

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сервісу для обліку продукції  
гуртовні «Electro»

Виконав: студент IV курсу, групи СНС-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Бунза В.П.  
(підпис) (прізвище та ініціали)

Керівник Гром'як Р.С.  
(підпис) (прізвище та ініціали)

Нормоконтроль Литвиненко Я.В.  
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.  
(підпис) (прізвище та ініціали)

Рецензент Чихіра І.В.  
(підпис) (прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

« 20 » червня 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Бундза Вадим Петрович  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-сервісу для обліку продукції гуртовні «Electro»

Керівник роботи Гром'як Роман Сильвестрович, канд. фіз-мат. наук, доцент кафедри КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 7 » лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 21 червня 2023р.

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ, Розділ 1. Аналіз літературних джерел для розробки сайту, 1.1 Найменування та область застосування, 1.2 Підстави для розробки, 1.3 Призначення розробки, 1.4 Вимоги до програми чи програмного виробу, 1.4.1 Вимоги до функціональних характеристик, 1.4.2 Вимоги до часових характеристик, 1.4.3 Вимоги до надійності, 1.4.4 Умови до експлуатації, 1.4.5 Вимоги до складу і параметрів технічних засобів, 1.4.6 Вимоги до інформаційної і програмної сумісності, 1.5 Вимоги до програмної документації, 1.6 Техніко-економічні показники, 1.7 Стадії та етапи розробки, 1.8 Порядок контролю та прийому, 1.9 Висновки до першого розділу, Розділ 2. Розробка технічного та робочого проекту, 2.1 Розробка загальної структури і варіантів використання програми, 2.2 Вибір бази даних, 2.3 Порівняння реляційних баз даних з не реляційними, 2.4 Опис структури бази даних програми, 2.5 Розробка серверної частини, 2.6 Розробка клієнтської частини, 2.7 Тестування програми і результати її виконання, 2.8 Висновки до другого розділу, Розділ 3. Безпека життєдіяльності, основи охорони праці, 3.1 Безпека життєдіяльності, 3.2 Охорона праці, 3.3 Висновок до третього розділу, Висновки, Перелік джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульний лист, 2. Мета, завдання. 3. Діаграма варіантів використання для програми «Electro», 4. Діаграма варіантів використання програми «Electro», 5. Структура колекцій, 6. Домашня сторінка, 7. Сторінка характеристики продукту, 8. Форма створення продукту, 9. Форма входу, 10. Головна форма обліку (Модератор/Адмін), 11. Сторінка обліку (продавець/адміністратор), 12. Висновки



## АНОТАЦІЯ

Розробка веб-сервісу для обліку продукції гуртовні «Electro» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Бундза Вадим Петрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // С. 61, рис. – 8, табл. – 3, слайди – 9, додат. – 0, бібліогр. – 22.

**Ключові слова:** node.js, javascript, mongodb, react, redux, saga, antd design

Дана робота присвячена розробці веб-сервісу для обліку продукції гуртовні "Electro". Метою проекту було створення ефективної системи, яка дозволяла швидко та точно виконувати операції з обліку товарів. У роботі були поставлені такі завдання: розробка бази даних для зберігання інформації про продукцію, створення серверної та клієнтської частин веб-сервісу, забезпечення безпеки даних та механізмів авторизації користувачів.

Для досягнення цих цілей використовувались сучасні технології програмування, такі як JavaScript, React, Node.js та MongoDB. Крім того, були використані популярні бібліотеки Redux, Saga та Antd Design для покращення функціональності та вигляду веб-сервісу.

В результаті роботи було розроблено функціональний та зручний у використанні веб-сервіс, який задовольняє потреби гуртовні "Electro" у системі обліку продукції. Розроблений сервіс дозволяє здійснювати операції з обліку товарів, зберігати та відстежувати всю необхідну інформацію про продукцію, забезпечує безпеку даних та механізми авторизації користувачів.

## ANNOTATION

Web Service Deveopment for Products Accounting in the "Electro" Wholesale Shop // Qualification work of the educational level "Bachelor" // Bundza Vadim Petrovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-41 // Ternopil, 2023 // P. 61, fig. – 8 , tabl. – 3, chair – 9, annexes – 0, references – 22.

**Keywords:** node.js, javascrip, mongodb, react, redux, saga, antd design

This work is devoted to the development of a web service for the accounting of the products of the "Electro" wholesaler. The goal of the project was to create an effective system that allowed quick and accurate performance of goods accounting operations. The following tasks were set in the work: development of a database for storing information about products, creation of server and client parts of the web service, ensuring data security and user authorization mechanisms.

Modern programming technologies such as JavaScript, React, Node.js and MongoDB were used to achieve these goals. In addition, the popular libraries Redux, Saga and Antd Design were used to improve the functionality and appearance of the web service.

As a result of the work, a functional and easy-to-use web service was developed that meets the needs of the "Electro" wholesale company in the product accounting system. The developed service allows you to carry out goods accounting operations, store and track all necessary information about products, ensures data security and user authorization mechanisms.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕСМА ( European Computer Manufacturers Association) – асоціація, діяльність якої присвячена стандартизації інформаційних і комунікаційних технологій.

AJAX (Asynchronous Javascript And Xml) – підхід до побудови користувацьких інтерфейсів веб-застосунків.

HTML (HyperText Markup Language) – стандартизована мова гіпертекстової розмітки документів у Всесвітній мережі.

XHTML (Extensible Hypertext Markup Language) – розширена мова гіпертекстової розмітки.

DOM (Document Object Model) – програмний інтерфейс, що дозволяє програмам і скриптам отримати доступ до вмісту HTML, XHTML і XML.

CSS (Cascading Style Sheets) – список інструкцій для браузера, – як і де відображати елементи веб-сторінки, написаний особливим чином.

DNS (Domain Name System) – комп'ютерна розподілена система для отримання інформації про домен.

FTP (File Transfer Protocol) – протокол передачі файлів.

SFTP (SSH File Transfer Protocol) – протокол прикладного рівня, призначений для копіювання і виконання інших операцій з файлами поверх надійного і безпечного з'єднання.

HTTP (Hyper Text Transfer Protocol) – протокол прикладного рівня передачі даних (у вигляді гіпертекстових документів).

HTT (Hypertext Transfer Protocol) – протокол передачі гіпертексту.

TCP/IP (Transmission Control Protocol/Internet Protocol) – протокол керування передачею.

PHP (Hypertext Preprocessor) – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.

CGI (Common Gateway Interface) – стандарт інтерфейсу, що використовується для зв'язку зовнішньої програми з веб-сервером

CVS (Concurrent Versions System) – програмний продукт, що відноситься до розряду систем управління версіями.

UML (Unified Modeling Language) – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

DFD (Data Flow Diagrams) – діаграми потоків даних.

СУБД (Системи управління базами даних) – сукупність програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують управління створенням і використанням баз даних.

DML (Data Manipulation Language) – сімейство комп'ютерних мов, що використовуються в комп'ютерних програмах або користувачами баз даних для отримання, вставки, видалення або зміни даних в базах даних.

## ЗМІСТ

ВСТУП .....	9
РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ДЛЯ РОЗРОБКИ САЙТУ .	12
1.1 Найменування та область застосування.....	12
1.2 Підстави для розробки .....	12
1.3 Призначення розробки.....	13
1.4 Вимоги до програми чи програмного виробу .....	15
1.4.1 Вимоги до функціональних характеристик .....	16
1.4.2 Вимоги до часових характеристик.....	16
1.4.3 Вимоги до надійності .....	18
1.4.4 Умови до експлуатації.....	19
1.4.5 Вимоги до складу і параметрів технічних засобів .....	20
1.4.6 Вимоги до інформаційної і програмної сумісності.....	21
1.5 Вимоги до програмної документації .....	21
1.6 Техніко-економічні показники.....	22
1.7 Стадії та етапи розробки.....	23
1.8 Порядок контролю та прийому .....	24
1.9 Висновки до першого розділу .....	25
РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ .....	27
2.1 Розробка загальної структури і варіантів використання програми....	28
2.2 Вибір бази даних.....	31
2.3 Порівняння реляційних баз даних з не реляційними.....	32
2.4 Опис структури бази даних програми.....	33
2.5 Розробка серверної частини .....	38
2.6 Розробка клієнтської частини .....	39
2.7 Тестування програми і результати її виконання .....	42
2.8 Висновки до другого розділу .....	50
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	53



3.1 Безпека життєдіяльності .....	53
3.2 Охорона праці .....	56
3.3 Висновок до третього розділу .....	57
ВИСНОВКИ.....	58
ПЕРЕЛІК ДЖЕРЕЛ .....	59

## ВСТУП

**Актуальність тематики:** В сучасному світі, де бізнес-процеси стають все більш складними і конкурентними, ефективне керування продукцією стає ключовим фактором успіху для багатьох компаній. Автоматизація процесів обліку продукції дозволяє зберегти час, ресурси та забезпечити точність даних, що має прямий вплив на ефективність підприємства. Тому розробка веб-сервісу для обліку продукції є актуальною і важливою темою в сучасному бізнес-середовищі.

**Мета роботи:** Метою цього проекту є створення ефективної та надійної системи обліку та керування продукцією для гуртовні "Electro". Наша мета полягає в розробці веб-сервісу, який допоможе автоматизувати процеси внесення, оновлення та видалення даних про товари, а також надасть зручний інтерфейс для користувачів.

### **Завдання:**

1. Розробити структуру бази даних, яка відповідає потребам обліку продукції.
2. Розробити серверну частину веб-сервісу, яка забезпечує зберігання та обробку даних про товари.
3. Розробити клієнтську частину веб-сервісу, яка надає зручний інтерфейс для користувачів та дозволяє здійснювати операції з обліком продукції.
4. Забезпечити безпеку даних та механізми авторизації та аутентифікації користувачів.
5. Провести тестування програмного продукту, щоб перевірити його функціональність та надійність.

Виконання цих завдань допоможе досягти мети проекту та надати компанії "Electro" ефективну систему обліку та керування продукцією, що покращить її бізнес-процеси та конкурентоспроможність.

Розробка веб-сервісу для обліку продукції гуртовні «Electro» є важливим завданням, яке дозволить автоматизувати процеси обліку, управління та контролю за продукцією. У цій роботі ми приступили до розробки програмного виробу, який буде надавати функціональність для ефективного ведення обліку продуктів, керування замовленнями та користувачами.

Перед початком розробки було проведено аналіз потреб і вимог до програмного виробу. Були визначені основні функціональні вимоги, вимоги до продуктивності, надійності та інші технічні вимоги. Була обрана база даних MongoDB для зберігання і керування даними про користувачів, продукти та замовлення.

Розробка включає створення серверної та клієнтської частини програми. В серверній частині були використані технології Node.js та Express.js для створення API, яке забезпечує обробку запитів, взаємодію з базою даних та надання необхідних даних клієнтській частині. В клієнтській частині були використані такі бібліотеки, як Redux, Saga та Antd Design, для керування станом додатку, асинхронної обробки запитів та створення користувацького інтерфейсу.

Після завершення розробки потрібно провести тестування програми, щоб перевірити її функціональність, стабільність та відповідність вимогам. Результати тестування показали, що програма працює належним чином і задовольняє встановлені вимоги.

В цій роботі була розроблена загальна структура програми, включаючи різні сторінки та форми для керування продуктами, замовленнями, користувачами та іншими аспектами обліку. Було описано структуру бази даних, включаючи таблиці для зберігання інформації про продукти, користувачів та інші дані.

У процесі розробки була врахована надійність, продуктивність та зручність використання програми. Були виконані вимоги до часових характеристик, надійності та інформаційної і програмної сумісності. Також

була розроблена технічна та робоча документація, яка описує процеси встановлення, використання та експлуатації програми.

В цілому, розробка веб-сервісу для обліку продукції гуртовні «Electro» є важливим кроком у поліпшенні ефективності та керованості бізнесу. Розроблений програмний виріб надає необхідні функції для обліку продукції, керування замовленнями та користувачами, сприяючи оптимізації робочих процесів та підвищенню продуктивності.

## РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ДЛЯ РОЗРОБКИ САЙТУ

### 1.1 Найменування та область застосування

Область застосування веб-сервісу для обліку продукції може охоплювати наступні аспекти:

1. Управління складом: Веб-сервіс може допомогти вам відстежувати наявність продуктів на складі, контролювати рівень запасів, отримувати сповіщення про недостатність товарів та оптимізувати управління запасами.

2. Замовлення та постачання: Ви можете використовувати веб-сервіс для розміщення замовлень у постачальників, відстеження стану замовлень та отримання інформації про відвантаження та поставки товарів.

3. Облік продукції: Система може зберігати детальну інформацію про кожен продукт, включаючи назву, опис, категорію, ціну, кількість тощо. Ви також можете відстежувати продажі, повернення та інші фінансові показники.

4. Генерація звітів: Веб-сервіс може надавати зручні засоби для генерації звітів, наприклад, про прибуток, продажі за період, популярність товарів тощо. Це допоможе вам аналізувати продуктивність вашого бізнесу та приймати обґрунтовані рішення.

5. Автоматизація процесів: Веб-сервіс може спростити рутинні завдання, такі як створення замовлень, генерація звітів, оновлення запасів тощо, що заощадить час та зусилля вашої команди.

### 1.2 Підстави для розробки

Підстави для розробки веб-сервісу для обліку продукції гуртовні «Electro» можуть включати наступні фактори:

1. Автоматизація та ефективність: Ручний облік продукції може бути складним і часоємним процесом. Розробка веб-сервісу дозволить

автоматизувати багато операцій, таких як ведення записів, генерація звітів і керування запасами, що пришвидшить процес та зменшить можливість помилок.

2. Точність та точність: Веб-сервіс для обліку продукції дозволить точно відстежувати кількість та стан товарів на складі. Це допоможе уникнути проблем з надлишком або нестачею товарів, забезпечуючи точність і надійність обліку.

3. Зручний доступ до даних: За допомогою веб-сервісу ви зможете отримати зручний доступ до даних про вашу продукцію з будь-якого місця та в будь-який час. Ви зможете відстежувати запаси, контролювати замовлення та отримувати актуальну інформацію про стан бізнесу.

4. Аналітика та прийняття рішень: Розробка веб-сервісу для обліку продукції дозволить нам збирати дані та генерувати звіти, що допоможе вам аналізувати продажі, тренди та ефективність продукції. Ця аналітична інформація допоможе вам приймати обґрунтовані рішення щодо управління запасами, маркетингових стратегій та розвитку бізнесу.

5. Створення конкурентної переваги: Використання сучасних технологій та розробка веб-сервісу може надати нашій гуртовні конкурентну перевагу. Зручний і ефективний облік продукції може поліпшити наші бізнес-процеси, забезпечити більшу задоволеність клієнтів та допомогти вам вибитися на ринку.

Ці підстави показують, як розробка веб-сервісу для обліку продукції може принести численні переваги вашій гуртовні «Electro» і поліпшити її ділові показники.

### **1.3 Призначення розробки**

Призначення розробки веб-сервісу для обліку продукції гуртовні «Electro» включає наступні аспекти:

1. **Управління інвентарем:** Розробка веб-сервісу дозволить ефективно керувати і контролювати ваш інвентар (продукцію). Ви зможете відстежувати наявність товарів на складі, контролювати рівень запасів, отримувати сповіщення про недостатність або перевищення запасів, а також планувати замовлення та постачання.

2. **Облік продукції:** Розробка веб-сервісу дозволить вести детальний облік продукції, включаючи інформацію про кожен продукт, таку як назва, опис, категорія, ціна, кількість тощо. Ви зможете створювати нові записи про продукти, редагувати існуючі записи, відстежувати продажі, повернення, а також отримувати звіти та статистику про продуктивність продукції.

3. **Замовлення та постачання:** Веб-сервіс дозволить ефективно керувати процесом замовлення та постачання. Ви зможете приймати замовлення від клієнтів, створювати замовлення до постачальників, відстежувати статус замовлень, отримувати інформацію про відвантаження та поставки товарів, а також контролювати терміни доставки.

4. **Звітність та аналітика:** Розроблений веб-сервіс надасть вам засоби для генерації різноманітних звітів та аналітичної інформації. Ви зможете отримувати звіти про продажі, прибуток, популярність продуктів, тенденції споживання, аналізувати даний потік, ідентифікувати сильні та слабкі сторони бізнесу та приймати обґрунтовані рішення щодо стратегії та розвитку.

5. **Зручний доступ та спільна робота:** Веб-сервіс забезпечить зручний доступ до інформації про продукцію з будь-якого місця та в будь-який час. Ви зможете спільно працювати з командою, надавати доступ до необхідної інформації різним користувачам та рівням доступу, а також забезпечувати безпеку та конфіденційність даних.

Отже, призначення розробки веб-сервісу для обліку продукції полягає в поліпшенні управління інвентарем, обліку продукції, замовленнях та постачанні, забезпеченні звітності та аналітики, а також у забезпеченні зручного доступу та спільній роботі.

## 1.4 Вимоги до програми чи програмного виробу

Вимоги до програми або програмного виробу для веб-сервісу обліку продукції гуртовні «Electro» можуть включати наступні аспекти:

1. Функціональні вимоги:
  - Реєстрація та авторизація користувачів з різними рівнями доступу (адміністратор, менеджер, співробітник).
  - Можливість створення, редагування та видалення записів про продукцію, включаючи назву, опис, категорію, ціну, кількість тощо.
  - Функціонал для керування інвентарем, включаючи відстеження запасів, контроль рівня запасів, сповіщення про недостатність або перевищення запасів.
  - Можливість створювати замовлення, відстежувати їх статус, контролювати поставки і відвантаження товарів.
  - Генерація звітів та аналітичної інформації про продажі, прибуток, популярність продуктів та інші показники продуктивності.
  - Забезпечення безпеки даних та конфіденційності.
2. Нефункціональні вимоги:
  - Зручний та інтуїтивно зрозумілий інтерфейс користувача, який дозволяє легко навігувати по системі та виконувати необхідні завдання.
  - Висока продуктивність та швидкість реакції системи для ефективного оброблення великої кількості даних.
  - Масштабованість системи, щоб забезпечити її ефективну роботу при зростанні обсягу даних та користувачів.
  - Сумісність з різними веб-браузерами та пристроями, щоб забезпечити доступність для широкого кола користувачів.
  - Надійність та стабільність системи, щоб уникнути втрати даних або недоступності сервісу.
3. Технічні вимоги:



- Використання веб-технологій для розробки, таких як HTML, CSS, JavaScript, а також мов програмування для бекенду, наприклад, Python, Java або PHP.
- Використання бази даних для зберігання даних про продукцію та іншу необхідну інформацію.
- Застосування методів захисту даних, таких як шифрування, автентифікація користувачів та забезпечення безпеки мережі.

Ці вимоги визначають основні функції, характеристики та технічні параметри, які потрібно враховувати при розробці програмного виробу для обліку продукції вашої гуртовні «Electro». Варто також враховувати специфічні потреби бізнесу та конкретні вимоги, що можуть виникнути в процесі консультації зі зацікавленими сторонами та користувачами.

#### **1.4.1 Вимоги до функціональних характеристик**

Програма передбачає такий функціонал:

1. Авторизацію.
2. Реєстрацію.
3. Облік товару.
4. Зберігання даних про товари в БД.
5. Зберігання даних про користувача в БД.
6. Роботу з хешом.
7. Графічний інтерфейс.

#### **1.4.2 Вимоги до часових характеристик**

Вимоги до часових характеристик веб-сервісу для обліку продукції гуртовні «Electro» можуть включати наступні аспекти:

1. Швидкодія: Система повинна забезпечувати швидку відповідь на запити користувачів та ефективну обробку даних. Це означає, що час відповіді системи на запити користувачів, наприклад, створення запису про продукт чи замовлення, має бути мінімальним і не перевищувати прийняті межі продуктивності.

2. Завантаження системи: Веб-сервіс повинен бути здатним оптимально працювати при різних навантаженнях, включаючи пікові навантаження. Система повинна бути розрахована на обробку одночасних запитів великої кількості користувачів та зберігання даних про продукцію з високою продуктивністю.

3. Час відновлення: У випадку виникнення помилок або відмови системи, важливо, щоб веб-сервіс мав мінімальний час відновлення. Це означає, що система повинна бути стійкою та здатною автоматично відновлюватися після відмови або непередбачуваної ситуації, щоб мінімізувати вплив на роботу гуртовні.

4. Резервне копіювання та відновлення даних: Система повинна мати механізми регулярного резервного копіювання даних про продукцію та можливість швидкого відновлення даних в разі втрати або пошкодження. Це забезпечить захист від втрати цінної інформації та можливість відновлення роботи системи у найкоротший термін.

5. Часові обмеження: У випадку наявності часових обмежень для певних операцій, наприклад, підтвердження замовлення в термін 24 години, система повинна забезпечувати відповідну функціональність та автоматизацію для виконання цих обмежень вчасно.

Ці вимоги до часових характеристик допоможуть забезпечити ефективну та швидку роботу веб-сервісу для обліку продукції гуртовні «Electro» та задовольнити вимоги бізнесу для користувачів.

А зокрема програма повинна викувати і обробляти запит на авторизацію чи реєстрацію за 2 сек., а запит на додавання чи видалення поля з бази даних

до 1 сек.

### **1.4.3 Вимоги до надійності**

Вимоги до надійності веб-сервісу для обліку продукції гуртовні "Electro" можуть включати наступні аспекти:

1. **Стабільність системи:** Система повинна бути стійкою та надійною, щоб уникнути непередбачуваних збоїв, відмов або зупинок. Це означає, що програмний виріб має бути ретельно протестованим та підтримувати стабільну роботу навіть при великому навантаженні або несприятливих умовах.

2. **Резервне копіювання та відновлення:** Система повинна мати механізми для регулярного резервного копіювання даних та можливість швидкого відновлення в разі втрати або пошкодження. Це забезпечить захист даних і можливість відновлення роботи системи у разі виникнення проблем.

3. **Захист даних:** Веб-сервіс повинен забезпечувати високий рівень безпеки та конфіденційності даних. Це означає застосування методів шифрування, автентифікації користувачів, контролю доступу та інших заходів для запобігання несанкціонованому доступу до інформації.

4. **Відновлення після відмови:** У випадку виникнення збоїв або відмов системи, важливо мати механізми для швидкого відновлення роботи. Це може включати автоматичне відновлення з резервних копій, використання резервних серверів або інші заходи для мінімізації перерв у роботі системи.

5. **Моніторинг та журналювання:** Система повинна мати функціонал для моніторингу та журналювання подій, помилок та активності. Це дозволить виявляти проблеми, відстежувати дії користувачів та забезпечувати можливість аналізу для поліпшення надійності та безпеки системи.

6. **Резервування та масштабованість:** Система повинна бути розроблена з урахуванням можливості резервування компонентів та

масштабованості, щоб забезпечити надійну роботу під час зростання обсягу даних та користувачів.

Ці вимоги до надійності допоможуть забезпечити стабільну та безперебійну роботу веб-сервісу для обліку продукції гуртовні «Electro» та забезпечити захист інформації та даних бізнесу.

Програма фільтрує вхідні дані на коректність для забезпечення стабільної роботи, та працює тільки з авторизованими користувачами чи адміністратором.

#### **1.4.4 Умови до експлуатації**

Розглянемо умови експлуатації веб-сервісу:

1. Комп'ютерне середовище: Система повинна працювати на комп'ютерах з підтримкою сучасних операційних систем, таких як Windows, macOS або Linux. Клієнтські комп'ютери повинні мати встановлені сумісні веб-браузери, що дозволяють користувачам взаємодіяти з веб-сервісом.

2. Інтернет-з'єднання: Для користування веб-сервісом потрібне стабільне та надійне підключення до Інтернету. Це може бути провідне або бездротове з'єднання з доступом до Інтернету.

3. Резервне копіювання: Рекомендується регулярно робити резервне копіювання даних, щоб уникнути втрати цінної інформації. Розробка механізмів автоматичного резервного копіювання та забезпечення доступу до них може бути корисною для експлуатації системи.

4. Технічна підтримка: Надання технічної підтримки користувачам є важливим аспектом експлуатації веб-сервісу. Варто мати механізми для звернення до технічної підтримки, наприклад, через електронну пошту, телефон або онлайн-чат, для вирішення запитів, проблем або питань, які виникають у користувачів.

5. Оновлення та підтримка: Розробка системи повинна передбачати можливість оновлення програмного забезпечення та його підтримки в разі виявлення помилок, вдосконалення функціональності або змін вимог користувачів. Важливо мати процедури для впровадження оновлень та забезпечення безперебійної роботи системи під час оновлень.

6. Безпека: Забезпечення безпеки системи та захисту від несанкціонованого доступу до даних є важливою умовою експлуатації. Потрібно встановлювати механізми автентифікації, контролю доступу, шифрування даних та інші заходи безпеки для забезпечення конфіденційності та цілісності інформації.

Ці умови експлуатації допоможуть забезпечити правильне функціонування та доступність веб-сервісу для обліку продукції гуртовні у відповідності до вимог вашого бізнесу та користувачів.

#### **1.4.5 Вимоги до складу і параметрів технічних засобів**

Для використання веб-сервісу обліку продукції гуртовні "Electro" потрібний наступний склад технічних засобів:

1. Монітор: Монітор з діагоналлю не менше 15 дюймів, який забезпечить зручний перегляд та взаємодію з веб-сервісом.
2. Клавіатура та мишка: Стандартна клавіатура та мишка, які дозволять користувачеві вводити дані та взаємодіяти з інтерфейсом веб-сервісу.
3. Персональний комп'ютер: IBM-сумісний персональний комп'ютер з наступними мінімальними характеристиками:
  - Процесор: Intel Pentium 4 або еквівалентний процесор.
  - Оперативна пам'ять: Мінімум 1 ГБ оперативної пам'яті, що забезпечить відповідну продуктивність системи при використанні веб-сервісу.
  - Жорсткий диск: Мінімум 10 ГБ пам'яті на жорсткому диску для зберігання програмного забезпечення та даних.

### **1.4.6 Вимоги до інформаційної і програмної сумісності**

Основні вимоги до інформаційної і програмної сумісності вашого веб-сервісу для обліку продукції гуртовні "Electro" включають наступне:

1. Мова програмування: Вихідні коди програми повинні бути реалізовані мовою JavaScript стандарту ECMAScript 2020 (ESMA Script 2020). Це забезпечить сумісність програмного виробу з сучасними браузерами та іншими JavaScript-оточеннями.

2. Середовище розробки: Вимагається використання середовища Qt Creator версії не нижче 4.7 для розробки програмного виробу. Qt Creator є інтегрованою середовищем розробки (IDE), яке забезпечує зручний інтерфейс та інструменти для програмування на мові JavaScript.

3. Операційна система: Програма повинна підтримуватися операційною системою сімейства Windows версії не нижче Windows XP. Це означає, що програмний виріб має бути сумісним з цими версіями операційної системи і працювати на них без проблем.

4. Вимоги до захисту інформації і програми: За вашими вказівками вимоги до захисту інформації і програми не пред'являються. Однак, рекомендується враховувати основні принципи безпеки програмного забезпечення, такі як захист від несанкціонованого доступу, використання шифрування для захисту конфіденційної інформації та резервне копіювання даних для запобігання втраті даних.

### **1.5 Вимоги до програмної документації**

Вимоги до програмної документації для веб-сервісу обліку продукції включають наступні пункти:

1. Інструкція інсталяції програми: Документ, який надає поетапні вказівки щодо встановлення та налаштування веб-сервісу. Ця інструкція має

включати вимоги до системних вимог, кроки встановлення та налаштування, інструкції щодо введення необхідних параметрів та підключення до бази даних, якщо така є.

2. Загальні відомості про можливості програми: Цей документ повинен містити опис функціональності та можливостей веб-сервісу обліку продукції. Він може включати інформацію про основні функції системи, зв'язок з базою даних, можливості реєстрації користувачів, роботу зі звітами та інші ключові функції.

3. Інструкція з експлуатації: Цей документ надає детальні вказівки щодо використання веб-сервісу. Він може містити опис інтерфейсу користувача, пояснення процесу роботи з системою, кроки створення, редагування та видалення даних, якщо це застосовно, а також інструкції щодо управління користувачами та доступом до системи.

Ці документи є важливою частиною програмної документації і допоможуть користувачам ефективно встановити, налаштувати та використовувати веб-сервіс обліку продукції гуртовні. Вони забезпечать зрозумілість та належне використання системи користувачами.

## **1.6 Техніко-економічні показники**

Розрахунок економічної ефективності і вартості розробки програмного продукту не проводиться. Приблизне число використань розробленої програми в рік: 200-250 раз.

## 1.7 Стадії та етапи розробки

### 1. Стратегічна стадія:

- Аналіз потреб користувачів і ринку: Визначення основних потреб цільової аудиторії, дослідження конкурентного середовища, оцінка ринкових можливостей та визначення унікальних особливостей продукту.
- Встановлення мети і цілей: Формулювання конкретних цілей, які потрібно досягти з розробкою програмного продукту, наприклад, покращення продуктивності, збільшення задоволеності користувачів, підвищення прибутковості тощо.
- Стратегія впровадження та підтримки: Визначення плану впровадження продукту на ринку, стратегії маркетингу, підтримки та обслуговування користувачів після випуску продукту.

### 2. Функціональні вимоги:

- Визначення основних функцій: Встановлення переліку функцій, які повинні бути реалізовані в програмному продукті, враховуючи потреби користувачів і вимоги ринку.
- Встановлення пріоритетів: Визначення важливості кожної функції для задоволення потреб користувачів і досягнення цілей проекту.
- Формалізація вимог: Описання функціональних вимог у формі документу, який може використовуватись для подальшої розробки та тестування продукту.

### 3. Аналіз:

- Архітектура та дизайн: Розробка загальної архітектури програмного продукту, включаючи вибір технологій, розподіл компонентів і модулів, встановлення структури бази даних і логіки програми.
- Розробка інтерфейсу користувача: Створення зручного і ефективного інтерфейсу, який відповідає потребам користувачів і забезпечує зручну навігацію та взаємодію з програмою.



- Деталізація вимог: Уточнення та деталізація вимог, розробка сценаріїв використання та встановлення особливостей функцій програмного продукту.

#### 4. Тестування:

- Планування тестування: Визначення стратегії тестування, створення тестових планів та сценаріїв, вибір необхідних тестових інструментів і оточення.

- Виконання тестування: Проведення тестів згідно з запланованими сценаріями, виявлення помилок, аналіз їх причин і виправлення.

- Приймальне тестування: Виконання тестів користувачами або замовниками для перевірки відповідності продукту вимогам і очікуванням.

### **1.8 Порядок контролю та прийому**

Порядок контролю та прийому розробленого програмного продукту може включати такі етапи:

1. Модульне тестування: Перевірка окремих модулів програми на відповідність встановленим вимогам. Це може включати запуск тестових сценаріїв, перевірку правильності обробки даних, виявлення та виправлення помилок.

2. Інтеграційне тестування: Перевірка взаємодії різних модулів програми та їхню спроможність працювати разом. Це включає перевірку передачі даних між модулями, правильну обробку взаємодії та ідентифікацію та виправлення можливих проблем.

3. Системне тестування: Виконання повних перевірок системи в цілому, включаючи всі її функціональні та нефункціональні аспекти. Це може включати тестування продуктивності, навантаження, безпеки та інших параметрів.

4. **Приймальне тестування:** Проведення тестування замовником або користувачами з метою перевірки, чи відповідає програмний продукт їхнім потребам і вимогам. Це дозволяє замовнику остаточно переконатися, що програма працює правильно та задовольняє їх очікування.

5. **Контроль якості:** Виконання всіх необхідних процедур контролю якості для перевірки, чи відповідає програмний продукт встановленим стандартам і нормам якості.

6. **Прийомка:** Остаточне затвердження і прийомка розробленого програмного продукту замовником або відповідними сторонами. На цьому етапі здійснюється остаточне формальне затвердження і передача програмного продукту в експлуатацію або використання.

Кожен етап контролю та прийому повинен бути документованим, а виявлені помилки або несуперечності повинні бути виправлені перед переходом до наступного етапу. Контроль та прийомка є важливими етапами процесу розробки програмного продукту для забезпечення якості і відповідності вимогам замовника.

## **1.9 Висновки до першого розділу**

Розробка веб-сервісу для обліку продукції гуртовні «Electro» призначена для автоматизації процесів обліку та управління продукцією компанії. Вона базується на потребі компанії в ефективному та точному обліку своєї продукції, збільшенні ефективності бізнес-процесів та покращенні керівництва прийняттям рішень.

Веб-сервіс повинен бути реалізований мовою JavaScript стандарту ESMA Script 2020. Для розробки використовується середовище QtCreator версії не нижче 4.7. Передбачається використання операційної системи сімейства Windows версії не нижче Windows XP.

Веб-сервіс повинен працювати швидко та ефективно, з мінімальними затримками при обробці даних та відгуком на запити користувачів, також повинен бути стабільним і надійним, забезпечуючи безперебійну роботу і запобігаючи втраті даних або несправності системи.

Він повинен бути доступним для користувачів з будь-якого пристрою з підтримкою веб-браузера та інтернет-підключенням. Розроблений веб-сервіс повинен бути легко встановлюватись та використовуватись без особливих технічних знань.

Для використання веб-сервісу необхідні такі технічні засоби: монітор з діагоналлю не менше 15", клавіатура, мишка, IBM-сумісний персональний комп'ютер з процесором Intel Pentium 4, 1 ГБ оперативної пам'яті та 10 ГБ пам'яті ЖД.

Вихідні коди програми повинні бути реалізовані мовою JavaScript стандарту ESMA Script 2020. В якості середовища розробки програми використовується середовище QtCreator версії не нижче 4.7. Системні програмні засоби та утиліти повинні забезпечуватися операційною системою сімейства Windows версії не нижче Windows XP.

Після завершення розробки веб-сервісу необхідно підготувати інструкцію з інсталяції програми, загальні відомості про можливості програми та інструкцію з експлуатації.

Контроль та прийом розробленого веб-сервісу включає модульне та системне тестування, перевірку відповідності вимогам і технічним характеристикам, а також виконання встановлених функцій та відповідність інструкціям та документації. Після успішного проходження контролю та прийому веб-сервіс вважається готовим до використання.

## РОЗДІЛ 2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

Розробка технічного та робочого проекту є важливим етапом в процесі створення веб-сервісу для обліку продукції гуртовні «Electro». Цей етап передбачає детальне визначення технічних аспектів розробки, проектування архітектури системи, розробку інтерфейсу користувача та встановлення вимог до функціональності програмного продукту. Нижче наведено загальний опис розробки технічного та робочого проекту:

### 1. Технічний проект:

- **Визначення архітектури системи:** Розробка архітектури включає визначення компонентів системи, їх взаємозв'язків та інтерфейсів. Також визначаються вимоги до масштабованості, безпеки та продуктивності системи.
- **Вибір технологій:** Визначення необхідних технологій для реалізації веб-сервісу, таких як мова програмування, фреймворки, бази даних, сервери тощо. Вибір технологій повинен відповідати вимогам проекту та забезпечувати ефективну та надійну роботу системи.
- **Проектування бази даних:** Розробка схеми бази даних, включаючи таблиці, структури даних та зв'язки між ними. Визначення правил цілісності даних та оптимізації запитів для забезпечення швидкого доступу до інформації.
- **Розробка інтерфейсу користувача:** Проектування зручного та інтуїтивно зрозумілого інтерфейсу користувача, який дозволяє здійснювати всі необхідні операції з обліку продукції. Враховуються принципи дизайну, взаємодії та навігації, щоб забезпечити зручну роботу для користувачів.
- **Розробка алгоритмів та функціональності:** Розробка необхідних алгоритмів та функцій для обліку продукції, включаючи додавання, зміну, видалення записів, обчислення статистики, генерацію звітів тощо. Враховуються вимоги до функціональності, ефективності та точності обробки даних.

### 2. Робочий проект:

- Розподіл завдань та графік робіт: Визначення окремих завдань, які потрібно виконати в рамках розробки, і розподіл їх між командою розробників. Створення графіка робіт, який визначає часові рамки для кожного етапу розробки.
- Реалізація програмного продукту: Розробка програмного коду відповідно до вимог і специфікацій, включаючи реалізацію функцій, модулів, інтерфейсів та логіки програми. Застосування найкращих практик програмування та тестування для забезпечення якості коду.
- Тестування та налагодження: Проведення різноманітних тестів для перевірки функціональності, коректності та продуктивності програмного продукту. Виявлення та виправлення помилок та недоліків. Налаштування системи для забезпечення її стабільної та надійної роботи.
- Документування: Підготовка технічної документації, включаючи інструкцію інсталяції, опис функціональності, документацію API, опис бази даних та іншу необхідну документацію для зручного використання і підтримки програмного продукту.

Розробка технічного та робочого проекту є важливим етапом, що дозволяє визначити всі технічні та функціональні аспекти програмного продукту, розподілити завдання між командою розробників та забезпечити якісну реалізацію та тестування продукту.

## **2.1 Розробка загальної структури і варіантів використання програми**

Розробка загальної структури програми включає в себе визначення основних компонентів, модулів та їх взаємозв'язків. Це допомагає організувати програму таким чином, щоб вона була логічно структурована та мала зрозумілий потік виконання.

Виходячи із технічного завдання, можна сформулювати варіанти використання програми для діючого суб'єкта – продавця. Варіанти використання включають наступне:

1. Авторизація:
  - Користувач запускає програму.
  - Система відображає форму авторизації.
  - Користувач вводить свої облікові дані (логін та пароль).
  - Система перевіряє правильність введених даних.
  - Якщо авторизація успішна, система надає доступ до функціоналу програми.
2. Виведення таблиці товарів:
  - Після успішної авторизації користувач отримує доступ до головного меню програми.
    - Користувач обирає опцію для перегляду таблиці товарів.
    - Система виводить таблицю, яка містить інформацію про наявні товари.
      - Користувач може переглядати дані про кожен товар, такі як назва, кількість, ціна тощо.
3. Видалення запису:
  - Користувач в таблиці товарів обирає конкретний запис, який потрібно видалити.
    - Система підтверджує намір користувача видалити вибраний запис.
    - Якщо користувач підтверджує видалення, система виконує відповідну операцію і оновлює таблицю товарів.
4. Редагування запису:
  - Користувач в таблиці товарів обирає конкретний запис, який потрібно відредагувати.
    - Система відкриває форму редагування, де користувач може змінити дані про товар.

- Після внесення змін користувач зберігає оновлені дані.
  - Система перевіряє правильність введених даних та оновлює таблицю товарів.
5. Додавання нового запису:
- Користувач у головному меню обирає опцію для додавання нового запису.
  - Система відкриває форму для введення даних про новий товар.
  - Користувач вводить інформацію про товар, таку як назва, кількість, ціна тощо.
  - Після введення даних користувач зберігає новий запис.
  - Система перевіряє правильність введених даних та оновлює таблицю товарів.

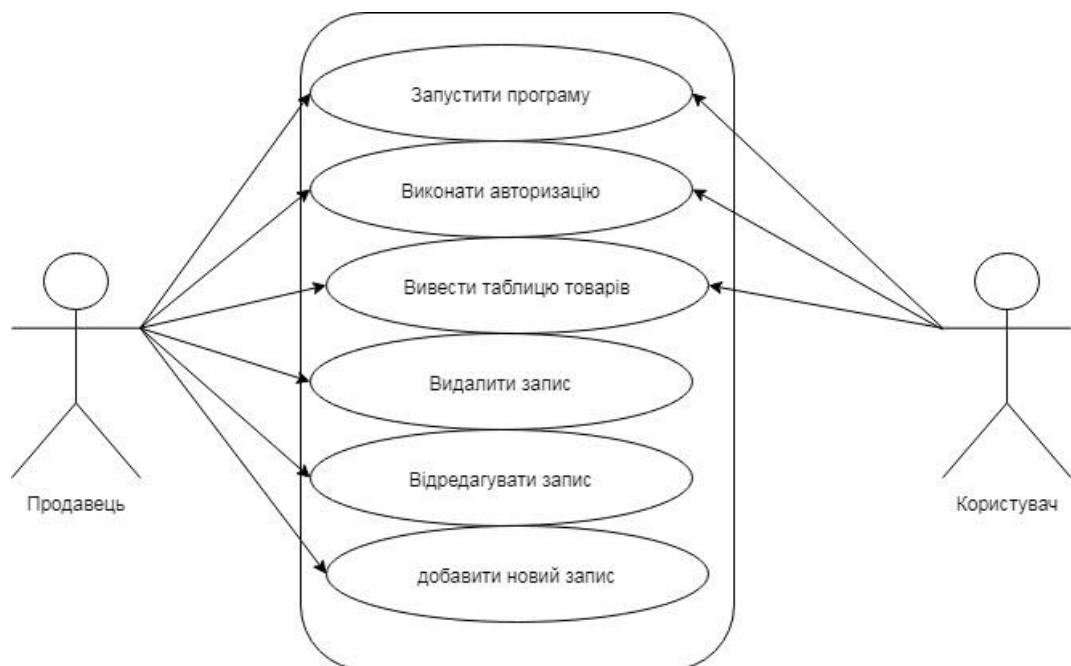


Рисунок 2.1 – Діаграма варіантів використання для програми «Electro»

На рисунку 2.1 зображено діаграму варіантів використання.

Також на етапі аналізу для моделювання варіантів використання, що мають декілька сценаріїв доцільно розробити UML-діаграми дій.



Рисунок 2.2 – Діаграма варіантів використання програми “Electro”

Цей тип діаграм демонструє керівні потоки від одних дій до інших, як показано на рисунку 2.2.

## 2.2 Вибір бази даних

Вибір бази даних MongoDB для розробки веб-сервісу для обліку продукції має свої переваги та особливості. Основні причини вибору MongoDB можуть включати:

1. Гнучкість структури даних: MongoDB використовує модель документів, що дозволяє зберігати дані у вигляді JSON-подібних документів. Це дозволяє вам легко змінювати структуру даних, додавати або видаляти поля без переробки всієї бази даних.
2. Горизонтальне масштабування: MongoDB підтримує горизонтальне масштабування, що дозволяє розподіляти дані на кілька серверів. Це дозволяє



підтримувати високу доступність та продуктивність в разі зростання обсягу даних.

3. Підтримка запитів: MongoDB має потужну систему запитів, включаючи можливість використовувати запити за полями, текстовий пошук, агрегаційні запити та інші операції. Це дозволяє ефективно взаємодіяти з базою даних та отримувати потрібну інформацію.

4. Висока продуктивність: MongoDB забезпечує швидкий доступ до даних та оптимізовану роботу з ними. Вона використовує кешування даних, індексацію та інші методи для підвищення продуктивності операцій з базою даних.

5. Спрощена реплікація та резервне копіювання: MongoDB має вбудовану підтримку реплікації, що дозволяє створювати дублікати бази даних для забезпечення високої доступності та відновлення в разі відмови. Вона також надає можливість резервного копіювання та відновлення даних.

### **2.3 Порівняння реляційних баз даних з не реляційними**

Порівняння реляційних баз даних (РБД) з нереляційними базами даних (NoSQL) включає декілька аспектів, які можуть впливати на вибір бази даних для вашого проекту. Ось деякі ключові відмінності між цими двома типами баз даних.

**Структура даних:** РБД використовують таблиці зі зв'язками між ними, що дозволяє зберігати структуровані дані з певною схемою. Нереляційні бази даних дозволяють зберігати неструктуровані або поліструктуровані дані у вигляді документів, ключ-значення, стовпчиків або графів.

**Масштабованість:** Нереляційні бази даних зазвичай легше масштабувати горизонтально, що дозволяє розподіляти дані на багато серверів і забезпечувати високу доступність та продуктивність. РБД, з іншого боку, частіше масштабуються вертикально, що означає збільшення ресурсів одного сервера.

Гнучкість структури даних: Нереляційні бази даних зазвичай мають більшу гнучкість у зміні структури даних, оскільки вони не вимагають строго дотримання схеми. РБД мають фіксовану схему, і зміна цієї схеми може бути складною та вимагати великої кількості змін у базі даних.

Підтримка операцій: РБД надають розширені можливості для виконання складних операцій, таких як з'єднання таблиць, агрегація, транзакції тощо. Нереляційні бази даних часто мають обмежену підтримку для цих операцій і зазвичай фокусуються на простоті та швидкості доступу до даних.

Використання в конкретних сценаріях: Вибір між РБД та нереляційними базами даних залежить від конкретних потреб вашого проекту. РБД, такі як MySQL, PostgreSQL або Oracle, зазвичай використовуються для додатків зі складними зв'язками даних, які вимагають транзакційної підтримки. Нереляційні бази даних, такі як MongoDB, Cassandra або Redis, частіше використовуються для проектів, де головними пріоритетами є швидкість, масштабованість та гнучкість.

## **2.4 Опис структури бази даних програми**

Структура бази даних програми може бути організована відповідно до вимог проекту. Оскільки ми вибрали базу даних MongoDB, наведемо загальний опис структури бази даних, який може бути використаний як початкова точка для нашого проекту. Проте, варто враховувати, що конкретна структура бази даних буде залежати від функціональних вимог нашого веб-сервісу.

MongoDB базується на моделі даних «документ-орієнтованої», де дані представлені у вигляді JSON-подібних документів. Основними складовими структури бази даних MongoDB є колекції (collections) і документи (documents).

Колекції є контейнерами для групування пов'язаних документів. Вони можуть бути порівняні з таблицями в реляційних базах даних. Кожна колекція має унікальне ім'я і може містити документи різної структури.

Документи є основними об'єктами даних в MongoDB. Вони представляються у вигляді JSON-подібних об'єктів, що містять ключі і значення. Кожен документ може мати різну структуру, але має унікальне поле «\_id», яке служить як унікальний ідентифікатор документа в колекції.

Поля в документах відображають різні властивості об'єктів. Вони можуть містити різні типи даних, такі як рядки, числа, масиви, вкладені документи тощо. Поля можуть також мати індекси, які полегшують швидкий доступ до даних.

MongoDB надає потужну мову запитів, яка дозволяє здійснювати пошук, оновлення, вставку і видалення документів в колекціях. Ви можете використовувати різні оператори запитів, умови, сортування, обмеження тощо для отримання потрібних даних.

Програма “Electro” включає в себе повноціну систему авторизації, яка буде відповідати всім сучасним, вимогам безпеки і функціональності, авторизація буде проходити з використанням технології “Json Web Token”. Проаналізувавши потенційні задачі програми можна прийти до висновку що використання сесій і відповідне виділення під них таблицю в БД в даному випадку буде поганою практикою, тому для зберігання токена буде використовуватись local storage, а сам токен буде передаватись на сервер для валідації через headers. Подібний підхід допоможе оптимізувати роботу сервісу оскільки cookies не будуть відправлятись на сервер з кожним запитом, також це заощадить пам'ять на сервері, тому що не прийдеться зберігати всю інформацію про дії користувача, його авторизацію чи щось в нашій базі даних.

Таблиця 2.1 представляє структуру колекції "passwordReset" в MongoDB, яка використовується для зберігання інформації про скидання пароля користувача. Нижче наведено опис полів цієї колекції:

- `expire (Дата)`: Це поле вказує на дату і час закінчення терміну дії запиту на скидання пароля. Зазвичай це буде значення типу `Date`, яке визначає точний момент, коли запит стане недійсним.

- `code` (Рядок): Це поле містить унікальний код або токен, який генерується для ідентифікації запиту на скидання пароля. Код може бути у вигляді рядка символів, який випадковим чином створюється при створенні запиту.

- `email` (Рядок): Це поле містить електронну адресу користувача, для якого був зроблений запит на скидання пароля. Воно допомагає ідентифікувати, для якого облікового запису потрібно змінити пароль.

Ця структура дозволяє зберігати дані про запити на скидання пароля, включаючи дату закінчення дії запиту, унікальний код і пов'язану електронну адресу користувача. За допомогою цієї колекції можна виконувати операції створення нових записів, оновлення і видалення запитів на скидання пароля.

Таблиця 2.1 – Структура колекції “passwordReset”

<code>expire</code>	<code>code</code>	<code>email</code>
Date	string	string

Таблиця 2.2 відображає структуру колекції "Users" в MongoDB, яка використовується для зберігання інформації про користувачів. Нижче наведено опис полів цієї колекції:

- `Name` (Рядок): Це поле містить повне ім'я користувача.
- `Email` (Рядок): Це поле зберігає електронну адресу користувача.
- `Password` (Рядок): Це поле містить захешований пароль користувача.
- `Nickname` (Рядок): Це поле містить нікнейм або псевдонім користувача.
- `Reference` (Рядок): Це поле містить посилання на інший об'єкт або документ, пов'язаний з користувачем.
- `Company` (Рядок): Це поле містить інформацію про компанію, з якою пов'язаний користувач.

- Enum ('admin', 'moder') (Рядок): Це поле використовується для визначення ролі користувача. Воно може приймати значення "admin" або "moder", що відповідають різним рівням привілеїв.
- Soldout (Рядок): Це поле містить інформацію про розпродані товари.
- Startcount (Рядок): Це поле містить початкову кількість товарів.
- Count (Рядок): Це поле містить поточну кількість товарів.
- Createdby (Рядок): Це поле містить інформацію про користувача, який створив запис.
- Createat (Рядок): Це поле містить дату і час створення запису.
- Editedat (Рядок): Це поле містить дату і час останньої редагування запису.
- Images (Рядок): Це поле містить посилання або інформацію про зображення, пов'язані з користувачем.

Ця структура дозволяє зберігати інформацію про користувачів, включаючи їх імена, електронні адреси, хешовані паролі, нікнейми, компанії, ролі, інформацію про товари, дати створення і редагування записів та інші додаткові властивості. За допомогою цієї колекції можна виконувати операції створення нових користувачів, оновлення, видалення та отримання інформації про користувачів.

Таблиця 2.2 – Структура колекції “Users”

Name	Email	Password	Nickname	Reference	Company	Enum('admin,'moder')
String	String	String	String	String	String	String
Soldout	Startcount	Count	Createdby	Createat	Editedat	Images

Таблиця 2.3 відображає структуру колекції "Products" в MongoDB, яка використовується для зберігання інформації про товари. Нижче наведено опис полів цієї колекції:

- Name (Рядок): Це поле містить назву товару.
- Type (Рядок): Це поле вказує на тип товару.
- Description (Рядок): Це поле містить опис товару.
- headImg (Рядок або Null): Це поле містить посилання або інформацію про головне зображення товару. Може мати значення рядка або нульового значення, якщо зображення відсутнє.
- Views (Число): Це поле відображає кількість переглядів товару.
- Likes (Число): Це поле відображає кількість лайків, отриманих товаром.
- Dislikes (Число): Це поле відображає кількість дизлайків, отриманих товаром.
- Soldout (Рядок): Це поле містить інформацію про розпродані товари.
- Startcount (Рядок): Це поле містить початкову кількість товарів.
- Count (Рядок): Це поле містить поточну кількість товарів.
- Createdby (Рядок): Це поле містить інформацію про користувача, який створив продукт.
- Createat (Рядок): Це поле містить дату і час створення запису про продукт.
- Editedat (Рядок): Це поле містить дату і час останньої редагування запису про продукт.
- Images (Рядок): Це поле містить посилання або інформацію про зображення, пов'язані з продуктом.

Таблиця 2.3 – Структура колекції “Products”

Name	Type	Decription	headImg	Views	Likes	Dislike
String	String	String	String or Null	Number	Number	Number
Soldout	Startcount	Count	Createdby	Createat	Editedat	Images

Ця структура дозволяє зберігати інформацію про продукти, включаючи їх назви, типи, описи, зображення, кількість переглядів, вподобання та несподобства, інформацію про розпродаж, кількість товару на складі, інформацію про створення та редагування записів, а також інші додаткові атрибути, які можуть бути корисними для управління та відстеження продуктів.

## 2.5 Розробка серверної частини

Розробка серверної частини веб-сервісу для обліку продукції включає створення та налаштування серверного програмного забезпечення, яке буде відповідати за обробку запитів від клієнтської частини і забезпечення взаємодії з базою даних.

Основні кроки розробки серверної частини можуть включати наступні етапи.

Вибір платформи або фреймворку для розробки серверної частини, який найкраще підходить до потреб проекту. Популярні варіанти включають Node.js, Python (за допомогою фреймворків, таких як Flask або Django), Ruby (за допомогою фреймворка Ruby on Rails) та інші.

Установка та конфігурація необхідного серверного програмного забезпечення, такого як веб-сервер (наприклад, Apache або Nginx) та база даних (наприклад, MongoDB).

Визначення і реалізація API (Application Programming Interface) для взаємодії з клієнтською частиною. API визначає доступні операції та формат

обміну даними між клієнтом і сервером. Зазвичай API реалізується за допомогою REST або GraphQL.

Розробка логіки обробки запитів, яка включає обробку даних, валідацію, обчислення та збереження в базу даних. Цей етап також може включати розробку алгоритмів, необхідних для виконання конкретних функцій, наприклад, обліку продукції, пошуку, фільтрації тощо.

Розробка механізмів аутентифікації та авторизації, щоб забезпечити доступ до функцій системи лише авторизованим користувачам. Це може включати використання токенів, сесій, шифрування даних тощо.

Реалізація автоматизованих та ручних тестів для перевірки працездатності, стабільності та безпеки серверної частини. Це допоможе виявити та виправити помилки та недоліки до випуску виробу в експлуатацію.

Враховуючи очікуване навантаження і обсяги даних, необхідно забезпечити масштабованість серверної частини, щоб вона могла ефективно працювати з ростом кількості користувачів та об'єму даних. Це може включати використання кешування, горизонтального та вертикального масштабування, розподіленої обробки тощо.

Ці етапи є загальними і можуть варіюватися залежно від конкретних вимог проекту та використовуваних технологій. Важливо також враховувати засоби контролю версій, належне документування коду та забезпечення безперебійної роботи сервера.

## **2.6 Розробка клієнтської частини**

Розробка клієнтської частини веб-сервісу для обліку продукції включає створення інтерфейсу користувача, який буде взаємодіяти з серверною частиною та дозволяти користувачам зручно використовувати функціональність сервісу. Основні кроки розробки клієнтської частини можуть включати наступні етапи:



1. **Дизайн інтерфейсу:** Розробка дизайну інтерфейсу, включаючи розташування елементів, стилізацію, кольорову схему та інші аспекти, що впливають на зовнішній вигляд та користувацький досвід. Важливо враховувати простоту використання та зручність для користувачів.

2. **Розробка фронтенду:** Використання відповідних технологій та фреймворків (наприклад, HTML, CSS, JavaScript або сучасні фреймворки, такі як React, Angular або Vue.js) для розробки фронтенду клієнтської частини. Це включає реалізацію різних компонентів, форм, таблиць та інших елементів інтерфейсу, які будуть використовуватись користувачами.

3. **Взаємодія з сервером:** Реалізація механізмів взаємодії з серверною частиною за допомогою API, що надається сервером. Це включає відправку запитів на сервер для отримання та відправлення даних, обробку відповідей сервера та оновлення відповідних частин інтерфейсу.

4. **Обробка даних:** Реалізація логіки для обробки та валідації даних, введених користувачем через інтерфейс. Це може включати перевірку правильності введених даних, обчислення значень, фільтрацію, сортування та інші операції з даними перед їх відправкою на сервер або відображенням на інтерфейсі.

5. **Обробка подій:** Реалізація обробки подій, таких як натискання кнопок, введення тексту, вибір елементів тощо. Це включає призначення обробників подій та виконання певних дій при їх спрацюванні, наприклад, оновлення вмісту сторінки, відправка запитів на сервер або зміна стану інтерфейсу.

6. **Тестування:** Виконання автоматизованих та ручних тестів для перевірки коректності роботи клієнтської частини, виявлення та виправлення помилок, а також перевірка відповідності функціональним вимогам.

7. **Оптимізація та вдосконалення:** Постійне вдосконалення клієнтської частини шляхом оптимізації швидкодії, вдосконалення користувацького досвіду та внесення поліпшень на основі отриманих відгуків користувачів.

Важливо також дотримуватись засад безпеки, забезпечувати захист передачі та збереження даних, а також дотримуватись принципів доступності та сумісності із різними пристроями та браузерами.

Бібліотеки Redux, Saga і Antd Design використовуються для розробки клієнтської частини веб-сервісу і надають додаткові можливості та інструменти для покращення управління станом додатку та вигляду інтерфейсу. Основні особливості цих бібліотек такі:

1. Redux: Redux є популярною бібліотекою для керування станом додатків у JavaScript. Він дозволяє зберігати стан додатку в централізованому store і здійснювати зміни стану за допомогою actions та reducers. Redux спрощує управління складними станами додатку та забезпечує передачу даних між компонентами.

2. Redux Saga: Redux Saga – це бібліотека, яка дозволяє використовувати генератори для управління побічними ефектами (такими як асинхронні запити до сервера) в Redux-додатках. Вона надає потужні інструменти для обробки асинхронності, керування побічними ефектами та сагами.

3. Antd Design: Ant Design – це комплект готових компонентів та стилів, розроблений на базі React, які дозволяють швидко створювати стильні та функціональні інтерфейси. Він містить набір готових компонентів, таких як кнопки, форми, таблиці, меню і багато інших, які можна легко використовувати та налаштовувати. Antd Design дозволяє швидше розробляти інтерфейси з дотриманням єдиного стилю.

Використання цих бібліотек полегшує розробку клієнтської частини, дозволяє ефективніше керувати станом додатку, обробляти асинхронні запити та створювати привабливі та функціональні інтерфейси.

## 2.7 Тестування програми і результати її виконання

Тестування програми є важливим етапом в розробці, оскільки воно дозволяє перевірити функціональність, надійність та якість програмного продукту. Нижче наведені різні види тестування, які можуть бути застосовані до веб-сервісу для обліку продукції «Electro», а також можливі результати їх виконання.

Виконання тестів, спрямованих на перевірку правильності роботи функціональних можливостей програми. Результати можуть бути представлені у вигляді звітів, які вказують на успішне виконання або виявлення помилок та недоліків.

Перевірка візуального вигляду та функціональності інтерфейсу користувача. Результати можуть бути відображені у вигляді скріншотів або звітів, які показують, чи відповідає інтерфейс заданим вимогам та чи є в ньому помилки або недоліки.

Використання автоматичних тестових скриптів для виконання тестів і перевірки функціональності програми. Результати можуть бути представлені у вигляді звітів, які показують успішність або невдачу виконання тестів.

Визначення продуктивності програми та її здатності обробляти великі обсяги даних або виконувати складні операції. Результати можуть бути представлені у вигляді звітів, які показують час відгуку, завантаження системи та інші показники продуктивності.

Перевірка вразливостей програми та виявлення потенційних проблем з безпекою. Результати можуть бути представлені у вигляді звітів, які показують виявлені проблеми безпеки та рекомендації щодо їх виправлення.

Перевірка, чи працює програма з попередніми версіями бази даних та інших залежностей. Результати можуть бути представлені у вигляді звітів, які показують, чи є проблеми зі сумісністю та що потрібно зробити для вирішення цих проблем.

Важливо регулярно проводити тестування, а також фіксувати результати, виявлені помилки та проблеми. Це дозволяє виявляти та виправляти проблеми ранньою стадією розробки, підвищує якість програми та користувацьке задоволення.

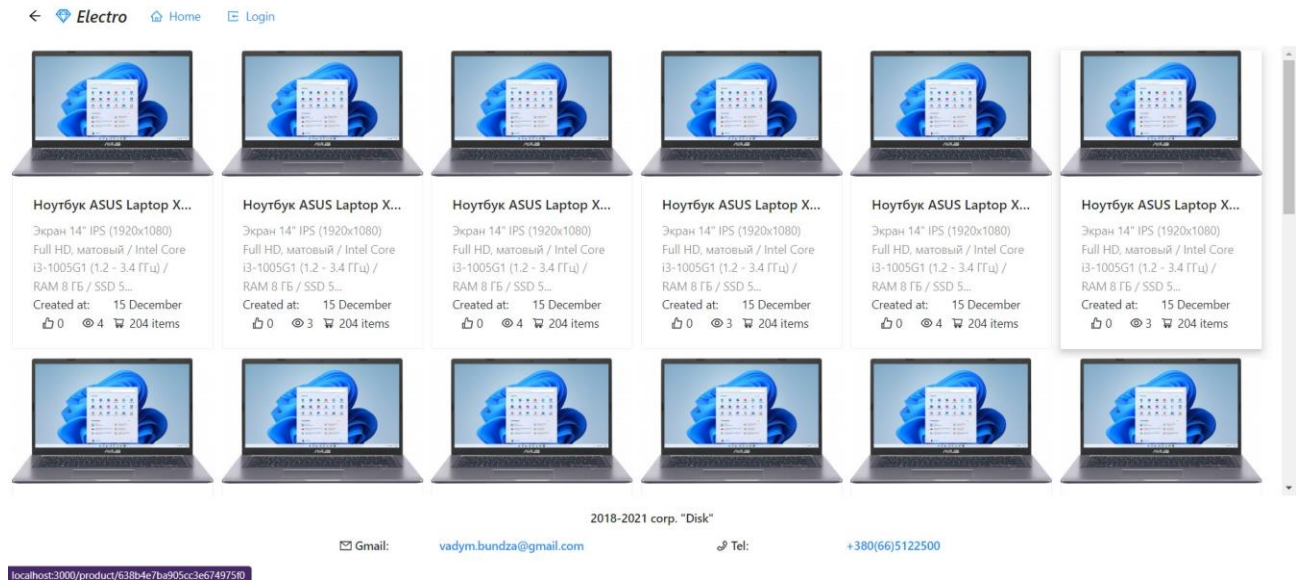


Рисунок 2.3 – Домашня сторінка

При першому переході на сайт користувача зустрине домашня сторінка з приємним дизайном. На домашній сторінці будуть зображені продукти та меню. Процес виконання зображено на рисунку 2.3.

У випадку якщо користувача зацікавила продукція він може на неї зайти клікнувши по ній. У такому випадку він попадає на сторінку де описані характеристик продукту. Процес виконання зображено на рисунку 2.4.

Характеристики продукту – це сторінка, яка надає детальну інформацію про основні властивості і функції продукту. На цій сторінці користувач може ознайомитися зі специфікаціями, можливостями та іншими важливими даними про продукт. Описуючи характеристики продукту, можна зосередитися на таких аспектах:

1. Назва продукту: Вказати офіційну назву продукту, яка використовується для ідентифікації.
2. Опис продукту: Представити короткий огляд продукту, включаючи його призначення, головні функції та користувацькі переваги. Цей опис повинен зацікавити користувача та зрозуміло пояснити, що робить продукт особливим.
3. Технічні характеристики: Надати детальні технічні характеристики, такі як розмір, вага, матеріали, роздільна здатність, підтримувані формати та інші технічні параметри, які є важливими для користувача.
4. Функціональні можливості: Пояснити, які функції та опції доступні в продукті. Вказати, які завдання він може виконувати та які переваги це надає користувачу.
5. Інтерфейс користувача: Описати, як виглядає інтерфейс користувача продукту, його навігація та основні елементи керування. Вказати, чи є інтерфейс інтуїтивно зрозумілим та легким у використанні.
6. Сумісність: Вказати, з якими операційними системами, пристроями або іншими програмними засобами продукт може працювати.
7. Підтримка та оновлення: Зазначити, які види підтримки надаються для продукту, включаючи технічну підтримку, оновлення програмного забезпечення та можливість отримання допомоги.
8. Ліцензія та ціна: Вказати тип ліцензії, за яким доступний продукт, а також інформацію про ціну або можливість безкоштовного використання.
9. Відгуки та рейтинги: Забезпечити можливість користувачам залишати свої відгуки та оцінки продукту, а також відображати існуючі відгуки користувачів.
10. Зображення продукту: Додати фотографії або зображення продукту, що допоможуть користувачам краще уявити його зовнішній вигляд або інтерфейс.

The screenshot displays the 'Electro' admin dashboard for a user named 'admin@test.com'. The main content area is a form for creating a new product. The form fields are as follows:

- Product name:** Apple iPhone 13 128GB Pink
- Type:** Another
- Description:** камера: 12 Мп + 12 Мп, фронтальная камера: 12 Мп / 128 Гб встроенной памяти / 3G / LTE / 5G / GPS / Nano-SIM, eSIM / iOS 15
- Head img:** <https://content1.rozetka.com.ua/goods/images/big/221214154.jpg>
- Count:** 21
- Images:** <https://content2.rozetka.com.ua/goods/images/big/221214151.jpg>

On the right side, there is a product preview card for 'Apple iPhone 13 128GB ...'. The card features an image of the phone and the following text:

- Экран (6.1", OLED (Super Retina XDR), 2532x1170) / Apple A15 Bionic / двойная основная камера: 12 Мп...
- Created at: 15 December
- 0 likes, 21 items

The footer of the page contains the following information:

- 2018-2021 corp. "Disk"
- Gmail: [vadym.bundza@gmail.com](mailto:vadym.bundza@gmail.com)
- Tel: +380(66)5122500

Рисунок 2.4 – Сторінка характеристики продукту

Загальна структура сторінки характеристик продукту може бути представлена у формі таблиці або списку, щоб зробити інформацію зручною для сприйняття користувачем. Важливо, щоб характеристики продукту були представлені чітко, конкретно і зрозуміло для цільової аудиторії.

При успішній авторизації користувач (адміністратор/модератор) перейде в форму, в якій він зможе добавляти товар. Процес виконання зображено на рисунку 2.5.

Створення продукту передбачає виконання різних етапів і процесів. Один з важливих етапів – це розробка форми створення продукту, яка дозволяє користувачам ввести необхідну інформацію та параметри для створення нового продукту.

Форма створення продукту повинна мати зручний та інтуїтивно зрозумілий дизайн, щоб користувачі змогли швидко та легко заповнити необхідні поля. Додатково, можна розглянути включення валідації даних, щоб переконатися, що користувач вводить вірну інформацію.

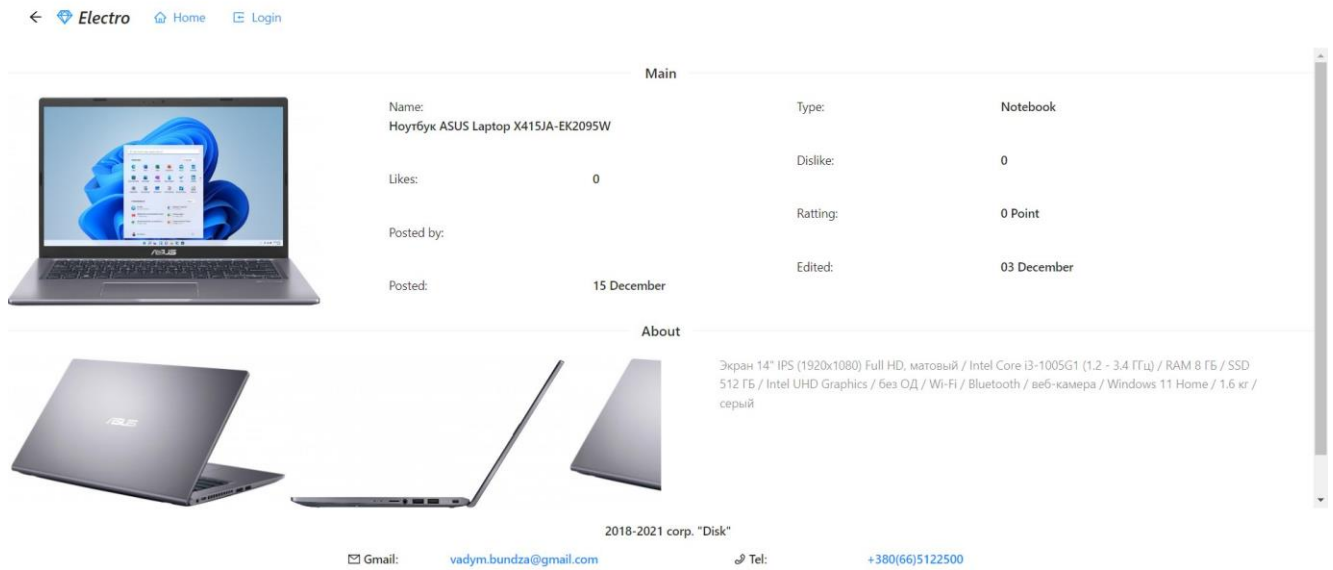


Рисунок 2.5 – Форма створення продукту

Для проходження успішною авторизацію користувачу потрібно ввести свої дані в відповідну форму. Процес виконання зображено на рисунку 2.6.

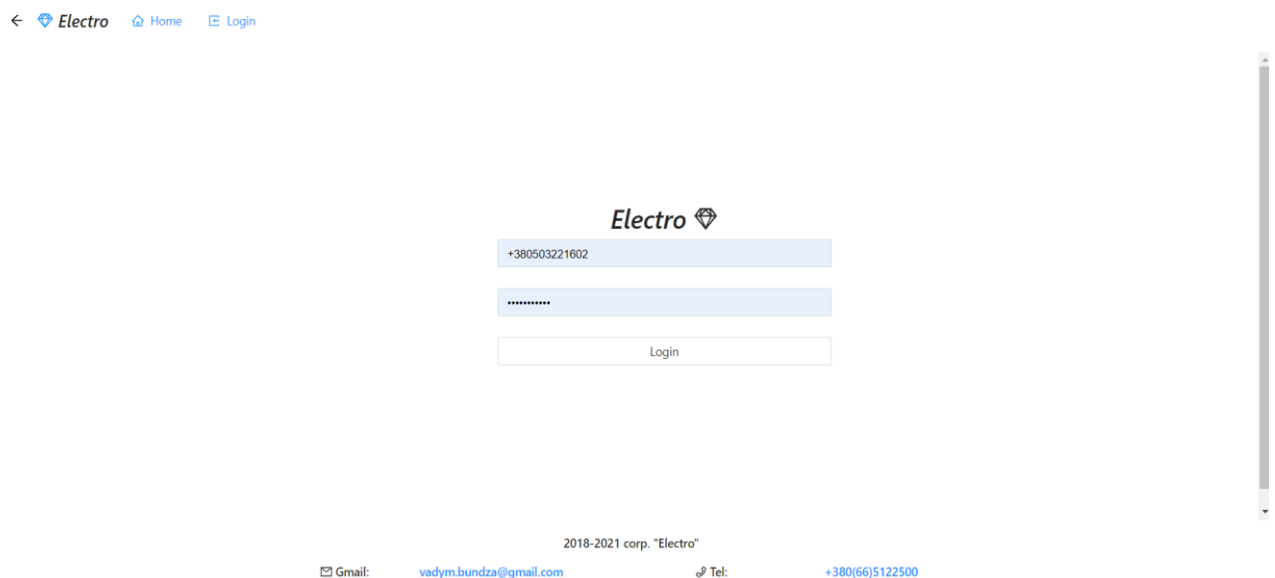


Рисунок 2.6 – Форма входу

Форма входу (логіну) є важливим елементом веб-додатків або мобільних додатків, який дозволяє користувачам аутентифікуватися та отримати доступ

до свого облікового запису. Основні елементи форми входу можуть включують наступні поля:

1. Логін (ім'я користувача або електронна пошта): Поле, в яке користувач вводить своє ім'я користувача або електронну пошту.
2. Пароль: Поле, в яке користувач вводить свій пароль. Зазвичай поле пароля приховується, щоб забезпечити конфіденційність.
3. Кнопка "Увійти" або "Вхід": Елемент, за допомогою якого користувач підтверджує свої облікові дані і намагається увійти в систему.

Форма входу повинна бути простою, зрозумілою та легко використовувати. Важливо забезпечити захист введених користувачем облікових даних, наприклад, за допомогою шифрування паролів та застосування механізмів безпеки, таких як CAPTCHA або двофакторна аутентифікація, для запобігання несанкціонованому доступу до облікових записів користувачів.

При потребі сортування продукції по різних полях користувач може скористатись фільтром. Процес виконання зображено на рисунку 2.7.

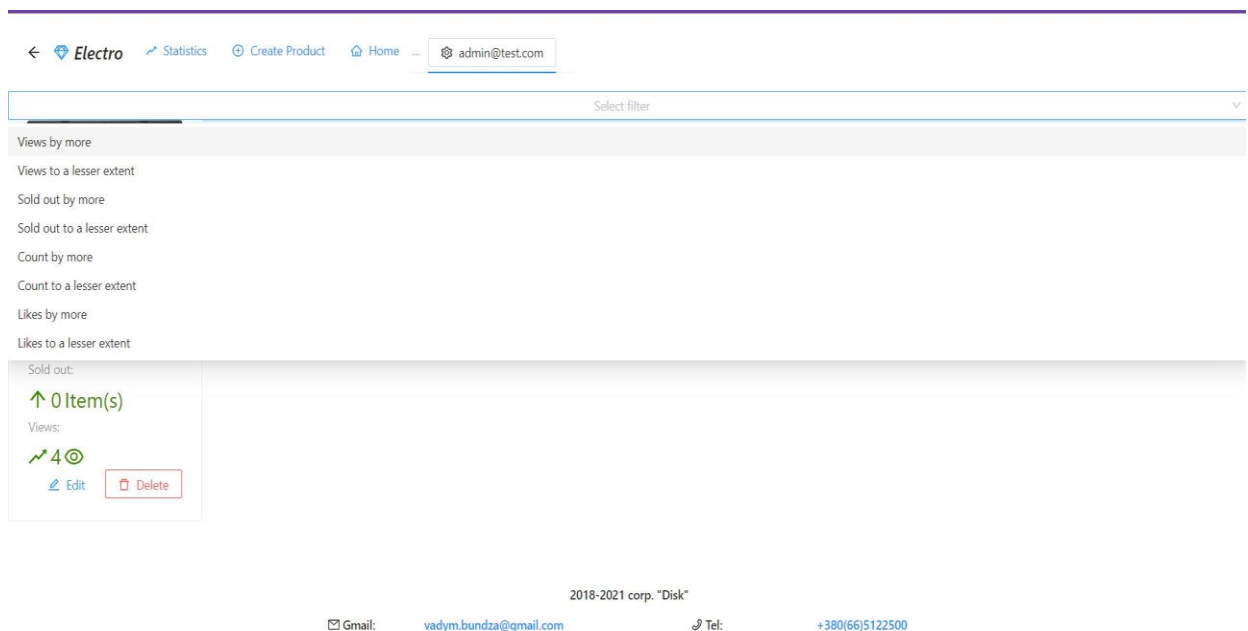


Рисунок 2.7 – Головна форма обліку (Модератор/Адмін)



Головна форма обліку для модератора або адміністратора є центральним інтерфейсом, який надає доступ до всіх основних функцій та операцій управління системою або платформою. Ця форма має багато різноманітних елементів, які дозволяють модератору або адміністратору виконувати різні завдання. Деякі з таких елементів можуть включати:

1. **Панель навігації:** Зазвичай розташована збоку або зверху і містить основні розділи, які модератор або адміністратор може вибрати. Це дозволяє швидко переходити між різними функціональними областями системи.

2. **Статистика та огляд:** Відображає основну статистику про систему, таку як загальна кількість користувачів, кількість активних сесій, загальний обсяг даних тощо. Це допомагає модератору або адміністратору відстежувати стан системи та виявляти потенційні проблеми або аномалії.

3. **Управління користувачами:** Надає можливість модератору або адміністратору переглядати, редагувати або видаляти користувачів. Це може включати можливості блокування облікових записів, встановлення ролей та прав доступу, вирішення скарг або запитів користувачів.

4. **Управління продуктами або контентом:** Дозволяє модератору або адміністратору додавати, редагувати або видаляти продукти, статті, зображення або будь-який інший контент, який присутній у системі. Це може включати можливості модерації, встановлення тегів або категорій, зміну статусу контенту тощо.

5. **Звіти та аналітика:** Надає модератору або адміністратору доступ до різноманітних звітів і аналітичних даних, які стосуються функціонування системи. Це можуть бути звіти про активність користувачів, звіти про продажі, звіти про помилки або відомості про системні параметри.

6. **Налаштування системи:** Дозволяє модератору або адміністратору налаштовувати параметри системи, такі як загальні налаштування, параметри безпеки, керування ролями та доступом користувачів, налаштування інтеграцій з іншими системами тощо.

Головна форма обліку для модератора або адміністратора повинна бути добре організованою, інтуїтивно зрозумілою та забезпечувати зручний доступ до всіх необхідних функцій та інструментів для ефективного управління системою.

Користувач будучи авторизованим має можливість редагувати/видаляти дані. Процес виконання зображено на рисунку 2.8.

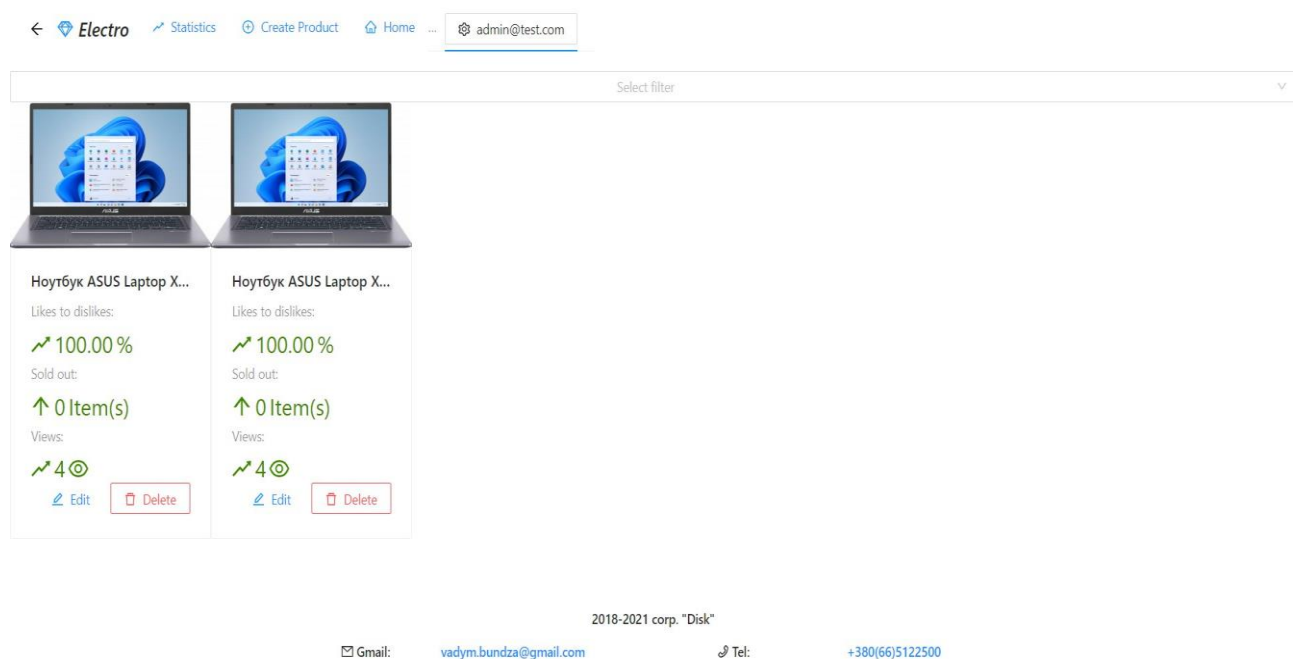


Рисунок 2.8 – Сторінка обліку (продавець/адміністратор)

Сторінка обліку для продавця або адміністратора є основним інтерфейсом, який дозволяє здійснювати облік та керування продуктами, замовленнями та іншими даними. Ця сторінка надає доступ до різних функцій та операцій, які виконуються продавцем або адміністратором. Деякі з можливих елементів та функцій на сторінці обліку можуть включати:

1. Загальна інформація: На сторінці обліку можуть бути відображені загальні дані, такі як статистика продажів, кількість замовлень, стан запасів тощо. Це дозволяє продавцеві або адміністратору отримувати швидкий огляд стану бізнесу.

2. Керування продуктами: Продавець або адміністратор може переглядати, додавати, редагувати або видаляти продукти. Це може включати встановлення цін, опису продукту, наявності на складі тощо. Продавець може також керувати категоріями або тегами, щоб групувати продукти.

3. Замовлення та доставка: Продавець або адміністратор можуть переглядати замовлення, відстежувати їх стан, змінювати статус замовлення, вказувати деталі доставки або відправлення товару, вирішувати питання пов'язані з обробкою замовлень.

4. Керування користувачами: Адміністратор може керувати користувачами системи, включаючи реєстрацію, авторизацію, блокування облікових записів, встановлення ролей та дозволів.

5. Звіти та аналітика: Сторінка обліку може містити звіти і аналітичні дані про продажі, прибуток, популярність продуктів, поведінку користувачів тощо. Це дозволяє адміністратору або продавцю аналізувати діяльність бізнесу та приймати відповідні рішення.

Сторінка обліку повинна бути зручною, з відображенням необхідних даних і функцій для ефективного керування бізнесом. Дизайн сторінки повинен бути чітким і зрозумілим, з урахуванням потреб користувачів та принципів зручності використання.

## **2.8 Висновки до другого розділу**

В ході розробки загальної структури програми було проведено аналіз функціональних вимог та потреб користувачів. Була створена логічна структура програми, яка включає основні модулі і компоненти системи, а також визначені основні варіанти використання програми. Це дало можливість чітко визначити функціональні можливості програми та структуру її інтерфейсу.

При виборі бази даних для проєкту було враховано його потреби щодо збереження та обробки інформації. Після порівняння різних варіантів було

прийняте рішення використовувати базу даних MongoDB. Ця нереляційна база даних надала гнучкість у зберіганні даних, легку масштабованість та швидкий доступ до інформації.

У підрозділі про порівняння реляційних баз даних з нереляційними були розглянуті переваги та недоліки обох типів баз даних. Враховуючи особливості проекту і його вимог до зберігання даних, було встановлено, що нереляційна база даних MongoDB відповідає краще потребам системи обліку продукції гуртовні "Electro".

Структура бази даних програми була ретельно спроектована, враховуючи основні сутності та взаємозв'язки між ними. Були визначені таблиці та поля, що відображають потрібну інформацію про товари, користувачів та інші сутності. Це забезпечує ефективне збереження та оптимальний доступ до даних.

Під час розробки серверної частини було створено необхідні API та сервіси для обробки запитів користувачів. Було реалізовано механізми авторизації та безпеки даних, що забезпечують захищений доступ до системи. Розробка серверної частини дозволила ефективно керувати даними, виконувати операції з обліку товарів та забезпечити відповідну взаємодію з базою даних.

У підрозділі про розробку клієнтської частини було описано процес створення веб-інтерфейсу користувача. Застосовано сучасні технології, такі як React, Redux і Saga, для створення функціонального та зручного інтерфейсу. Була розроблена форма створення продукту, форма входу та інші необхідні компоненти. Це дало змогу забезпечити зручну роботу з системою та швидкий доступ до потрібних функцій.

Під час тестування програми були перевірені різні аспекти її функціонування. Були проведені модульні тести для перевірки окремих компонентів та функцій, а також інтеграційні тести для взаємодії між серверною та клієнтською частинами. Результати тестування свідчать про правильну роботу системи, відповідність функціональних вимог та стабільність її роботи.

У цьому розділі було успішно вирішено завдання розробки технічного та робочого проекту. Відповідно до запланованих критеріїв, була розроблена структура програми, обрана відповідна база даних, порівняні реляційні та нереляційні бази даних, описана структура бази даних програми, реалізовано серверну та клієнтську частини, проведено тестування та отримано результати її виконання.

## РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Безпека життєдіяльності

#### *Надання першої долікарської допомоги при порушенні дихання і серцевої діяльності*

У потерпілих іноді порушуються дихання й серцева діяльність. Порушення дихання виникає при закупорюванні трахеї та бронхів блювотними масами, згустками крові (у разі поранення шиї, обличчя), водою при утопленні, коли не надходить повітря до легень. Крім того, дихання може припинитися, коли людина непритомніє, і через розслаблення м'язів у неї западає корінь язика, перекриваючи вхід у гортань; при порушенні функції центру дихання внаслідок тяжких черепно-мозкових травм, крововиливу в мозок, великої кількості переломів ребер; при ушкодженні легень і плеври. У потерпілого з'являється спочатку блідість, потім синюшність шкірних покривів обличчя. Дихальні рухи прискорюються, стають нерегулярними (іноді рідшають), а через деякий час, якщо не надається потрібна допомога, припиняються зовсім.

Насамперед перевіряють порожнину рота і намагаються відновити прохідність верхніх дихальних шляхів. Для цього вказівним пальцем, обмотаним бинтом чи носовою хусточкою, звільняють рот від блювотних мас, слизу, згустків крові, виймають знімні зубні протези. Коли запав язик, голову потерпілого слід повернути набік і підтримати щелепу.

Коли ж потерпілий не дихає, треба перевірити, чи не припинилася в нього серцева діяльність, і негайно розпочати штучне дихання. Для цього його кладуть спиною на жорстку поверхню (на підлогу автомашини, на дорогу), підмостивши під лопатки зроблений з одягу валик. Потерпілого розгинають і закидають його голову назад так, щоб максимально відвести підборіддя від грудної клітки, розстібають комір і пояс, що утруднюють дихання.

Найефективнішим способом штучного дихання є вдування повітря в ніс або рот потерпілого. Такий спосіб називають «рот до рота» або «рот до носа». У ніс вдувають повітря тоді, коли у хворого ушкоджені губи, нижня або верхня щелепа.

Штучне дихання способом "рот до рота" роблять так. Людина, яка надає допомогу, стає на коліна біля голови потерпілого, краще з правого боку, і кладе ліву руку на його лоб, затискуючи пальцями потерпілому ніс; прикривши його рот марлею або носовою хусточкою, робить глибокий вдих, а потім, притиснувшись ротом до рота потерпілого, вдуває в нього повітря, роблячи енергійний видих. Внаслідок такого вдування, що замінює вдих, повітря надходить у легені потерпілого. Згодом завдяки еластичності легеневої тканини і грудної клітки настає пасивний видих. При цьому рот потерпілого повинен бути відкритим. Вдувати повітря слід ритмічно, з однаковим інтервалом, 12-15 разів на хвилину.

Так само роблять штучне дихання способом «рот до носа», тільки повітря вдувають через ніс, а рот потерпілого закривають. Ефективність штучного дихання підвищується, коли застосовувати спеціальні S-подібні повітроводи, які забезпечують прохідність верхніх дихальних шляхів.

Штучне дихання слід виконувати доти, доки у потерпілого не відновиться самостійне ритмічне дихання. Якщо штучне дихання розпочинають робити своєчасно і проводять його правильно, то самостійне дихання відновлюється через 1...2 хвилини. Однак часом штучне дихання потрібно проводити 1-1,5 години, принаймні доти, доки на місце нещасного випадку не прибудуть медичні працівники.

Той, хто надає допомогу, особливо при надто енергійному вдуванні повітря, може навіть відчути запаморочення, слабкість, знепритомніти. Тому, приблизно через кожних 1-2 хвилини особи, які виконують штучне дихання, змінюють одна одну. Це підвищить ефективність допомоги потерпілому.

При тяжких ушкодженнях часом припиняється дихання і серцева діяльність. Тоді у потерпілого настає так звана клінічна смерть. У нього не промацується пульсація сонної артерії на шиї, не прослуховується серцебиття, припиняється дихання, зіниці розширюються, шкірні покриви й слизові оболонки губ – синюшні.

Через 5-6 хвилин після настання клінічної смерті відбуваються необоротні зміни в нервових клітинах центрів головного мозку, що керують життєво важливими функціями організму. Внаслідок цього настає біологічна смерть. З клінічної смерті потерпілого можуть вивести лише негайні (не пізніше як через 4-10 хвилин після її настання) й енергійні заходи, спрямовані на відновлення серцевої діяльності та дихання. Необхідно пам'ятати, що кожна втрачена секунда може стати фатальною. Порядок дій у цьому випадку:

- впевнитися у відсутності пульсу на сонній артерії;
- звільнити грудну клітку від одягу та розстібнути пояс;
- прикрити двома пальцями клиновидний відросток;
- нанести різкий удар кулаком по грудині;
- перевірити, чи з'явився пульс. Якщо пульс відсутній, удар можна повторити.

Якщо після вказаних вище дій пульс не прослуховується, негайно приступити до виконання штучного дихання та непрямого масажу серця.

Роблячи непрямий масаж серця, потерпілого кладуть спиною на тверду поверхню. Той, хто надає допомогу, стає на коліна збоку (краще справа) біля потерпілого і, поклавши кисті рук одну на другу в нижній частині грудини, робить енергійний поштовх, натискаючи на груднину так, щоб вона зміщувалася приблизно на 4-5 см у напрямі хребта. При цьому стискається серце між грудиною і хребтом, кров із порожнини серця виштовхується в кровоносні судини. Оскільки стінка грудної клітки еластична, то вона повертається у вихідне положення, а порожнини серця наповнюються кров'ю. Поштовхи слід робити ритмічно, близько 60 разів на хвилину.



У разі відсутності дихання непрямий масаж серця поєднують із штучним диханням. Це потребує чималих зусиль. Тому бажано, щоб потерпілим у стані клінічної смерті допомогу надавало двоє людей, одна робить непрямий масаж серця, друга – штучне дихання. Після кожного вдихання повітря в легені потерпілого чотири рази натискають на грудину. Під час вдихання повітря не можна стискати грудну клітку.

Показником ефективності непрямих масажу серця і штучного дихання є порожевіння шкірних покривів, звуження зіниць, поява на великих артеріях (стегновій, сонній) пульсових поштовхів синхронно натисканню на грудину і, нарешті, відновлення самостійного дихання й серцебиття. Непрямий масаж серця і штучне дихання слід проводити до відновлення серцевої діяльності й дихання. При цьому необхідно не втрачати пильність та не забувати про можливість нової зупинки серця та дихання. Щоб не пропустити цей момент, необхідно стежити за зіницями, кольором шкіри, диханням, перевіряти частоту і ритмічність пульсу.

При розробці веб-сайту Тернопільського обласного осередку Федерації пауерліфтингу України враховані “Державні санітарні норми і правила роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин” ДСанПіН 3.3.2.007-98.

## **3.2 Охорона праці**

### *Вимоги до виробничих приміщень для експлуатації ВДТ*

Широке промислове та побутове використання ПК актуалізувало питання охорони праці їхніх користувачів. Найбільш повним нормативним документом щодо забезпечення охорони праці користувачів ПК є “Державні санітарні норми і правила роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин” ДСанПіН 3.3.2.007-98.

Дотримання вимог цих правил значно знижує наслідки несприятливої дії на працівників шкідливих та небезпечних факторів, які супроводжують роботу з відео-дисплейними матеріалами, зокрема можливість зорових, нервово-емоційних переживань, серцево-судинних захворювань. Роботодавець зобов'язаний забезпечити гігієнічні й ергономічні вимоги щодо організації робочих приміщень для експлуатації ВДТ, робочого середовища, робочих місць з ВДТ, режиму праці і відпочинку при роботі з ВДТ.

Відповідно до встановлених гігієнічно-санітарних вимог ДСанПіН 3.3.2.007-98 роботодавець зобов'язаний забезпечити в приміщеннях з ВДТ оптимальні параметри виробничого середовища. У таблиці 4.1 наведено норми мікроклімату для приміщень з ВДТ.

Природне освітлення в приміщеннях з ВДТ здійснюється через вікна, орієнтовані переважно на північ або північний схід і забезпечує коефіцієнт природної освітленості не нижче ніж 1,5 %. Для захисту від прямих сонячних променів, які створюють прямі та відбиті відблиски з поверхні екранів ПК і клавіатури передбачені сонцезахисні пристрої, вікна повинні мати жалюзі або штори.

### **3.3 Висновок до третього розділу**

В третьому розділі кваліфікаційної роботи розглянуто питання надання першої долікарської допомоги при порушенні дихання і серцевої діяльності та вимоги до виробничих приміщень для експлуатації ВДТ.

## ВИСНОВКИ

В ході виконання кваліфікаційної роботи освітнього рівня «Бакалавр» було успішно розроблено веб-сервіс для обліку продукції гуртовні «Electro». Завдяки цьому проекту було досягнуто декількох важливих висновків:

1. Завдяки автоматизації обліку продукції, компанія "Electro" змогла покращити ефективність своїх процесів управління та контролю. Система дозволяє швидко та точно відстежувати наявність товарів, здійснювати редагування, видалення та додавання нових записів.

2. Використання сучасних технологій та бібліотек, таких як Redux, Saga та Antd Design, дозволило розробити зручний та інтуїтивно зрозумілий інтерфейс користувача. Це полегшує роботу користувачам та сприяє покращенню їх досвіду взаємодії з програмою.

3. Застосування бази даних MongoDB дозволяє ефективно зберігати та керувати великими обсягами даних, що забезпечує швидкий доступ до інформації і зменшує час реакції системи.

4. Тестування програми дозволило виявити та виправити помилки та недоліки, що сприяє покращенню якості та стабільності програмного продукту.

Загалом, розробка веб-сервісу для обліку продукції гуртовні «Electro» принесла позитивні результати і виконала поставлені завдання. Вона дозволяє компанії покращити свою ефективність, зручність управління та контролю за продукцією.

## ПЕРЕЛІК ДЖЕРЕЛ

1. JavaScript [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/JavaScript> – Дата доступу: 12.04.2023. – Загол. с екрана.
2. JavaScript [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript> – Дата доступу: 12.04.2023. – Загол. з екрану.
3. jQuery [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/JQuery> – Дата доступу: 15.04.2023. – Загол. з екрану.
4. CSS [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/CSS> – Дата доступу: 15.04.2023. – Загол. з екрану.
5. HTML [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/HTML> – Дата доступу: 15.04.2023. – Загол. з екрану.
6. PHP [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/PHP> – Дата доступу: 18.04.2023. – Загол. з екрану.
7. WebStorm [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/WebStorm> – Дата доступу: 18.04.2023. – Загол. з екрану.
8. MySQL [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/MySQL> – Дата доступу: 25.04.2023. – Загол. з екрану.
9. PhpStorm [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/PhpStorm> – Дата доступу: 28.04.2023. – Загол. з екрану.
10. PSPad [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/PSPad> – Дата доступу: 28.04.2023. – Загол. з екрану.

11. Система керування базами даних [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Система\\_керування\\_базами\\_даних](https://uk.wikipedia.org/wiki/Система_керування_базами_даних) – Дата доступу: 02.05.2023. – Загол. з екрану.

12. Веб-технології та веб-дизайн [Електронний ресурс] – Режим доступу: <http://victoria.lviv.ua/html/wp/index.html> – Дата доступу: 30.05.2023. – Загол. с екрана.

13. Моделі потоків даних (DFD-моделі): призначення, місце застосування в системному аналізі, правила побудови, приклади [Електронний ресурс] – Режим доступу: <http://victoria.lviv.ua/html/wp/index.html> – Дата доступу: 15.05.2023.

14. Основні методи аналізу виробничого травматизму [Електронний ресурс] – Режим доступу: <http://pidruchniki.com/15341220/bzhd/> – Дата доступу: 12.04.2023. – Загол. з екрану.

15. Ергономічні вимоги до організації робочих місць [Електронний ресурс] – Режим доступу: <http://pidruchniki.com/14821111/bzhd> – Дата доступу: 12.04.2023. – Загол. з екрану.

16. Охорона праці користувачів персональних комп'ютерів (ПК) [Електронний ресурс] – Режим доступу: <http://library.if.ua/book/9/1016.html> – Дата доступу: 20.04.2023.

17. Загальні вимоги до виробничих приміщень для експлуатації ВДТ ЕОМ та ПЕОМ [Електронний ресурс] – Режим доступу: <http://mybiblioteka.su/11-115960.html> – Дата доступу: 20.04.2023.

18. Перша долікарська допомога [Електронний ресурс] – Режим доступу: [http://ua-referat.com/Перша\\_долікарська\\_допомога](http://ua-referat.com/Перша_долікарська_допомога) – Дата доступу: 22.04.2023.

19. Надання першої допомоги потерпілим [Електронний ресурс] – Режим доступу: <http://referatu.net.ua/newreferats/448/187221> – Дата доступу: 20.04.2023.

20. Основи охорони праці [Електронний ресурс] – Режим доступу: <http://ua.textreferat.com/referat-2127-1.html> – Дата доступу: 20.04.2023.

21. Гандзюк, М. П. Основи охорони праці [Текст] : підручник / М. П. Гандзюк, Є. П. Желібо, М. О. Халімовський ; за ред. М. П. Гандзюка ; МОН України. – 4-е видання. – К. : Каравела, 2013. – 384 с. – ISBN966-8019-01-6.

22. Гогіташвілі, Г. Г. Основи охорони праці [Текст] : навчальний посібник / Г. Г. Гогіташвілі, В. М. Лапін ; 4-те вид. випр. і доп. – К. : Знання, 2013. – 302 с. – ISBN 978-966-346-400-8.