

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

СИСТЕМА КОНТРОЛЮ ЯКОСТІ ПРОДУКТІВ НА ОСНОВІ  
ДАТЧИКІВ IOT

PRODUCT QUALITY CONTROL BASED ON IOT SENSORS

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи КІ-42

**Романюк Ярослав Анатолійович**

(підпис)

Керівник:

к.т.н., доцент

**Мельник Катерина Вікторівна**

(підпис)

Кваліфікаційну роботу

допущено до захисту

«\_\_\_» червня 2024 р.

Гарант освітньої програми:

к.т.н., доцент

**Лавренчук Світлана Василівна**

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н. Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Романюку Ярославу Анатолійовичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Система контролю якості продуктів на основі датчиків IoT*

Керівник роботи *к.т.н., доцент Мельник Катерина Вікторівна*

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *11.06.2024р.*

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

*Вступ*

*1. Аналіз сучасних методів та апаратів для відстеження якості продуктів, способ*

*Ефективного та раціонального використання продуктів і відслідковування їх придатності*

*2. Теоретична частина*

*3. Практична частина*

*4. Висновки*

5. Перелік графічного (ілюстративного) матеріалу:

*UI дизайн застосунку*

*Основні елементи коду*

*Спрощена схема робочих елементів застосунку*

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Мельник К.В., доцент</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Мельник К.В., доцент</i>		
<i>Практична реалізація об'єкта проектування</i>	<i>Мельник К.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		___%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Розділ 1 Ознайомлення із Сферою IoT</i>	До 15.02.2024 р.	
2.	<i>Розділ 2 Аналіз ефективності Інструментів</i>	До 15.03.2024 р.	
3.	<i>Розділ 3 Побудова додатку у взаємодії із старт холодильником</i>	До 04.05.2024 р.	
4.	<i>Висновки та пропозиції</i>	До 07.05.2024 р.	
5.	<i>Формування списку використаних джерел</i>	До 10.05.2024 р.	
6.	<i>Формування додатків</i>	До 15.05.2024 р.	
7.	<i>Оформлення ілюстративного матеріалу</i>	До 20.05.2024 р.	
8.	<i>Нормоконтроль</i>	До 01.06.2024 р.	
9.	<i>Інструментальна перевірка на академічний плагіат</i>	До 04.06.2024 р.	
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	До 11.06.2024 р.	

Здобувач вищої освіти

(підпис)

Романюк Я.А.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Мельник К. В.

(прізвище, ініціали)

## АНОТАЦІЯ

Романюк Я. А. Контроль якості продуктів на основі датчиків IoT за допомогою взаємодії із застосунком.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 41 с.

Кваліфікаційна робота складається з вступу, трьох розділів, графічних зображень, висновків, списку використаних джерел.

Перший розділ містить огляд предметної області, обговорюючи фундаментальні концепції датчиків IoT та їх застосування в контролі якості продукції. Він охоплює типи використовуваних датчиків IoT, їх функціональні можливості та різні галузі, де вони реалізовані. Розділ також містить практичні приклади, що ілюструють використання датчиків IoT для моніторингу та забезпечення якості продукції. Крім того, він переглядає існуючі додатки та системи (наприклад, розумні холодильники, системи моніторингу якості, сенсорні мережі).

В другому розділі приведенне обґрунтування обраних засобів розробки. Обрано засоби: PHP та AWS DynamoDB. Розглянуто питання ООП в PHP.

Третій розділ присвячено опису розробки застосунку на основі IoT технологій який буде взаємодіяти із цими пристроями. Описанні можливості користувача та наведенні візуальні приклади.

Ключові слова: IoT сенсори, розумний холодильник, PHP, застосунок, UI.

## ANNOTATION

Romanyuk Y. A. Quality control of products based on IoT sensors using interaction with the application.

Bachelor's qualifying thesis of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 41p.

The qualification work consists of an introduction, three sections, graphic images, conclusions, a list of used sources.

The first section provides an overview of the subject area, discussing the fundamental concepts of IoT sensors and their applications in product quality control. It covers the types of IoT sensors used, their functionality and the various industries where they are implemented. The chapter also includes practical examples illustrating the use of IoT sensors for product quality monitoring and assurance. In addition, it reviews existing applications and systems (such as smart refrigerators, quality monitoring systems, sensor networks) [4, 5].

The second section provides an overview of selected development tools. Tools selected: PHP and AWS DynamoDB. The question of OOP in PHP is considered.

The third section is devoted to the description of the development of an application based on IoT technologies that will interact with these devices. Describes user capabilities and provides visual examples.

Keywords: IoT, sensors, smart refrigerator, PHP, application, UI.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 ІОТ ЯК ОБ'ЄКТ ДОСЛІДЖЕННЯ .....	9
1.1 Еволюція та вплив Інтернету речей (ІоТ).....	9
1.2 Види датчиків та їх використання.....	10
РОЗДІЛ 2 АНАЛІЗ ЕФЕКТИВНОСТІ ІНСТРУМЕНТІВ .....	13
2.1 Ознайомлення із датчиками .....	13
2.2 PHP мова вибору .....	15
2.3 Аналіз проблем та їх вирішення.....	18
2.4 Ілюстрація User Interface із описом.....	21
РОЗДІЛ 3 ПОБУДОВА ДОДАТКУ У ВЗАЄМОДІЇ ІЗ СМАРТ ХОЛОДИЛЬНИКОМ.....	24
3.1 Процес передачі даних ключові компоненти.....	24
3.2 Покращення користувача у застосунку побудованому на ІОТ: використання даних і функції програми Smart Refrigerator .....	24
3.3 Використання JSON.....	26
3.4 Опис реалізації застосунку без UI.....	28
3.5 Опис роботи застосунку .....	34
ВИСНОВКИ.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37

## ВСТУП

Тенденція сучасної цифровізації розвивається майже щодня, суттєво впливаючи на споживчі звички та попит на інноваційні продукти. З постійним розвитком технологій споживча культура змінюється, що вимагає вдосконалення контролю якості продукції. Зокрема, інтеграція технології IoT (Інтернет речей) у побутові прилади, такі як холодильники, пропонує перспективне рішення для моніторингу та підтримки якості продукції. Можливість відстежувати та забезпечувати свіжість швидкопсувних товарів за допомогою даних у реальному часі від датчиків IoT, вбудованих у розумні холодильники, відповідає критичній потребі сучасного ринку [6].

Забезпечення якості та безпеки швидкопсувних продуктів має вирішальне значення в різних галузях промисловості, зокрема у зберіганні харчових продуктів і роздрібній торгівлі. Традиційні методи моніторингу якості продукції часто недостатні для вирішення динамічного характеру умов навколишнього середовища в складських одиницях. Технологія IoT пропонує трансформаційний підхід, забезпечуючи безперервний моніторинг і контроль у реальному часі, тим самим значно підвищуючи надійність і ефективність управління якістю продукції [1].

Оскільки ринок розумної побутової техніки розширюється, зростає потреба в передових програмах, які можуть повністю використовувати можливості пристроїв IoT.

Розумні холодильники, оснащені датчиками, можуть надавати важливі дані про умови всередині пристрою, такі як температура та вологість, які є важливими для підтримки якості швидкопсувних товарів. Цей моніторинг у реальному часі може допомогти запобігти псуванню, забезпечити безпеку харчових продуктів і оптимізувати використання ресурсів відповідно до глобальних цілей сталого розвитку.

Метою роботи є розробка програми, яка спрямована на ефективне використання продуктів за раунок розумного та своєчасного використання.

Об'єктом розробки є додаток, який би спрощує користувачу менеджмент хачорових продуктів. Передача даних до застосунку із холодильника та зовнішніх баз даних за допомогою API у форматі JSON. Робота із даними та їх зберігання методами PHP [1, 3].

Предметом дослідження є методи взаємодії датчиків IoT, задіяних у розумних холодильниках, методи передачі даних, взаємодія із зовнішніми базами даних методами JSON API та мережі інтернет. Організація даних у доступному для користувача форматі за допомогою методів PHP у застосунку.

Завдання, які необхідно виконати:

- здійснити огляд доступних інструментів для реалізації застосунку в сфері IoT для керування розумним холодильником.
- розробити адаптивну інтерфейсну програму, використовуючи HTML, CSS, PHP і JS для відображення даних про стан продуктів та забезпечити зв'язок додатку із зовнішніми базами даних.
- візуалізувати інструменти подачі даних, щоб надавати користувачам чітке уявлення про стан збережених продуктів.

## РОЗДІЛ 1

### ІОТ ЯК ОБ'ЄКТ ДОСЛІДЖЕННЯ

#### 1.1 Еволюція та вплив Інтернету речей (ІоТ)

Інтернет речей (ІоТ) представляє зміну парадигми в тому, як сприймаємо цифрову сферу та взаємодіємо з нею. Це революційна концепція, яка змінила не лише технологічний ландшафт, але й те, як ми живемо та працюємо. За своєю суттю ІоТ є великою мережею взаємопов'язаних пристроїв, кожен із вбудованими датчиками, програмним забезпеченням та іншими технологіями, здатними збирати та обмінюватися даними.

Концепція ІоТ ґрунтується на ідеї розширення доступу Інтернету за межі традиційних комп'ютерних пристроїв, таких як комп'ютери та смартфони. По суті, він передбачає світ, де предмети повсякденного використання, від побутових приладів до промислового обладнання, мають можливість безперервно спілкуватися та обмінюватися інформацією.

Рушійною силою розвитку Інтернету речей є постійно розширюване підключення за допомогою Інтернету. Оскільки доступ до Інтернету став все більш поширеним, він створив основу для нової ери, коли пристрої могли спілкуватися один з одним у режимі реального часу, створюючи мережу взаємопов'язаних екосистем. Фундаментальний принцип простий, але глибокий: наділивши неживі об'єкти здатністю збирати та обмінюватися даними, ми можемо підвищити ефективність, продуктивність і навіть якість життя.

ІоТ проявляється в різних областях, починаючи від розумних будинків і міст і закінчуючи промисловими додатками та охороною здоров'я.

У розумному домі, наприклад, пристрої з підтримкою Інтернету речей, такі як термостати, холодильники та камери безпеки, працюють у тандемі, створюючи інтелектуальне взаємопов'язане середовище. Ці пристрої можна віддалено контролювати, забезпечуючи власникам будинків безпрецедентний рівень зручності та енергоефективності.

У промисловості IoT відкрив епоху Індустрії 4.0, де виробничі процеси переповнені інтелектом і підключенням. Датчики, вбудовані в обладнання, збирають дані про продуктивність, потреби в технічному обслуговуванні та ефективність роботи, що дозволяє прогнозувати технічне обслуговування та оптимізувати графіки виробництва. Це не тільки скорочує час простою, але й максимізує використання ресурсів, що зрештою призводить до економії коштів і підвищення конкурентоспроможності.

Сектор охорони здоров'я став свідком трансформаційного впливу інтеграції IoT. Носимі пристрої, такі як фітнес – трекери та розумні годинники, відстежують життєво важливі показники та показники здоров'я, дозволяючи людям вживати профілактичних заходів для свого благополуччя. У лікарнях пристрої IoT оптимізують догляд за пацієнтами за допомогою моніторингу в реальному часі обладнання, стану пацієнтів і прийому ліків.

Однак із величезним потенціалом Інтернету речей виникають значні проблеми та міркування. Питання про безпеку та конфіденційність є першочерговими, оскільки величезна кількість даних, що генеруються пристроями Інтернету речей, стає потенційною мішенню для кіберзагроз. Також необхідно вирішити питання стандартизації та взаємодії, щоб забезпечити безперебійний зв'язок між пристроями різних виробників.

Підсумовуючи, Інтернет речей – це не просто технологічна тенденція; це трансформаційна сила, яка змінює наш світ. Поєднуючи фізичну та цифрову сфери, IoT має потенціал для безпрецедентного прогресу в ефективності, стійкості та якості життя. Що позитивно впливає на людину та сприяє її контрибуції до суспільства.

## **1.2 Види датчиків та їх використання**

Занурення в типи, що розширюють можливості технології IoT. Інтернет речей (IoT) виник як технологічний джаггернаут, який бездоганно поєднує фізичний і цифровий світи. У центрі цього трансформаційного ландшафту

лежать датчики – неоспівані герої, які розширюють можливості екосистем Інтернету речей. Ці датчики, різноманітні за своїми функціями, діють як сенсорні органи IoT, сприяючи збору даних, які сприяють прийняттю інтелектуальних рішень. У цьому есе досліджується безліч типів датчиків, які лежать в основі технології IoT.

Розкриття спектру задіяних приладів.

Датчики температури всюди присутні в програмах IoT. Від систем клімат контролю в розумних будинках до моніторингу промислових процесів, ці датчики фіксують температуру навколишнього середовища. Точність має вирішальне значення, будь то підтримка ідеальної температури в серверній кімнаті або забезпечення цілісності швидкопсувних товарів під час транспортування.

Датчики руху є пильними охоронцями безпеки в IoT. У розумних будинках ці датчики виявляють рухи, запускаючи сигналізацію або активуючи камери спостереження. У промислових умовах вони підвищують безпеку, контролюючи рух машин, забезпечуючи дотримання експлуатаційних стандартів.

Датчики наближення визначають близькість об'єкта без фізичного контакту. Вони зазвичай зустрічаються в автоматичних змішувачах або ліфтах і є невід'ємною частиною оптимізації споживання енергії. В IoT ці датчики сприяють безконтактній взаємодії, наприклад автоматичному відкриванню дверей або адаптивному освітленню на основі присутності людини [36].

Датчики вологості відіграють ключову роль у моніторингу навколишнього середовища. Від сільського господарства, де вони сприяють точному зрошенню, до галузей, які вимагають контрольованого рівня вологості, ці датчики сприяють підтримці оптимальних умов. У сфері охорони здоров'я вони допомагають зберігати чутливе медичне обладнання.

Датчики світла, або фотодетектори, вимірюють інтенсивність світла в їх оточенні. Програми IoT використовують ці датчики для систем адаптивного освітлення, енергозбереження, а в розумних містах для оптимізації вуличного освітлення на основі умов навколишнього освітлення [33].

У промисловості та в розумних будинках газові датчики виявляють наявність шкідливих газів, забезпечуючи безпеку. Наприклад, у розумних містах ці датчики контролюють якість повітря, надаючи дані в реальному часі для заходів із контролю забруднення.

Датчики тиску вимірюють прикладену до них силу та знаходять застосування в різноманітних сферах. Від моніторингу промислового обладнання до медичних пристроїв, як – от вимірювачів артеріального тиску, ці датчики мають вирішальне значення. В IoT вони сприяють прогнозованому обслуговуванню, виявляючи аномалії в рівнях тиску.

Звукові датчики вловлюють аудіосигнали, забезпечуючи такі програми, як розпізнавання голосу, системи безпеки та моніторинг шумового забруднення. В IoT вони знаходять застосування в розумних будинках для голосових помічників і в міському плануванні для аналізу шумових моделей [31].

Датчики зображення, включаючи камери, фіксують візуальну інформацію. В IoT системи відеоспостереження використовують датчики зображення, що забезпечує безпеку. Крім того, ці датчики дозволяють розширені програми, такі як розпізнавання обличчя та виявлення об'єктів [22, 32, 35].

Біометричні датчики фіксують унікальні фізичні чи поведінкові риси для ідентифікації. В IoT вони покращують заходи безпеки, від розпізнавання відбитків пальців у смартфонах до сканування райдужної оболонки ока для контролю доступу [24].

Датчик камери з штучним інтелектом точно розпізнає продукти, пропонує віддалений доступ і розвивається через взаємодію користувача.

Розуміння різноманітності датчиків у технології IoT має важливе значення для оцінки багатогранних програм, які надають ці пристрої. У міру того, як технології продовжують розвиватися, ландшафт датчиків буде розвиватися, стимулюючи інновації та розширюючи горизонти того, що можна досягти у сфері взаємопов'язаних пристроїв. У симбіотичних стосунках між датчиками та IoT можливості такі ж великі та динамічні, як і дані, які вони спільно збирають та інтерпретують [25, 26, 28].

## РОЗДІЛ 2

### АНАЛІЗ ЕФЕКТИВНОСТІ ІНСТРУМЕНТІВ

#### 2.1 Ознайомлення із датчиками

Прагнучи створити інноваційну та зручну для користувача розумну систему для холодильників, було ретельно розглянуто питання вибору та інтеграції датчиків, необхідних для оптимізації управління продуктами харчування та відстеження запасів. В основі цієї конструкції є кілька ключових датчиків, кожен з яких виконує певну мету та сприяє загальній функціональності та ефективності пристрою.

Датчики температури відіграють вирішальну роль у підтримці оптимальних умов зберігання швидкопсувних продуктів у холодильнику. Постійно контролюючи рівень температури, ці датчики гарантують, що їжа залишається свіжою та безпечною для споживання. Коливання температури можуть вказувати на потенційні проблеми, такі як несправність систем охолодження або протікання дверних ущільнювачів, що дозволяє користувачам швидко вживати заходів щодо їх усунення. Крім того, дані про температуру, зібрані цими датчиками, можна використовувати для розрахунку залишкового терміну придатності продуктів, надаючи користувачам цінну інформацію про якість їжі і термін придатності [7, 8].

Детектори руху використовуються для виявлення та відстеження взаємодії користувача з холодильником, наприклад відкриття та закриття дверей. Фіксуючи рухи всередині пристрою, ці датчики дозволяють системі фіксувати частоту доступу та визначати шаблони поведінки при використанні. Цю інформацію можна використовувати для оптимізації споживання енергії, посилення заходів безпеки та навіть персоналізації взаємодії з користувачем на основі звичок використання [14].

Датчики наближення використовуються для визначення наявності предметів у безпосередній близькості від внутрішньої частини холодильника. Завдяки точному вимірюванню відстані ці датчики дозволяють приладу

визначати, коли предмети поміщаються або виймаються з певних відділень. Ці дані є безцінними для управління запасами, оскільки дозволяють системі відстежувати рух товару в режимі реального часу та відповідно оновлювати списки запасів. Крім того, датчики наближення можуть запускати автоматичні дії, такі як регулювання освітлення або активація контролю температури на основі взаємодії користувача [9, 11, 12].

Датчики вологості необхідні для моніторингу рівня вологості в холодильнику, особливо в контексті збереження фруктів, овочів та інших продуктів, що швидко псуються. Вимірюючи рівень вологості, ці датчики допомагають запобігти псуванню їжі та підтримувати оптимальні умови зберігання. Крім того, дані про вологість можна використовувати для динамічного налаштування параметрів температури та вологості, гарантуючи, що делікатні речі зберігаються в середовищі, яке сприяє довготривалій свіжості [10].

Однією з найбільш інноваційних функцій розумного холодильника є вбудовані датчики зображення, які зазвичай називають камерами. Ці датчики фіксують візуальні дані вмісту холодильника, дозволяючи користувачам дистанційно контролювати запаси їжі та визначати продукти, які потребують поповнення. Крім того, датчики зображення забезпечують розширені функції, такі як розпізнавання об'єктів і відстеження запасів за допомогою ШІ та алгоритмів глибокого навчання [34]. Використовуючи візуальні дані, користувачі можуть отримати цінну інформацію про свої звички споживання їжі, отримувати персоналізовані рекомендації та з легкістю активно керувати інвентарем [2, 30].

Прагнучи створити інноваційну та зручну для користувача розумну холодильну систему, було ретельно розглянуто питання вибору та інтеграції датчиків, необхідних для оптимізації управління продуктами харчування та відстеження запасів. В основі цієї конструкції є кілька ключових датчиків, кожен з яких виконує певну мету та сприяє загальній функціональності та ефективності пристрою.

Використовуючи візуальні дані, користувачі можуть отримати цінну інформацію про свої звички споживання їжі, отримувати персоналізовані рекомендації та з легкістю активно керувати своїм інвентарем.

## **2.2 PHP мова вибору**

Під час розробки системи розумного холодильника було обрано PHP як основну мову програмування через її універсальність, простоту та розгалужену екосистему бібліотек і фреймворків. Інтуїтивно зрозумілий синтаксис і легкість читання PHP роблять його ідеальним вибором для швидкого створення прототипів і розробки, забезпечуючи ефективну ітерацію та вдосконалення функцій програмного забезпечення. Крім того, надійна підтримка PHP для обробки даних, розпізнавання зображень і мережевого зв'язку робить його ідеальним для впровадження складних функцій, таких як відстеження запасів, рекомендації рецептів і дизайн інтерфейсу користувача. Крім того, крос-платформна сумісність PHP забезпечує повну інтеграцію з існуючими платформами та службами IoT, забезпечуючи взаємодію та масштабованість у різноманітних апаратних середовищах [15, 27].

Переваги інтеграції мобільних додатків.

Рішення розробити спеціальну програму для мобільних пристроїв, яка доповнює систему розумного холодильника, впливає з бажання надати користувачам бездоганний та інтуїтивно зрозумілий інтерфейс для керування запасами їжі та звичками споживання. За допомогою мобільного додатку користувачі можуть дистанційно контролювати вміст свого холодильника, отримувати сповіщення та попередження в реальному часі, а також отримувати персоналізовані рекомендації та пропозиції рецептів на основі доступних інгредієнтів. Крім того, програма забезпечує безперебійну взаємодію з датчиками та функціями холодильника, дозволяючи користувачам вводити дані, оновлювати списки запасів і легко налаштовувати налаштування.

Завдяки інтеграції інтелектуального холодильника з мобільним додатком користувачі можуть насолоджуватися більшою зручністю, гнучкістю та контролем над своїми процедурами управління їжею, що зрештою призведе до більш ефективного та приємного процесу приготування їжі.

Переваги інтегрованого дизайну.

Інтегрована конструкція системи розумного холодильника пропонує кілька явних переваг перед традиційними приладами та автономними рішеннями. Поєднуючи датчики температури, детектори руху, датчики наближення, датчики вологості, датчики зображення та зручний мобільний додаток, прилад надає користувачам вичерпну інформацію про їхні харчові запаси, звички споживання та умови зберігання [16]. Цей цілісний підхід до управління продуктами харчування дозволяє користувачам приймати більш обґрунтовані рішення щодо вибору дієти, зменшувати харчові відходи та оптимізувати свої списки покупок і планування їжі. Крім того, інтеграція штучного інтелекту та алгоритмів глибокого навчання покращує інтелект і адаптивність системи, дозволяючи їй постійно навчатися та вдосконалюватися з часом на основі відгуків користувачів і взаємодії. Загалом інтегрований дизайн розумного холодильника є значним прогресом у технології побутової техніки, пропонуючи користувачам безпрецедентну зручність, ефективність і контроль над своїми кухонними процедурами.

Покращене управління їжею за допомогою сенсорів на основі штучного інтелекту. Впровадження датчиків на основі штучного інтелекту, зокрема датчиків зображення або камер, революціонізує управління продуктами харчування в розумному холодильнику. Ці датчики, оснащені вдосконаленим штучним інтелектом і алгоритмами глибокого навчання, дозволяють системі точно розпізнавати та класифікувати харчові продукти, що зберігаються в приладі. Аналізуючи візуальні дані, отримані камерами, ідентифікуючи різні типи продуктів, молочних продуктів, напоїв та інших товарів, надаючи користувачам повний огляд запасів. Цей рівень автоматизації спрощує процес оновлення інвентарних списків і відстеження свіжості продукту, усуваючи

потребу в ручному введенні та знижуючи ризик людської помилки.

Крім того, датчики на основі штучного інтелекту сприяють безперервному навчанню та вдосконаленню, дозволяючи системі з часом адаптуватися до вподобань і поведінки користувачів. Коли користувачі взаємодіють із розумним холодильником і вносять корективи в списки запасів, додаток навчається на основі цих взаємодій, удосконалюючи свої алгоритми та рекомендації, щоб краще відповідати потребам і вподобанням користувачів. Наприклад, якщо користувач часто вживає певні види продуктів або дотримується певних дієтичних обмежень, додаток може визначити пріоритетність відповідних рекомендацій і відповідним чином адаптувати свої пропозиції.

Крім того, інтеграція датчиків на основі штучного інтелекту покращує здатність системи ефективно виявляти харчові відходи та зменшувати їх кількість. Відстежуючи свіжість продукту та терміни придатності, додаток може сповіщати користувачів про продукти, термін придатності яких закінчується, і пропонувати рецепти чи ідеї страв, щоб використати ці інгредієнти, перш ніж вони зіпсуються. Крім того, аналізуючи моделі споживання та відгуки користувачів, додаток може оптимізувати стратегії управління запасами, допомагаючи користувачам мінімізувати надмірні запаси та зменшити непотрібні покупки продуктів харчування.

Загалом, використання сенсорів на основі штучного інтелекту є значним прогресом у технології управління харчовими продуктами, пропонуючи користувачам безпрецедентну зручність, ефективність і контроль над своїми кухонними процедурами. Використовуючи потужність штучного інтелекту та глибокого навчання розумного холодильника змінює спосіб взаємодії користувачів зі своїми харчовими запасами, надаючи їм змогу приймати розумніші рішення, зменшувати харчові відходи та насолоджуватися більш екологічним і приємним процесом приготування їжі.

Підсумовуючи, дизайн розумної холодильної системи являє собою комплексне та інноваційне рішення сучасних проблем управління харчовими продуктами.

Використовуючи комбінацію передових датчиків, технологій на базі штучного інтелекту та зручного мобільного додатка, додаток надає користувачам безпрецедентний контроль, зручність та ефективність в управлінні запасами їжі та звичками споживання. Інтеграція датчиків температури, детекторів руху, датчиків наближення, датчиків вологості та датчиків зображення дає змогу в режимі реального часу відстежувати умови зберігання харчових продуктів, точно відстеження запасів і персоналізовані рекомендації на основі вподобань користувача та дієтичних вимог.

Крім того, вибір PHP як основної мови програмування сприяє швидкій розробці та бездоганній інтеграції складних функціональних можливостей, а мобільний додаток покращує доступність користувача та взаємодію з системою розумного холодильника. Разом ці компоненти утворюють згуртовану та інтелектуальну екосистему, яка оптимізує процеси управління харчовими продуктами, зменшує харчові відходи та дає користувачам можливість робити більш здорові та екологічні рішення у своєму повсякденному житті.

Зрештою, дизайн розумної холодильної системи є прикладом потенціалу технологій Інтернету речей для революції в традиційній побутовій техніці, пропонуючи користувачам більшу зручність, ефективність і спокій в управлінні постачанням їжі. Завдяки постійним інноваціям і вдосконаленню розумні холодильники мають потенціал змінити наш підхід до зберігання їжі, планування їжі та споживання, прокладаючи шлях до більш стійкого та пов'язаного майбутнього на кухні.

### **2.3 Аналіз проблем та їх вирішення**

Початковий етап планування передбачав комплексне дослідження існуючих технологій, тенденцій ринку та потреб споживачів. Розуміння можливостей датчиків, баз даних і варіантів підключення було найважливішим для створення надійного рішення. Концепція інтеграції датчиків у холодильник

для відстеження харчових продуктів стала багатообіцяючим підходом, пропонуючи моніторинг у режимі реального часу та розуміння на основі даних.

Одним із перших завдань, з якими зіткнулися під час розробки, був вибір відповідних датчиків для холодильника. Кожен датчик мав відповідати певним вимогам, таким як моніторинг температури, визначення наближення та захоплення зображення. Поєднання точності, економічності та сумісності становило значну перешкоду, що вимагало ретельної оцінки та тестування.

Інтеграція датчиків із холодильником передбачала ретельний розгляд фізичного розміщення та проводки. Забезпечення безперебійного зв'язку між датчиками, центральним процесором і програмою вимагало ретельного підключення та протоколів підключення. Кабелі Ethernet і бездротові адаптери використовувалися для встановлення надійного з'єднання між холодильником і процесором.

Наступним критичним аспектом було ввімкнення підключення Wi-Fi для холодильника, що дозволило йому спілкуватися з програмою та зовнішніми серверами. Налаштування параметрів Wi-Fi, впровадження протоколів шифрування та оптимізація продуктивності мережі були важливими кроками для забезпечення безпечної та надійної передачі даних.

Додаток є інтерфейсом, через який користувачі взаємодіяли з розумним холодильником, надаючи їм інформацію в режимі реального часу про їхні запаси їжі, терміни придатності та пропозиції щодо рецептів.

Розробка програми вимагала ретельного розгляду принципів взаємодії з користувачем (UX) та інтерфейсу (UI), щоб забезпечити інтуїтивно зрозумілу навігацію та безперебійну взаємодію. Каркас і прототипування відіграли вирішальну роль у вдосконаленні макета, функцій і функціональності програми перед тим, як перейти до повномасштабної розробки.

Однією з ключових особливостей програми була її здатність синхронізуватися з датчиками холодильника та базою даних, надаючи користувачам актуальну інформацію про їхні продукти харчування. Це вимагало надійної розробки серверної частини для обробки даних, зберігання та

синхронізації між програмою та холодильником.

Реалізація логіки для програми включала складні алгоритми для аналізу свіжості їжі, створення рекомендацій щодо рецептів і оновлення статусу запасів на основі введення користувача. Алгоритми машинного навчання були використані для підвищення точності розпізнавання їжі та прогнозування терміну придатності з часом, дозволяючи системі адаптуватися до вподобань і звичок користувачів.

Інтеграція між додатком і розумним холодильником була досягнута за допомогою поєднання API, веб – сервісів і спеціальних протоколів зв'язку. Це сприяло безперебійному обміну даними між двома середовищами, дозволяючи користувачам дистанційно контролювати свій запас їжі та отримувати своєчасні сповіщення та повідомлення [29].

Незважаючи на ретельне планування та виконання, процес розробки не обійшовся без проблем. Технічні збої, проблеми сумісності та невдачі в налагодженні вимагали повторного тестування та вдосконалення, щоб забезпечити стабільний і надійний продукт.

Інтеграція пристрою з датчиками холодильника вимагала ретельного розгляду апаратної сумісності та підключення. Різноманітні датчики, включаючи датчики температури, датчики наближення та датчики вологості, були стратегічно розміщені всередині холодильника для моніторингу різних аспектів умов зберігання продуктів. Ці датчики були підключені до пристрою за допомогою комбінації дротового та бездротового протоколів, забезпечуючи надійну передачу даних та мінімальне втручання в роботу холодильника.

Сам пристрій служив центральним центром для збору та обробки даних датчиків, керування базою даних харчових продуктів і полегшення зв'язку з додатком і зовнішніми серверами. Його компактний дизайн і низьке енергоспоживання зробили його ідеальним для інтеграції в існуючі моделі холодильників без шкоди для функціональності чи естетики.

Підключення пристрою до Wi-Fi було необхідним для забезпечення віддаленого доступу та керування за допомогою супровідної програми. Це

включало конфігурацію пристрою для безпечного підключення до домашньої мережі користувача з використанням протоколів шифрування та механізмів автентифікації для захисту конфіденційних даних від несанкціонованого доступу.

Взаємодія з програмою була полегшена завдяки інтуїтивно зрозумілому інтерфейсу користувача та адаптивному дизайну, що дозволяло користувачам отримувати доступ до свого запасу їжі, переглядати терміни придатності та легко отримувати персоналізовані рекомендації щодо рецептів. Логіка програми була організована навколо функцій, орієнтованих на користувача, таких як сканування штрих – кодів, голосові команди та ручне введення, що забезпечує гнучкість і зручність для користувачів із різноманітними уподобаннями та дієтичними потребами.

Пріоритетом дизайну програми є простота та ясність із чіткими макетами, інтуїтивно зрозумілою навігацією та візуально привабливою графікою. Розширені функції, такі як прогнозна аналітика та алгоритми машинного навчання, були плавно інтегровані в інтерфейс програми, надаючи користувачам практичну інформацію та персоналізовані рекомендації на основі їхніх моделей споживання їжі та вподобань.

Підсумовуючи, розробка розумного холодильника та супровідної програми передбачала ретельне планування, інноваційний дизайн і суворе тестування, щоб забезпечити безперебійну роботу користувача та надійну роботу. Завдяки використанню передових технологій і принципів дизайну, орієнтованого на користувача, інтегроване рішення пропонує користувачам зручний і ефективний спосіб керування запасами їжі та оптимізації роботи на кухні.

## **2.4 Ілюстрація User Interface із описом**

Програма холодильника ініціалізує та відображає інформаційну панель зі статусом холодильника в реальному часі. Він отримує дані з AWS DynamoDB,

сортує та відображає продукти у форматі списку з такими деталями, як назва, кількість і термін придатності (рис 2.1).

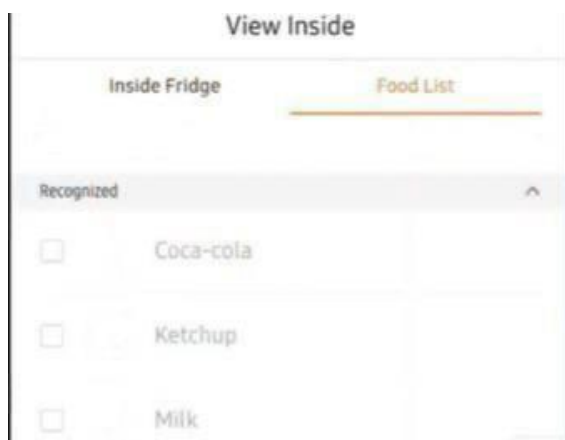


Рисунок 2.1 – Список продуктів

На цій же сторінці є вкладка Inside Fridge, що дозволяє поглянути актуальне наповнення холодильника що зображено на рисунку 2.2.

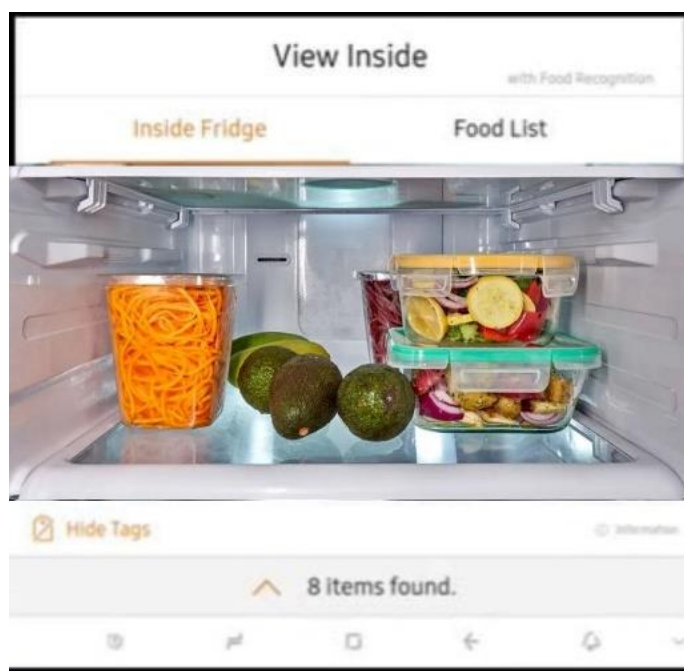


Рисунок 2.2 – Огляд продуктів у реальному часі

У додатку є вкладка рекомендованих рецептів, яка аналізує доступні продукти для створення персоналізованих пропозицій рецептів (рис. 2.3)

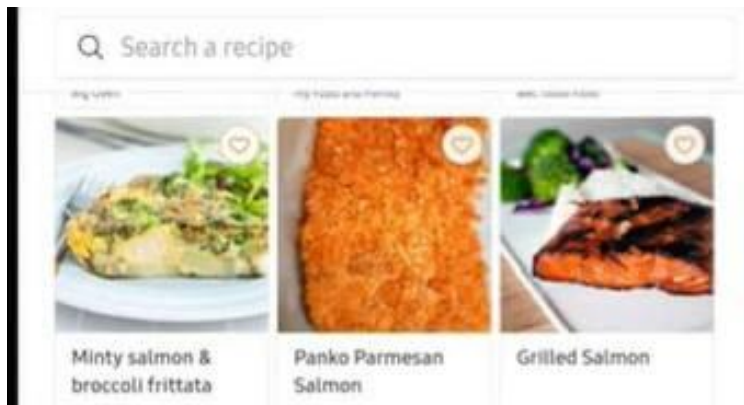


Рисунок 2.3 – Рекомендація рецептів

Программа надає push – повідомлення про продукти, термін придатності яких закінчився або скоро закінчиться, забезпечуючи проактивне управління та скорочуючи харчові відходи (рис. 2.4).



Рисунок 2.4 – Сповідження терміну придатності

## РОЗДІЛ 3

### ПОБУДОВА ДОДАТКУ У ВЗАЄМОДІЇ ІЗ СМАРТ ХОЛОДИЛЬНИКОМ

#### 3.1 Процес передачі даних ключові компоненти

Поглинання даних: JSON – подібний файл, що містить відсортовані дані датчиків, надходить до AWS DynamoDB. Це досягається за допомогою AWS SDK або API, які забезпечують повну інтеграцію з різними мовами програмування та платформами.

Коли нові дані потрапляють у DynamoDB, це запускає серію подій, які сповіщають програму про оновлення. Цей механізм сповіщень у режимі реального часу гарантує.

Програма отримує оновлені дані з DynamoDB за допомогою запитів або сканувань. Гнучкі можливості надсилання запитів DynamoDB дозволяють програмі отримувати конкретні точки даних на основі уподобань користувача або попередньо визначених критеріїв.

Після отримання даних програма обробляє їх, щоб отримати відповідну інформацію та відповідно оновити інтерфейс користувача. Це може включати обчислення, форматування та представлення даних у зручній формі.

Оновлені дані відображаються користувачеві в режимі реального часу через інтерфейс користувача програми. Це гарантує, що користувачі мають доступ до найновішої інформації про своє середовище IoT, дозволяючи їм приймати обґрунтовані рішення та вживати відповідних дій [17, 19, 20].

#### 3.2 Покращення користувача у застосунку побудованому на ІОТ: використання даних і функції програми Smart Refrigerator

У сфері додатків IoT інтеграція інтелектуальних пристроїв із зручними мобільними додатками кардинально змінила спосіб взаємодії людей із середовищем. Одним із таких прикладів є інтелектуальна програма для холодильників, розроблена для оптимізації управління швидкопсувними

товарами, використовуючи дані датчиків у реальному часі та розширені функції для підвищення зручності та ефективності користувача.

Використання та сортування даних.

Отримавши дані датчиків у режимі реального часу від розумного холодильника, програма бере на себе важливе завдання сортування та організації цієї інформації для використання користувачами. Використовуючи AWS DynamoDB для зберігання та пошуку даних, програма використовує складні алгоритми для класифікації продуктів на основі різних параметрів, таких як температура, вологість і термін придатності.

Завдяки інтелектуальним механізмам сортування програма гарантує, що користувачі можуть легко отримувати доступ до своїх запасів швидкопсувних товарів і керувати ними. Продукти відображаються у зручному для користувача форматі списку, надаючи такі важливі відомості, як назва, кількість і термін придатності з першого погляду.

Такий організований підхід спрощує процес управління запасами та сприяє проактивному прийняттю рішень щодо споживання та поповнення продуктів.

Рекомендовані рецепти.

Окрім відображення списку збережених продуктів, програма покращує взаємодію з користувачем, пропонуючи вкладку, присвячену рекомендованим рецептам. Використовуючи алгоритм підбору за налаштування користувача, програма пропонує рецепти на основі вибраного продукту зі списку.

Аналізуючи доступні інгредієнти та вподобання користувачів, програма генерує персоналізовані рекомендації щодо рецептів, адаптованих до індивідуальних смаків і дієтичних вимог. Ця функція не тільки заохочує користувачів максимально використовувати наявні запаси, але й надихає на креативність у плануванні та приготуванні їжі.

Додавання вручну терміну придатності та найменування продукту.

Щоб ще більше розширити можливості користувачів у управлінні своїми запасами, програма дозволяє вручну додавати терміни придатності до продуктів.

Користувачі можуть легко ввести термін придатності для кожного товару, забезпечуючи точне відстеження свіжості продукту та терміну придатності.

Крім того, програма пропонує зручну функцію, яка дозволяє користувачам натискати на продукт, який переглядається через камеру, і призначати йому назву. Використовуючи технологію розпізнавання зображень і дані користувача, програма ідентифікує продукт і додає його до списку запасів із відповідними деталями, такими як назва, кількість і термін придатності.

Щоб забезпечити безперебійну синхронізацію між додатком і інтелектуальним холодильником, усі внесені вручну та зміни дані про продукт автоматично оновлюються в режимі реального часу. Цей процес синхронізації використовує хмарне сховище та протоколи зв'язку, забезпечуючи послідовність і точність на обох платформах.

Крім того, програма містить push – повідомлення, щоб попередити користувачів про прострочені продукти або терміни придатності, що наближаються. Проактивно сповіщаючи користувачів про потенційні харчові відходи або проблеми з безпекою, додаток дає їм змогу вчасно вжити заходів, наприклад спожити продукт або додати його до списку покупок для заміни.

Підсумовуючи, додаток для розумного холодильника є прикладом конвергенції технології Інтернету речей і розробки мобільних додатків для надання комплексного рішення для управління швидкопсувними товарами. Завдяки ефективному використанню даних, персоналізованим рекомендаціям та інтуїтивно зрозумілим функціям програма покращує зручність для користувачів, сприяє екологічності харчових продуктів і дає змогу користувачам приймати обґрунтовані рішення щодо своїх харчових звичок.

### **3.3 Використання JSON**

JSON розшифровується як JavaScript Object Notation. Це дані, збережені у файлі .json і складаються з серії пар ключ/значення.

```
{ "ключ": "значення" }
```

Значення будь-якого ключа JSON може бути рядком, логічним значенням, числом, нулем, масивом або об'єктом. Коментарі не дозволені в JSON.

Хоча JSON нагадує об'єкт або масив, JSON – це рядок. Серіалізований рядок, що означає, що пізніше його можна проаналізувати та декодувати в типи даних.

Читання даних, збережених у форматі JSON за допомогою PHP, наведено у лістингу 3.1. По-перше, щоб детальніше зрозуміти, що JSON – це просто рядок, запишемо JSON у рядок PHP і застосуємо його до змінної під назвою \$data.

### Лістинг 3.1 – Читання даних збережених у форматі JSON за допомогою PHP

---

```
$data = '{
    "name": "Andriy",
    "race": "Elf"
}';
```

---

Кінець лістингу 3.1

Потім ми використаємо функцію `json_decode()`, щоб перетворити рядок JSON на об'єкт PHP (лістинг 3.2).

### Лістинг 3.2 – Функція `json_decode()`

---

```
$character = json_decode($data);
# Тепер ми можемо отримати доступ до нього як до об'єкта PHP.
echo $character->name;
```

---

Кінець лістингу 3.2

Ось результат.

Andriy

### 3.4 Опис реалізації застосунку без UI

У лістингу 3.3 зображено налаштування звіту про помилки для відображення всіх помилок і визначається шлях до файлу JSON (products.json), який зберігає дані про продукт. Це гарантує виявлення будь – яких проблем під час розробки та забезпечує єдину точку відліку для файлу даних продукту.

Лістинг 3.3 – Звіт про помилки

---

```
error_reporting(E_ALL);
ini_set('display_errors', 1);
define('PRODUCTS_FILE', 'products.json');
```

---

Кінець лістингу 3.3

У лістингу 3.4 зображено функцію завантаження продукту з файлу JSON, якщо він існує. Він зчитує вміст файлу та декодує дані JSON у масив PHP, роблячи дані продукту доступними для маніпулювання та відображення в програмі.

Лістинг 3.4 – Функція завантаження продукту

---

```
function loadProducts() {
    if (file_exists(PRODUCTS_FILE)) {
        $json = file_get_contents(PRODUCTS_FILE);
        return json_decode($json, true);
    }
    return [];
}
```

---

Кінець лістингу 3.4

У лістингу 3.5 зображено функцію зберігання поточного стану продуктів у файлі JSON. Він перетворює PHP – масив продуктів назад у рядок JSON і записує

його в products.json, забезпечуючи збереження будь – яких змін у даних продукту.

### Лістинг 3.5 – Зберігання поточного стану продуктів

---

```
function saveProducts($products) {
    $json = json_encode($products, JSON_PRETTY_PRINT);
    file_put_contents(PRODUCTS_FILE, $json);
}
```

---

Кінець лістингу 3.5

У лістингу 3.6 зображено рядок ініціалізування змінної \$products, завантажуючи поточний список продуктів із файлу JSON за допомогою функції loadProducts. Це гарантує, що програма має найновіші дані про продукт, доступні для подальших операцій.

### Лістинг 3.6 – Рядок ініціалізує змінну \$products

---

```
$products = loadProducts();
```

---

Кінець лістингу 3.6

У лістингу 3.7 зображено додавання нового продукту. Він перевіряє, чи було подано форму для додавання продукту, а потім створює новий продукт з унікальним ідентифікатором, назвою, терміном придатності та кількістю. Новий продукт додається до масиву, а оновлений список зберігається у файлі JSON.

### Лістинг 3.7 – Зображено додавання нового продукту

---

```
if (isset($_POST['add_product'])) {
    $newProduct = [
        'id' => uniqid(),
        'name' => $_POST['name'],
        'expiry_date' => $_POST['expiry_date'],
        'quantity' => (int)$_POST['quantity']
    ];
    $products[] = $newProduct;
    saveProducts($products);
}
```

---

Кінець лістингу 3.7

У лістингу 3.8 зображено перевірку терміну придатності кожного продукту. Він розраховує кількість днів до закінчення терміну придатності продукту та додає сповіщення, якщо продукт закінчився протягом 5 днів або залишилося менше 10 днів до псування. Функція `array_unique` гарантує відсутність дублікатів сповіщень.

### Лістинг 3.8 –Зображено перевірку терміну придатності

---

```
function checkExpiringProducts($products) {
    $notifications = [];
    $currentDate = new DateTime();
    foreach ($products as $product) {

        $expiryDate = new DateTime($product['expiry_date']);
        $interval = $currentDate->diff($expiryDate)->days;

        if ($interval <= 5) {
            $notifications[] = "Product '{$product['name']}' is expiring in $interval
day(s).";
        }

        if ($interval <= 10) {
            $notifications[] = "Product '{$product['name']}' has under 10 days left to
spoil.";
        }
    }
    return array_unique($notifications);
}
```

---

### Кінець лістингу 3.8

У лістингу 3.9 зображено блок який обробляє редагування існуючого продукту. Він переглядає продукти, щоб знайти той, що має відповідний ідентифікатор, і оновлює його назву, термін придатності та кількість значеннями з форми. Потім оновлений список зберігається у файлі JSON.

## Лістинг 3.9 – Зображено блок який обробляє редагування продукту

---

```

if (isset($_POST['edit_product'])) {
    foreach ($products as &$amp;product) {
        if ($product['id'] === $_POST['product_id']) {
            $product['name'] = $_POST['name'];
            $product['expiry_date'] = $_POST['expiry_date'];
            $product['quantity'] = (int)$_POST['quantity'];
            break;
        }
    }
    saveProducts($products);
}

```

---

Кінець лістингу 3.9

У лістингу 3.10 зображено блок, який обробляє видалення продукту за назвою. Він відфільтровує продукт із вказаною назвою зі списку та зберігає решту продуктів назад у файл JSON. Ця операція гарантує, що вказаний продукт буде видалено з даних.

## Лістинг 3.10 – Зображено блок який обробляє видалення продукту

---

```

if (isset($_POST['delete_product'])) {
    $productName = $_POST['product_name'];
    $products = array_filter($products, function($product) use ($productName) {
        return $product['name'] !== $productName;
    });
    saveProducts($products);
}

```

---

Кінець лістингу 3.10

У лістингу 3.11 зображена функція яка обчислює кількість днів, що залишилися до закінчення терміну придатності продукту. Він бере термін придатності продукту, порівнює його з поточною датою та повертає кількість днів, що залишилися. Це використовується для визначення терміновості повідомлень.

## Лістинг 3.11 – Зображено функцію яка обчислює кількість днів

---

```
function daysLeftToSpoil($expiryDate) {
    $currentDate = new DateTime();
    $expiryDate = new DateTime($expiryDate);
    return $currentDate->diff($expiryDate)->days;
}
```

---

Кінець лістингу 3.11

У лістингу 3.12 зображена функція перетворює масив продуктів у відформатований рядок JSON і відображає його в зручному для читання форматі. Це забезпечує швидкий спосіб переглянути весь набір даних про продукт у форматі JSON на веб – сторінці.

## Лістинг 3.12 – Функція перетворює масив у відформатований рядок

---

```
function displayProducts($products) {
    echo '<pre>' . json_encode($products, JSON_PRETTY_PRINT) . '</pre>';
}
```

---

Кінець лістингу 3.12

У лістингу 3.13 зображена функція відображає спрощений список продуктів із зазначенням лише їх назви, кількості та днів, що залишилися до псування. Він форматує кожен продукт у елемент списку, що полегшує користувачам швидкий перегляд статусу кожного продукту.

## Лістинг 3.13 – Відображає спрощений список продуктів

---

```
function displayFormattedProducts($products) {
    echo '<ul>';
    foreach ($products as $product) {
        $daysLeft = daysLeftToSpoil($product['expiry_date']);
        echo "<li>{$product['name']} ({$product['quantity']}) - {$daysLeft} days left to spoil</li>";
    }
    echo '</ul>';
}
```

---

Кінець лістингу 3.13

У лістингу 3.14 зображена функція, яка виділяє продукти, до псування яких залишилося менше 10 днів, і вибирає для кожного рекомендований рецепт. Він запитує зовнішній API, використовуючи назву продукту, отримує перший рецепт і відображає назву страви та інструкції з приготування, надаючи практичні поради щодо використання продукту.

### Лістинг 3.14 – Функція яка виділяє продукти

---

```
function highlightProductsUnder10Days($products) {
    $productsToHighlight = array_filter($products, function($product) {
        return daysLeftToSpoil($product['expiry_date']) <= 10;
    });

    foreach ($productsToHighlight as $product) {
        $productName = urlencode($product['name']);
        $apiResponse =
file_get_contents("https://www.themealdb.com/api/json/v1/1/search.php?s=
{$productName}");
        $apiData = json_decode($apiResponse, true);

        if (!empty($apiData['meals'])) {
            $recipe = $apiData['meals'][0];
            $meal = $recipe['strMeal'];
            $instructions = $recipe['strInstructions'];
            echo "<p>It is recommended to use {$product['name']}:</p>";
            echo "<p><strong>{$meal}</strong></p>";
            echo "<p>{$instructions}</p>";
        }
    }
}
```

---

### Кінець лістингу 3.14

Структура HTML для веб – інтерфейсу програми Smart Fridge. Містить форми для додавання, редагування та видалення товарів, а також відображає поточні продукти, продукти, що наближаються до псування, і повідомлення. Різні функції викликаються для заповнення даними на веб – сторінці.

### 3.5 Опис роботи застосунку

Основні робочі елементи застосунку зображено у спрощеній формі на рисунку 3.1. Секція Current Products показує список продуктів, які є у наявності. Відображаються ці продукти у форматі name (quantity) – Number days left to spoil.

**Smart Fridge Product Manager**

**Current Products**

- fish (1) - 0 days left to spoil
- beef (1) - 4 days left to spoil

**Add New Product**

Product Name:

Expiry Date:

Quantity:

**Edit Product**

Product ID:

New Product Name:

New Expiry Date:

New Quantity:

**Delete Product**

Product Name:

**Products Under 10 Days to Spoil**

It is recommended to use fish:

**Fish pie** - 01 Put the potatoes in a large pan of cold salted water and bring to the boil. Lower the heat, cover, then simmer gently for 15 minutes until tender. Drain, then return to the pan over a low heat for 30 seconds to drive off any excess water. Mash with 1 tbsp olive oil, then season. 02 Meanwhile put the milk in a large sauté pan, add the fish and bring to the boil. Remove from the heat, cover and stand for 3 minutes. Remove the fish (reserving the milk) and pat dry with kitchen paper, then gently shake into an ovenproof dish, discarding the skin and any bones. 03 Heat the remaining oil in a pan, stir in the flour and cook for 30 seconds. Gradually stir in 200-250ml of the reserved milk (discard the rest). Grate in nutmeg, season, then bubble until thick. Stir in the cream. 04 Preheat the oven to 190°C (fan 170°C gas 5). Grate the artichokes and add to the fish with the leek, prawns and herbs. Stir the lemon zest and juice into the sauce, then pour over. Mix gently with a wooden spoon. 05 Spoon the mash onto the fish mixture, then use a fork to make peaks, which will crisp and brown as it cooks. Sprinkle over the cheese, then bake for 35-40 minutes until golden and bubbling. Serve with wilted greens.

It is recommended to use beef:

**Beef Asado** - 0 Combine beef, crushed peppercorns, soy sauce, vinegar, dried bay leaves, lemon, and tomato sauce. Mix well. Marinate beef for at least 30 minutes. 1 Put the marinated beef in a cooking pot along with remaining marinade. Add water. Let boil. 2 Add Knorr Beef Cube. Stir. Cover the pot and cook for 40 minutes in low heat. 3 Turn the beef over. Add tomato paste. Continue cooking until beef tenderizes. Set aside. 4 Heat oil in a pan. Fry the potato until it browns. Turn over and continue frying the opposite side. Remove from the pan and place on a clean plate. Do the same with the carrots. 5 Save 3 tablespoons of cooking oil from the pan where the potato was fried. Sauté onion and garlic until onion softens. 6 Pour-in the sauce from the beef stew. Let boil. Add the beef. Cook for 2 minutes. 7 Add butter and let it melt. Continue cooking until the sauce reduces to half.

**Notifications**

- Product fish is expiring in 0 day(s).
- Product fish has under 10 days left to spoil.
- Product beef is expiring in 4 day(s).
- Product beef has under 10 days left to spoil.

Рисунок 3.1 – Зображено елементи застосунку

Наступна секція Add New Product, надає можливість додати новий продукт. Для цього необхідно вказати найменування продукту, термін придатності та кількість цього продукту. Натиснувши кнопку Add new product, новий продукт буде додано до списку доступних

Секція Edit Product, надає можливість оновлювати дані про продукт. А саме за допомогою Product ID ми можемо обрати певний продукт, перейменувати за допомогою поля New Name, зазначити зміну у терміні придатності у секції New Expiry Date за необхідності та змінити кількість яка є в наявності, зазначивши її у полі New Quantity.

Секція Delete Product використовує ім'я продукту як ID. Вписавши туди ім'я та натиснувши клавішу Delete Product ми видалимо продукт із списку.

Секція Products Under 10 Days to Spoil демонструє нам продукти у яких термін придатності менше 10 днів. А також рекомендує нам рецепт із використанням цього продукту. Ми достаємо рецепт за допомогою Api із

спеціалізованого сайту з базою даних наповненою рецептами. Рецепт підбирається за наявності продукту у інгредієнтах.

Секція Notifications відпраляє нам сповіщення, в якому сказано що певний продукт має термін придатності який вичерпується за 10 днів. Тому слід скористатись цим продуктом. Сповіщення дублюється двічі на день.

## ВИСНОВКИ

Здійснено огляд доступних інструментів для реалізації застосунку в сфері IoT для керування розумним холодильником. Обрано PHP мову для написання застосунку, через використання ООП, що значно полегшує написання коду та зменшує його об'єм. Також обрано JSON Арі, як метод передачі даних, який є безпечним та простим у розумінні комп'ютеру та людині.

Розроблено адаптивну інтерфейсну програму, використовуючи HTML, CSS, PHP і JS для відображення даних про стан продуктів та забезпечено зв'язок додатку із зовнішніми базами даних. Додаток побудовано на мові PHP, подані результати за допомогою HTML та CSS. Додаток отримує данні про рецепти за допомогою JSON Арі із зовнішньої баз даних. Програма має можливість ефективно оновлювати данні про продукти з участю користувача та отримувати поради з використання продуктів.

Візуалізовано інструменти подачі даних, щоб надати користувачам чітке уявлення про стан збережених продуктів. Доступне ведення даних користувачем та за допомогою арі, данні представленні у доступному форматі для взаємодії із користувачем. Кожен окремий елемент застосунку має детальний опис та перелік можливостей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Comprehensive Review on Food Waste Reduction Based on IoT and Big Data Technologies. URL: <https://doi.org/10.3390/su15043482> (дата звернення: 27.04.2024).
2. A S. G. IoT based smart refrigerator and shopping system prototype. IEEE Xplore. URL: <https://ieeexplore.ieee.org/document/10434662> (date of access: 06.02.2024).
3. Amazon is reportedly working on a smart fridge that tracks what's inside. URL: <https://techcrunch.com/2021/10/05/amazon-is-reportedly-working-on-a-smart-fridge-that-tracks-whats-inside> (дата звернення: 03.05.2024).
4. Amazon is reportedly working on a smart fridge. URL: <https://www.theverge.com/2021/10/5/22711240/amazon-smart-fridge-cameras-go-stores>. (дата звернення: 10.04.2024).
5. An intelligent IoT – based control and traceability system to forecast and maintain water quality in freshwater fish farms. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0168169919305320> (дата звернення: 19.05.2024).
6. Analysis of Different Techniques Used to Reduce The Food Waste Inside The Refrigerator. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9792896> (дата звернення: 12.05.2024).
7. Artificial Intelligence for Product Quality Inspection toward Smart Industries: Quality Control of Vehicle Non – Conformities. URL: <https://ieeexplore.ieee.org/abstract/document/9080396> (дата звернення: 14.05.2024).
8. Challenges of Testing Complex Internet of Things (IoT) Devices and Systems. URL: <https://ieeexplore.ieee.org/abstract/document/8919324> (дата звернення: 17.05.2024).
9. Data Quality Best Practices in IoT Environments. URL: <https://ieeexplore.ieee.org/abstract/document/8590201> (дата звернення: 15.04.2024).

10. Data Quality Management in the Internet of Things. URL: <https://www.mdpi.com/1424-8220/21/17/5834> (дата звернення: 10.05.2024).
11. Design of Low – Cost Object Identification Module for Culinary Applications. URL: <https://doi.org/10.1088/1742-6596/1964/6/062088> (дата звернення: 05.04.2024).
12. Development of IoT – Based Sensor Tag for Smart Factory. URL: <https://www.zelusinternational.com/index.php/IRJECE/article/view/372> (дата звернення: 18.04.2024).
13. End – to – end industrial IoT platform for Quality 4.0 applications. URL: <https://www.sciencedirect.com/science/article/pii/S0166361521001986> (дата звернення: 08.05.2024).
14. ENHANCING Enhancing Textile Quality Control With Iot Sensors: A Case Study Of Automated Defect Detection. URL: <https://globalmainstreamjournal.com/index.php/IJMISDS/article/view/113> (дата звернення: 11.05.2024).
15. F H. K. Heat sources in a biosafety cabinet compromise experimental and user protection. BSC MythBusters white paper. URL: <https://ieeexplore.ieee.org/document/10511172> (date of access: 03.04.2024).
16. Hnatiuc P. A. An intelligent iot – based food quality monitoring approach using low – cost sensors. MDPI. URL: <https://doi.org/10.3390/sym11030374> (date of access: 01.03.2024).
17. Implementation of Iot Based Smart Cooking Environment. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9063796> (дата звернення: 21.03.2024).
18. Intelligent Refrigerator using Machine Learning and IoT. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9792896> (дата звернення: 14.05.2024).
19. Internet of Things, IoT. URL: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot> (дата звернення: 15.05.2024).

20. IoT enabled food recommender with NIR system. URL: <https://doi.org/10.1016/j.matpr.2020.11.302> (дата звернення: 06.04.2024).
21. IoT, Або Інтернет Речей, Що Це Таке. URL: <https://hub.kyivstar.ua/news/iot-abo-internet-rechey> (дата звернення: 04.04.2024).
22. Kumar V. Fruit recognition using the convolutional neural network. AIM. URL: <https://analyticsindiamag.com/fruit-recognition-using-the-convolutional-neural-network/> (date of access: 26.04.2024).
23. Planar and printed antennas for Internet of Things-enabled environment: Opportunities and challenges. URL: <https://doi.org/10.1002/dac.4940> (дата звернення: 26.05.2024).
24. Quality Control in the Context of Industry 4.0. URL: [https://link.springer.com/chapter/10.1007/978-3-030-14973-4\\_17](https://link.springer.com/chapter/10.1007/978-3-030-14973-4_17) (дата звернення: 19.04.2024).
25. Quality management in product recovery using the Internet of Things: An optimization approach. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0166361513002364> (дата звернення: 14.04.2024).
26. Smart Refrigerator System: Enabling Healthy Cooking. URL: [https://doi.org/10.1007/978-981-99-0293-4\\_87](https://doi.org/10.1007/978-981-99-0293-4_87) (дата звернення: 23.03.2024).
27. Smart Refrigerator: An IOT and Machine learning based Approach. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9154025> (дата звернення: 03.05.2024).
28. Sneha S. G. IoT based smart refrigerator and shopping system prototype.IEEEExplore. URL:<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10434662> (date of access: 03.04.2024).
29. TheMealDB URL: <https://www.themealdb.com/api.php> (дата звернення: 13.05.2024).
30. Tiny Machine Learning for High Accuracy Product Quality Inspection. URL: <https://ieeexplore.ieee.org/abstract/document/9969601/> (дата звернення: 12.04.2024).

31. Vrinda G. J. Blood guard: an IoT based innovative rfid – enhanced blood storage system with biometric security and real – time temperature monitoring. IEEE Xplore. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10511172> (date of access: 07.05.2024).
32. База даних фотографі фруктів та овочів. URL: <https://www.kaggle.com/datasets/mbkinaci/fruit-images-for-object-detection> (дата звернення: 17.04.2024).
33. В Г. О. Ефективне використання інтернету речей (іот). doi. URL: <https://doi.org/10.32782/2522-1256-2023-39-06> (дата звернення: 02.04.2024).
34. Навчання нейронної мережі. URL: <https://analyticsindiamag.com/fruit-recognition-using-the-convolutiona-neural-network> (дата звернення: 23.04.2024).
35. Операційна система Raspberry Pi. URL: <https://www.raspberrypi.com/software/> (дата звернення: 13.04.2024).
36. Що таке інтернет речей і як він працює? URL: <https://server-shop.ua/ua/the-internet-of-things-and-the-scope-of-its-use.html> (дата звернення: 14.05.2024).