

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему:

Інформаційна система для аналізу терміну придатності товарів

Виконав: студент

IV курсу, групи СНЗ-41

спеціальності

122 Комп'ютерні науки

(шифр і назва спеціальності)

	Бонюкевич А.Р. (підпис) (прізвище та ініціали)
Керівник	Матійчук Л.П. (підпис) (прізвище та ініціали)
Нормоконтроль	Шимчук Г.В. (підпис) (прізвище та ініціали)
Завідувач кафедри	Боднарчук І.О. (підпис) (прізвище та ініціали)
Рецензент	Петрик М.Р. (підпис) (прізвище та ініціали)

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)

« 07 » червня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

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

Студенту Бонюкевич Андрій Русланович
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система для аналізу терміну придатності товарів

Керівник роботи Матійчук Любомир Павлович, доцент, професор кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 07 » травня 2025 року № 4/7-449

2. Термін подання студентом завершеної роботи 23 червня 2025 р.

3. Вихідні дані до роботи Наукові публікації, електронні ресурси, підручники, посібники з тематики дослідження.

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

Вступ. 1. Аналіз предметної області та постановка завдання дослідження. 1.1 Необхідність відстеження терміну придатності товарів. 1.2 Аналіз відомих систем в даній предметній області. 1.3 Постановка завдання дослідження. 1.4 Висновки до першого розділу.

2. Проектування інформаційної системи для відстеження терміну придатності товарів.

2.1. Проектування архітектури інформаційної системи. 2.2. Проектування інтерфейсу користувача. 2.3. Структура JSON даних. 2.4. Висновки до другого розділу. 3. Програмна реалізація та тестування інформаційної системи. 3.1. Вибір засобів та програмна реалізація аутентифікації інформаційної системи. 3.2. Програмна реалізація взаємодії з серверною частиною реалізація React модулів. 3.3. Програмна реалізація адаптивного інтерфейсу.

4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел. Додатки.

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

1.Тема. 2. Актуальність теми. 3. Мета та завдання дослідження. 4. Аналіз предметної області.

5. Огляд існуючих систем. 6.Архітектура системи. 7. Проектування інтерфейсу користувача.

8. Структура даних JSON. 9. Реалізація клієнтської частини. 10. Адаптивний інтерфейс.

11. Система аутентифікації. 12. Функціонал системи. 13. Тестування системи. 14. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	доцент кафедри МТ Сенчишин Віктор Степанович	02.06.2025	05.06.2025

7. Дата видачі завдання 07 травня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	07.05.2025	Виконано
2.	Підбір та опрацювання літературних джерел по темі кваліфікаційної роботи	08.05.2025-15.05.2025	Виконано
3.	Виконання дослідження щодо інформаційної системи для аналізу терміну придатності товарів	16.05.2025-22.05.2025	Виконано
4.	Оформлення розділу «Аналіз предметної області та постановка завдання дослідження»	23.05.2025-01.06.2025	Виконано
5.	Оформлення розділу «Проектування інформаційної системи для відстеження терміну придатності товарів»	23.05.2025-01.06.2025	Виконано
6.	Оформлення розділу «Програмна реалізація та тестування інформаційної системи»	23.05.2025-01.06.2025	Виконано
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	02.06.2025-05.06.2025	Виконано
8.	Виконання завдання до підрозділу «Основи охорони праці»	02.06.2025-05.06.2025	Виконано
9.	Оформлення кваліфікаційної роботи	06.06.2025-10.06.2025	Виконано
10.	Нормоконтроль	11.06.2025-13.06.2025	Виконано
11.	Перевірка на плагіат	16.06.2025	Виконано
12.	Попередній захист кваліфікаційної роботи	18.06.2025	Виконано
13.	Захист кваліфікаційної роботи	19.06.2025	

Студент

(підпис)

Бонюкевич А.Р.

(прізвище та ініціали)

Керівник роботи

(підпис)

Матійчук Л.П.

(прізвище та ініціали)

АНОТАЦІЯ

Інформаційна система для аналізу терміну придатності товарів // Кваліфікаційна робота освітнього рівня «Бакалавр» // Бонюкевич Андрій Русланович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНЗ-41 // Тернопіль, 2025 // С. 56 , рис. – 36, табл. – 4, додат. – 2 , бібліогр. – 36 .

Ключові слова: Інформаційна система, терміну придатності товарів, структура JSON, реалізація React модулів, інтерфейс користувача, тестування інформаційної системи.

Кваліфікаційна робота висвітлює дослідження, щодо розробки клієнтської частини системи відстеження терміну придатності товарів. В першому розділі охарактеризовано предметну область та напрямки дослідження. Визначено набір функцій, необхідних для бізнес-процесу, що лягли в основу створення схеми управління цим процесом. Додатково, проведено аналіз вимог до програмного забезпечення, та поставлено завдання дослідження.

Вдругому розділі кваліфікаційної роботи були узагальнено функціональні та нефункціональні вимоги до клієнтської частини системи, розглянуті варіанти використання системи. Розроблена архітектура веб-системи, визначені компоненти системи, їх взаємодія та функціонал. Спроектowana структура даних у форматі JSON, яка забезпечує гнучкий обмін інформацією та адаптацію до нових вимог.

Розроблена веб-система для відстежування термінів придатності продукції з використанням Visual Studio Code, React, Node.js, MobX, Axios, Clsx, Lodash, Moment, React-table, Vite, ZXing, HTML, Sass та TypeScript. Забезпечена безпечна аутентифікація з JWT та адаптивна верстка з використанням Mobile-first, Flexbox та BEM. Axios використано для передачі даних, а ZXing – для сканування штрих-кодів, та здійснено тестування інформаційної системи.

ANNOTATION

Information System for Analyzing the Expiration Date of Products // Qualification work of the educational level "Bachelor" // Bonyukevich Andriy Ruslanovich // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNz-41 // Ternopil, 2025 // P. 56 , fig. – 36 , tab. – 4 , annexes. – 2, references – 36.

Keywords: Information system, shelf life of goods, JSON structure, implementation of React modules, user interface, testing of the information system.

The qualification work highlights the research on the development of the client part of the system for tracking the shelf life of goods. The first section describes the subject area and areas of research. A set of functions necessary for the business process was determined, which formed the basis for the creation of a management scheme for this process. Additionally, an analysis of software requirements was conducted, and the research was conducted.

In the second section of the qualification work, functional and non-functional requirements for the client part of the system were summarized, and options for using the system were considered. The architecture of the web system was developed, the system components, their interaction and functionality were determined. A data structure in JSON format was designed, which provides flexible information exchange and adaptation to new requirements.

A web system for tracking product expiration dates was developed using Visual Studio Code, React, Node.js, MobX, Axios, Clsx, Lodash, Moment, React-table, Vite, ZXing, HTML, Sass and TypeScript. Secure authentication with JWT and adaptive layout using Mobile-first, Flexbox and BEM were provided. Axios was used for data transmission, and ZXing was used for barcode scanning, and the information system was tested.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ	7
1.1 Необхідність відстеження терміну придатності товарів	7
1.2 Аналіз відомих систем в даній предметній області	9
1.3 Постановка завдання дослідження	15
1.4 Висновки до першого розділу	23
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВІДСТЕЖЕННЯ ТЕРМІНУ ПРИДАТНОСТІ ТОВАРІВ	24
2.1. Проектування архітектури інформаційних системи	24
2.2. Проектування інтерфейсу користувача	27
2.3. Структура JSON даних	30
2.4. Висновки до другого розділу	33
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	34
3.1. Вибір засобів та програмна реалізація аутентифікації інформаційної системи	34
3.2. Програмна реалізація взаємодії з серверною частиною реалізація React модулів	36
3.3. Програмна реалізація адаптивного інтерфейсу	43
3.4. Тестування інформаційної системи	44
3.5. Висновки до третього розділу	51
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	53
4.1 Питання щодо безпеки життєдіяльності	53
4.2 Питання з основ охорони праці	56
4.3 Висновок до четвертого розділу	60
ВИСНОВКИ	61
ПЕРЕЛІК ДЖЕРЕЛ	62
ДОДАТОКИ	66

ВСТУП

Актуальність теми. Сьогодні в роздрібній торгівлі та виробництві продуктів харчування забезпечення якості та безпеки продукції відіграє ключову роль. Відстеження термінів придатності стає все важливішим аспектом цього процесу, при цьому традиційні методи контролю поступаються місцем автоматизованим системам. Ці системи не тільки оптимізують процеси контролю, але й знижують ризики продажу простроченої продукції, підвищуючи операційну ефективність і забезпечуючи високий стандарт якості та безпеки.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка клієнтської частини системи відстеження терміну придатності товарів, шляхом автоматизації дублюючих операцій, що виконуються «вручну», оптимізації взаємодії учасників процесу відстеження.

Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- виконати аналіз предметної області, описати вимоги до системи відстеження терміну придатності продукції;
- спроектувати клієнтську частину системи;
- виконати реалізацію клієнтської частини системи;
- провести тестування клієнтської частини системи.

Практичне значення одержаних результатів. Розробка та впровадження систем відстеження термінів придатності стають невід'ємною частиною сучасного бізнесу. Такі системи не лише підвищують ефективність роботи підприємств, але й запобігають продажу простроченої продукції, що важливо для підтримання репутації компанії та забезпечення її конкурентоспроможності на ринку. В роботі реалізована клієнтська частини інформаційної системи.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Автоматизація відстеження терміну придатності товарів

Термін придатності товарів є важливим і невід'ємним елементом у сфері виробництва, обігу та споживання товарів на всіх етапах їх життєвого циклу. Він визначає період, протягом якого товар зберігає свої основні властивості, відповідає встановленим вимогам щодо безпеки, якості, ефективності та інших технічних характеристик. Термін придатності, залежно від типу товару, може значно змінюватися, оскільки для різних груп продукції встановлені різні умови зберігання, що зумовлено специфікою їх фізико-хімічних властивостей та умовами використання.

Забезпечення належного контролю терміну придатності товарів має критичне значення, оскільки продаж товарів, що вийшли з терміну придатності, може призвести до серйозних наслідків для споживачів, включаючи небезпеку для здоров'я та безпеки. Зокрема, для харчових продуктів, лікарських засобів та косметичних товарів втрата якості або токсичність після закінчення терміну придатності може призвести до отруєнь, хвороб або неефективності ліків. Таким чином, правильне управління термінами придатності сприяє не лише підтриманню високого рівня безпеки, але й збереженню довіри споживачів і репутації компаній.

Контроль терміну придатності товарів є важливим аспектом в управлінні якістю продукції. Сучасні підходи до визначення та відстеження терміну придатності включають не лише традиційні методи, такі як візуальний огляд або застосування спеціальних індикаторів, але й автоматизацію процесів через використання сучасних інформаційних технологій. Включення в цей процес автоматизованих систем, таких як програми для моніторингу терміну придатності, штучний інтелект, системи баз даних, а також програмне забезпечення для управління запасами, дозволяє забезпечити ефективність і точність контролю. Завдяки таким технологіям стає можливим оперативно та

безпомилково відстежувати терміни придатності товарів, зменшуючи ризик помилок, які можуть виникнути при ручному обліку.

Система моніторингу термінів придатності товарів є важливою складовою у забезпеченні якості та безпеки продукції. Вона охоплює не тільки зберігання даних про кожен окремий товар і його термін придатності, а й включає механізми для перевірки правильності умов зберігання та своєчасного оновлення інформації про стан продукції. Крім того, використання новітніх методів аналізу та прогнозування дозволяє знижувати ризики від використання прострочених товарів і оптимізувати управлінські процеси на підприємствах.

Особливу увагу слід приділяти специфікаціям термінів придатності для харчових продуктів, лікарських засобів, косметичних товарів та інших груп продукції, що мають обмежений термін зберігання. Для кожної категорії товарів вимоги до зберігання можуть істотно відрізнятися: для харчових продуктів часто враховуються температурні режими, вологість, упаковка, в той час як для ліків додатково враховуються впливи на хімічний склад, такі як світло або повітря. З метою дотримання нормативних вимог і збереження високої якості продукції, необхідно проводити регулярні перевірки і моніторинг умов зберігання.

Розвиток та впровадження сучасних інформаційних систем для обліку та моніторингу термінів придатності товарів дозволяє значно підвищити ефективність управління запасами та знизити ймовірність виникнення негативних наслідків від продажу товарів, термін придатності яких минув. Ці системи, як правило, інтегровані з платформами для управління запасами та обліку продукції, а також з іншими інформаційними технологіями, що дозволяє здійснювати моніторинг на різних етапах постачання товарів. Наприклад, автоматизовані системи можуть використовувати історичні дані та прогностичні моделі для точного прогнозування термінів придатності залежно від умов зберігання або транспортування, що значно покращує операційну ефективність.

Впровадження таких технологій дозволяє компаніям знижувати витрати, пов'язані з втратами через псування товарів, забезпечувати відповідність до регламентів безпеки та якості і підвищувати задоволеність споживачів. У свою чергу, це має важливе значення для покращення конкурентоспроможності компаній на ринку, адже своєчасне і точне управління термінами придатності є однією з ключових складових успіху в сучасних умовах бізнесу.

Загалом, предметна область терміну придатності товарів охоплює різні аспекти, від безпеки продукції до ефективного управління виробничими і дистрибуційними процесами, і є важливою складовою стратегії якості в різних секторах економіки.

1.2 Аналіз відомих систем в даній предметній області

Відстеження та управління термінами придатності товарів є критичним компонентом забезпечення якості та безпеки продукції. З огляду на різноманітність товарів, що вимагають контролю термінів придатності, існує кілька систем, розроблених для автоматизації цього процесу. Нижче проведемо порівняльний аналіз відомих систем, що використовуються для моніторингу та управління термінами придатності.

1. Система SAP Extended Warehouse Management (EWM), рисунок 1.1.

Опис: SAP EWM є однією з найпоширеніших систем управління складом, яка також включає функціональність для управління термінами придатності товарів. Вона дозволяє автоматично відслідковувати терміни придатності товарів на складі, здійснювати їх облік і контроль на всіх етапах ланцюга постачання.

Переваги: Інтеграція з іншими бізнес-системами SAP дозволяє ефективно обробляти дані в межах всього підприємства. Забезпечує автоматичне управління термінами придатності за допомогою штрих-кодів і RFID-технологій. Можливість налаштування складних сценаріїв для управління складом та термінами придатності.

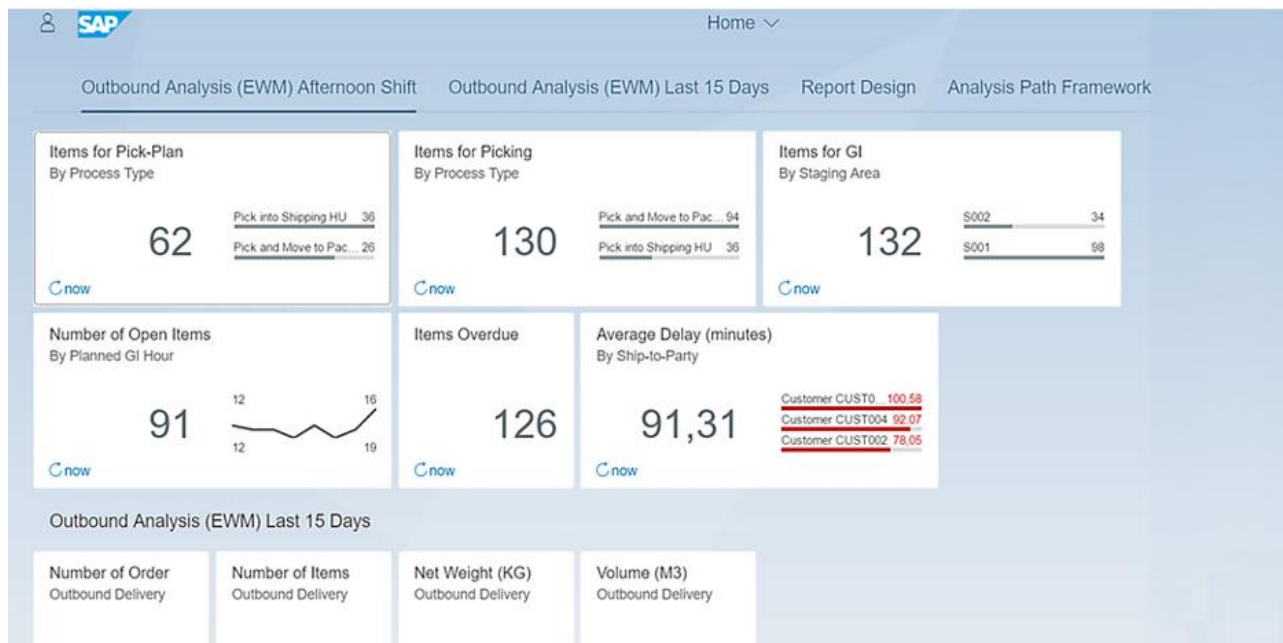


Рис. 1.1 – Система SAP Extended Warehouse Management

Недоліки. Висока вартість впровадження та обслуговування. Потрібно значне налаштування під специфіку підприємства. Може бути складною для використання на малих підприємствах через високу складність.

2. Oracle Warehouse Management (WMS), рисунок 1.2. Опис: Oracle WMS є ще однією потужною системою для управління складами, яка включає функції моніторингу термінів придатності товарів. Вона підтримує інтеграцію з іншими корпоративними системами для забезпечення повного контролю над ланцюгами постачання.

Переваги. Підтримує відстеження термінів придатності за допомогою автоматизованих процесів. Інтеграція з іншими системами Oracle для обробки даних і оптимізації ланцюга постачання. Висока точність даних завдяки використанню штрих-кодів та RFID.

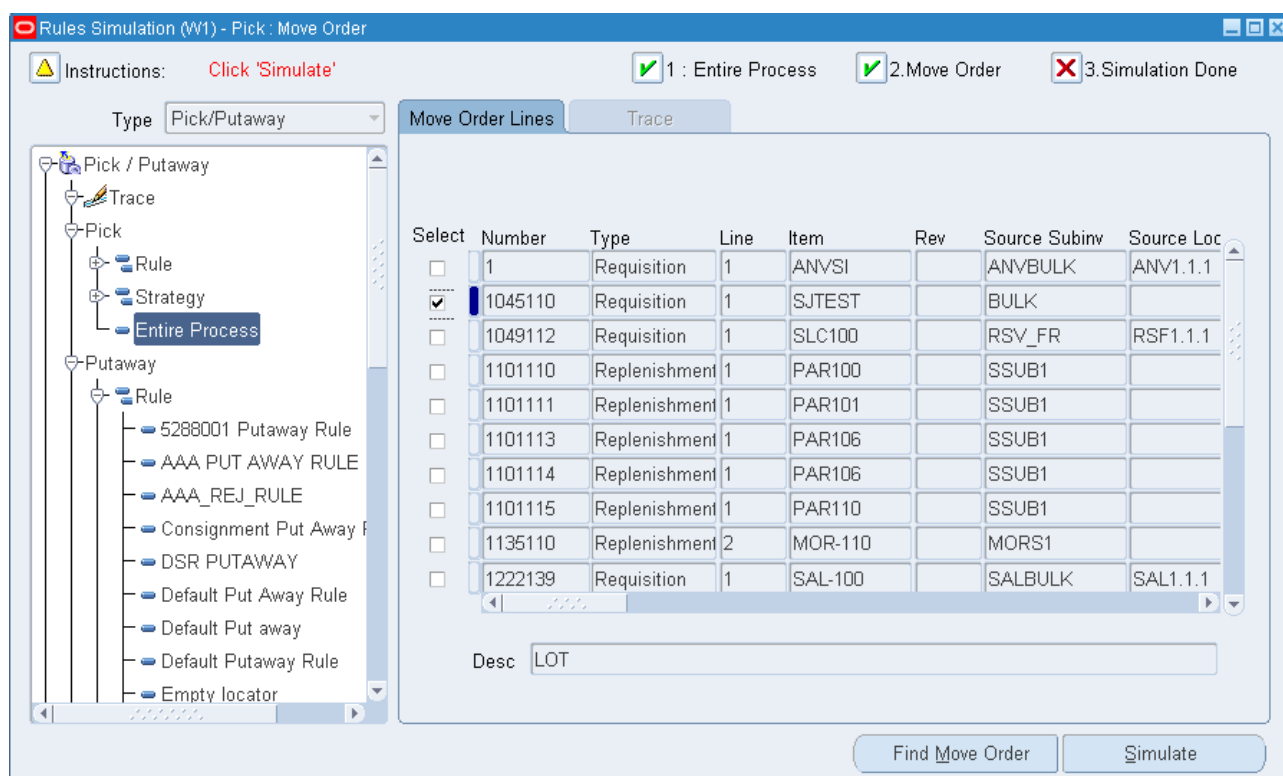
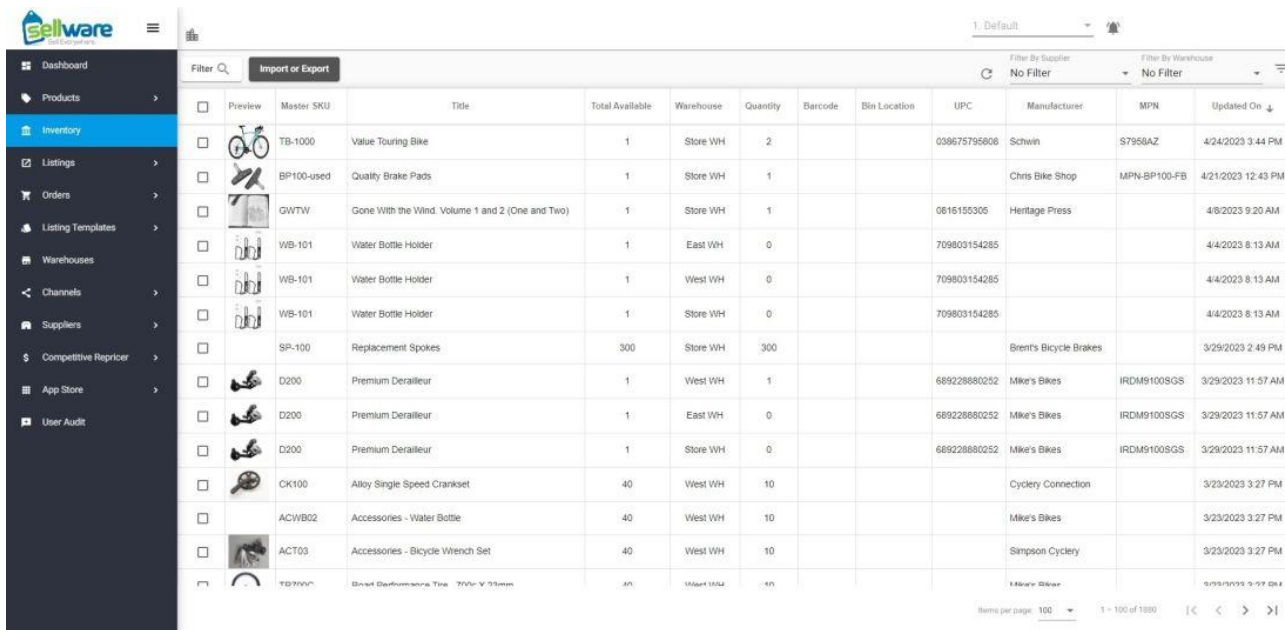


Рис. 1.2 – Система Oracle Warehouse Management

Недоліки. Як і SAP EWM, має високі вимоги до ресурсів для впровадження. Залежність від Oracle-платформи для повноцінної інтеграції. Потрібно спеціалізоване навчання для ефективного використання.

3. Fishbowl Inventory, рисунок 1.3. Опис: Fishbowl Inventory — це популярна система для малого та середнього бізнесу, яка пропонує функціональність для моніторингу термінів придатності товарів. Вона спеціалізується на управлінні запасами та контролі термінів придатності, забезпечуючи простоту в інтеграції та використанні.

Переваги. Доступна для малого і середнього бізнесу завдяки відносно низькій вартості. Легкість впровадження і використання. Підтримка управління термінами придатності і автоматичне оновлення статусу продуктів.



Preview	Master SKU	Title	Total Available	Warehouse	Quantity	Barcode	Bin Location	UPC	Manufacturer	MPN	Updated On
	TB-1000	Value Touring Bike	1	Store WH	2			039675795808	Schwinn	ST958AZ	4/24/2023 3:44 PM
	BP100-used	Quality Brake Pads	1	Store WH	1				Chris Bike Shop	MPN-BP100-FB	4/21/2023 12:43 PM
	GWTW	Gone With the Wind, Volume 1 and 2 (One and Two)	1	Store WH	1			0616155305	Heritage Press		4/8/2023 9:20 AM
	WB-101	Water Bottle Holder	1	East WH	0			709803154285			4/4/2023 8:13 AM
	WB-101	Water Bottle Holder	1	West WH	0			709803154285			4/4/2023 8:13 AM
	WB-101	Water Bottle Holder	1	Store WH	0			709803154285			4/4/2023 8:13 AM
	SP-100	Replacement Spokes	300	Store WH	300				Brent's Bicycle Brakes		3/29/2023 2:49 PM
	D200	Premium Derailleur	1	West WH	1			689228880252	Mike's Bikes	IRDM9100SGS	3/29/2023 11:57 AM
	D200	Premium Derailleur	1	East WH	0			689228880252	Mike's Bikes	IRDM9100SGS	3/29/2023 11:57 AM
	D200	Premium Derailleur	1	Store WH	0			689228880252	Mike's Bikes	IRDM9100SGS	3/29/2023 11:57 AM
	CK100	Alloy Single Speed Crankset	40	West WH	10				Cyclery Connection		3/23/2023 3:27 PM
	ACWB02	Accessories - Water Bottle	40	West WH	10				Mike's Bikes		3/23/2023 3:27 PM
	ACT03	Accessories - Bicycle Wrench Set	40	West WH	10				Simpson Cyclery		3/23/2023 3:27 PM
	T20000	Black Performance Tire 700c x 35mm	40	West WH	10				Mike's Bikes		3/23/2023 3:27 PM

Рис. 1.3 – Система Fishbowl Inventory

Недоліки. Обмежені можливості масштабування для великих підприємств. Недостатня гнучкість для складних налаштувань. Можливі проблеми з інтеграцією в разі складних бізнес-процесів.

4. Infor WMS, рисунок 1.4. Опис: Infor WMS — це система управління складами, яка дозволяє автоматизувати процеси управління запасами та контролювати терміни придатності товарів. Вона також надає розширені функціональні можливості для управління ланцюгами постачання та розподілу.

Переваги. Пропонує великий набір функцій для управління складом і термінами придатності. Можливість інтеграції з іншими бізнес-системами. Підтримка мобільних пристроїв для зручності роботи на складі.

Недоліки. Висока вартість, що може бути невідомою для малого бізнесу. Потрібно велику кількість часу на впровадження та навчання персоналу. Може бути складною для користувачів, які не мають досвіду роботи з великими корпоративними системами.

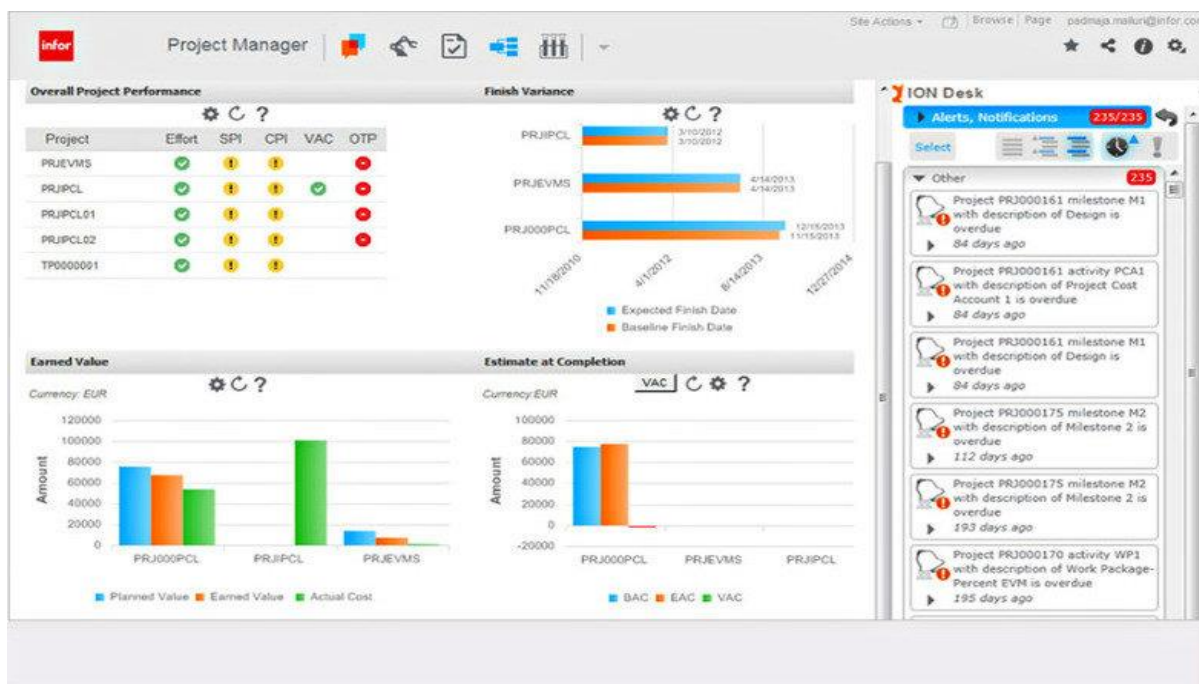


Рис. 1.4 – Система Infor WMS

5. Zoho Inventory, рисунок 1.5. Опис: Zoho Inventory — це онлайн-система для управління запасами та моніторингу термінів придатності товарів. Вона призначена для малих і середніх підприємств, забезпечуючи простоту в управлінні запасами та відстеженні термінів придатності.

Переваги. Простота впровадження і використання. Доступна ціна для малого бізнесу. Вбудовані функції для відстеження термінів придатності товарів і управління їх обігом.

Недоліки. Можливості для масштабування обмежені для великих підприємств. Відсутність розширених функцій для складних процесів моніторингу. Може бути недостатньо для підприємств з високими вимогами до інтеграції та автоматизації.

6. RFID-based systems (наприклад, Impinj), рисунок 1.6. Опис: Системи на основі технології RFID використовуються для відстеження товарів з термінами придатності в реальному часі. Вони дозволяють автоматично збирати дані про термін придатності за допомогою RFID-міток, що забезпечує точність і швидкість обробки.

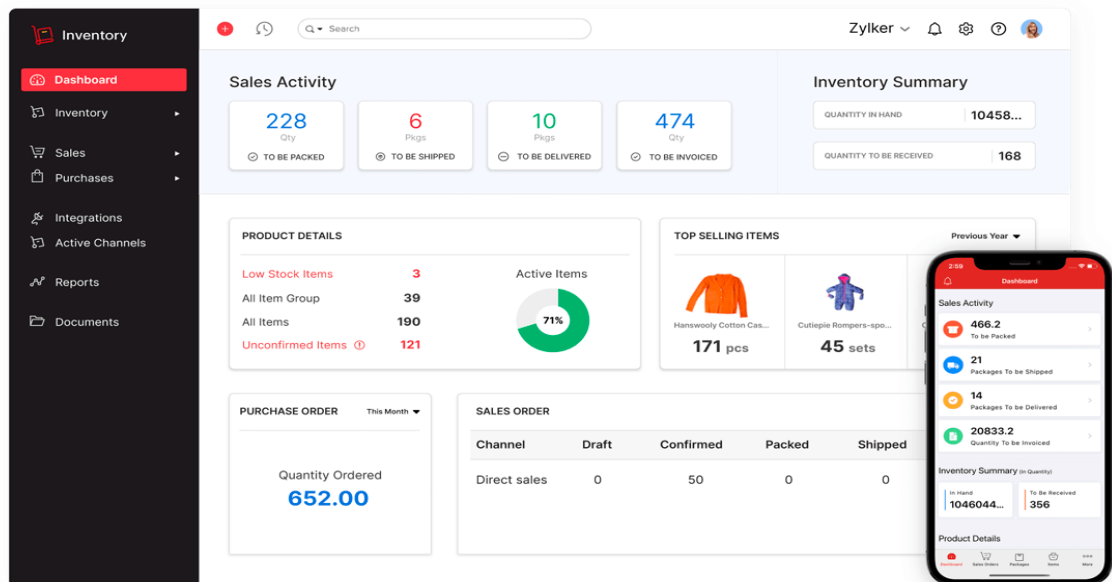


Рис. 1.5 – Система Zoho Inventory

Переваги. Висока точність даних завдяки автоматизованому збору інформації. Можливість відстежувати товари в реальному часі без потреби в ручному втручанні. Інтеграція з іншими системами для поліпшення управлінських процесів.

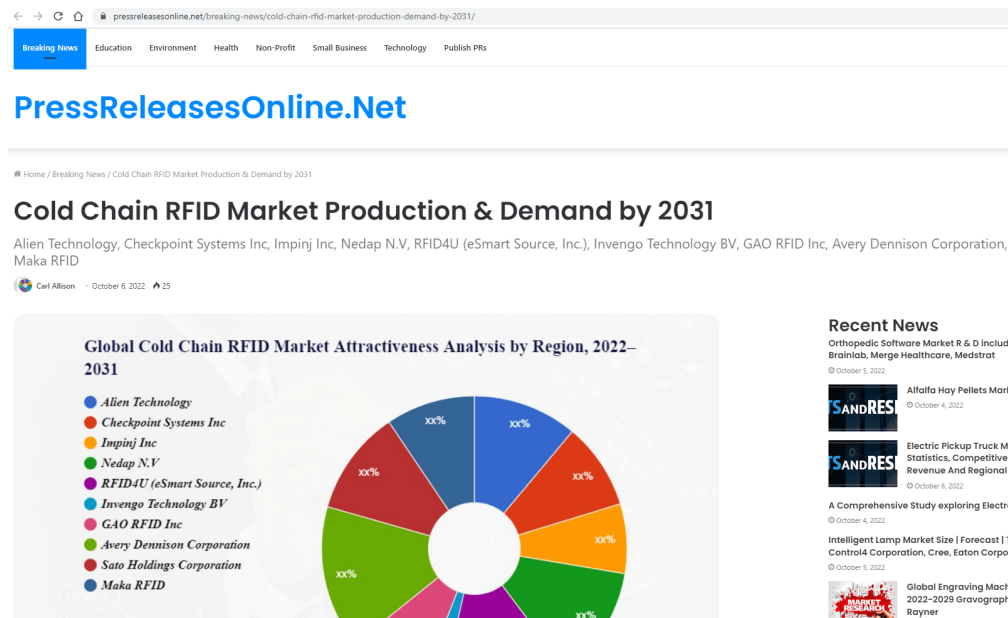


Рис. 1.6 – RFID-based systems

Недоліки. Висока вартість впровадження через потребу в спеціалізованих мітках і зчитувачах. Може бути складною для підприємств, що не мають досвіду роботи з RFID-технологіями. Потребує високого рівня технічної підтримки та обслуговування.

Кожна з вищезазначених систем має свої особливості, переваги та обмеження. Вибір системи для управління термінами придатності товарів залежить від розміру підприємства, складності бізнес-процесів і бюджету. Встановлення автоматизованої системи з відстеження термінів придатності дозволяє значно покращити ефективність управління запасами, зменшити ризики, пов'язані з втратою якості продукції, і забезпечити високий рівень безпеки для споживачів.

1.3 Постановка завдання дослідження

Аналіз предметної області дозволив виділити наступні функціональні та нефункціональні вимоги.

Функціональні вимоги. На основі проведеного аналізу були визначені функціональні вимоги, що визначають ключові функції та сервіси, які повинна забезпечувати система для задоволення потреб користувачів і досягнення цілей проєкту.

Були виділені наступні функціональні вимоги:

1. Система не повинна надавати доступ неавторизованим користувачам. Авторизація повинна здійснюватися за логіном та паролем.
2. Система повинна дозволяти користувачам сканувати штрих-коди та QR-коди за допомогою сканера.
3. Система повинна дозволяти користувачам сканувати штрих-коди та QR-коди за допомогою камери.
4. Система повинна дозволяти користувачам вводити дані про продукцію та її термін придатності.

5. Система повинна дозволяти користувачам створювати цехи, на яких виготовляється власна продукція підприємства.

6. Система повинна дозволяти користувачам створювати типи продукції, що виготовляються або надходять на підприємство.

7. Система повинна надавати можливість переглядати список усієї продукції, що використовується на підприємстві.

8. Система повинна надавати можливість переглядати дані про всю зареєстровану продукцію.

9. Система повинна дозволяти користувачам переглядати відслідковувану продукцію з інформацією про її термін придатності.

10. Система повинна дозволяти користувачам видаляти продукцію зі списку відслідкованої продукції після того, як її прибрали з прилавка.

11. Система повинна дозволяти користувачам фільтрувати список відслідкованої продукції за цехами.

12. Система повинна дозволяти користувачам фільтрувати список зареєстрованої продукції за цехами.

13. Система повинна дозволяти користувачам експортувати список зареєстрованої продукції за цехами для друку.

14. Система повинна дозволяти користувачам створювати акаунти в веб-додатку.

15. Система повинна дозволяти користувачам реєструватися в веб-додатку.

16. Система повинна дозволяти користувачам авторизуватися в веб-додатку.

17. Система повинна дозволяти користувачам виходити з облікового запису веб-додатку.

18. Система повинна дозволяти користувачам управляти ролями акаунтів веб-додатку.

19. Система повинна дозволяти користувачам створювати організації в веб-додатку.

20. Система повинна дозволяти користувачам управляти організаціями в веб-додатку.

21. Система повинна дозволяти користувачам блокувати акаунти в веб-додатку.

22. Система повинна дозволяти користувачам розблокувати акаунти в веб-додатку.

Нефункціональні вимоги. На основі проведеного аналізу були визначені нефункціональні вимоги, які визначають якісні характеристики та обмеження системи.

Були виділені наступні нефункціональні вимоги:

1. Система повинна забезпечувати безпеку користувацьких даних, а також використовувати HTTPS для захищеної передачі даних.

2. Система повинна бути доступною постійно, щоб користувачі могли мати доступ до неї в будь-який час.

3. Система повинна бути повністю протестована для забезпечення стабільної роботи при її оновленнях.

Варіанти використання системи. Для проектування додатку був використаний графічний опис для об'єктного моделювання UML. За складеним списком вимог до проектованої системи була побудована модель взаємодії акторів з програмною системою у вигляді діаграми варіантів використання, рисунки 1.7-1.10, для різних ролей.

Було виділено 4 актори, можливі варіанти використання для яких представлені нижче.

Неавторизований користувач – це людина або програма, яка не пройшла процес аутентифікації і не має доступу до системи.



Рис. 1.7 – Діаграма варіантів використання для неавторизованого користувача

Працівник – це користувач, який виконує конкретні обов'язки на підприємстві, такі як складовщик, кухар або продавець.

Менеджер – це користувач, відповідальний за управління операціями, управління персоналом і забезпечення ефективної роботи підприємства в цілому.

Адміністратор – це користувач, що має привілеї для управління налаштуваннями, доступом, безпекою та іншими аспектами функціонування додатку.

Неавторизований користувач може виконати авторизацію в додатку (увійти в акаунт для отримання прав користувача).

Працівник має обмежений доступ до функціоналу системи і може виконувати в системі наступні дії:

Сканування за допомогою сканера – сканування штрих-кодів та QR-кодів з допомогою пристроїв зі сканером.

Сканування за допомогою камери – сканування штрих-кодів та QR-кодів за допомогою камери смартфона або планшета.

Реєстрація надійшої продукції – оцінка органолептичних характеристик продукту (по шкалі від 1 до 5), введення ваги або кількості продукції, терміну придатності та температури.



Рис. 1.8 – Діаграма варіантів використання для працівника

Перегляд списку відслідковуваної продукції – перегляд списку зареєстрованої продукції, яка знаходиться на прилавках магазину, для отримання інформації про термін її придатності.

Видалення відслідковуваної продукції – процес виключення зі списку відслідковуваної продукції тієї продукції, яка або прострочена і непридатна для продажу, або вже була продана.

Фільтрація за цехами – функція, що дозволяє фільтрувати продукцію за цехами, в яких вона була виготовлена або до яких вона відноситься.

Менеджер, крім функціоналу працівника, має доступ до наступних функцій:

Перегляд списку цехів – перегляд списку назв цехів, на яких підприємство виготовляє продукцію або з яких її отримує.

Створення цехів – можливість додавання нового цеху, з якого надходить продукція.

Перегляд списку зареєстрованої продукції – перегляд списку з даними про всю продукцію, яка надійшла в продаж.

Експорт списку зареєстрованої продукції – можливість експортувати дані про зареєстровану продукцію в файл для друку.

Перегляд списку типів продукції підприємства – перегляд списку всіх типів продукції, що продається підприємством.

Створення типу продукції підприємства – додавання нового типу продукції з вказівкою назви, цеху, де він виготовлений, та коду для сканування.

Перегляд списку налаштувань – перегляд списку керованих параметрів.

Зміна часу спрацьовування звукових сповіщень – зміна налаштування часу спрацьовування звукових сповіщень до моменту закінчення терміну придатності продукції.

Увімкнення та вимкнення звукових сповіщень – зміна налаштування для увімкнення або вимкнення звукових сповіщень.

Фільтрація за датою – функція, що дозволяє фільтрувати продукцію за датою реєстрації.

Зміна пароля – функція зміни пароля поточного акаунта в налаштуваннях з введенням поточного та нового пароля.

Адміністратор може виконувати в системі наступні дії. Перегляд списку акаунтів – перегляд списку акаунтів, зареєстрованих у системі.



Рис. 1.9 – Діаграма варіантів використання для менеджера

Фільтрація списку акаунтів – можливість фільтрувати список акаунтів за заблокованими та незаблокованими.

Створення акаунтів – можливість зареєструвати новий акаунт з вказівкою ПІБ працівника, логіну, пароля, організації та ролі.

Редагування акаунтів – можливість змінити ПІБ працівника та роль.

Блокування акаунтів – можливість заблокувати акаунт, обмеживши його доступ до системи.

Розблокування акаунтів – можливість розблокувати акаунт, відновивши його доступ до системи.



Рис. 1.10 – Діаграма варіантів використання для адміністратора

Перегляд списку ролей – перегляд списку ролей у системі.

Створення ролей – можливість додати нову роль в системі з вказівкою назви та прав доступу.

Редагування ролей – можливість змінити назву та доступи, які матиме акаунт з цією роллю.

Перегляд списку організацій – перегляд списку організацій у системі.

Створення організацій – можливість додати нову організацію в системі з вказівкою назви.

1.4 Висновки до першого розділу

В даному розділі охарактеризовано предметну область та напрямки діяльності. Здійснено визначення набору функцій, необхідних для бізнес-процесу, що лягли в основу створення схеми управління цим процесом. Додатково, проведено аналіз вимог до програмного забезпечення.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ДЛЯ АНАЛІЗУ ТЕРМІНУ ПРИДАТНОСТІ ТОВАРІВ

2.1 Проектування архітектури інформаційної системи

Система є клієнт-серверним застосунком, що складається з таких частин: веб-інтерфейсу, бази даних і веб-застосунку, поділених на мікросервіси, такі як основна веб-система, сервіс керування акаунтами та сервіс автентифікації. Для представлення архітектури системи була побудована діаграма компонентів (рисунок 2.1).

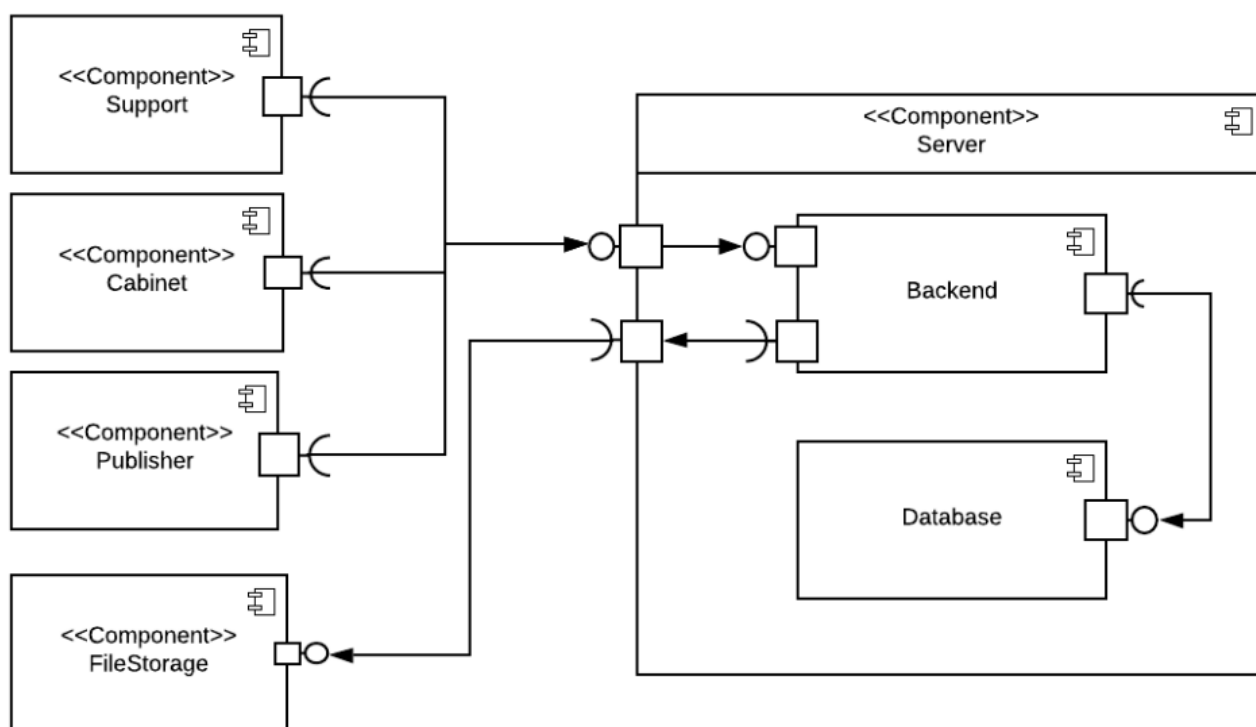


Рис. 2.1 – Діаграма компонентів інформаційної системи

На діаграмі представлені компоненти системи:

1. Веб-інтерфейс сервісу відстеження термінів придатності продукції – інтерфейс системи, який дозволяє зовнішнім клієнтам взаємодіяти з веб-

системою. Він підтримує можливість перегляду даних, їх фільтрації, зміни, додавання, входу та виходу з акаунта, завантаження та скачування файлів.

2. Веб-інтерфейс сервісу автентифікації – інтерфейс системи, що дозволяє зовнішнім клієнтам взаємодіяти з веб-системою через інтернет-браузер. Він надає можливість користувачам, які мають акаунт в системі, отримати до неї доступ шляхом автентифікації.

3. Веб-інтерфейс сервісу керування акаунтами – інтерфейс системи, який дозволяє зовнішнім клієнтам взаємодіяти з веб-системою через інтернет-браузер.

4. Backend – API-сервіс веб-системи відстеження термінів придатності продукції, який відповідає за обробку даних, що надходять з клієнтської частини, зберігання даних та реалізацію функціоналу.

Для наочного представлення сторінкової структури додатку була складена діаграма, зображена на рисунку 2.2.

В інформаційній структурі представлені такі сторінки:

1. Сканування штрих-кода – сторінка для сканування штрих-кода сканером;
2. Сканування QR-кода – сторінка для сканування QR-кода камерою;
3. Зміна пароля – сторінка, що відображає форму зміни пароля поточного акаунта;
4. Відслідковувана продукція – сторінка для відслідковування зареєстрованої продукції, що надає інформацію про свіжу, зіпсовану продукцію та продукцію, у якої термін придатності скоро спливає;
5. Бракеражний журнал – сторінка, що відображає всю інформацію про зареєстровану будь-коли продукцію, з можливістю фільтрації за цехами, періодами та експорту в файл Excel;
6. Список цехів – сторінка, що відображає інформацію про цехи або постачальників продукції, з можливістю створення нового цеху;

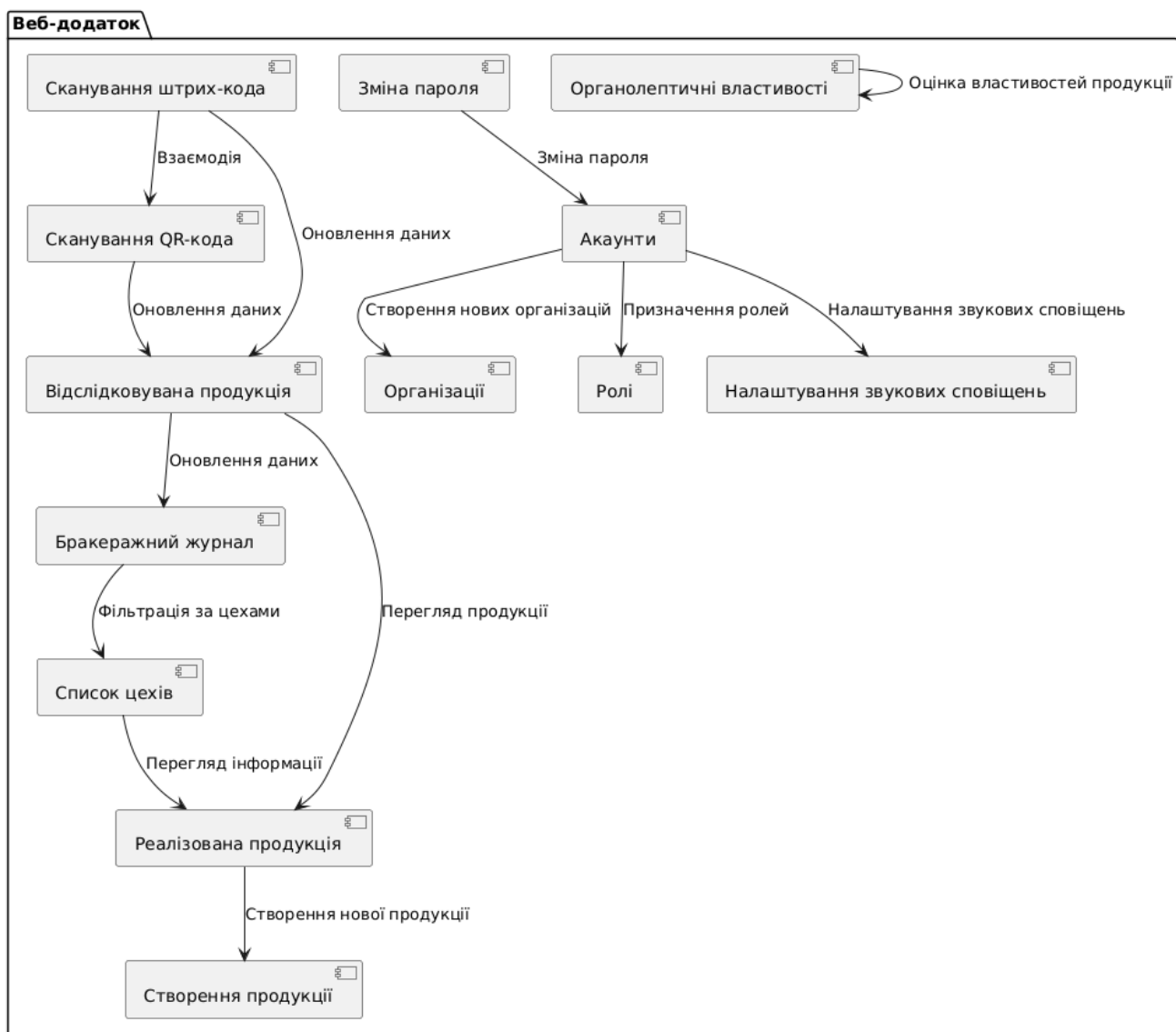


Рис. 2.2. – Інформаційна архітектура системи

7. Реалізована продукція – сторінка, що відображає інформацію про всю реалізовану продукцію підприємства, з можливістю фільтрації за цехами та створення нової продукції;

8. Створення продукції – сторінка, що дозволяє створити новий тип реалізованої продукції;

9. Акаунти – сторінка, що відображає інформацію про акаунти, зареєстровані в системі, з можливістю блокування, редагування та створення нового акаунта;

10. Організації – сторінка, що відображає інформацію про організації, зареєстровані в системі, з можливістю створювати нові організації;

11. Ролі – сторінка, що відображає інформацію про ролі, доступні в системі, з можливістю створення нових ролей та призначення дозволів цим ролям;

12. Налаштування звукових сповіщень – сторінка, що відображає і дозволяє редагувати налаштування звукових сповіщень, такі як включення чи вимкнення сповіщень та час їх спрацьовування до закінчення терміну придатності продукції;

13. Органолептичні властивості – сторінка для реєстрації продукції, з можливістю оцінки властивостей продукції за такими параметрами, як запах, колір, вигляд, консистенція, температура та час зберігання.

2.2. Проектування інтерфейсу користувача

Під час проектування користувацьких інтерфейсів зовнішній дизайн системи може бути представлений у різних формах: вайрфреймів, прототипів або мокапів [9]. Ці методи відрізняються за рівнем точності уявлення, що безпосередньо впливає на вартість виконання. Необхідний рівень точності визначається конкретною ситуацією, враховуючи складність системи, її призначення та інші фактори.

Вайрфрейм є дизайном з низьким рівнем точності, який демонструє основну структуру інформації та надає базову візуалізацію взаємодії між інтерфейсом і користувачем.

Прототипи відрізняються середньою або високою точністю та інтерактивністю, вони використовуються для моделювання кінцевої взаємодії між користувачем і інтерфейсом.

Мокапи надають найвищу точність уявлення дизайну, часто являючи собою графічні начерки або навіть завершений графічний дизайн. Мокапи забезпечують статичну візуалізацію контенту.

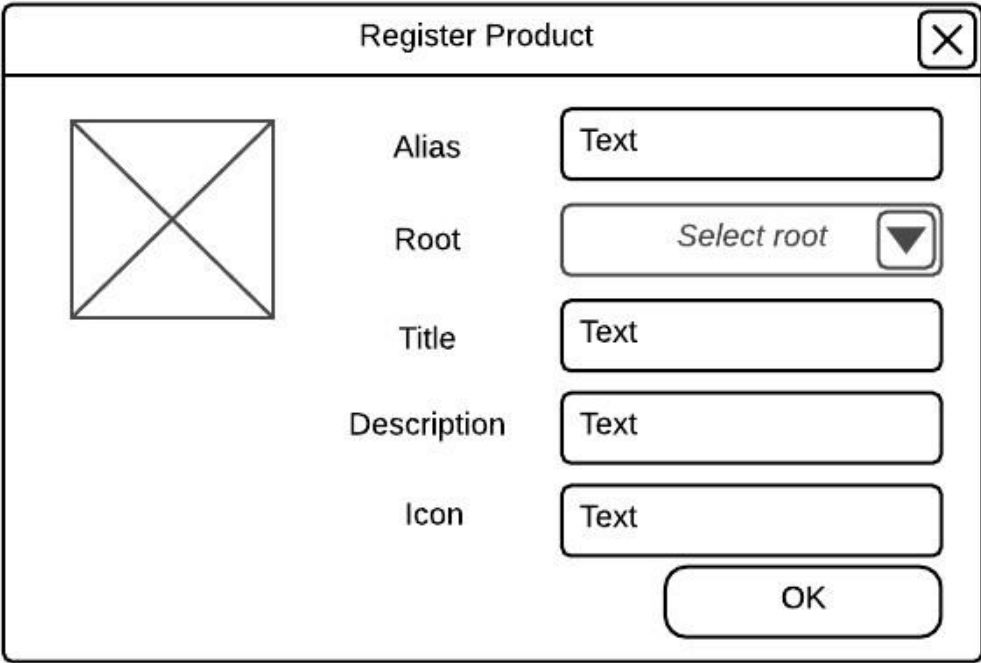
Оскільки розроблювана система не передбачає складного графічного інтерфейсу та на етапі розробки першої версії не є обов'язковим детальне UX-тестування, для проектування інтерфейсу був вибраний підхід використання вайрфреймів.

У правому верхньому кутку екрану розміщений логотип профілю користувача, що увійшов до системи, натискання на який відкриває доступ до функцій управління профілем.

У центрі екрана відображається вміст сторінок, здебільшого у вигляді списків сутностей. Наприклад, на сторінці "Продукти" представлений список продуктів, на сторінці "Компанії" – список компаній, і так далі. Також на сторінках є поля для пошуку, кнопки для оновлення та додавання сутностей.

Додавання та редагування сутностей здійснюється через діалогові вікна, основні з яких описані нижче. На рисунку 2.3 показано діалог реєстрації продукту, що містить такі поля:

- Alias – короткий псевдонім додаваного продукту;



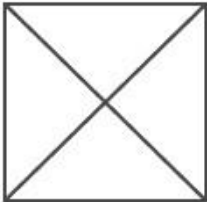
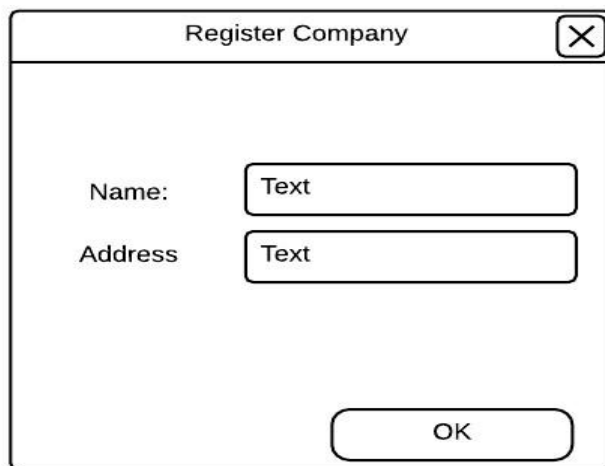
Register Product		
	Alias	Text
	Root	Select root ▼
	Title	Text
	Description	Text
	Icon	Text
OK		

Рис. 2.3 – Діалог реєстрації нового продукту

- Root – список кореневого проекту у разі реєстрації;
- Title – назва;

- Description – опис;
- Icon – шлях до іконки.

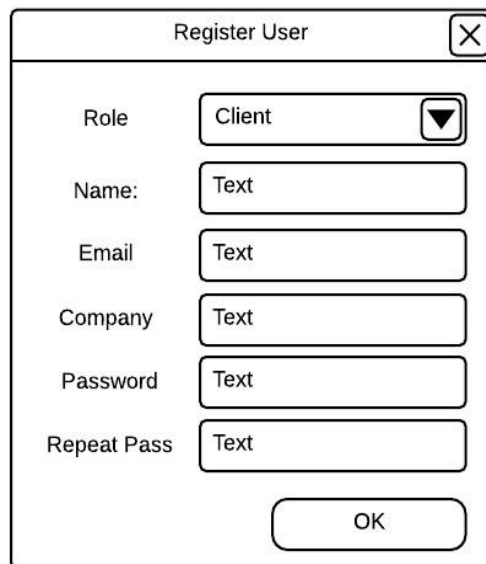
На рисунку 2.4 зображено діалог реєстрації нової компанії.



The image shows a dialog box titled "Register Company" with a close button (X) in the top right corner. Inside the dialog, there are two text input fields. The first is labeled "Name:" and the second is labeled "Address". Both fields contain the placeholder text "Text". At the bottom right of the dialog is an "OK" button.

Рис. 2.4 – Сторінка реєстрації нової компанії

На рисунку 2.5 зображено сторінку реєстрації нового користувача в системі.



The image shows a dialog box titled "Register User" with a close button (X) in the top right corner. Inside the dialog, there are several input fields. The first is a dropdown menu labeled "Role" with "Client" selected and a downward arrow button. Below it are five text input fields labeled "Name:", "Email", "Company", "Password", and "Repeat Pass", each containing the placeholder text "Text". At the bottom right of the dialog is an "OK" button.

Рис. 2.5 – Діалог додавання нового користувача

На рисунку 2.6 представлено сторінку оформлення проблемних товарів після здійснення продажу продукту клієнту та формування відповідних скарг.

Register Order

Company Select company ▼

Buyer Select buyer ▼

Select	Name	Amount	Cost
☒	Sandbox	1	
☐	MySelfie	0	
☐	Floorium	0	

OK

Рис. 2.6 – Діалог реєстрації проблемного замовлення

У вікні є елементи для вибору клієнта та компанії, яку він представляє. Також вікно містить таблицю, в якій можна вибирати продані продукти, вказувати кількість одиниць та їх термін придатності.

2.3 Структура JSON даних

Розроблювана клієнтська частина системи відстеження термінів придатності продукції отримує дані від серверної частини у вигляді JSON-структур і не має прямого доступу до баз даних системи.

В таблиці 2.1 наведені основні JSON-дані сервісу автентифікації, які будуть використовуватися для реалізації веб-інтерфейсу.

Таблиця 2.1 – Структура JSON-даних сервісу автентифікації

Найменування	Тип	Опис
login	string	логін
tenantid	number	ідентифікатор організації
accountid	number	ідентифікатор акаунта
currentpassword	string	поточний пароль
newpassword	string	новий пароль

В таблиці 2.2 наведені основні JSON-дані сервісу керування акаунтами, які будуть використовуватися для реалізації веб-інтерфейсу.

Таблиця 2.2 – Структура JSON-даних сервісу керування акаунтами

Найменування	Тип	Опис
firstname	string	ім'я
lastname	string	прізвище
middlename	string	по батькові
isBlocked	boolean	флаг блокування акаунта
tenantName	string	назва організації
roles	array	ролі
permissions	array of string	дозволи

В таблиці 2.3 наведена основна частина JSON-даних сервісу відстеження термінів придатності продукції, які будуть використовуватися для реалізації веб-інтерфейсу.

Таблиця 2.3 – Структура JSON-даних сервісу відстеження термінів придатності продукції

Найменування	Тип	Опис
id	number	ідентифікатор
name	string	найменування
shopName	string	найменування цеха
isPiece	boolean	штучність продукції
barcodes	array	штрих-коди продукції
productId	number	ідентифікатор продукції
lifetimeInMinutes	number	термін придатності в хвилинах
barcode	string	штрих-код продукції
trackedProductCount	number	вага продукції
temperature	object	органолептичні властивості продукції
organolepticProps	string	органолептичні властивості
organolepticProps: color	number	колір
organolepticProps: appearance	number	вигляд
organolepticProps: smell	number	запах
organolepticProps: consistency	number	консистенція

На рисунку 2.7. представлено структуру даних для кожної з підсистем у вигляді схеми даних.

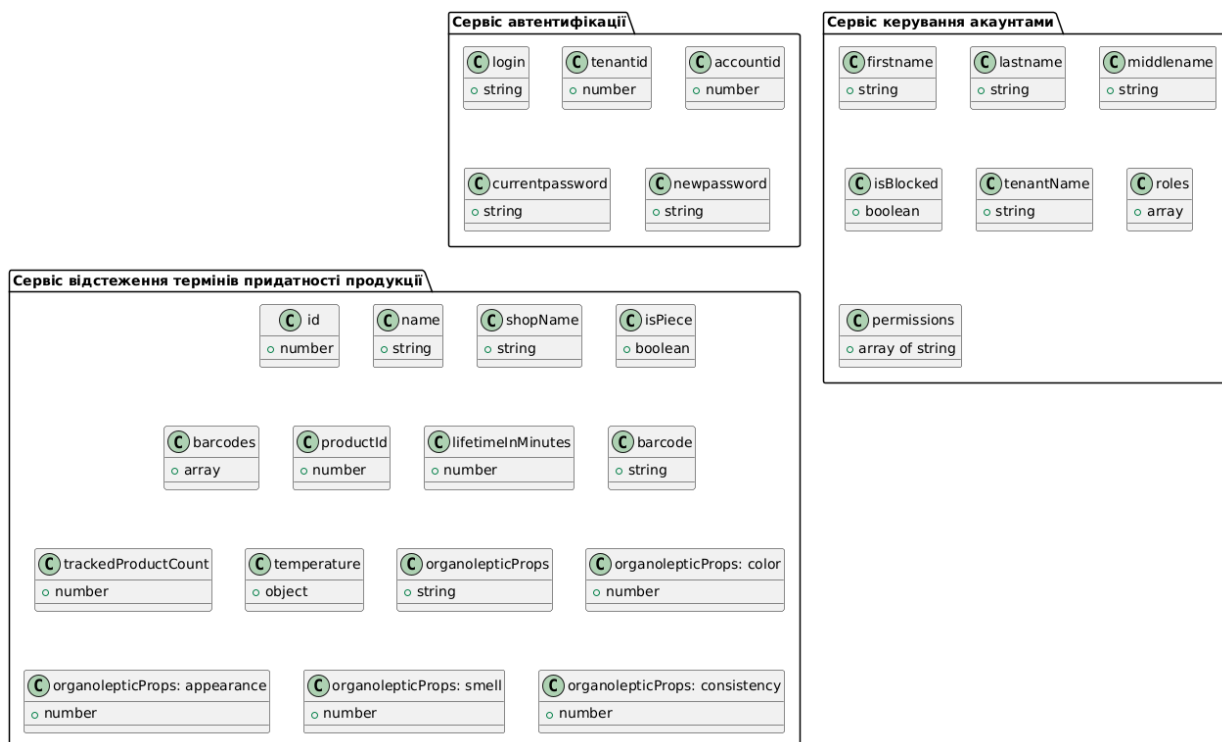


Рис. 2.7 – Схема аналізованих даних

2.4. Висновки до другого розділу

На основі розгляду аналогічних систем з функціоналом відстеження термінів придатності продукції та технологій розробки, були розроблені функціональні та нефункціональні вимоги до клієнтської частини системи, підготовлений дизайн інтерфейсу системи відстеження терміну придатності продукції, розглянуті варіанти використання системи.

Розроблена архітектура веб-системи, визначені компоненти системи, їх взаємодія та функціонал. Спроектowana структура даних у форматі JSON, яка забезпечує гнучкий обмін інформацією та адаптацію до нових вимог.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Вибір засобів реалізації інформаційної системи

Розробка системи була реалізована з використанням наступних технологій:

Visual Studio Code 1.84.2 (рисунок 3.1.) – редактор коду, що надає широкі можливості для розробки програмного забезпечення.

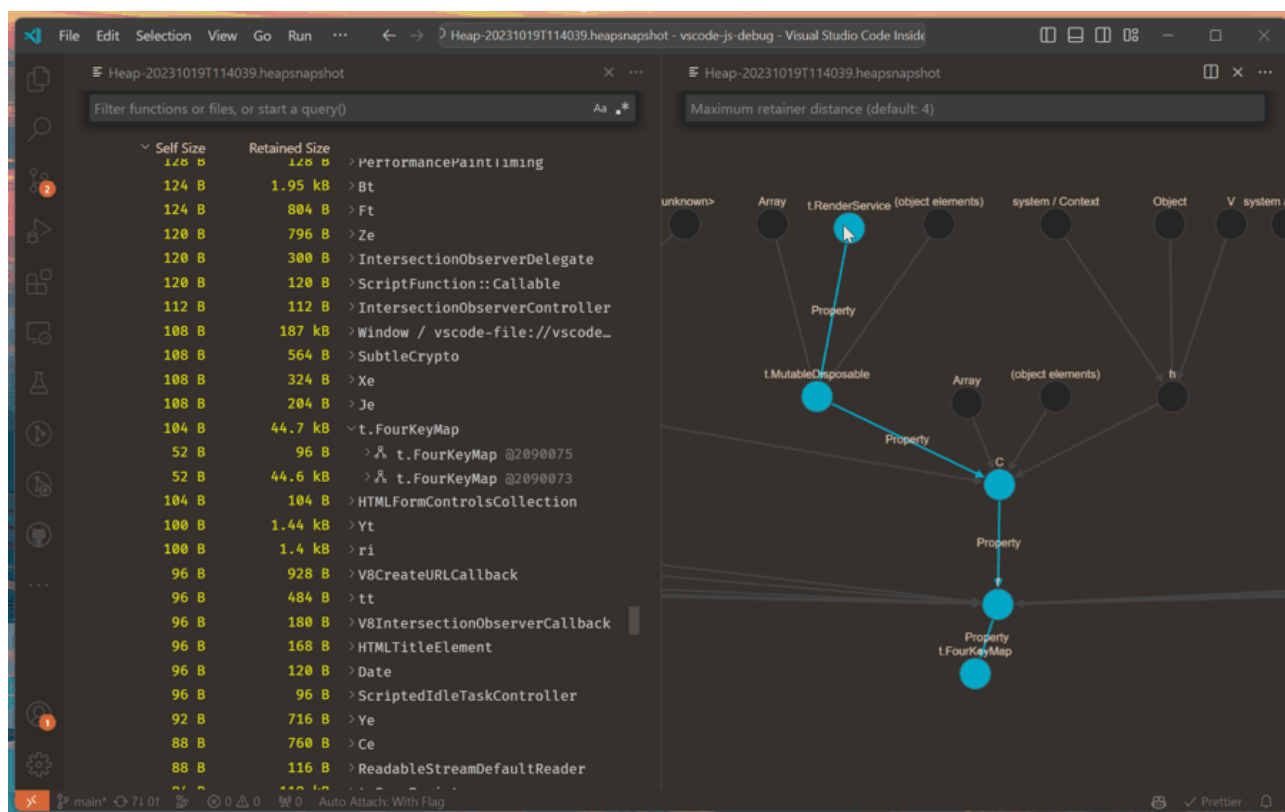


Рис. 3.1 – Серовище розробки Visual Studio Code 1.84.2

React 18.2.0 – бібліотека JavaScript для розробки користувацьких інтерфейсів, що використовує компонентний підхід.

Node.js 18.17.1 (рисунок 3.2) – середовище виконання JavaScript, яке використовується для запуску серверного коду.

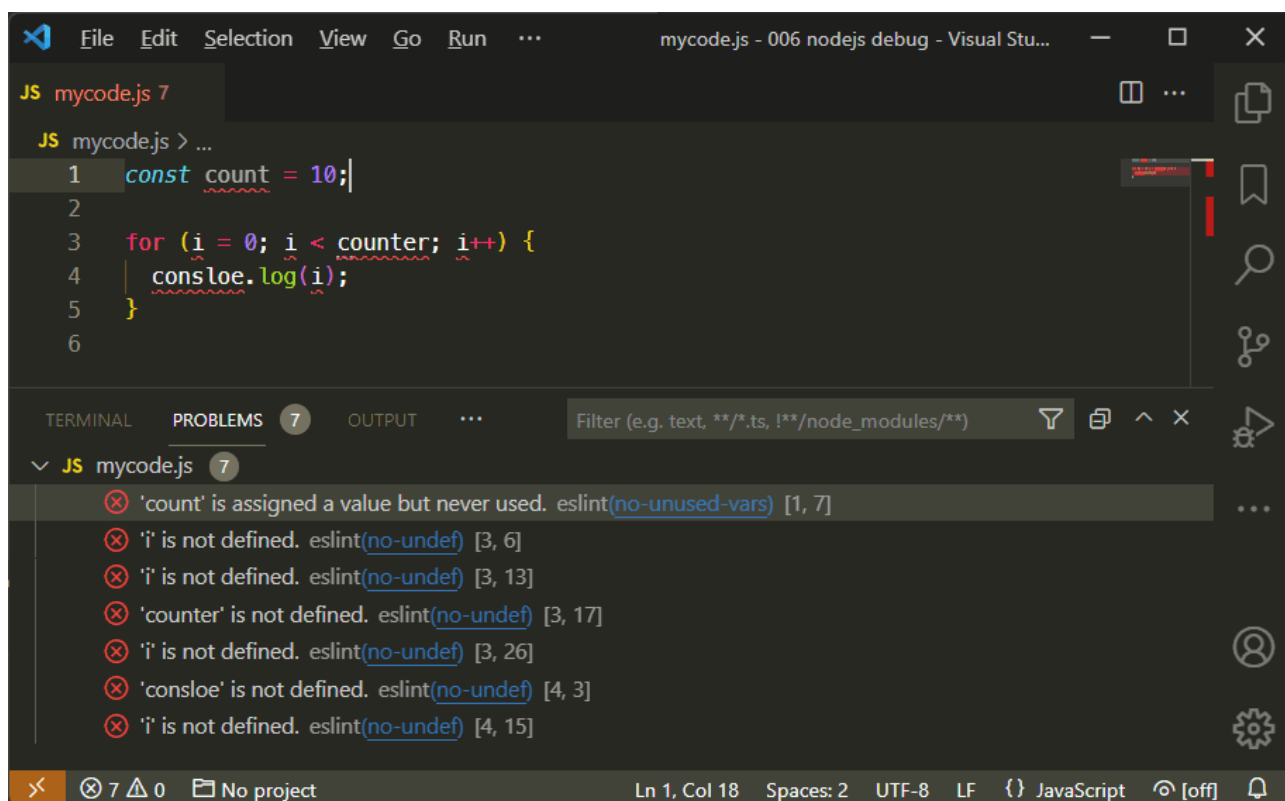


Рис. 3.2 – Node.js

MobX 6.9.0 – бібліотека для управління станом додатку в React.

Axios 1.4.0 – бібліотека для виконання HTTP-запитів з JavaScript.

Clsx 1.2.1 – утиліта для генерації класів для React-компонентів на основі умов.

Lodash 4.17.20 – утиліти JavaScript для роботи з масивами, об'єктами та функціями.

Moment 2.29.4 – бібліотека для зручної роботи з датами та часом в JavaScript.

React-table 7.8.0 (рисунок 3.3.) – бібліотека для створення гнучких таблиць у React.

Vite 3.2.0 – плагін для перетворення SVG в React-компоненти.

ZXing 0.20.0 – бібліотека для роботи з штрихкодами та QR-кодами в JavaScript, заснована на класичних алгоритмах комп'ютерного зору та обробки зображень.

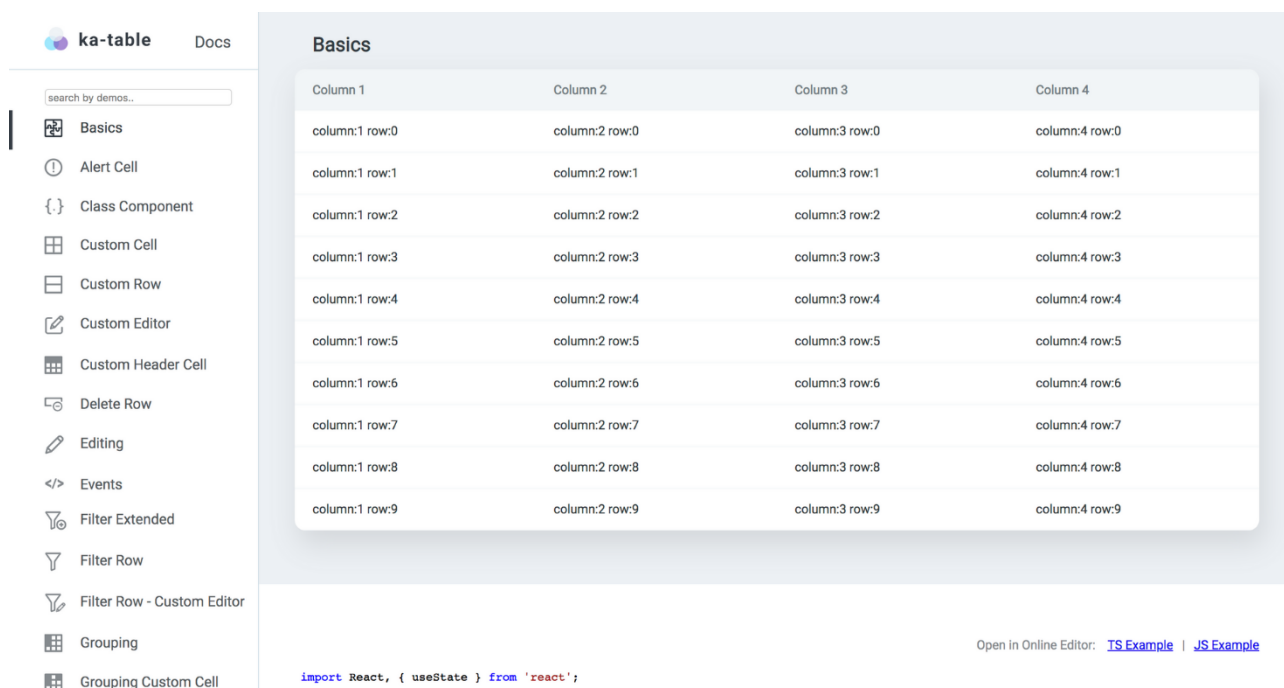


Рис. 3.3 – React-table 7.8.0

HTML – стандартна мова розмітки, що використовується для створення та структурування веб-сторінок.

Sass – метамова CSS з розширеним функціоналом, який включає змінні, вкладені правила та інші покращення.

TypeScript – мова програмування, яка розширює можливості JavaScript шляхом додавання статичної типізації.

3.2 Програмна реалізація взаємодії з серверною частиною реалізація React модулів

JWT (JSON Web Token) — це компактний і самодостатній спосіб представлення інформації між двома сторонами у вигляді об'єкта JSON. Він використовується для передачі тверджень (claims) між клієнтом і сервером.

JWT складається з трьох частин: заголовка (header), корисного навантаження (payload) та підпису (signature).

JWT застосовується в механізмах аутентифікації та авторизації веб-застосунків. Робота з JWT включає кілька етапів:

1. Отримання JWT. Коли користувач успішно автентифікується на сервері (наприклад, ввівши правильні облікові дані), сервер створює JWT для цього користувача. Цей токен повертається у відповіді сервера після успішної аутентифікації.

2. Зберігання JWT. Отриманий JWT зберігається на клієнтській стороні для подальшого використання. Він зберігається у локальному сховищі (localStorage) або в куки (cookies).

3. Включення JWT у кожен запит до сервера. Після того, як JWT отримано і збережено, він має бути включений у кожен запит до сервера. Це робиться шляхом додавання заголовка Authorization зі значенням Bearer <JWT> у кожен HTTP-запит.

4. Перевірка JWT на сервері. При отриманні запиту на сервері сервер має перевірити переданий JWT. Це включає перевірку підпису токена, а також перевірку його терміну дії.

5. Обробка і оновлення JWT. JWT має термін дії, після якого він стає недійсним. За потреби сервер може повернути новий JWT у відповіді на запит користувача (наприклад, якщо поточний JWT вичерпав термін дії або потребує оновлення з інших причин), і клієнт повинен оновити збережений JWT на новий.

6. Вихід з акаунта і видалення JWT. При виході користувача з системи (logout) або після закінчення терміну дії токена, JWT має бути видалено зі сховища на клієнтській стороні для безпеки. Приклад роботи з токеном наведено в лістингу на рисунку 3.4.

```

1 function setLoggedOut() {
2     tokenProvider.setTokenPair(null, null);
3 }
4
5 function getAuthToken() {
6     return tokenStorage.getTokenValue();
7 }
8
9 function getRefreshToken() {
10    return refreshTokenStorage.getTokenValue();
11 }

```

Рис. 3.4 – Приклад роботи з токеном

У цьому прикладі показано видалення токена, його отримання та оновлення.

Для реалізації клієнтської частини веб-системи сервером надається API системи. Запит до сервера формується у вигляді об'єкта формату JSON і відправляється асинхронно з використанням ключового слова **await**, щоб програма дочекалася завершення запиту перед продовженням виконання інших дій у функції. Була використана бібліотека **Axios**, яка надає зручний інтерфейс для відправки запитів. Приклад запиту наведено на рисунку 3.5.

```

1 async function loadProductsAsync () {
2     try {
3         const { data } = await api.get(loadProductsUrl)
4         setProducts(data);
5     } catch (error: any) {
6         errorBoundary.setErrorInfo({
7             code: error.response.status,
8         });
9     }
10 }

```

Рис. 3.5 – Приклад запиту до серверної частини

Наводиться приклад коду, який реалізує запит до серверної частини для отримання списку продукції. Загальний вигляд запиту до сервера представлений у Додатку А.

Розглянемо запит на отримання інформації про продукцію та відповідь сервера

Метод API GET/POST

- header: токен
- RESPONSE: JSON-об'єкт

Методи API. Всі методи поділяються на такі блоки:

1. products – методи блоку «продукція»;
2. products log – методи блоку «zareestrovana produkciya»;
3. shops – методи блоку «цехи»;
4. tracked products – методи блоку «vidslidkovuvana produkciya»;
5. accounts – методи блоку «akaunti»;
6. roles – методи блоку «rolі»;
7. tenants – методи блоку «organizacii».

Загалом використовується:

- 4 методи блоку products,
- 3 методи блоку products log,
- 3 методи блоку shops,
- 4 методи блоку tracked products,
- 6 методів блоку accounts,
- 5 методів блоку roles,
- 3 методи блоку tenants.

У додатку наведено лістинг запиту на отримання інформації про продукцію та відповідь сервера. На рисунку 3.5. представлено скріншот з даними цього запиту.

	id	name	category	price	stock	description	product_id	action	timestamp	details	location
1	1	Product A	Category 1	100.0	50.0	Description of Product A					
2	2	Product B	Category 2	150.0	30.0	Description of Product B					
3	1						1.0	Added	2025-05-03T12:00:00	Product A added to the inventory	
4	2						2.0	Updated	2025-05-02T10:30:00	Price of Product B updated	
5	1	Shop 1									Location 1
6	2	Shop 2									Location 2
7	1						1.0				
8	2						2.0				
9	1										
10	2										
11	1	Admin									
12	2	User									
13	1	Organization 1									
14	2	Organization 2									

Рис. 3.6 – Приклад запиту і відповіді у формі таблиці

На рисунку 3.7 представлено запит на додавання продукції, а на рисунку 3.8 відповідно згенеровану відповідь.

```
1 POST /tracked-products
2 {
3   header: JSON Web Token,
4   body:{
5     "productId": 6398,
6     "lifetimeInMinutes": 720,
7     "barcode": "1234567896",
8     "trackedProductCount": 10,
9     "temperature": 100,
10    "organolepticProps": {
11      "color": 3,
12      "appearance": 2,
13      "smell": 5,
14      "consistency": 4}}
15 RESPONSE:{"id":6}
16
```

Рис. 3.7 – Приклад запиту на додавання продукції

Результати API Запиту на Трекінг Продукції

Відповідь на запит:

```
{  "id": 6
}
```

Деталі запити:

Параметр	Значення
productId	6398
lifetimeInMinutes	720
barcode	1234567896
trackedProductCount	10
temperature	100
organolepticProps:	• Color: 3 • Appearance: 2 • Smell: 5 • Consistency: 4

Ідентифікатор трекінгової продукції: 6

Рис. 3.8 – Приклад відповіді на запит

У заголовку також передається токен клієнта. Тіло запиту містить інформацію про id типу реєструємої продукції, її термін придатності, код, температуру, кількість та органолептичні властивості, такі як запах, колір, консистенція та вигляд.

Односторінкове додаток (SPA) завантажує лише одну HTML-сторінку і оновлює її вміст динамічно, що покращує користувацький досвід. У React для цього створюються компоненти, які відображають різні частини інтерфейсу залежно від стану додатка. Перерисовка компонентів автоматично відбувається при зміні стану або пропсів. Розробка на React базується на створенні компонентів, які керуються станом (state) і пропсами (props), що дозволяє легко керувати даними і оновлювати інтерфейс без перезавантаження сторінки. У лістингу на рисунку 3.9 наведено приклад стейта для зберігання інформації про код продукції.

```

1  import { makeAutoObservable } from "mobx";
2  type BarcodeInfoType = {
3      id: number;
4      name: string;
5      barcodes: Array<number>;
6  };
7  class BarcodeInfoState {
8      private _barcodeInfo: Array<BarcodeInfoType> | null = null;
9      private _barcodeNumber: string = ``;
10     private _isOpen = false;
11     private _ProductName: string = ``;
12     constructor() {
13         makeAutoObservable(this);
14     }
15     get isOpenModal() {
16         return this._isOpen;
17     }
18     setProductName(ProductName: string) {
19         this._ProductName = ProductName;
20     }
21     get ProductName() {
22         return this._ProductName;
23     }
24     setOpenModal(isOpen: boolean) {
25         this._isOpen = isOpen;
26     }
27     get barcodeInfo() {
28         return this._barcodeInfo;
29     }
30     get barcodeNumber() {
31         return this._barcodeNumber;
32     }
33     getInfoBarcode(info: Array<BarcodeInfoType> | null) {
34         this._barcodeInfo = info;
35     }
36     updateBarcodeNumber(updatedNumber: string) {
37         this._barcodeNumber = updatedNumber;
38     }
39     export default BarcodeInfoState;

```

Рис. 3.9 – Приклад стейта для зберігання інформації про код продукції

Цей приклад коду представляє клас `BarcodeInfoState`, який використовується для управління станом інформації про штрих-коди. У класі визначені приватні поля: `_barcodeInfo`, `_barcodeNumber`, `_isOpen` та `_ProductName`, що відповідають за інформацію про штрих-коди, номер штрих-коду, статус модального вікна та назву продукту відповідно. Клас також містить методи для доступу до цих полів (геттери) та для їх оновлення (сеттери). Метод `getInfoBarcode` використовується для отримання інформації про штрих-коди, а `updateBarcodeNumber` — для оновлення номера штрих-коду. Конструктор класу використовує функцію `makeAutoObservable` з пакету `mobx`.

для автоматичного створення геттерів і сеттерів для всіх полів класу, що полегшує управління станом.

3.3 Програмна реалізація адаптивного інтерфейсу

Для реалізації адаптивного інтерфейсу використовувалися різні технології та підходи.

HTML та CSS медіа-запити, які дозволяють налаштовувати стилі залежно від характеристик пристрою, таких як розмір екрану або орієнтація.

Flexbox та Grid Layout, що надають зручні інструменти для управління розташуванням елементів на веб-сторінці, допомагаючи створити адаптивний макет.

Mobile-first підхід, який є стратегією розробки, що починається з оптимізації для мобільних пристроїв і потім покращує відображення для більших екранів.

Розбиття на компоненти, що є принципом розробки, передбачаючи поділ інтерфейсу на незалежні компоненти для легкого масштабування та повторного використання коду.

BEM підхід, що є методологією іменування класів CSS, заснованою на блоках, елементах та модифікаторах, для створення зрозумілої та підтримуваної структури стилів.

На рисунку 3.10 представлено приклад адаптивної верстки сторінки типів продукції.

```

1  .all-products-list-page {
2    $this: &;
3    padding: 20px 20px 0;
4    &__head {
5      display: flex;
6      flex-direction: column;
7      justify-content: flex-end;
8      margin-bottom: 30px;
9    }
10   &__button {
11     margin-top: 20px;
12     margin-left: auto;
13     width: 120px;
14     text-align: center;
15     @include tablet {
16       width: auto;}}}

```

Рис. 3.10 – Приклад адаптивної верстки сторінки типів продукції

Цей приклад коду є прикладом адаптивної верстки компонента сторінки типів продукції, до якого далі підключаються інші компоненти, такі як таблиці та фільтри. У цьому прикладі використовуються Mobile-first підхід, Flexbox та BEM підхід.

3.4 Тестування інформаційної системи

Клієнтська частина тестується за допомогою юніт-тестів, E2E-тестів та компонентних тестів у інструменті автоматизації зовнішнього тестування Cypress.

Юніт-тести: для тестування окремих модулів або компонентів використовуються юніт-тести. Метою цих тестів є перевірка коректності роботи окремих функцій, класів чи модулів в ізоляції від інших частин програми.

E2E-тести: для перевірки роботи додатку на рівні користувацького інтерфейсу використовуються E2E-тести (end-to-end). Вони імітують дії користувача в браузері та перевіряють, що додаток поводить себе як очікується в реальних умовах.

Компонентні тести: тести, які перевіряють окремі компоненти користувацького інтерфейсу. Вони перевіряють, що компоненти правильно відображаються та взаємодіють із зовнішніми даними чи подіями.

Розробка веб-системи проводилася з використанням методу TDD [26], який передбачає написання тестів перед написанням коду програми. Це дозволяє забезпечити повне покриття коду тестами та покращити його якість завдяки ранньому виявленню помилок.

Таким чином, вся клієнтська частина веб-системи покрита автоматизованими тестами, включаючи юніт-тести для окремих модулів, E2E-тести для перевірки роботи програми в цілому та, можливо, компонентні тести для перевірки окремих компонентів користувацького інтерфейсу.

Наведені приклади тестів. У лістингу на рисунку 3.11 представлено юніт-тест для тестування роботи компонента productsState.

```
const INITIAL_DATA: ProductsInfo = {
  expiredTrackedProductsCount: 0,
  expiringTrackedProductCount: 0,
  list: [
    {
      id: 123456,
      name: `Салат`,
      createdAtUtc: `2023-12-05T10:00:00`,
      expiresAtUtc: 2023-12-05T14:00:00,
      minutesBeforeExpiration: 60,
      status: 0,
      weight: 0,
    },
  ],
};

it(`SHOULD return all products WHEN initialized`, () => {
  productsState.initialize({
    productsInfo: INITIAL_DATA,
  });
  expect(productsState.allProductsInfo.expiredTrackedProductsCount)
    .to.deep.eq(0);
  expect(productsState.allProductsInfo.list)
    .to.has.lengthOf(1);
  expect(productsState.allProductsInfo)
    .to.deep.eq(INITIAL_DATA);
});
```

Рис. 3.11 – Юніт-тест для тестування правильної ініціалізації продуктів у productsState

Цей тест використовує підготовлені дані для ініціалізації в компонент `productsState`. Далі проводиться перевірка даних: що в них немає прострочених продуктів, що завантажені дані лише про один продукт та що завантажені дані повністю відповідають наданим.

У лістингу на рисунку 3.12. представлено E2E-тест для тестування сторінки всієї реалізованої продукції.

Цей тест перевіряє коректність роботи всіх функцій сторінки реалізованої продукції, таких як створення нового типу продукції та його відображення в таблиці.

```
describe('Create a product and make sure it appears in the products table',
() => {
  const code = generateProductCode();
  beforeEach('Authorize', () => {
    cy.authByApi();
    cy.clearProductsByApi();
  });
  after('Clean', () => {
    cy.clearProductsByApi();
  });
  it('GIVEN allProducts page
  WHEN creating a product
  THEN make sure it's in the products table and remove it from the table',
() => {
    AllProductsPage.visit();
    AllProductsPage.createNewProduct(code);
    AllProductsPage.checkNewProduct(code);
  });
});
```

Рис. 3.12 – E2E-тест для тестування сторінки реалізованої продукції

У лістингу на рисунку 3.13. представлено компонентний тест для перевірки компонента сторінки всієї продукції.

```

describe(`AllProductsListPage`, () => {
  it(`SHOULD render all products list WHEN open page`, () => {
    mountComponent();
    cy.getByData(`all-products-list-page`)
      .should(`exist`);
  });
  it(`SHOULD render no products message WHEN there are no products`, () => {
    cy.intercept(`GET`, (`${API_ROOT}/products`), []).as(`call-1`);
    mountComponent();
    cy.getByData(`no-products-message`)
      .contains(`Продукції немає`);
  }
});
it(`SHOULD render add button WHEN open page`, () => {
  mountComponent();
  cy.getByData(`all-products-list-page-button`)
    .should(`exist`);
}
);
}
);

```

Рис. 3.13 – Компонентний тест для перевірки компонента сторінки всієї продукції.

Цей тест перевіряє коректність роботи компонента сторінки реалізованої продукції. Перевіряється правильне відображення елементів сторінки. При змінах у коді запускаються всі тести для перевірки працездатності змін. У таблиці 3.1. наводиться частина звіту по проходженню тестів.

Таблиця 3.1. – Частина звіту по проходженню тестів

Тести	Час	Виконано	Не виконано	Пропущена
BarcodeInfoState	44 ms	2	0	0
BarcodeScanner	10 ms	7	0	0
Clock	69 ms	2	0	0
ErrorModal	27 ms	2	0	0
FilterMenu	178 ms	4	0	0
ProductsList	10 ms	18	0	0
AllProductsListPage	106 ms	3	0	0
CreateProductPage	20 ms	10	0	0

На даний момент розроблено 151 тест, загальний час виконання яких складає 21 секунду. В кінці звітів, якщо всі тести виконані, наводиться підсумкова строка, зображена на рисунку 3.14.

✓ All specs passed!	00:21	151	0	0
---------------------	-------	-----	---	---

Рис. 3.14 – Підсумкова строка звіту виконання тестів.

Система представляє односторінковий веб-додаток зі наступними вкладками:

1. Сторінка для сканування через термінал;
2. Сторінка для сканування камерою;
3. Сторінка відслідковуваної продукції;
4. Сторінка зареєстрованої продукції (журнал);
5. Сторінка списку відділів;
6. Сторінка повного списку продукції.

Сторінка (рисунок 3.15) включає таблицю з даними про товар та відповідні штрихкоди, а також можливість сканувати штрихкоди за допомогою камери.

Список Товарів та їх Штрихкодів

Перелік Товарів

Назва Товару	Штрихкод
Продукт 1	123456789012
Продукт 2	987654321098
Продукт 3	112233445566
Продукт 4	223344556677
Продукт 5	334455667788

Сканування Штрихкоду

Використовуйте камеру для сканування штрихкоду товару:



Рис. 3.15 – Сторінка для сканування камерою

На сторінці є таблиця, яка містить список товарів та їх відповідні штрихкоди. Нижче таблиці розташована секція для сканування штрихкодів через камеру. Сучасний дизайн з чистими лініями, кольоровими фонами для заголовків і таблиць, а також ефектами при наведенні.

Відео для сканування штрихкоду має сучасне виглядове оформлення з кнопкою для початку сканування.

Використовується бібліотека jsQR для сканування штрихкодів через камеру. Кнопка "Почати сканування" запускає доступ до камери, а після знаходження штрихкоду зупиняється сканування і відображається знайдений штрихкод.

У коді використовується API доступу до камери через `navigator.mediaDevices.getUserMedia()` для сканування штрихкоду. За допомогою бібліотеки jsQR реалізується функція для аналізу зображення, що надходить з відеопотоку та виявлення штрихкоду.

На рисунку 3.16 представлено сторінку з функціоналом для опису процесів відслідковування терміну придатності продукції.

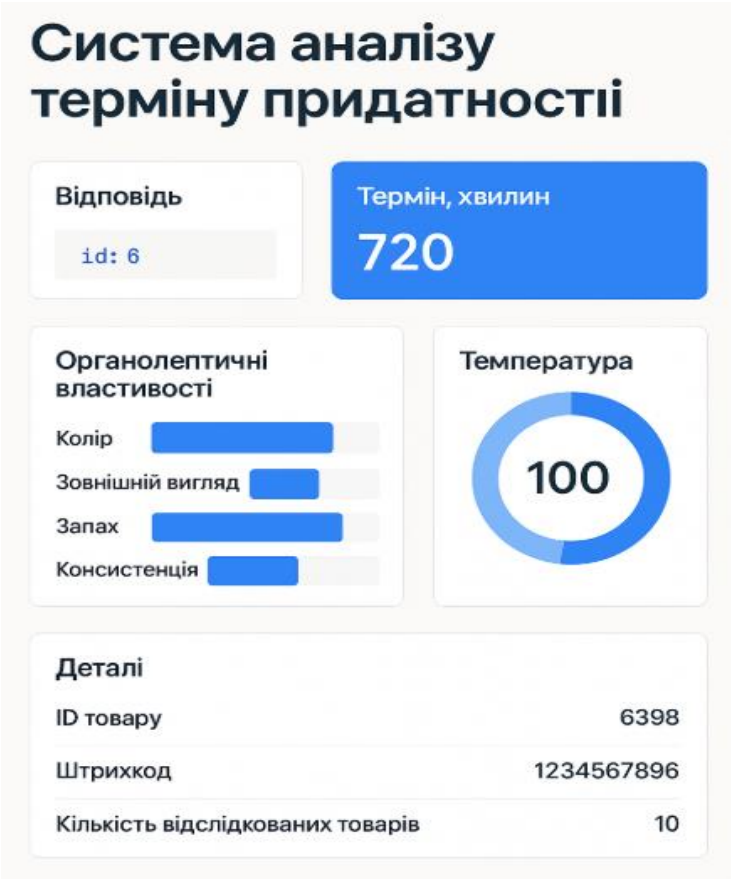


Рис. 3.16 – Сторінка відслідковуваної продукції

Для створення сторінки журналу зареєстрованої продукції, де можна переглядати список товарів, їхні терміни придатності та іншу інформацію, давайте розробимо HTML шаблон з можливістю додавання, перегляду та фільтрації товарів. Для цієї сторінки використаємо таблицю для відображення інформації та прості форми для введення даних.

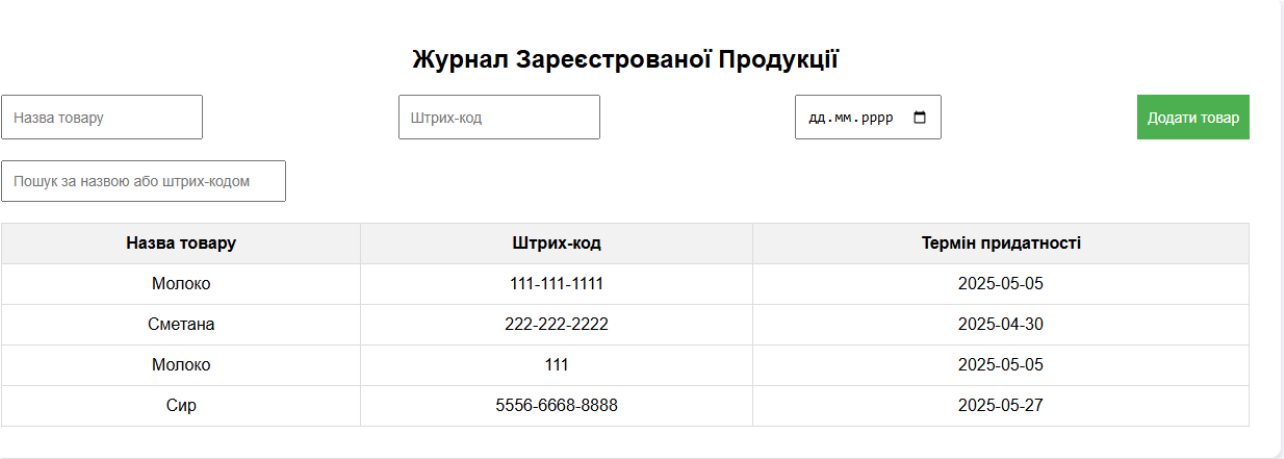


Рис. 3.17 – Сторінка зареєстрованої продукції (журнал)

На рисунку 3.18 представлено вкладку, яка містить інформацію про відділи, можливість їх додавання, те редагування.

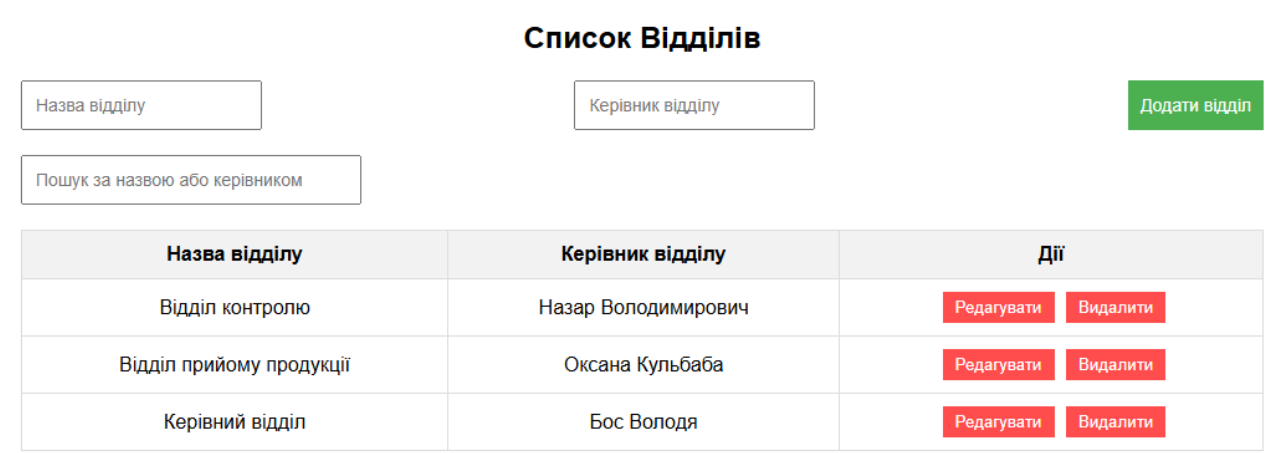


Рис. 3.18 – Сторінка списку відділів

На рисунку 3.19 представлено сторінку яка містить інформацію про повний список аналізованої продукції.

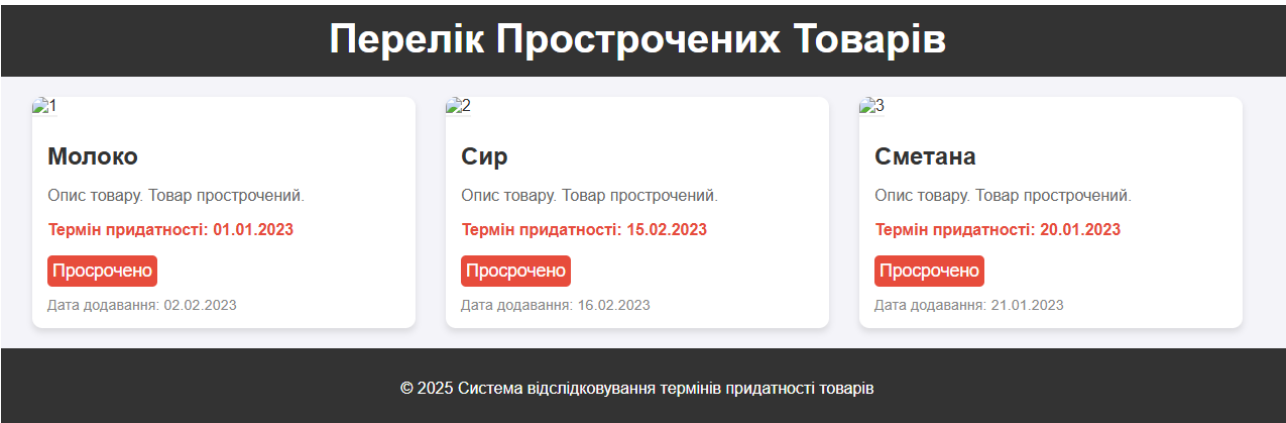


Рис. 3.19 – Сторінка повного списку простроченої продукції

3.5 Висновки до третього розділу

Розроблена веб-система для відстежування термінів придатності продукції з використанням Visual Studio Code, React, Node.js, MobX, Axios,

Clx, Lodash, Moment, React-table, Vite, ZXing, HTML, Sass та TypeScript. Забезпечена безпечна аутентифікація з JWT та адаптивна верстка з використанням Mobile-first, Flexbox та BEM. Axios використано для передачі даних, а ZXing — для сканування штрих-кодів.

При розробці веб-системи з використанням методу TDD (Test-Driven Development) застосовувалися юніт-тести та E2E-тести для перевірки модулів та взаємодії компонентів, а також тестувалися інтерфейси для забезпечення їх коректності. Автоматизація тестування за допомогою Cypress підвищила якість, скоротила час ручного тестування та забезпечила надійність системи.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1. Основні принципи конструювання робочого місця користувача ЕОМ.

Ергономіка (від грецьк. *έρων* наука про пристосування посадових $\square$ у традиційному розумінні $\square$ роботи») обов'язків, робочих місць, обладнання та комп'ютерних програм задля створення найбільш безпечних та ефективних умов праці для людини, виходячи з фізичних і психічних особливостей людського організму.

Більш широке визначення ергономіки, яке було прийняте в 2010 році Міжнародною асоціацією ергономіки (IEA) (Міжнародною ергономічною це наукова дисципліна, що вивчає $\square$ асоціацією), звучить так: «Ергономіка взаємодію людини та інших елементів системи, а також сфера діяльності щодо застосування теорії, принципів, даних і методів цієї науки для забезпечення благополуччя людини та оптимізації загальної продуктивності системи».

З цього визначення випливають такі головні завдання ергономіки:

1. Проведення досліджень, спрямованих на пристосування елементів системи "людина – трудовий процес" до природних фізичних і психічних можливостей працівника.
2. Прагнення до забезпечення таким шляхом умов для максимальної ефективності праці.
3. Прагнення запобігти всім можливим загрозам для здоров'я працівника.
4. Прагнення до оптимальної витрати біологічних ресурсів у процесі праці.

Загальні ергономічні вимоги для організації робочого місця користувача ПЕОМ. Ці вимоги встановлюють основні параметри робочого місця, оснащеного дисплеєм, і враховують особливість виконуваних робіт.

Параметри робочого місця повинні бути наступними.

Площа кабінету, в якому буде проходити робота повинна бути не менш 6 м², а об'єм не менш 24 м³. Для внутрішньої обробки приміщення повинні використовуватися дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі – 0,7-0,8; для стін – 0,5-0,6; для підлоги – 0,3-0,5.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання. Конструкція крісла повинна забезпечувати підтримку раціональної робочої пози під час роботи з відео-дисплейним терміналом (Далі ВДТ) і ПЕОМ, дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини для попередження розвитку втоми працюючого. Поверхня сидіння, спинки та інших елементів стільця (крісла) повинна бути напівм'якою, з покриттям, що не електризується, неслизьке та повітронепроникне, що забезпечує легке очищення від забруднення.

Висота робочої поверхні столу, за відсутності можливості її регулювання повинна складати 725 мм. Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм, не менше 450 мм в глибину на рівні колін і на рівні простягнутої ноги – не менше 650 мм. Робоче місце має бути обладнане підставкою для ніг, має ширину не менше 300 мм, глибину не менше 400 мм, регулювання по висоті в межах 150 мм за кутом нахилу опорної поверхні підставки до 20 градусів.

Відстань від очей користувача до екрану дисплея має становити 500-700 мм. Кут зору 10-20°, але не більше 40°; кут між верхнім краєм дисплея і рівнем очей користувача має становити не менше 10°. Кращим є розташування екрану перпендикулярно до лінії зору користувача.

Робочі місця по відношенню до світлових прорізів повинні розташовуватися не ближче 3 м так, щоб природне світло падало збоку, переважно зліва. Освітленість також впливає на стан здоров'я і працездатність людини. У відповідності зі СНіП 11-4-79 встановлені наступні вимоги до освітленості:

Для штучного освітлення:

- Комбіноване освітлення – освітленість 1500 лк;
- Загальне освітлення – освітленість 400 лк.

Для природного освітлення:

- Верхнє або комбіноване освітлення – коефіцієнт природної освітленості (далі КПО) 10%;
- Бічне освітлення – КПО 3.5%.

Для суміщеного освітлення:

- Верхнє або комбіноване освітлення – КПО 3-6%;
- Бічне освітлення – КПО 1.1-2%.

До основних показників, що визначають умови здорової роботи, належать: фон, контраст об'єкта з фоном, видимість, показник осліпленості, коефіцієнт пульсації освітленості.

Фон характеризується коефіцієнтом відбиття. Контраст об'єкта з фоном (К) характеризується співвідношенням яскравості розглянутого об'єкта (точки, лінії, знаки) і фону. Оскільки роботи користувача ПЕОМ відносяться до категорії 1а – легкі фізичні роботи (роботи проводяться сидячи і супроводжуються незначним фізичним напруженням, з енерговитратами до 120 ккал / годину), необхідно дотримуватися наступних норм: коефіцієнт відображення більше 0,4, тобто світлий фон; контраст об'єкта з фоном великий і середній при К більше 0,2 (згідно СНіП 11-4-79).

У полі зору користувача ПЕОМ має бути забезпечений відповідний розподіл яскравості. Відношення яскравості екрана до яскравості оточуючих його поверхонь не повинно перевищувати у робочій зоні 3:1 (СНіП 11-4-79). У зв'язку з цим дисплей ПЕОМ повинен відповідати наступним вимогам:

- Яскравість свічення екрану не менше 100 кд/м;
- Мінімальний розмір світної точки для кольорового дисплея не більше 0,6 мм ;
- Контрастність зображення знаку – не менше 0,8;
- Низькочастотне тремтіння зображення в діапазоні 0,05-1,0 Гц повинно знаходитися в межах 0,1 мм;

- Екран повинен мати покриття антивідблиску;
- Відеомонітор повинен бути обладнаний поворотним майданчиком, що дозволяє переміщати відеотермінал в горизонтальній і вертикальній площинах в межах 130-220 мм і змінювати кут нахилу на 10-15 мм.

Коефіцієнт відбиття світла матеріалами і обладнанням всередині приміщень має велике значення для освітлення: чим більше світла відбивається від поверхонь, тим вище освітленість. Коефіцієнт відображення відповідно повинен бути для: стелі 60-70%, стін 40-50%, підлоги 30%, для інших поверхонь 30-40%.

Результати досліджень показують, що найбільшою мірою негативний фізіологічний вплив на операторів ПК пов'язаний з дискомфорними зоровими умовами через неправильно спроектоване освітлення. Згідно СНіП II-4-79 освітленість на горизонтальній площині робочого місця оператора ЕОМ повинна складати 400 лк при висоті цієї площини 0,8 м над підлогою.

4.2 Управління охороною праці.

Управління охороною праці - це підготовка, прийняття та реалізація рішень щодо здійснення організаційних, технічних, санітарно-гігієнічних і лікувально-профілактичних заходів, спрямованих на забезпечення здоров'я та працездатності людини під час праці.

Система управління охороною праці (СУОП) є складовою частиною загальної системи керування підприємством. При автоматизованій системі управління, управління охороною праці є її складовою частиною, або підсистемою. Управління охороною праці передбачає участь в цьому процесі практично всіх служб і підрозділів підприємства.

Об'єктом управління є діяльність структурних підрозділів, яка спрямована на створення безпечних і здорових умов праці. Управління охороною праці на підприємстві в цілому здійснює його керівник (власник), а в підрозділах (цехах, відділах, службах) - їх керівники або головні фахівці.

Координує всю цю діяльність служба охорони праці. Задачі служби охорони праці та її функції викладені в "Типовому положенні про службу охорони праці", яке затверджено наказом Комітету Держнаглядохоронпраці від 3 серпня 1993 р. № 73.

Служба охорони праці створюється на підприємствах, установах, організаціях незалежно від форми власності та видів діяльності для виконання правових, організаційно-технічних, санітарно-гігієнічних, соціально-економічних і лікувально-профілактичних заходів, спрямованих на запобігання нещасним-випадкам, професійним захворюванням і аваріям в процесі праці. Для здійснення вищезазначених цілей служба охорони праці повинна вирішувати такі завдання:

- а) забезпечувати безпеку виробничих процесів, устаткування, будівель і споруд;
- б) забезпечувати працюючих засобами індивідуального та колективного захисту;
- в) здійснювати професійну підготовку і підвищення кваліфікації працівників з питань охорони праці, вести пропаганду безпечних методів праці;
- г) забезпечувати оптимальні режими праці і відпочинку працюючих;
- д) вимагати професійного добору виконавців для певних видів робіт.

Служба охорони праці створюється на підприємствах, установах та організаціях із числом працюючих 50 чоловік і більше. В організаціях з меншою кількістю працюючих цю службу може представляти інженер, призначений за сумісництвом. Працівники служби охорони праці повинні мати вищу спеціальну освіту з охорони праці, а також практичний досвід у відповідній галузі виробництва. За важливістю діяльності та оплатою праці вони прирівнюються до працівників провідних відділів та служб підприємства або установи. Підпорядковується служба охорони праці безпосередньо керівнику підприємства (власнику).

У системі управління охороною праці підприємства (СУОП), яку здійснює служба охорони праці разом з керівництвом підприємства, основними

чинниками є: законодавство України про охорону праці і про працю, міжгалузеві і галузеві нормативні акти про охорону праці і "Положення про службу охорони праці".

Основними функціями, що розробляє і втілює служба охорони праці, є :

1. Створення ефективної СУОП, яка б сприяла удосконаленню діяльності кожного структурного підрозділу і кожної посадової особи.

2. Здійснення оперативно-методичного керівництва роботою з охорони праці.

3. Розробка разом з структурними підрозділами заходів щодо забезпечення норм безпеки, гігієни праці та виробничого середовища або їх підвищення, якщо вони досягнуті, а також підготовка розділу "Охорона праці" у колективному договорі.

4. Розробка методики запровадження інструктажу з питань охорони праці і його проведення.

5. Забезпечення працюючих правилами, стандартами, нормами, положеннями, інструкціями та іншими нормативними актами.

6. Проведення паспортизації цехів, дільниць, робочих місць щодо відповідності їх вимогам охорони праці.

7. Здійснення оперативного та поточного контролю за станом охорони праці підприємства.

8. Розслідування, облік, аналіз нещасних випадків, професійних захворювань і аварій, а також розрахунок шкоди від цих подій.

9. Участь у підготовці та складанні статистичних звітів підприємства з питань охорони праці.

10. Розробка перспективних та поточних планів роботи підприємства щодо створення безпечних та нешкідливих умов праці.

11. Планування та контроль витрат коштів на охорону праці з фонду охорони праці.

12. Пропаганда та агітація безпечних та нешкідливих умов праці шляхом проведення консультацій, конкурсів, бесід, лекцій, наочної агітації та роботи методичного кабінету.

13. Організація навчання, підвищення кваліфікації та перевірки знань з питань охорони праці посадових осіб.

14. Участь в роботі комісії з питань охорони праці підприємства та допомога в опрацюванні необхідних матеріалів та реалізації її рекомендацій.

15. Участь в комісіях по введенню в дію цехів, дільниць, нового устаткування або після його капітального ремонту.

16. Забезпечення працюючих колективними та індивідуальними засобами захисту від шкідливих: та небезпечних факторів виробництва, лікувальнопрофілактичним харчуванням, миючими засобами, санітарнопобутовими приміщеннями, надання передбачених законодавством пільг і компенсацій, пов'язаних із важкими і шкідливими умовами праці.

17. Контроль за дотриманням вимог трудового законодавства щодо використання праці неповнолітніх, інвалідів та жінок, проходженням попередніх, періодичних, щорічних обов'язкових та інших передбачених відповідними документами медичних оглядів працівниками підприємства.

18. Контроль за дотриманням чинного законодавства, міжгалузевих, галузевих та інших нормативних актів, виконання посадових інструкцій, проведення інструктажів на робочому місці, виконання приписів органів державного нагляду, наказів, розпоряджень, а також заходів до усунення причин нещасних випадків і аварій, які зазначені в актах розслідувань.

19. Контроль за відповідністю нормативним актам про охорону праці машин, механізмів, устаткування, транспортних засобів, технологічних процесів, засобів протиаварійного колективного та індивідуального захисту працюючих, наявність технологічної документації на робочих місцях.

Для виконання вищевказаних функцій служба охорони праці повинна мати відповідну інформаційну базу, засоби зв'язку, сучасну оргтехніку, комп'ютерне забезпечення і висококваліфікований інженерний склад

працівників. Крім того, служба охорони праці повинна мати засоби впливу на виробничу діяльність підприємства.

Такий вплив передбачений "Положенням про службу охорони праці". Так, працівники служби охорони праці мають право видавати керівникам підприємств, установ, організацій та їх підрозділам обов'язкові для виконання приписи щодо усунення наявних недоліків. Припис спеціаліста з охорони праці, у тому числі про зупинення робіт, може скасувати в письмовій формі лише посадова особа, якій підпорядкована служба охорони праці.

Посадові особи мають вимагати усунення від роботи працівників, які не пройшли медичний огляд, навчання, інструктаж, перевірку знань з охорони праці, чи не мають допуску до відповідних робіт, чи порушують нормативні акти про охорону праці, надсилати керівнику підприємства подання про притягнення до відповідальності працівників, що порушують вимоги щодо охорони праці.

Окрім адміністративних заходів рекомендується принцип матеріального заохочення працівників, які сумлінно ставляться до виконання виробничих обов'язків і беруть активну участь у підвищенні безпеки та поліпшенні умов праці. Положення про матеріальне заохочення розробляється службою охорони праці і затверджується керівником підприємства (власником). Працівники служби охорони праці не можуть залучатися до виконання функцій, не передбачених Законом "Про охорону праці" і "Типовим положенням про службу охорони праці".

4.3 Висновок до четвертого розділу

В даному розділі описано основні принципи конструювання робочого місця користувача ЕОМ, зазначено діючі вимоги щодо ергономіки робочого місця. А також розкрито поняття управління охороною праці.

ВИСНОВКИ

В рамках кваліфікаційної роботи була розроблена система, яка автоматизує відстежування термінів придатності продукції, ведення обліку виробленої продукції, а також надає можливість розподілу функціональності відповідно до ролей співробітників.

Були виконані поставлені завдання:

1. Виконано аналіз предметної області, описано вимоги до системи відстежування терміну придатності продукції.
2. Спроектовано структуру клієнтської частини системи.
3. Виконано реалізацію клієнтської частини системи.
4. Проведено тестування клієнтської частини системи.

Наразі триває інтеграція розробленої системи на підприємствах. У майбутньому планується продовжити розробку та удосконалення системи, впроваджуючи додаткові функції та покращуючи існуючі. Це може включати в себе розширення можливостей моніторингу та аналітики, додавання нових інструментів для управління продукцією та термінами придатності, а також покращення інтерфейсу користувача для більш зручного та інтуїтивно зрозумілого використання. Крім того, планується оновлення системи з урахуванням змінюваних потреб бізнесу та вимог ринку, щоб забезпечити її актуальність та конкурентоспроможність у довгостроковій перспективі.

ПЕРЕЛІК ДЖЕРЕЛ

1. Web application - Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Web_application.
2. Web Architecture 101 [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/storyblocks-engineering/web-architecture-101-a3224e126947>.
3. Static Vs Dynamic Websites [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wix.com/blog/2021/11/static-vs-dynamic-website/>.
4. A Guide to Different Types of Website Structures [Електронний ресурс] – Режим доступу до ресурсу: <https://xd.adobe.com/ideas/process/informationarchitecture/different-types-of-website-structures>.
5. Лаврищева К.М. Програмна інженерія / Лаврищева К.М. –К: Академперіодика, 2008. – 319с.
6. Харченко О.Г. Інструментальний засіб порівняльного оцінювання і багатокритеріального вибору архітектури програмних систем / О.Г.Харченко, І.О.Боднарчук, І.Е.Райчев, І.О.Галай // Інженерія програмного забезпечення. – 2015. –№1(21). – С. 10–24.
7. Харченко, О., Яцишин, В., & Боднарчук, І. (2013). Експертна система проектування архітектури програмного забезпечення. Комп'ютерні Технології Друкарства, (29), 10–26.
8. Матійчук, Л., Готович, В., & Бонар, В. (2025). Порівняння ефективності методів некерованого машинного навчання для виявлення аномалій в obd2 даних. Measuring and computing devices in technological processes, (1), 407-414.
9. Харченко О.Г. Службовий твір Комп'ютерна програма “Архітектор програмних систем” / О.Г.Харченко, І.Е.Райчев, О.А.Щербак, Б.С.Павленко, І.О.Боднарчук // Свідectво про реєстрацію авторського права на твір №59631.

Видане державною службою інтелектуальної власності України 13.05.2015, м.Київ, НАУ.

10. Боднарчук, І., Харченко, О., Хоміцький, Б., & Шимчук, Г. (2019). Проектування архітектури програмних систем в проектах з гнучкими методами управління. Матеріали XXI Наукової Конференції Тернопільського Національного Технічного Університету Імені Івана Пулюя, 46–48.

11. Харченко, О. Г., Галай, І. О., Бондарчук, І. О., & Яцишин, В. В. (2010). Проектування архітектури WEB-застосування на основі моделі якості проектування. Інженерія Програмного Забезпечення, 4(4), 26.

12. Кириченко А. В. Динамічні сайти на HTML, CSS, JavaScript та Bootstrap. Практика, практика та лише практика. / А. В. Кириченко, Є. В. Дубовик. Наука та техніка, 2018. – 272 с.

13. Матійчук Л., Готович В., Бонар В. Порівняння ефективності методів некерованого машинного навчання для виявлення аномалій в OBD2 даних. Вимірювальна та обчислювальна техніка в технологічних процесах. Хмельницький національний університет. (1), 2025. С. 407–414.

14. Коноваленко І. В. Платформа .NET та мова програмування C# 8.0 : навч. посіб. / І. В. Коноваленко, П. О. Марущак. – Тернопіль : В. А. Паляниця, 2020 – 320 с.

15. Комп'ютерні мережі: [навчальний посібник] / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник.– Львів: «Магнолія 2006», 2013–256с.

16. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі. Книга 1 [навчальний посібник]. Львів : «Магнолія 2006», 2013. 256 с.

17. Моргунов Е. П. PostgreSQL. Основи мови SQL: [навч. посіб.] [Електронний ресурс]

18. Придатко О. В., Бурак Н. Є., Дзень В. Є., Кунинець М. С. Адаптивна інформаційно-довідкова система "UniBell" як складова частина

проєкту "Smartyніверситет". Науковий вісник НЛТУ України. 2020, т. 30, № 5. С. 105–113.

19. Литвин В.А. Принципи створення сучасних веб-сайтів. – Одеса: Астропринт, 2019 р. с. 234.

20. Мороз Ю.Б. Захист інформації в комп'ютерних системах та мережах. – Київ: Наукова думка, 2021 р. с. 289.

21. Google Developers. Progressive Web Apps. [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/web/progressive-web-apps>. Strutynska, I. (2019). Метрики цифрової трансформації бізнесу: світові та вітчизняні реалії. Галицький економічний вісник, 61(6), 30–45.

22. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник. Тернопіль: : ТНТУ 2020 – 62 с.

23. Strutynska, I. V. (2018). Informatsiini tekhnolohii orhanizatsii biznesu–imperatyv innovatsiinoho rozvytku biznes-struktur [Information technologies of business organization an imperative of innovative development of business structures]. Galician Economic Journal, 55(2), 40–49.

24. Конспект лекцій з курсу «Програмування для мобільних пристроїв» для здобувачів освітнього ступеня «бакалавр», спеціальності 126 «Інформаційні системи та технології», денної форми навчання. Укладачі: Готович В.А., Михайлович Т. В. ТНТУ, 2020 р.

25. Системи баз даних та знань. Книга 1. Організація баз даних та знань: [навч. посіб.] / А. Ю. Берко, О. М. Верес, В. В. Пасічник. – Львів: «Магнолія2006», 2019. – 584 с.

26. Якість та тестування ПЗ. URL: <https://dl.tntu.edu.ua/index.php>.

27. Firebase Test Lab [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/docs/test-lab>.

28. What is automation testing? URL: <https://www.globalapptesting.com/blog/what-is-automation-testing>.

29. Кодекс цивільного захисту України [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/5403-17#Text>

30. Про затвердження Положення про підсистему реагування на надзвичайні ситуації, проведення аварійно-рятувальних та інших невідкладних робіт єдиної державної системи цивільного захисту : наказ Міністерства внутрішніх справ України від 04.05.2016 №356.

31. Про затвердження Положення про штаб з ліквідації наслідків надзвичайної ситуації та видів оперативно-технічної і звітної документації штабу з ліквідації наслідків надзвичайної ситуації : наказ Міністерства внутрішніх справ України від 26.12.2014 №1406.

32. Про організацію роботи штабу з ліквідації наслідків надзвичайної ситуації та забезпечення його готовності : наказ Державної служби України з надзвичайних ситуацій від 16.03.2015 №149.

33. Рекомендації для роботодавців щодо організації виконання робіт підвищеної небезпеки під час воєнних (бойових) дій [Електронний ресурс] – Режим доступу: <https://dsp.gov.ua/faq/rekomendatsii-dlia-robotodavtsiv-shchodo-orhanizatsiivikonannia-robit-pidvyshchenoi-nebezpeky-pid-chas-voiennykh-boiovykh-dii/>

34. Кулалаєва Н.В. Ручні та пересувні засоби пожежогасіння: основні типи, будова та безпечне використання : навчальний посібник / В.О.Михайлюк, Б.Д.Халмурадов. Київ : Центр учбової літератури, 2011. 189 с.

35. Андрейчук Н.І. Охорона праці : навч. посіб. / Н.І. Андрейчук, Ю.В. Кіт, С.В. Шибанов, О.В. Шерстньова. Львів : Видавництво Львівська політехніка, 2012. 276 с.

36. Бедрій Я.І. Основи охорони праці : навч. посіб. 4-е вид. перероб. і доп. Тернопіль : Навчальна книга – Богдан, 2014. 240 с

ДОДАТКИ

Лістинг основних модулів інформаційної системи

1. App.js — Основний компонент для рендерингу:

```
import React, { useState } from 'react';
import ProductList from './ProductList';
import './styles.css';

const App = () => {
  // Стан для фільтрації продуктів та для збереження товарів
  const [filter, setFilter] = useState("all");
  const [products, setProducts] = useState([
    { id: "p1", name: "Молоко", expiryDate: "2023-12-15", status: "valid",
quantity: 100, price: 25.50 },
    { id: "p2", name: "Хліб", expiryDate: "2023-11-10", status: "expired",
quantity: 50, price: 12.30 },
    { id: "p3", name: "Йогурт", expiryDate: "2023-12-20", status: "valid",
quantity: 75, price: 20.00 },
    { id: "p4", name: "Масло", expiryDate: "2023-10-01", status: "expired",
quantity: 30, price: 50.00 }
  ]);

  const handleFilterChange = (event) => {
    setFilter(event.target.value);
  };

  const addProduct = (newProduct) => {
    setProducts([...products, newProduct]);
  };

  const updateProduct = (updatedProduct) => {
    const updatedProducts = products.map(product =>
      product.id === updatedProduct.id ? updatedProduct : product
    );
    setProducts(updatedProducts);
  };
};
```

```

return (
  <div className="container">
    <header>
      <h1>Система Аналізу Терміну Придатності Товарів</h1>
    </header>
    <div className="filter">
      <label htmlFor="filter">Фільтрувати за статусом:</label>
      <select id="filter" onChange={handleFilterChange}>
        <option value="all">Всі</option>
        <option value="valid">Дійсні</option>
        <option value="expired">Просрочені</option>
      </select>
    </div>
    <ProductList
      products={products}
      filter={filter}
      updateProduct={updateProduct}
      addProduct={addProduct}
    />
  </div>
);
};

export default App;

```

2. ProductList.js — Компонент для відображення списку продуктів:

```

import React from 'react';
import ProductItem from './ProductItem';

const ProductList = ({ products, filter }) => {
  const filteredProducts = products.filter(product => {
    if (filter === "all") return true;
    return product.status === filter;
  });

  return (
    <div className="product-list">

```

```

        {filteredProducts.map(product => (
          <ProductItem key={product.id} product={product} />
        ))}
      </div>
    );
  };

export default ProductList;

import React, { useState } from 'react';
import ProductItem from './ProductItem';

const ProductList = ({ products, filter, addProduct, updateProduct }) => {
  const [newProduct, setNewProduct] = useState({
    name: "",
    expiryDate: "",
    status: "valid",
    quantity: 0,
    price: 0
  });

  const handleAddProductChange = (e) => {
    setNewProduct({
      ...newProduct,
      [e.target.name]: e.target.value
    });
  };

  const handleAddProductSubmit = (e) => {
    e.preventDefault();
    const productId = `p${Date.now()}`;
    addProduct({ id: productId, ...newProduct });
    setNewProduct({ name: "", expiryDate: "", status: "valid", quantity: 0, price:
0 });
  };

  const filteredProducts = products.filter(product => {
    if (filter === "all") return true;
    return product.status === filter;
  });

```

```

});

return (
  <div className="product-list">
    <form onSubmit={handleAddProductSubmit} className="add-product-
form">
      <h2>Додати новий товар</h2>
      <input
        type="text"
        name="name"
        value={newProduct.name}
        onChange={handleAddProductChange}
        placeholder="Назва товару"
        required
      />
      <input
        type="date"
        name="expiryDate"
        value={newProduct.expiryDate}
        onChange={handleAddProductChange}
        required
      />
      <input
        type="number"
        name="quantity"
        value={newProduct.quantity}
        onChange={handleAddProductChange}
        placeholder="Кількість"
        required
      />
      <input
        type="number"
        name="price"
        value={newProduct.price}
        onChange={handleAddProductChange}
        placeholder="Ціна"
        required
      />
      <button type="submit">Додати товар</button>
    </form>
  </div>
);

```

```
</form>
```

```

    {filteredProducts.map(product => (
      <ProductItem
        key={product.id}
        product={product}
        updateProduct={updateProduct}
      />
    ))}
  </div>
);
};

```

```
export default ProductList;
```

3. ProductItem.js — Компонент для відображення окремого товару:

```
import React from 'react';
```

```

const ProductItem = ({ product }) => {
  return (
    <div className="product-item">
      <h3>{product.name}</h3>
      <p><strong>Термін придатності:</strong> {product.expiryDate}</p>
      <p><strong>Статус:</strong> {product.status === 'expired' ? <span
className="expired">Просрочено</span>
                                :
                                <span
className="valid">Дійсне</span>}</p>
      <p><strong>Кількість:</strong> {product.quantity}</p>
      <p><strong>Ціна:</strong> {product.price} UAH</p>
    </div>
  );
};

```

```
export default ProductItem;
```

4. styles.css — Стили для компонентів:

```

body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f9;
  margin: 0;

```

```
padding: 0;  
}
```

```
.container {  
  width: 80%;  
  margin: 0 auto;  
  padding: 20px;  
}
```

```
header {  
  background-color: #333;  
  color: white;  
  padding: 15px;  
  text-align: center;  
}
```

```
h1 {  
  margin: 0;  
  font-size: 2.5em;  
}
```

```
.filter {  
  margin-top: 20px;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
.filter label {  
  font-size: 1.1em;  
}
```

```
.product-list {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 20px;  
  margin-top: 20px;  
}
```

```
.product-item {  
  background-color: white;  
  border-radius: 10px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  width: 300px;  
  padding: 15px;  
  transition: transform 0.3s ease-in-out;  
}  
  
.product-item:hover {  
  transform: scale(1.05);  
}  
  
.product-item h3 {  
  margin: 0;  
  font-size: 1.5em;  
}  
  
.product-item p {  
  font-size: 1.1em;  
}  
  
.valid {  
  color: green;  
}  
  
.expired {  
  color: red;  
}
```

Json структура основних файлів

1. products.json — Файл, що містить інформацію про продукти:

```
{
  "products": [
    {
      "id": "p1",
      "name": "Молоко",
      "barcode": "123456789012",
      "expiryDate": "2023-12-15",
      "addDate": "2023-11-01",
      "status": "valid",
      "quantity": 100,
      "price": 25.50,
      "category": "Молочні продукти",
      "image": "images/milk.jpg"
    },
    {
      "id": "p2",
      "name": "Хліб",
      "barcode": "987654321098",
      "expiryDate": "2023-11-10",
      "addDate": "2023-10-05",
      "status": "expired",
      "quantity": 50,
      "price": 12.30,
      "category": "Хлібобулочні вироби",
      "image": "images/bread.jpg"
    }
  ]
}
```

2. users.json — Файл, що містить дані про користувачів системи:

```
{
  "users": [
```

```

{
  "userId": "u1",
  "username": "admin",
  "password": "admin123",
  "role": "administrator",
  "email": "admin@example.com",
  "createdAt": "2023-01-01"
},
{
  "userId": "u2",
  "username": "user1",
  "password": "password1",
  "role": "user",
  "email": "user1@example.com",
  "createdAt": "2023-03-12"
}
]
}

```

3. transactions.json — Файл, що зберігає історію транзакцій:

```

{
  "transactions": [
    {
      "transactionId": "t1",
      "productId": "p1",
      "action": "add",
      "quantity": 50,
      "date": "2023-11-01",
      "userId": "u1",
      "status": "successful"
    },
    {
      "transactionId": "t2",
      "productId": "p2",
      "action": "remove",
      "quantity": 20,
      "date": "2023-10-10",

```

```
    "userId": "u2",  
    "status": "successful"  
  }  
]  
}
```

4. settings.json — Файл налаштувань системи:

```
{  
  "settings": {  
    "defaultExpirationPeriod": 30,  
    "notificationEmail": "notifications@example.com",  
    "maxProductAlert": 50,  
    "theme": "dark",  
    "currency": "UAH"  
  }  
}
```