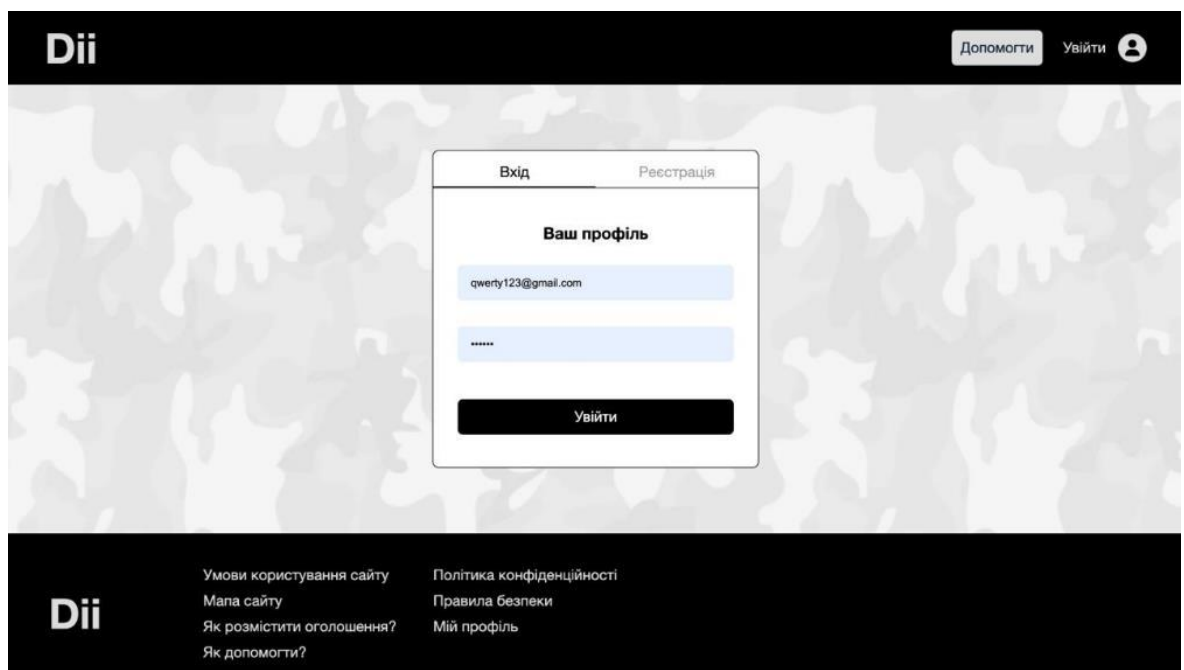


Рисунок 3.11. Форма входу користувача в персональну секцію сервісу

Після входу на сайт, відкривається стартова сторінка ресурсу web-сервісу з продажу електротоварів. Головна сторінка проєктованого ресурсу web-сервісу з продажу електротоварів має дані про користувача.

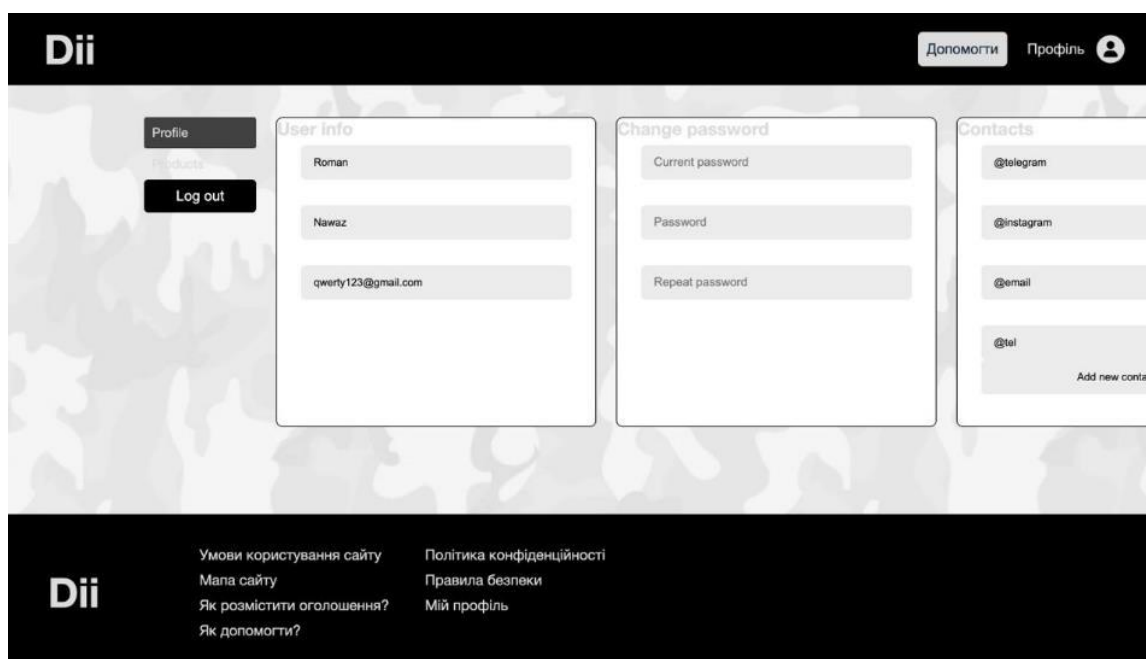


Рисунок 3.12. Сторінка редагування персональних даних користувача ресурсу web-сервісу з продажу електротоварів

Дана секція ресурсу дозволяє редагувати персональні дані користувача, підключені соціальні профілі та пароль доступу до web-сервісу. Пароль зберігається у базі даних в зашифрованому вигляді.

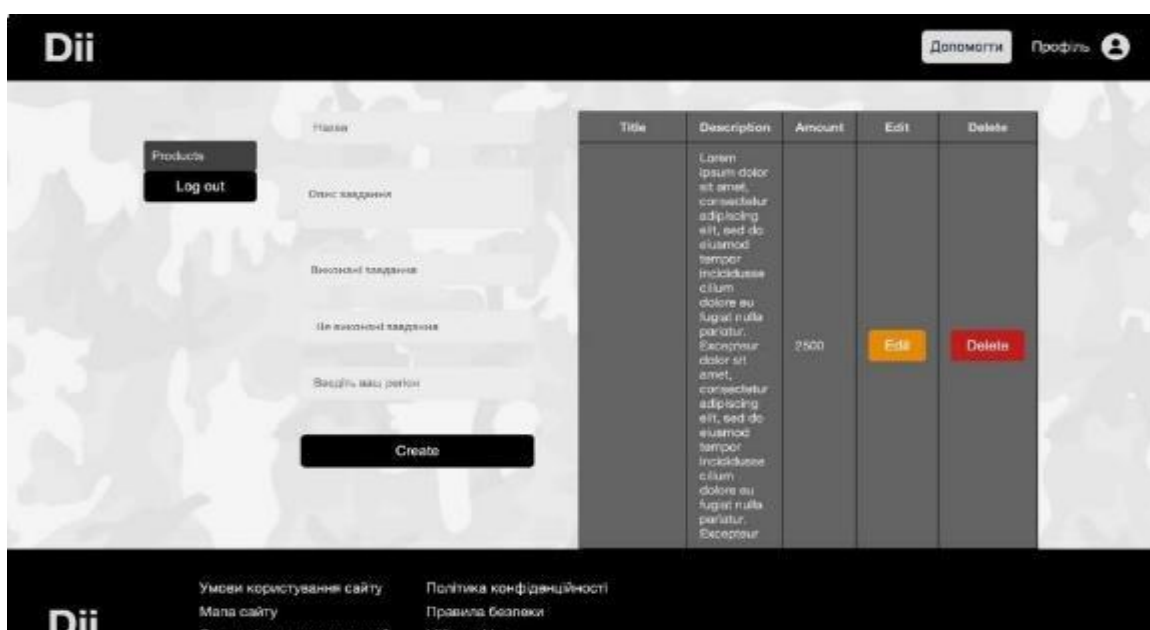


Рисунок 3.13. Редагування опису користувача

Дана секція проєктованого ресурсу web-сервісу з продажу електротоварів дозволяє редагувати опис про користувача, видаляти його та змінювати його параметри. Створення нового блоку опису в системі можливе при натисканні кнопки Create.

3.5. Розгортання програмного забезпечення проєктованої системи

Формулювання апаратних рішень щодо розгортання та роботи ресурсу web-сервісу з продажу електротоварів буде насамперед залежати не від апаратних конфігурацій сервера а від засобів, які оперують виконанням роботи додатку. Тут мається на увазі програмний комплекс роботи web-сервера від якого залежить якість роботи додатку. Формулювання вимог, щодо апаратного рішення, залежить від багатьох факторів роботи ресурсу web-сервісу з продажу електротоварів. Правильне функціонування проєктованої інформаційної системи неможливе без використання відповідного програмного забезпечення. Програмне забезпечення для роботи інформаційної системи являє собою комплекс взаємозв'язаних програмних реалізацій що підтримують функціонування самого сервісу.

Для функціонування роботи проєктованої системи потрібне програмне забезпечення яке керує його роботою та дозволяє під'єднуватися безпосередньо портативно або стаціонарно до web-ресурсу. Для цього нами спроектовано фронтенд- та бекенд-частини проєкту. Функціонал роботи додатку створений мовою програмування JavaScript та бібліотеки React.js що безпосередньо вимагає інтерпретатора коду.

Для підключення додатку потрібні повноцінний web-сервер. Web-сервер містить в собі встановлене програмне забезпечення яке відповідає за роботу як протоколів з'єднання так і самого функціонування додатку. Комп'ютер-сервер повинен містити в собі налаштування та підключені програмні засоби, що забезпечують роботу web-сервісів та встановлення сторонніх додатків на стороні бекенд-ресурсу web-сервісу з продажу електротоварів. Оскільки

проєктований web-додаток призначений для роботи у мережі Інтернет, тому серверні програми будуть вимагати певних апаратних ресурсів для його обслуговування.

Для оперування роботою web-сервера, тобто комп'ютера віддаленого доступу, потрібний web-сервер. Найпоширенішим web-сервером сьогодення є Apache. У цьому випадку буде використовуватись web-сервер Apache версії 2.4.54.

Робота додатку неможлива без використання бази даних. Щоб мати можливість збереження даних та їх оперування потрібно щоб додаток мав можливість підтримки доступу до файлової системи. Найпоширенішим рішенням щодо використання бази даних є MySQL-сервер бази даних. Також всі вище наведені компоненти повинні бути налаштовані та функціонувати як єдина система у цілому.

Отже, для оптимальної роботи проєктованої інформаційної системи web-сервісу з продажу електротоварів потрібна висока пропускна можливість каналу передачі даних web-сервера, тобто щоб швидкість завантаження даних із сервера була досить високою при великій кількості одночасних користувачів з усіх регіонів. Це є ключовою умовою при формуванні апаратних характеристик та рішень. У нашому випадку, додаток не в'являє собою надпотужний сервіс, що оперує великими обсягами інформації, тому швидкодія роботи інформаційної системи не вимагає великих обсягів апаратних засобів.

Після того як буде написаний весь програмний код необхідно створити файли проєкту, встановити їх на сервер у відповідні директорії та розпочати тестування та розгортання. Тестування сервісу – це перевірка фронтенд- та бекенд-проєкту різними методами і способами на працездатність після його встановлення. Тестування необхідно як новому, так і вже працюючому сервісу, для отримання гарантії його працездатності.

У процесі розробки програми проводилося поетапне тестування з метою виявлення програмних помилок і невідповідностей технічному завданню. Для цього були створені емулятори смартфона і планшету з різними діагоналями

екрану для різних версій Android. Тестований програмний продукт послідовно запускається на цих емуляторах, аналізується його поведінка, і при необхідності за результатами аналізу вносяться зміни в код.

3.6. Тестування інформаційного продукту інтернет-магазину

Після розробки програмного продукту web-сервісу з продажу електротоварів виконаємо етап тестування(рис 3.14), на якому перевіряємо якість розробленого продукту. Тестування буває функціональне та нефункціональне. Під час функціонального тестування проводиться перевірка на коректність роботи усіх функцій, які були описані у технічному завданні. На етапі нефункціонального тестування проводиться перевірка характеристик розробленого продукту.

Перевірка працездатності програми полягає в основному у тестуванні всіх функціональних частин та системи в цілому. У нашому випадку тестування програми є доцільним у наступному:

- Робота сесії користувача.
- Тестування стабільності з'єднання із базою даних.
- Тестування швидкості з'єднання із базою даних.
- Тестування кросплатформенності сервісу.
- Тестування працездатності технологій.
- Тестування роботи таблиць бази даних.

Для тестування окремих модулів ресурсу web-сервісу з продажу електротоварів роботи з базою даних в текст програми можуть бути внесені спеціальні функції(табл. 3.1.), що дозволять аналізувати базу даних і, при підозрі на помилку, виводити запит в системний журнал. Вони також відомі як юніт-тести.

```
public class ConectBd : MonoBehaviour
{
    public static string path = "host=127.0.0.1;port=3306;user=root;password=;database=unity";
}
```

Рисунок 3.14. Тестування з'єднання з БД

Наприклад, при змінах в базі даних проводиться перевірка цілісності бази даних (перевірка на відповідність ключів – індексів), після чого при необхідності виводиться запит.

Таблиця 3.1.

Тестування інформаційної системи web-сервісу з продажу електротоварів

TC_ID	Test Scenario / Test Step Name	Actions	ER	Статус
Prerequisite. Маємо доступ до інтернету та сайт відкрито у вікні браузера Chrome				
TC1.1	Перехід по вкладкам на сторінці	Натиснути лівою клавішею миші на вкладку Головна Натиснути лівою клавішею миші на вкладку Галерея	Перехід на відповідну вкладку	+
TC1.2	Перевірка посилання	На Головній сторінці натиснути на посилання (слово Контакти)	Перехід на вкладку Контакти	+
TC1.3	Відправлення заявки на отримання послуги	В поле вводу Ваше ім'я ввести Ваше ім'я. В поле вводу Ваш email ввести Ваш email адрес. В поле вводу Тема ввести кілька слів, що до теми вашого інтересу. В поле вводу Опис заявки ввести текст, де Ви розкриєте тему вашого запиту Натиснути кнопку Надіслати	Повідомлення про успішне відправлення заявки	+

З розробленим web-сервісом варто провести такі тести:

- Тестування usability (перевірка зручності користування сервісом web-сервісу з продажу електротоварів). У ході такого тестування визначається якість виконання і зручність інтерфейсу web-сервісу, а так само проводяться роботи з виявлення можливих помилок в структурі. Результати дають можливість визначити, наскільки правильно використовує проєктовану інформаційну систему середньостатистичний користувач, і як швидко він може знайти потрібні функції;

- Виявлення помилок. Кожна активність була піддана юніт-тестуванню з метою виявлення помилок, викликаних невідповідністю очікуваних і отриманих параметрів. Для цього для кожної активності був створений спеціальний юніт-клас, що посилає в активність різні вірні і невірні параметри. При аномальній поведінці доступу до ресурсу web-сервісу з

продажу електротоварів або його збоїв. Аналізувалася поведінка, а виявлені помилки виправлялися;

- Тестування на стійкість до великих навантажень. Цей тест імітує одночасне користування великою кількістю користувачів (сотень або навіть тисяч) проєктованої інформаційної системи для визначення працездатності ресурсу web-сервісу з продажу електротоварів при великих навантаженнях або ж інтенсивна, довгочасна робота web-сервісу в умовах невеликих ресурсів. Таке тестування обов'язкове для проєктованого сервісу з передбачуваною великою аудиторією. У ході тестування перевіряється не стільки сам ресурс, скільки комплексну роботу апаратної частини сервера, модулів, програмного ядра та інших компонентів ресурсу web-сервісу з продажу електротоварів;

- Тестування на некоректні дані. У базу даних навмисно вносилися неприпустимі дані у відповідні поля, які могли бути невірно інтерпретовані програмою проєктованої інформаційної системи. Потім аналізувалася поведінка активності сервісу під час обробки неприпустимих даних;

- Тестування на сумісність з різними версіями Android. Додаток було запущено на пристроях, що працюють під управлінням різних версій Android з метою виявлення особливостей роботи програми, запущеного в різних операційних системах. Під час проведення даного тестування web-сервіс було успішно запущено на смартфонах з такими операційними системами: Android 4.4.4 (Kit Kat), Android 5.0 (Lollipop), Android 5.1 (Lollipop), Android 6.0 (Marshmallow).

Для роботи із базою даних потрібне постійне і надійне з'єднання. Для використання постійного з'єднання із базою даних було використано xml функцію SQLite_connect. Дана функція встановлює постійний зв'язок із базою даних а також при збої транзакцій бази даних автоматично відновлює їх. Маніпуляція таблицями бази даних виконується лише через програмний код xml. Тестовий запис інформації у базу даних показав хороший результат виконання транзакцій у часі, що дає змогу розробити оптимізований алгоритм тестування.

Тестування працездатності технологій обміну даними. Тестування технології полягало у надсиланні тестових запитів на сервер із передачею параметрів. Параметри визначались як тестова інформація для передачі скриптам web-сервера. Тестування швидкості надсилання запитів показало що всі запити досить швидко відправлялись та оброблялись тестовими скриптами на сервері. У нашому прикладі піл час надсилання часових циклічних запитів до сервера локальної машини використовуючи технологію React.js не було виявлено затримки та відповіді від сервера-приймача. Окрім того React.js не створює додаткового навантаження на web-сервер а також з легкістю дає можливість контролю отриманої інформації від сервера.

Робота сесії користувача на ресурсі web-сервісу з продажу електротоварів. При реєстрації чи або авторизації користувача завіюється запрограмований код створення сесії користувача. При тестовому навантаженні реєстрації користувачів, сесії були автоматизовано присвоєні всім користувачам без явної затримки. Це дає змогу оцінити з якісної сторони алгоритм аутентифікації у системі. Працездатність роботи алгоритму роботи сесій користувача була на найвищому рівні за всіма показниками.

Тестування роботи таблиць бази даних. При використанні бази даних у тестовому варіанті наповнення даними не було виявлено затримок та простою. Таблиці заповнювались даними без зайвих проблем та зауважень. Кожна таблиця бази даних має режим хешування даних, тому час ви конання запитів до бази даних значно менший.

Тестування кросплатформенності ресурсу web-сервісу з продажу електротоварів. При запуску додатку на різних версіях Android не було суттєвої різниці у якості відображення веб сторінки та її контенту. Всі web-сторінки відображались коректно та злагоджено, що свідчить про адаптивність додатку у різних сімейств Android.

Першою важливою задачею тестування функціональності інтерфейсу було отримання токенів користувача в кожному з сервісів по-черзі. Для цього,

ми повинні створити авторизації, отримати від користувача номер телефона та надіслати його через Rest API до кожного сервісу.

Можна побачити графічний інтерфейс авторизації на мобільному пристрої який наданий на рисунку 3.15.

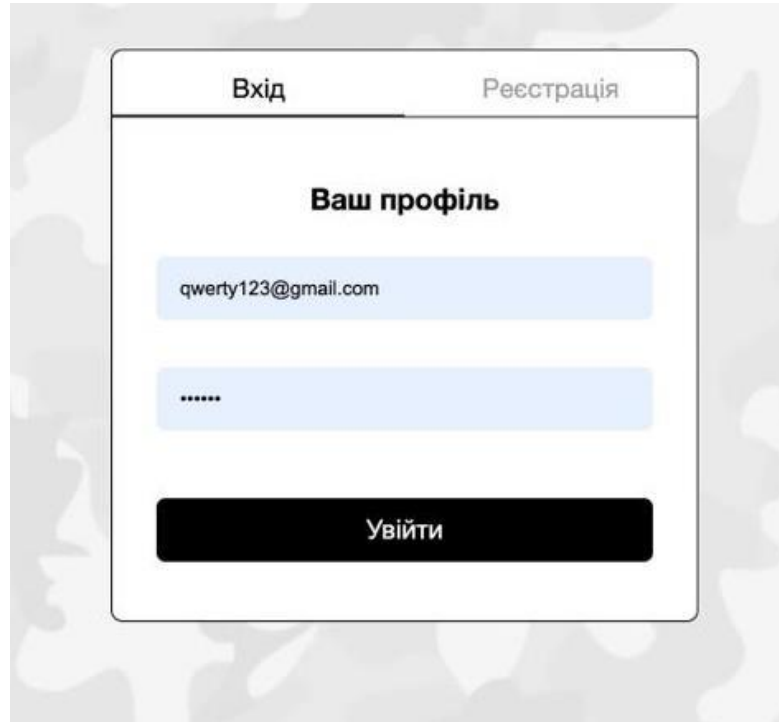


Рисунок 3.15. Форма входу на мобільному пристрої

Далі, після авторизації, нам у відповідь надходить токен користувача, який дозволяє нам робити всі інші запити.

Узагальнюючи, можна сказати, що система виконує свої функції та має можливість до масштабування та додавання нових функцій. Після тестування даного програмного продукту було встановлено, що він не має критичних або ж блокуючих помилок.[1-3,36]

ВИСНОВКИ ДО РОЗДІЛУ 3

Нами було створено інформаційне забезпечення ресурсу web-сервісу з продажу електротоварів для з дотриманням таких принципів: цілісність інформації; достовірність інформації; контроль інформації; захист від несанкціонованого доступу; єдність та гнучкість; стандартизація та уніфікація; адаптивність; мінімізація вводу та виводу інформації.

У проєктованого сервісу інформаційного забезпечення web-сервісу з продажу електротоварів реалізований зручний користувацький інтерфейс та наступні можливості для користувачів: за натиском кнопки «Замовити» користувач, що увійшов у систему може замовити обраний товар.

ВИСНОВКИ

Отже, сьогодні майже у всіх сферах життєдіяльності використовуються ресурси глобальної мережі Інтернет. Розробка і створення порталів відноситься до категорії найважливіших і найбільш актуальних заходів, які дозволяють фірмам і організаціям досягти поставлених цілей.

Метою кваліфікаційної роботи було створення ІС комерційного web-ресурсу. Головною проблемою існуючих аналогів є недостатня обізнаність користувача про сам портал. Головним елементом ІС є простота, зручне отримання вичерпної інформації з картки товару та можливість дізнатися детально про портал.

Завдяки розробці web-сервісу для користувачів ми вирішимо такі проблеми як: відстеження історій замовлень, збереження необхідних контактів, групування заявок, верифікація осіб які використовують основні функції на сайті (користувачі, адміністратори). З цією метою нами проведено вибір сучасних інструментів та аналіз методологій для розробки web-сервісів та бази даних для них.

Обрано: мову розмітки гіпертексту HTML; Scss – скриптову метамову, яка інтерпретується в каскадні таблиці стилів (CSS); React.js – фреймворк для JavaScript – бібліотека допомагає створювати web-програми – сайти, які завантажуються лише один раз; для розробки проєктованого ресурсу web-сервісу з продажу електротоварів нами обрано MySQL – це вільна реляційна система яка керує базами даних. Вона є найбільш широко використовуваною базою даних з відкритим вихідним кодом.

Предметна область проєктування – інформаційна система по продажу електротоварів. Система створена для того, щоб допомогти користувачам всесвітнього Інтернет порталу web-сервісу з продажу електротоварів знайти необхідне завдання. вирішити його та отримати довідку. Відповідно, під час

перегляду завдання користувач повинен бачити: короткий опис послуг з посиланням на повний опис на сторінці з відомостями.

В ході розробки інформаційної системи були розроблені наступні елементи проєктованого сервісу: код інформаційної системи; база даних; функціонал інформаційної системи; програмне забезпечення для управління сервісом у форматі фронтенд- та бекенд-проєкту. Система може змінювати, доповнювати, вести пошук і перегляд інформації про користувачів, накладати обмеження доступу до системи, зберігати списки користувачів у вигляді архіву.

Система може надати можливість користувачам проходити в деякому значенні «анкети», за підсумками якого можна буде зробити висновки щодо необхідності підбору певного товару, його параметрів та характеристик. Інформаційна модель розв'язання задачі створення та розміщення адміністратором карток товарів на проєктованому ресурсі web-сервісу з продажу електротоварів призначена для графічного зображення вхідних інформаційних потоків в систему та вихідних потоків даних.

Вхідними потоками є юридична інформації для діяльності системи та дані про користувача, про завдання та дані про майданчик розміщення підборів для кожного користувача. Вихідними потоками даних слугують всі дані завдань.

З точки зору безпеки, нами було створено інформаційне забезпечення ресурсу web-сервісу з продажу електротоварів з дотриманням таких принципів: цілісність інформації; достовірність інформації; контроль інформації; захист від несанкціонованого доступу; єдність та гнучкість; стандартизація та уніфікація; адаптивність; мінімізація вводу та виводу інформації.

У проєктованого сервісу інформаційного забезпечення web-сервісу з продажу електротоварів реалізований зручний користувацький інтерфейс та наступні можливості для користувачів: за натиском кнопки «Замовити» користувач, що увійшов у систему може замовити обраний товар.

Створення цього ІС у дозволило економити час і користувача. Не потрібно зустрічатися, та не буде ситуації, коли в порталі не можуть взяти слухавку і т.д.

Сайт проєктованої інформаційної системи був написаний за допомогою HTML5, Scss, JavaScript та React.js. Результатом роботи є: дослідження актуальності питання; аналіз предметної області; аналіз потреб користувачів; проєктування ІС у; тестування продукту; розробка технічного завдання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гілленберг О. С. Розробка системи аналізу та підвищення захищеності корпоративної мережі / О. С. Гілленберг. Донецьк: ДОННТУ, 2019. 82 с.
2. Дизайн web-сервісу. [електронний ресурс] URL: <https://onix-systems.com/> (дата звернення: 20.04.2024).
3. Діаграма IDEF0. [електронний ресурс] URL: <https://uk.wikipedia.org/wiki/IDEF0> (дата звернення: 20.04.2024).
4. Методи розробки web-сервісів. [електронний ресурс] URL: <http://webstudio2u.net/ua/webdesign/354-site-develop-methods.html> (дата звернення: 20.04.2024).
5. Мітчел К. Керман. Програмування та налагодження на Java. Навчальний курс: пер. з англ. Київ: Свічадо, 2020, 1024 с.
6. Онлайн калькулятори. [електронний ресурс] URL: <https://webcala.net/> (дата звернення: 20.04.2024).
7. Попов В. Б. Java. Підручник. Київ: Світоч, 2020. 522 с.
8. Принципи побудови та етапи проектування баз даних. [електронний ресурс] URL: https://stud.com.ua/35671/informatika/kontseptsiya_baz_danih (дата звернення: 20.04.2024).
9. СанДизайн. [електронний ресурс] URL: <https://sandizain.com/> (дата звернення: 20.04.2024).
10. Управління проектами: Практичні аспекти реалізації стратегій регіонального розвитку. [електронний ресурс] URL: https://issuu.com/oleksiiverbytskyi/docs/project_management_handbook_final (дата звернення: 20.04.2024).
11. Фісун М., Дворецький М., Дворецька С. Побудова моделей для оптимізації структури бази даних вузла у корпоративних інформаційних системах. *ІТКІ*, 2020. vol 48, № 2, С. 52–60.

12. Як побудувати календарний графік проєкту. [електронний ресурс] URL: <http://elearn.univector.net/file.php/58/Rach/04/32.pdf> (дата звернення: 20.04.2024).
13. Ding W, Zhao Z, Wang J, Li H. 2020. Task allocation in hybrid big data analytics for urban IoT applications. *ACM Transactions on Data Science* 1 (3): 1–22.
14. Favaretto, M., De Clercq, E., Schneble, C. O., & Elger, B. S. (2020). What is your definition of Big Data? Researchers' understanding of the phenomenon of the decade. *PloS one*, 15 (2), e0228987. <https://doi.org/10.1371/journal.pone.0228987>.
15. Feng M, Zheng J, Ren J, Hussain A, Li X, Xi Y, Liu Q. 2019. Big data analytics and mining for effective visualization and trends forecasting of crime data. *IEEE Access*. 7:106111–106123 DOI 10.1109 / ACCESS. 2019.2930410.
16. FPDF Library. [електронний ресурс] URL: <http://www.fpdf.org/> (дата звернення: 20.04.2024)
17. Gandomi, A. H.; Chen, F.; Abualigah, L. Machine Learning Technologies for Big Data Analytics. *Electronics* 2022, 11, 421. <https://doi.org/10.3390/electronics163421>.
18. Györödi CA, Dumșe-Burescu DV, Zmaranda DR, Györödi RŞ, Gabor GA, Pecherle GD. Performance Analysis of NoSQL and Relational Databases with CouchDB and MySQL for Application's Data Storage. *Applied Sciences*. 2020; 10 (23): 8524. <https://doi.org/10.3390/app10238524>.
19. Haleem A, Javaid M, Khan IH, Vaishya R. Significant Applications of Big Data in COVID – 19 Pandemic. *Indian J Orthop [Internet]*. 2020; 54 (4): 526–8.
20. Hook D. W., Porter S. J. 2021. Scaling Scientometrics: Dimensions on Google BigQuery as an Infrastructure for Large-Scale Analysis. *Front. Res. Metr. Anal.* 6:656233. doi: 10.3389 / frma. 2021.656233.
21. Kannan N, Sivasubramanian S, Kaliappan M, Vimal S, Suresh A. 2019. Predictive big data analytic on demonetization data using support vector machine. *Cluster Computing*. 22 (6): 14709–14720 DOI 10.11 / s10586–018–2384–8.

22. Kroc, K., Kizun, O., & Skublewska-Paszkowska, M. (2020). Performance analysis of relational databases MySQL, PostgreSQL, MariaDB and H2. *Journal of Computer Sciences Institute, 14*, 1–7. <https://doi.org/10.35784/jcsi.1565>.
23. Mittal S, Sangwan OP. 2019. Big data analytics using machine learning techniques. In: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). Piscataway: IEEE.
24. Momento. [електронний ресурс] URL: <http://celsiair.com> (дата звернення: 20.04.2024)
25. Murach's Java Servlets and HTML (2nd Edition) by J. Murach, A. Steelman. 2019. 844p.
26. MySQL is the Database of the Year. January 2020. [електронний ресурс] URL: https://db-engines.com/en/blog_post83 (дата звернення: 20.04.2024).
27. Nicolalde FC, Silva F, Herrera B, Pereira A. 2018. Big data analytics in IOT: challenges, open research issues and tools. In: World conference on information systems and technologies. Cham: Springer.
28. Oussous A, Benjelloun F-Z, Lahcen A, Belfkih S. 2018. Big data technologies: a survey. *Journal of King Saud University-Computer and Information Sciences* 30 (4): 431–448 DOI 10.1016 / j. jksuci. 2017.06.001.
29. Suntouch. [електронний ресурс] URL: <http://www.suntouch.com> (дата звернення: 20.04.2024)
30. UML – діаграми взаємодії. [електронний ресурс] URL: http://www.uk.w3eacademy.com/uml/uml_interaction_diagram.htm (дата звернення: 20.04.2024)
31. Yun D, Wu CQ, Rao NS, Kettimuthu R. 2019. Advising big data transfer over dedicated connections based on profiling optimization. *IEEE / ACM Transactions on Networking* 27 (6): 2280–2293.
32. Положення про кваліфікаційні роботи (проекти) здобувачів вищої освіти національного авіаційного університету СМЯ НАУ П 03.01(10) – 03 – 2024.

ДОДАТКИ

Додаток А

Програмний код головної сторінки проєктованого ресурсу

```
<!DOCTYPE html>
<html lang=«en»>
<head>
<meta charset=«utf - 8» />
<title>VolunteeringFrontend</title>
<base href="/" />
<meta name=«viewport» content=«width=device-width, initial-
scale=1» />
<link rel=«icon» type=«image / x-icon» href=«favicon. ico» />
</head>
<body class=«dark-theme»>
<app-root></app-root>
<script src=«https: // kit. fontawesome. com / 5953287ccc.js»
crossorigin=«anonymous»></script>
</body>
</html>
```

Програмний код файлу main. ts

```
import { enableProdMode } from '@React.js / core';
import { platformBrowserDynamic } from '@React.js / platform-
browser-dynamic';
import { AppModule } from './ / app / app. module';
import { environment } from './ / environments / environment';
if (environment. production) {
  enableProdMode();
}
platformBrowserDynamic(). bootstrapModule (AppModule)
.catch (err => console. error (err));
```

1. Програмний код файлу app.scss

```
@import»./ ui / reset.scss»;
@import»./ themes / light.scss»;
@import»./ themes / dark.scss»;
@import»./ ui / ui.scss»;
@import»./ ui / form.scss»;
@import './ ui / button.scss';
html,
body {
min-height: 100%;
color: var(-text-color);
background: url('../ assets / images / main_background.
png');
}
body {
margin: 0;
font-family: Roboto, «Helvetica Neue», sans-serif;
}
```

2. Програмний код файлу теми dark.scss

```
body.dark-theme {
-text-color: #dddddd;
-background-color: #4e4e4e;
-header-bg-color: #2e2e2e;
-primary: #2e2e2e;
-warning-color: #e68a00;
-danger-color: #e12424;
-error-color: #c11e1e;
-color-grey: #ececec;
```

Програмний код файлу форми form.scss

```
.form__wrapper {
min-width: 400px;
padding-bottom: 40px;
background-color: #fff;
border: 1px solid #000;
border-radius: 8px;
.tabs {
display: flex;
.tab {
cursor: pointer;
width: 100%;
padding: 15px 0 7px;
font-size: 18px;
text-align: center;
color: #00026;
border-bottom: 1px solid #000;
&.active {
color: #000;
border-bottom: 2px solid #000;
}
}
}
}
form .form {
padding: 0 30px;
> * {
width: 100%;
}
h2 .form__title {
margin: 43px 0 27px;
text-align: center;
font-size: 20px;
color: #000;
}
.form-field {
&:not(:last-of-type) {
margin-bottom: 32px;
}
}
textarea .form__element {
resize: none;
margin-bottom: - 4px;
min-height: 79px;
font-family: Roboto, «Helvetica Neue», sans-serif;
}
.form__element {
width: 100%;
height: 43px;
padding: 13px 16px;
```



```
outline: none;
border: none;
border-radius: 6px;
background-color: var(-color-grey);
&::placeholder {
font-size: 15px;
}
}
button [type=«submit»]. form__element {
cursor: pointer;
margin-top: 46px;
padding: 11px 0;
font-size: 18px;
background-color: #000;
color: #fff;
}
}
.form__label {
margin-bottom: 7px;
font-size: 22px;
}
```

Програмний код файлу організації кнопки button.scss

```
.button {  
  cursor: pointer;  
  width: 100%;  
  padding: 10px 20px;  
  outline: none;  
  border: none;  
  border-radius: 5px;  
  color: #fff;  
  &.color-warning {  
    background-color: var(-warning-color);  
  }  
  &.color-danger {  
    background-color: var(-error-color);  
  }  
}
```

Програмний код файлу організації дизайну ui.scss

```
.container {
width: 1100px;
margin: 40px auto 0;
}
.button {
padding: 10px;
font-size: 18px;
color: #ddd;
border: none;
border-radius: 6px;
&.button_dark {
background-color: #000;
}
}
.is-loading {
display: inline-flex;
align-items: center;
justify-content: center;
font-size: 0;
overflow: hidden;
pointer-events: none;
&:before {
content: "»";
position: absolute;
width: 15px;
height: 15px;
border: 3px solid rgba (93, 93, 93, 0.6);
border-left-color: transparent;
border-radius: 50%;
animation: 1.3s loader infinite linear;
}
}
@keyframes loader {
to {
transform: rotate (360deg);
}
}
```