

**МАГІСТЕРСЬКА РОБОТА**

**МР. ШМ - 08.00.00.000 ПЗ**

**Група ШМ-23-1**

**Боднарчук Віталій**

**2024**

**Івано-Франківський національний технічний університет нафти і газу**

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

**Боднарчук Віталій Володимирович**

(прізвище, ім'я, по батькові)

УДК 004.942  
(індекс)

## **МАГІСТЕРСЬКА РОБОТА**

**Моделі, алгоритми та веб-технології автоматизації планування**

**харчування на основі індивідуальних потреб**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Боднарчук В.В.**

(підпис, ініціали та прізвище здобувача освітнього ступеня)

**Науковий керівник Романишин Тарас Любомирович, к.т.н., доцент**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Допущено до захисту**

Завідувач кафедри

доц. **Бандура В.В.**

(посада) (підпис) (дата) (ініціали та прізвище)

**Нормоконтроль**

доц. **Вовк Р.Б.**

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

**Івано-Франківський національний технічний університет нафти і газу**

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доц.

В.В. Бандура

“ 04 ” вересня 2024 р.

# ЗАВДАННЯ

## НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Боднарчуку Віталію Володимировичу**

(прізвище, ім'я, по-батькові)

**1. Тема магістерської роботи “Моделі, алгоритми та веб-технології автоматизації планування харчування на основі індивідуальних потреб”**

керівник проекту (роботи) Романишин Тарас Любомирович, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781 /7

**2. Строк подання студентом проекту (роботи) 17 грудня 2024 р.**

**3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних систем і програмного забезпечення**

**4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Дослідження предметної області автоматизації планування харчування

2. Розробка моделей, алгоритмів і методології планування

3. Підготовка даних для індивідуалізованого планування харчування

4. Імплементация веб-технологій автоматизації процесу

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

1. Формула на визначення добової норми калорій (рис. 1.2)

2. Приклад готового збалансованого раціону (таб. 1.1)

3. Візуалізація вибору фізичної активності (рис. 2.1)

4. План НАССР (рис. 2.3)

5. Візуалізація подачі інформації про раціони (рис. 2.4)

## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2024 р.

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження предметної області автоматизації планування харчування	15.10.2024	виконано
4	Розробка моделей, алгоритмів і методології планування	08.11.2024	виконано
5	Підготовка даних для індивідуалізованого планування харчування	20.11.2024	виконано
6	Імплементация веб-технологій автоматизації процесу	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр

\_\_\_\_\_ (підпис)

Керівник роботи

\_\_\_\_\_ (підпис)

## АНОТАЦІЯ

**Магістерська робота:** 77 с., 21 рис., 2 табл., 30 джерел.

**Тема:** Моделі, алгоритми та веб-технології автоматизації планування харчування на основі індивідуальних потреб.

**Мета роботи:** Розробка та впровадження моделей, алгоритмів і веб-технологій для автоматизації процесу планування харчування на основі індивідуальних характеристик користувачів.

**Об'єкт дослідження:** Процеси автоматизації планування харчування на основі аналізу індивідуальних потреб.

**Предмет дослідження:** Моделі, алгоритми та веб-технології для автоматизованого планування харчування з урахуванням індивідуальних параметрів і потреб.

### **Результати дослідження**

У роботі запропоновано методику інтеграції автоматизованого планування харчування із сучасними веб-технологіями, зокрема розроблено інтерфейс для інтеграції з платформою WordPress.

### **Висновок**

В результаті проведених досліджень вдалося сформулювати теоретичні та практичні основи автоматизації планування харчування, розробити ефективні алгоритми та запропонувати інноваційні підходи до використання веб-технологій у цьому процесі.

**АВТОМАТИЗАЦІЯ ПЛАНУВАННЯ ХАРЧУВАННЯ,  
ІНДИВІДУАЛЬНІ ХАРЧОВІ ПОТРЕБИ, МОДЕЛЮВАННЯ ТА  
АЛГОРИТМИ, ВЕБ-ТЕХНОЛОГІЇ, БАЗИ ДАНИХ ПРОДУКТІВ,  
ОПТИМІЗАЦІЯ МЕНЮ, АНАЛІЗ ДАНИХ, ІНТЕГРАЦІЯ З  
WORDPRESS.**

## ABSTRACT

**Master Thesis:** 77 pp., 21 fig., 2 tab., 30 sources.

**Thesis Subject:** Models, algorithms, and web technologies for automating meal planning based on individual needs.

**Purpose of work:** Development and implementation of models, algorithms, and web technologies to automate the meal planning process based on individual user characteristics.

**Object of research:** Processes of automating meal planning based on the analysis of individual needs.

**Subject of research:** Models, algorithms, and web technologies for automated meal planning, considering individual parameters and needs.

### **Research results**

The paper proposes a methodology for integrating automated meal planning with modern web technologies, in particular, the WordPress platform.

### **Conclusion**

The master's research managed to formulate theoretical and practical foundations for automating meal planning, develop efficient algorithms, and propose innovative approaches to leveraging web technologies in this process.

**MEAL PLANNING AUTOMATION, INDIVIDUAL NUTRITIONAL NEEDS, MODELING AND ALGORITHMS, WEB TECHNOLOGIES, FOOD DATABASES, MENU OPTIMIZATION, DATA ANALYSIS, WORDPRESS INTEGRATION.**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	9
ВСТУП.....	10
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ ХАРЧУВАННЯ НА ОСНОВІ ІНДИВІДУАЛЬНИХ ПОТРЕБ .....	13
1.1. Опис проблеми дослідження .....	13
1.2. Поняття індивідуального харчового плану на основі автоматизованого аналізу потреб .....	14
1.2.1. Визначення і таксономія.....	15
1.2.2. Інструменти для створення індивідуального харчового плану .....	17
1.2.3. Набори даних для порівняльного аналізу.....	18
1.3. Процес збору та підготовки даних .....	19
1.4 Концепція автоматизації планування харчування як інструменту оптимізації.....	23
Висновки до розділу .....	25
РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МОДЕЛЕЙ, МЕТОДІВ ТА МЕТОДОЛОГІЇ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ ХАРЧУВАННЯ.....	27
2.1. Формула Сан-Жеора як основа розрахунків.....	27
2.2. Алгоритми оптимізації харчових раціонів .....	31
2.3. Огляд методології дослідження .....	35
2.4. Підготовка даних для автоматизації харчового планування .....	38
2.4.1. Збір даних про користувача .....	38
2.4.2. Очищення та стандартизація інформації.....	38
2.5. Побудова моделей автоматизації .....	38
2.5.1 Параметри моделі індивідуалізації харчових планів .....	39
2.5.2. Налаштування алгоритмів автоматизації .....	40
2.6. Валідація та тестування інструментів.....	41
2.6.1. Методи перевірки точності розрахунків .....	41

2.6.2. Оцінка ефективності використання .....	42
2.6.3. Валідація на основі контрольних даних .....	42
Висновки до розділу .....	43
<b>РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗАЦІЇ</b>	
<b>ПЛАНУВАННЯ ХАРЧУВАННЯ .....</b>	<b>44</b>
3.1. Загальна стратегія реалізації рішення.....	44
3.2. Реалізація програми імпорту даних .....	47
3.2.1. Контекст, область застосування та підхід до імпорту .....	48
3.2.2 Інтеграція WordPress із Next.js.....	52
3.2.3. Налаштування WPGraphQL для доступу до даних .....	55
3.2.4. Фіналізація імпорту даних .....	56
3.3. Моделювання автоматизованого планування харчування .....	57
3.3.1. Контекст, область застосування та підхід до моделювання .....	57
3.3.2. Реалізація структурних блоків на Next.js .....	59
3.3.3. Вибір та використання алгоритмів для оптимізації раціону .....	61
3.3.4. Тестування алгоритмів і моделі .....	63
3.3.5. Валідація результатів планування .....	65
3.4. Виклики в процесі автоматизації планування .....	66
3.4.1. Проблеми імпорту та обробки даних з WordPress .....	68
3.4.2. Оптимізація продуктивності на Vercel .....	69
Висновки до розділу .....	71
<b>ВИСНОВКИ .....</b>	<b>73</b>
<b>ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....</b>	<b>75</b>

## **ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

MP – Meal Planning

BMR – Basal Metabolic Rate

TDEE – Total Daily Energy Expenditure

WP – WordPress

WC – WooCommerce

DB – Database

API – Application Programming Interface

NLP – Natural Language Processing

RDA – Recommended Dietary Allowance

UI – User Interface

UX – User Experience

BMI – Body Mass Index

CSV – Comma-Separated Values

REST – Representational State Transfer

JSON – JavaScript Object Notation

LTV – Lifetime Value

## ВСТУП

### **Актуальність теми.**

Сучасний ритм життя диктує необхідність автоматизації багатьох аспектів, у тому числі процесу планування харчування. Індивідуальний підхід до раціону базується на обліку енергетичних потреб, харчових вподобань, стану здоров'я та особистих цілей, таких як підтримка ваги або покращення фізичної форми. Проте ручне створення таких планів є трудомістким та схильним до помилок, що ускладнює процес для користувачів.

Автоматизація планування харчування на основі індивідуальних параметрів дозволяє значно полегшити цей процес, зробити його більш доступним та ефективним [1]. Завдяки сучасним веб-технологіям та алгоритмам, зокрема інтеграції з платформою WordPress та її модулями, можна створити інтуїтивно зрозумілий сервіс для формування персоналізованих харчових планів.

Розв'язання цієї проблеми є важливим у контексті глобальної тенденції до здорового способу життя. Автоматизовані системи планування харчування мають великий потенціал у таких напрямках:

- забезпечення індивідуальних рекомендацій відповідно до наукових даних;
- зменшення часу на підготовку раціонів;
- зручність у використанні як для звичайних користувачів, так і для професіоналів, наприклад, дієтологів [2].

Таким чином, створення моделей, алгоритмів і веб-технологій для автоматизації цього процесу є не лише актуальною науковою задачею, а й важливим практичним завданням, яке сприяє розвитку галузі охорони здоров'я та технологій у сфері харчування.

**Мета дослідження** - розробка та впровадження моделей, алгоритмів і веб-технологій для автоматизації планування харчування на основі індивідуальних потреб.

**Об'єкт дослідження** - процеси автоматизації планування харчування з використанням інформаційних та веб-технологій.

**Предмет дослідження** - методи, моделі та інструменти автоматизації планування харчування з урахуванням індивідуальних параметрів користувачів.

Задачі дослідження:

- провести аналіз предметної області автоматизації планування харчування, визначити основні концепції та методи;
- розробити структуру бази даних харчових продуктів та алгоритми їх підбору відповідно до індивідуальних потреб;
- інтегрувати розроблену систему з веб-платформою WordPress;
- провести тестування і валідацію алгоритмів на відповідність заданим критеріям точності та ефективності.

**Методи дослідження**

Для досягнення мети використано:

- алгоритми розрахунку енергетичних потреб (наприклад, рівняння Mifflin-St Jeor);
- методи обробки даних, включаючи роботу з базами даних і веб-інтерфейсами;
- сучасні веб-технології для інтеграції з WordPress та модулем WooCommerce.

**Наукова новизна отриманих результатів**

Запропоновано новий підхід до автоматизації планування харчування, який поєднує адаптивні алгоритми, інтеграцію з WordPress та персоналізацію раціонів на основі індивідуальних параметрів.

**Практичне значення результатів**

Розроблена система може бути впроваджена як інструмент для автоматизації роботи дієтологів або як доступний сервіс для широкого загалу. Це сприяє підвищенню зручності користування, якості рекомендацій та ефективності управління харчуванням у повсякденному житті.

**Структура магістерської роботи.** Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 77 сторінок, і містить 21 рисунок, 2 таблиці, список використаних джерел із 30 найменувань.

# РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ ХАРЧУВАННЯ НА ОСНОВІ ІНДИВІДУАЛЬНИХ ПОТРЕБ

## 1.1. Опис проблеми дослідження

У сучасному світі здоровий спосіб життя набуває все більшої популярності, а відповідно зростає попит на індивідуальні підходи до харчування. Планування раціону на основі персоналізованих даних, таких як вік, вага, ріст, рівень фізичної активності, стан здоров'я та харчові вподобання, є важливим елементом досягнення збалансованого харчування. Однак цей процес залишається складним і трудомістким для багатьох людей через необхідність обробки значного обсягу інформації.

Більшість сучасних підходів до складання раціону базуються на ручному аналізі даних, що є не тільки ресурсозатратним, але й схильним до помилок [3]. Крім того, універсальні дієти часто не враховують індивідуальних особливостей, що знижує їхню ефективність. Такий підхід ускладнює створення раціонального харчування, яке було б не лише корисним, але й зручним у використанні для користувачів.

Зі стрімким розвитком цифрових технологій з'явилася можливість автоматизації цього процесу шляхом інтеграції сучасних веб-рішень та алгоритмів обробки даних. Проте більшість доступних рішень є фрагментованими або мають обмежений функціонал, що не дозволяє створювати повноцінні індивідуальні плани харчування. Основні проблеми, що виникають у цьому контексті:

- недостатня інтеграція інструментів автоматизації;
- відсутність доступних і зручних для користувачів рішень, які б забезпечували персоналізований підхід до харчування.

Одним із перспективних напрямів є використання веб-технологій для автоматизації процесу планування харчування. Інтеграція таких систем із

платформою WordPress дозволяє створити зручний для користувача інтерфейс із доступом до бази даних продуктів, функціоналом для розрахунку енергетичних потреб і формування раціону.

Однак під час розробки таких систем виникають численні виклики. Наприклад, необхідність адаптації алгоритмів до різних груп користувачів, забезпечення швидкого доступу до великих обсягів даних. Особливо це актуально для інтеграції з платформами електронної комерції, такими як WooCommerce, де персоналізація є ключовим фактором для покращення користувацького досвіду [4].

Для подолання цих проблем необхідно створити комплексну систему, яка включатиме:

- 1) базу даних продуктів із відповідною інформацією про поживну цінність;
- 2) алгоритми для розрахунку калорійності та складу раціону відповідно до індивідуальних параметрів користувача;
- 3) інтеграцію з веб-платформами для забезпечення зручного інтерфейсу користувача;
- 4) тестування та оцінку точності роботи системи на практичних прикладах.

Метою цього дослідження є розробка і впровадження веб-системи для автоматизації планування харчування на основі персоналізованих даних. Дослідження зосереджене на розробці алгоритмів, інтеграції їх із платформою WordPress і валідації отриманих результатів. Завдяки цьому можна значно підвищити ефективність створення індивідуальних харчових планів і зробити їх доступними для широкого загалу користувачів.

## **1.2. Поняття індивідуального харчового плану на основі автоматизованого аналізу потреб**

Визначення індивідуального харчового плану є важливим етапом для досягнення збалансованого харчування, яке враховує специфічні потреби кожної людини. Автоматизований аналіз цих потреб є важливим інструментом

у сучасних технологіях харчування, оскільки він дозволяє на основі даних про здоров'я, фізичну активність, метаболізм та інші індивідуальні фактори створювати персоналізовані плани харчування, що забезпечують найкращий результат для користувача.

Основним завданням автоматизованого аналізу є обробка великих обсягів текстових даних, які описують харчові продукти, їхні властивості, калорійність, вітаміни та мікроелементи, що містяться в їжі. Для цього використовуються методи обробки природної мови (NLP), що дозволяють вилучати потрібну інформацію з текстових описів продуктів та інших джерел. Такий аналіз забезпечує точність визначення потреб людини в конкретних компонентах харчування, на основі яких можна побудувати індивідуальний харчовий план.

Концепція індивідуального харчового плану включає кілька етапів: аналіз здоров'я користувача, визначення його харчових потреб, підбір продуктів та їх порцій для досягнення бажаного результату. Всі ці етапи можуть бути автоматизовані за допомогою алгоритмів машинного навчання та тематичного моделювання, що дозволяють не тільки врахувати наявні дані, але й прогнозувати можливі зміни в потребах користувача з часом.

У цьому розділі буде розглянуто основні принципи автоматизованого аналізу потреб і їх використання для формування індивідуальних харчових планів. Також будуть представлені методи, інструменти та ресурси, які використовуються для цього процесу, а також оцінка ефективності застосування цих технологій у реальних умовах.

### *1.2.1. Визначення і таксономія*

Індивідуальний харчовий план можна визначити як персоналізовану стратегію харчування, яка враховує потреби конкретної людини на основі різних факторів, таких як вік, стать, фізична активність, здоров'я, метаболізм, а також особисті переваги та обмеження. Визначення індивідуальних харчових

потреб є основою для створення таких планів і може бути поділене на кілька етапів.

Індивідуальний харчовий план можна розглядати в двох аспектах: як процес і як результат. Процес включає етапи збору даних про користувача, їх аналізу, побудови рекомендацій та постійного моніторингу ефективності харчування. Результатом є конкретний план, що включає перелік продуктів, їх кількість і частоту вживання, з урахуванням різних харчових компонентів і калорійності. Такі плани можуть бути адаптовані під конкретні цілі користувача, такі як схуднення, набір маси, підтримка здоров'я або профілактика хвороб.

Таксономія індивідуальних харчових планів базується на класифікації потреб, які можуть бути різними в залежності від мети, стану здоров'я та фізичної активності людини. Одним із основних підходів є визначення основних категорій харчових компонентів: білки, жири, вуглеводи, вітаміни та мінерали. Додатково враховуються такі параметри, як калорійність, індекс маси тіла, рівень фізичної активності, а також спеціальні дієти для людей з певними захворюваннями (наприклад, для діабетиків або людей з харчовими алергіями) [5].

Процес автоматизованого аналізу харчових потреб включає використання алгоритмів машинного навчання та обробки природної мови для збору і аналізу даних. Важливим елементом цього процесу є видобуток та очищення текстових даних про продукти харчування з різних джерел, таких як бази даних про харчову цінність продуктів, наукові дослідження та медичні рекомендації. Ці дані використовуються для створення точних індивідуальних рекомендацій, які враховують усі аспекти потреб користувача.

Таксономія індивідуальних харчових планів, на основі автоматизованого аналізу потреб, забезпечує чітке і ефективне формулювання рекомендацій, що дозволяють досягти бажаних результатів у короткостроковій та довгостроковій перспективі.

### 1.2.2. Інструменти для створення індивідуального харчового плану

В останні роки для автоматизованого створення індивідуальних харчових планів було розроблено кілька інструментів, які використовують різноманітні методи аналізу, щоб допомогти людям підібрати раціон відповідно до їхніх потреб. Цей розділ надасть огляд деяких з найбільш актуальних інструментів, які застосовуються для створення індивідуальних харчових планів [6].

Одним з перших інструментів, який став основою для подібних розробок, є MyFitnessPal. Цей додаток дозволяє користувачам вводити свої особисті дані, такі як вага, вік, рівень фізичної активності, і на основі цього генерує індивідуальний харчовий план, враховуючи кількість калорій та необхідні поживні речовини. MyFitnessPal має велику базу даних про продукти харчування, що дозволяє створювати точні плани на основі реальних продуктів. Робота додатку показана на рисунку 1.1.



Рис. 1.1. Робота додатку MyFitnessPal

Іншим популярним інструментом є Eat This Much, який автоматично генерує харчові плани на основі заданих параметрів користувача. Цей сервіс також враховує обмеження по дієтах, такі як веганство, безглютенова дієта чи дієти з обмеженням калорій. Завдяки алгоритмам, які оптимізують вживання продуктів з урахуванням макро- і мікроелементів, Eat This Much дозволяє

створювати збалансовані плани, що підходять для будь-якої мети — схуднення, підтримка ваги чи набір м'язової маси.

Одним з новітніх інструментів є PlateJoy, який використовує алгоритми машинного навчання для того, щоб персоналізувати раціон користувача ще більше. Цей сервіс враховує додаткові фактори, такі як біохімія організму, алергії, схильності до хвороб і навіть метаболічні особливості. PlateJoy пропонує індивідуальні рецепти та харчові плани, що мають на меті покращення здоров'я користувача та допомогу в досягненні конкретних цілей, таких як зниження ваги або покращення рівня енергії.

Ці інструменти використовують алгоритми машинного навчання, обробку даних, а також принципи здорового харчування для того, щоб створювати індивідуальні плани харчування, які можуть бути адаптовані до змін у фізичному стані або меті користувача. Крім того, всі ці інструменти взаємодіють з користувачами через зручні інтерфейси, що дозволяє зібрати дані про харчування та надавати рекомендації в автоматичному режимі.



 <b>MEN</b>	$\text{BMR} = (10 \times \text{weight [kg]}) + (6.25 \times \text{height [cm]}) - (5 \times \text{age [yrs]}) + 5$
 <b>WOMEN</b>	$\text{BMR} = (10 \times \text{weight [kg]}) + (6.25 \times \text{height [cm]}) - (5 \times \text{age [yrs]}) - 161$

Рис. 1.2. Базова формула на визначення добової норми калорій

### 1.2.3. Набори даних для порівняльного аналізу

Для порівняльного аналізу ефективності індивідуальних харчових планів, створених на основі автоматизованого аналізу потреб, важливо використовувати різноманітні набори даних. Це дає можливість оцінити точність, ефективність та релевантність рекомендованих харчових планів для різних груп людей з різними потребами, а також порівняти різні методи створення харчових планів.

Одним з основних наборів даних, який може бути використаний для цієї мети, є набір даних, що містить інформацію про демографічні характеристики, фізичну активність, медичні показники, а також індивідуальні вподобання щодо харчування. Такі набори даних дозволяють моделювати потреби в калоріях і макроелементах для різних груп населення, зокрема для людей, що ведуть сидячий спосіб життя, а також для тих, хто займається спортом [7].

Щоб зробити порівняльний аналіз між різними інструментами автоматизованого створення харчових планів, можна використовувати такі набори даних, як National Health and Nutrition Examination Survey (NHANES) та інші відкриті ресурси, що містять дані про харчування та здоров'я населення. Ці набори даних включають інформацію про споживання калорій, вітамінів, мінералів, а також про медичні історії учасників, що дозволяє створити точніші прогнози для індивідуальних харчових планів.

Ще одним варіантом є використання наборів даних, що містять рецепти та харчові інгредієнти, зокрема бази даних, які містять калорійність продуктів, їх макро- і мікроелементи, що дозволяє автоматично складати плани харчування з урахуванням усіх вимог до дієти. Це може включати, наприклад, набори даних, зібрані на основі популярних дієт, таких як середземноморська або вегетаріанська.

Для порівняльного аналізу можна також використовувати результати тестувань програмних інструментів, які реалізують автоматичне складання харчових планів, порівнюючи їх з реальними даними про ефективність різних дієт, отриманими з наукових досліджень або клінічних випробувань. Це дасть змогу виявити переваги та недоліки кожного інструменту та підходу до створення індивідуальних харчових планів.

### **1.3. Процес збору та підготовки даних**

Процес збору та підготовки даних є ключовим етапом для створення індивідуальних харчових планів на основі автоматизованого аналізу потреб.

Оскільки успіх розробки ефективних харчових рекомендацій безпосередньо залежить від якості даних, цей етап вимагає ретельного підходу до збору, очищення та підготовки інформації. У цьому підрозділі буде описано основні етапи цього процесу, а також інструменти та методи, що використовуються для забезпечення точності та релевантності отриманих даних.

Збір даних для створення індивідуальних харчових планів включає такі ключові джерела:

- демографічні дані: інформація про вік, стать, зріст, вагу, фізичну активність та інші параметри, які визначають потреби в калоріях і макроелементах. Ці дані можуть бути отримані через анкети або з наявних баз даних, таких як медичні записи або опитування населення;

- медичні дані: історії хвороб, наявність хронічних захворювань, алергії на певні продукти та інші медичні умови, які можуть впливати на харчові потреби. Важливо отримати доступ до надійних медичних записів або провести відповідні опитування для збору цієї інформації;

- дані про харчування: інформація про поточне харчування користувача, його уподобання, дієти, які він дотримується, а також його реакція на різні продукти та харчові стратегії. Це можуть бути анкети, що заповнюються користувачами, або оброблені дані з додатків для моніторингу харчування;

- інгредієнти та рецепти: дані про інгредієнти, їх калорійність, вміст макро- та мікроелементів, що необхідні для розрахунку харчового плану. Ці дані можна отримати з баз даних продуктів або з наукових джерел, що містять інформацію про харчову цінність продуктів.

Таблиця 1.1.

Огляд таблиці готового раціону

	ПН	Вт	СР	ЧТ	ПТ	СБ	НД
Сніданок 1	Мусовий десерт "Три шоколади"	Бананові панкейки з ягідним кюлі	Кокосовий чізкейк	Трайфл "Снікерс" зі згущеним молоком на фруктозі та арахісом	Морквяний рулет з вершково-сирним кремом та ягодами	Фруктовий салат	Яблука запечені з цукатами та родзинками

Продовження таблиці 1.1.

Сніданок 2	Маффіни з моцарелою та шинкою	Запечене яйце на зерновому хлібі з індичкою шинкою та зеленню	Англійський сніданок "Скрамбл з індичими ковбасками та квасолею та чіпсами з бекону	Яйця з начинкою з шинки та сиру з грінками	Шпинатні млинці з рисового борошна з шинкою та моцареллою	Сирні пиріжки з яйцем та зеленню	Бріош з індичкою та кукурудзою в сирному соусі
Обід	Томлена телятина в томатному соусі з гречаною кашою	Боул з куркою, пастою орзо, овочами та пармезаном	Хек "під овочами" в томаті з булгуром	Пшенична каша з куркою в соусі карі	Рис з куркою та перцем	Спагетті з м'ясними кульками під соусом бешамель	Червона риба та кускус у вершковому соусі під сиром
Перекус	Лаваш на грилі з куркою, міксом салатів, свіжими овочами та сиром сулугуні (начинку не змішувати)	Свіжий салат з молодою капустою, редисом, огірком, яйцем	Салат з сирними кульками та овочами	Фреш салат з куркою, овочами та сиром	Легкий салат з твердим сиром; кукурудзою; айсбергом, перцем та печерицями	Салат з запеченим філе, вяленими томатами та фетою	Жульєн в млинці з омлету
Вечеря	Мікс запеченої капусти з куркою під соусом бешамель	Курячі нагетси та пюре з сочевиці	Курячі котлетки з творогом під вершковим соусом з картоплею пюре	Телятина з тушкованою квасолею з овочами в томаті	Гуляш з курячого мяса в томатно-гірчичному соусі та пюре з нутом	Шашлик з овочами гриль	Курочка з сиром та пармезаном та молодою картоплею

У таблиці 1.1 показано індивідуальний харчовий план на основі розробленого раціону.

Після збору даних наступним етапом є їх очищення та підготовка до аналізу. Очищення даних передбачає:

- видалення пропущених значень: дані можуть містити пропущені або некоректні записи, які потребують корекції. Це можуть бути пропущені значення для певних параметрів, наприклад, для інформації про медичні показники чи фізичну активність;

- стандартизація одиниць виміру: різні джерела можуть використовувати різні одиниці виміру для параметрів, таких як вага, висота, кількість калорій в їжі тощо;

- нормалізація та масштабування: в деяких випадках значення можуть потребувати нормалізації для подальшого використання в математичних моделях або алгоритмах машинного навчання. Це дозволяє уникнути проблем,

пов'язаних з великою різницею між величинами різних параметрів (наприклад, вага і рівень фізичної активності) [8].

Нижче представлена функція, яка реалізує нормалізацію кількості калорій відповідно до рівня фізичної активності користувача. Ця функція дозволяє масштабувати дані таким чином, щоб вони могли використовуватися для визначення загальної добової витрати енергії (TDEE). Завдяки цьому забезпечується точність розрахунків, які враховують індивідуальні особливості користувача, такі як фізична активність та інші фактори способу життя.

Лістинг 1.1. Функція на JavaScript, яка нормалізує кількість калорій під фізичну активність користувача.

```
const calculateTDEE = (bmr: number, activity: string) => {
  switch (activity) {
    case "verylow":
      return bmr * 1.2;
    case "low":
      return bmr * 1.375;
    case "medium":
      return bmr * 1.55;
    case "high":
      return bmr * 1.725;
    case "veryhigh":
      return bmr * 1.9;
    default:
      return 0;
  }
};
```

Наступна функція демонструє, як нормалізувати необхідну кількість калорій відповідно до цілі користувача, будь то схуднення, підтримка ваги або набір маси. Вона враховує індивідуальні потреби та дозволяє адаптувати раціон харчування для досягнення бажаного результату.

Лістинг 1.2. Функція на JavaScript для нормалізації необхідної кількості калорій для цілі або мети користувача

```
const calculateCalories = (goal: string, tdee: number) => {
  switch (goal) {
```

```
case "loss":
    return tdee * 0.8;
case "keep":
    return tdee;
case "gain":
    return tdee * 1.2;
default:
    return 0;
}
};
```

Після очищення даних наступним етапом є їх підготовка для подальшого аналізу. Це може включати:

- інтеграція зовнішніх даних, таких як таблиці калорійності продуктів або харчова база даних, для створення повної та коректної картографії їжі;
- форматування даних у зручний для обробки вигляд, наприклад, у вигляді таблиць, які можуть бути легко використані в програмних інструментах для побудови індивідуальних харчових планів;

Оскільки дані, що збираються для створення індивідуальних харчових планів, можуть містити чутливу інформацію про здоров'я та особисті уподобання користувачів, важливо забезпечити їх анонімність та захист [9]. Згідно з нормами GDPR (Загальний регламент захисту даних) або іншими законодавчими актами, що стосуються захисту персональних даних, необхідно забезпечити:

- анонімізацію або псевдонімізацію даних, щоб уникнути ідентифікації користувачів;
- впровадження політик безпеки для зберігання даних, включаючи шифрування та обмеження доступу до чутливої інформації.

#### **1.4 Концепція автоматизації планування харчування як інструменту оптимізації**

Автоматизація планування харчування грає важливу роль у створенні ефективних інструментів для персоналізації раціонів, орієнтуючись на індивідуальні потреби користувачів. Сучасні технології дозволяють

автоматично формувати харчові плани, що базуються на кількох основних аспектах, таких як фізіологічні характеристики людини, рівень фізичної активності, стан здоров'я та персональні вподобання. Важливість автоматизації в цьому контексті полягає у забезпеченні точності, зручності та індивідуального підходу до кожного користувача, що дозволяє оптимізувати процес харчування.

Процес автоматизації планування харчування включає використання алгоритмів, які враховують безліч факторів, що впливають на індивідуальні потреби користувача. Одним з основних інструментів є формула Сан-Жеора, яка дозволяє визначити базові енергетичні потреби організму, враховуючи такі параметри, як вік, стать, вага та рівень фізичної активності. Однак для досягнення більш точних результатів, система повинна враховувати і інші аспекти, такі як специфічні потреби в мікроелементах та вітамінах, алергії чи харчові вподобання.

Для автоматизації планування харчування використовуються математичні моделі та алгоритми, які дозволяють точно розрахувати кількість калорій, макро- та мікроелементів, необхідних користувачу. Основною метою цих моделей є оптимізація складу раціону так, щоб забезпечити не тільки досягнення поставлених цілей (наприклад, схуднення або набір м'язової маси), а й підтримку загального здоров'я. Алгоритми, основані на принципах балансування нутрієнтів, дозволяють адаптувати харчовий план залежно від поточних результатів, забезпечуючи більшу гнучкість і ефективність.

Автоматизація планування харчування дозволяє швидко і зручно коригувати харчові плани залежно від змін у фізичному стані користувача або його цілей. Наприклад, при зміні рівня фізичної активності або при досягненні певних результатів у тренуваннях, система може внести корективи в кількість і склад продуктів, щоб забезпечити оптимальні умови для досягнення нових цілей.

Інтеграція автоматизованого планування харчування з іншими технологіями — це важливий елемент розвитку такого підходу. Це може

включати взаємодію з додатками для моніторингу фізичної активності, пристроями для вимірювання метаболічних показників або навіть інтерфейсами для обліку сну та стресу. Завдяки такій інтеграції автоматизоване планування харчування може стати частиною більш широкої екосистеми здоров'я користувача, забезпечуючи комплексний підхід до підтримки його фізичного стану [10].

З розвитком технологій, зокрема штучного інтелекту, автоматизація планування харчування набуває нових можливостей. Штучний інтелект дозволяє аналізувати великі обсяги даних і на основі цього створювати ще більш персоналізовані плани, враховуючи навіть генетичні особливості користувачів. Завдяки цьому можна буде не тільки досягати кращих результатів, а й мінімізувати ризики здоров'я, пов'язані з неправильним харчуванням.

### **Висновки до розділу**

У цьому розділі розглянуто концепцію автоматизації планування харчування як інструменту для оптимізації раціонів з урахуванням індивідуальних потреб користувачів. Основну увагу приділено використанню формули Сан-Жеора для розрахунку енергетичних потреб, а також алгоритмам, що дозволяють створювати персоналізовані плани харчування, що відповідають конкретним цілям користувачів.

Однією з основних переваг автоматизованого планування харчування є здатність адаптувати раціони до змінюваних умов життя та здоров'я користувача. Використання формули Сан-Жеора для розрахунку базових енергетичних потреб дозволяє з максимальною точністю визначити необхідну кількість калорій для підтримки нормальної життєдіяльності, враховуючи індивідуальні особливості. Однак, на відміну від традиційних методів, де людина повинна вручну коригувати свій раціон, автоматизовані системи здатні оперативно реагувати на зміни — наприклад, на коливання фізичної

активності або на зміну в харчових перевагах. Такі системи можуть самостійно запропонувати коригування раціону в залежності від різних факторів, таких як вік, стать, рівень фізичних навантажень чи наявність певних захворювань. Це дозволяє не тільки забезпечити ефективність харчового плану, але й значно полегшити процес підтримки здорового способу життя, оскільки користувач отримує рекомендації, які точно відповідають його поточним потребам. Крім того, використання таких інструментів може допомогти людям з конкретними медичними потребами, наприклад, з діабетом або серцево-судинними захворюваннями, у складанні меню, яке враховує їхню специфічну дієту, що знижує ризик помилок при самостійному плануванні харчування.

Зроблено акцент на важливості адаптації харчового плану до фізіологічних характеристик людини, рівня її фізичної активності та здоров'я, що дозволяє не тільки досягати поставлених цілей (наприклад, схуднення або набір м'язової маси), але й підтримувати загальний стан здоров'я. Одним із ключових аспектів є можливість коригування харчового плану в реальному часі, враховуючи зміни в стані користувача.

Автоматизація процесу планування харчування дає змогу значно полегшити процес досягнення здорових результатів, зберігаючи високу точність у розрахунках та забезпечуючи зручність для користувачів. Важливою складовою цієї автоматизації є інтеграція з іншими інструментами, що дозволяє створювати більш ефективні системи для управління харчовими планами.

Розвиток таких технологій дозволить в майбутньому створювати більш індивідуалізовані та ефективні рішення для планування харчування, що буде корисним не тільки для користувачів, а й для розробників додатків, які прагнуть забезпечити комплексний підхід до здоров'я.

## РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ МОДЕЛЕЙ, МЕТОДІВ ТА МЕТОДОЛОГІЇ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ ХАРЧУВАННЯ

### 2.1. Формула Сан-Жеора як основа розрахунків

Формула Міффіна-Сан Жеора є сучасним підходом до розрахунку базової метаболічної норми (BMR), яка є одним із ключових параметрів у побудові індивідуального плану харчування. Цей метод був розроблений у 1990-х роках і швидко здобув популярність завдяки своїй точності та адаптивності до різних груп людей. Головна перевага формули полягає в її універсальності: вона враховує такі важливі індивідуальні показники, як вага, ріст, вік і стать.

Формула Міффіна-Сан Жеора була розроблена на заміну класичній формулі Харріса-Бенедикта, яка тривалий час використовувалась у дієтології. Попри свою точність, формула Харріса-Бенедикта виявилась менш ефективною для сучасної популяції через зміни у способі життя, середній масі тіла та рівні фізичної активності. Формула Міффіна-Сан Жеора враховує ці зміни, що робить її більш точною для сучасних користувачів [11].

BMR є основою для розрахунків усіх калорійних потреб людини. Це показник енергії, необхідної для підтримання основних функцій організму в стані спокою, таких як дихання, серцебиття, функціонування мозку та інших органів. Без розрахунку BMR неможливо точно визначити загальну добову потребу в калоріях (TDEE), яка враховує фізичну активність і інші енергетичні витрати.

Обчислення BMR для чоловіків:

$$BMR = 10 \cdot \text{вага (кг)} + 6.25 \cdot \text{ріст (см)} - 5 \cdot \text{вік (років)} + 5 \quad (2.1)$$

Обчислення BMR для жінок:

$$BMR = 10 \cdot \text{вага (кг)} + 6.25 \cdot \text{ріст (см)} - 5 \cdot \text{вік (років)} - 161 \quad (2.2)$$

Ці формули враховують статеві відмінності, які впливають на швидкість обміну речовин, наприклад, через відмінності у відсотку м'язової маси.

Використання формули Міффіна-Сан Жеора дозволяє:

- індивідуалізувати підхід до харчування. Різні люди мають різні потреби залежно від їхніх фізичних характеристик, і ця формула дає змогу враховувати такі відмінності;

- планувати калорійність раціону. Формула є базою для створення програм харчування, спрямованих на зниження ваги, підтримання поточного стану або набір маси;

- оптимізувати фізичну форму. За допомогою точного обчислення можна запобігти дефіциту або надлишку калорій, що важливо для спортсменів і людей, які ведуть активний спосіб життя [12].

Для визначення загальної добової потреби в калоріях (Total Daily Energy Expenditure, TDEE) BMR множиться на коефіцієнт фізичної активності:

Дуже низька активність (сидячий спосіб життя):

$$TDEE = BMR \cdot 1.2 \quad (2.3)$$

Низька активність (легкі фізичні вправи 1-3 рази на тиждень):

$$TDEE = BMR \cdot 1.375 \quad (2.4)$$

Середня активність (помірні фізичні вправи 3-5 разів на тиждень):

$$TDEE = BMR \cdot 1.55 \quad (2.5)$$

Висока активність (інтенсивні вправи 6-7 разів на тиждень):

$$TDEE = BMR \cdot 1.725 \quad (2.6)$$

Дуже висока активність (дуже інтенсивні вправи або фізична робота):

$$TDEE = BMR \cdot 1.9 \quad (2.7)$$

На рисунку 2.1 зображено візуальне представлення графічного інтерфейсу, яке допомагає визначити точну добову потребу в калоріях індивідуально для кожної людини.

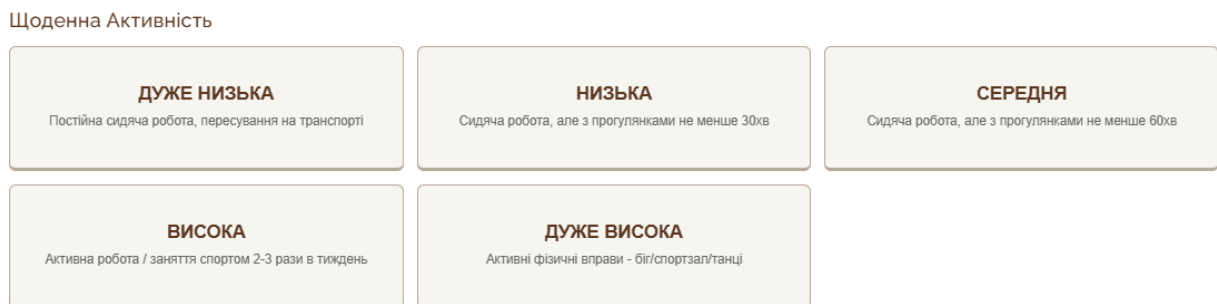


Рис. 2.1. Візуалізація вибору фізичної активності

Формула базується на емпіричних дослідженнях, які показали, що основні показники метаболізму тісно пов'язані з масою тіла, ростом, віком і статтю. Чоловіки, зазвичай, мають вищий рівень BMR через більший відсоток м'язової маси, тоді як у жінок ця цифра нижча через вищий вміст жирової тканини [13].

Окрім цього, вік відіграє значну роль у зниженні метаболізму: з кожним десятиліттям після 20 років базова метаболічна норма знижується в середньому на 2-3%. Цей фактор враховується у формулі через мультиплікатор віку.

## Переваги формули Міффіна-Сан Жеора:

1) точність. У порівнянні з попередніми формулами, ця методика дає більш точні результати для сучасної популяції, що підтверджено численними дослідженнями;

2) простота використання. Формула легко застосовується і не потребує складних обчислень чи спеціального обладнання;

3) гнучкість. Формула адаптується до різних цілей: підтримання ваги, набір або зниження маси тіла.

Попри свою універсальність, формула має певні обмеження:

- не враховує м'язову та жирову масу окремо. Формула враховує лише загальну масу тіла, що може впливати на точність розрахунків для людей із надмірною вагою або високим рівнем м'язової маси;

- не враховує метаболічні відмінності. Існують індивідуальні фактори, такі як гормональний баланс, які можуть впливати на BMR, але не враховуються у формулі;

- залежність від правильної оцінки фізичної активності. Визначення TDEE вимагає точного вибору коефіцієнта фізичної активності, що іноді є суб'єктивним.

У рамках проекту було реалізовано програмну інтеграцію формули Міффіна-Сан Жеора для автоматичного обчислення BMR та TDEE. Це забезпечило зручність і точність розрахунків для користувачів, що ввели свої фізичні параметри.

За допомогою Next.js як фреймворка була створена інтерактивна платформа, яка дозволяє:

- 1) автоматично збирати дані користувача (вага, ріст, вік, стать);
- 2) швидко обчислювати BMR і TDEE у режимі реального часу;
- 3) пропонувати індивідуальні рекомендації щодо калорійності раціону залежно від обраної мети.

Формула Міффіна-Сан Жеора стала ключовим компонентом у розробці системи автоматизації харчового планування. Вона не лише забезпечила

основу для точних розрахунків, але й дала змогу користувачам отримувати швидкі та персоналізовані результати, що підвищує ефективність створення індивідуальних планів харчування.

Такий підхід значно спрощує процес планування для користувачів, дозволяючи їм зосередитися на досягненні своїх цілей, будь то зниження ваги, підтримання фізичної форми чи набір маси.

## **2.2. Алгоритми оптимізації харчових раціонів**

Алгоритми оптимізації харчових раціонів відіграють важливу роль в автоматизації процесу розрахунку індивідуальних планів харчування. Метою таких алгоритмів є забезпечення балансу між енергетичними витратами людини та її потребами у харчуванні для досягнення конкретних цілей — зниження ваги, підтримка поточної ваги чи набір маси. Окрім того, алгоритми повинні враховувати не лише кількість калорій, а й мікро- та макроелементи, такі як вітаміни, мінерали, білки, жири та вуглеводи.

Оптимізація харчового раціону — це багатокроковий процес, який включає збір інформації, обчислення потреб та побудову збалансованих раціонів, що задовольняють вимоги користувача. В основі цієї задачі лежить математичне моделювання, яке включає різноманітні алгоритми, здатні знаходити оптимальне рішення для заданих умов.

Найпоширенішими підходами до оптимізації харчових раціонів є:

- лінійне програмування;
- генетичні алгоритми;
- алгоритми на основі глибокого навчання;
- алгоритми на основі теорії багатокритеріальної оптимізації.

Лінійне програмування (ЛП) є одним із найстаріших і найбільш використовуваних методів для оптимізації раціонів, оскільки дозволяє формулювати задачу у вигляді лінійних рівнянь і нерівностей. У рамках цієї задачі раціон розглядається як набір продуктів, а кожен продукт має певні

значення калорійності, білків, жирів, вуглеводів, вітамінів і мінералів. Завдання оптимізації полягає в тому, щоб зафіксувати сумарні потреби людини в цих компонентах і при цьому мінімізувати або максимізувати певну функцію, наприклад, мінімізувати вартість харчування або калорійність, або досягти заданого балансу між макроелементами [14].

У формулюванні лінійного програмування використовуються наступні параметри:

- цільова функція, яка визначає, що необхідно оптимізувати, наприклад, мінімізація вартості харчування;
- обмеження, що відповідають на питання про кількість і тип макро- та мікроелементів у раціоні.

Наприклад, задача мінімізації вартості харчування може виглядати так:

$$C = \sum_{i=1}^n x_i \cdot p_i \quad (2.8)$$

де:

$C$  – вартість раціону;

$x_i$  – кількість продукту  $i$  в раціоні;

$p_i$  – ціна продукту  $i$ .

Обмеження на кількість калорій, білків, жирів та вуглеводів можуть бути записані як:

$$\sum_{i=1}^n x_i \cdot c_{i,j} \geq a_j \quad (2.9)$$

де:

$c_{i,j}$  – кількість елементу  $j$  (наприклад білків) в продукті  $i$ ;

$a_j$  – мінімальна кількість елементу  $j$ , яку потрібно забезпечити в раціоні.

Генетичні алгоритми (ГА) є популярним методом для вирішення задач оптимізації, зокрема таких складних, як оптимізація харчових раціонів. Вони базуються на принципах природного відбору та еволюції і використовують механізми, які включають відбір, схрещування та мутації для знаходження оптимальних рішень.

Процес включає:

- створення популяції раціонів, де кожен раціон є потенційним рішенням;
- оцінка фітнесу кожного раціону, що вимірюється через функцію корисності, яка враховує відповідність раціону індивідуальним вимогам;
- селекція найкращих раціонів для створення нової популяції;
- схрещування і мутація, щоб забезпечити різноманітність і можливість виявлення нових рішень.

Генетичні алгоритми дуже ефективні в умовах великих і складних просторах рішень, де традиційні методи, такі як лінійне програмування, можуть не дати оптимальних результатів через великі обмеження.

З розвитком штучного інтелекту та машинного навчання виникли нові підходи до оптимізації харчових раціонів. Алгоритми на основі глибокого навчання (deep learning) здатні обробляти великі обсяги даних і на основі патернів виявляти найбільш ефективні стратегії харчування для різних груп користувачів.

Одним із таких підходів є використання нейронних мереж для моделювання відносин між продуктами харчування та їхніми властивостями. Наприклад, нейронна мережа може бути навчена на великому наборі даних про різні харчові продукти та їхні поживні властивості, щоб потім надавати рекомендації на основі введених користувачем параметрів (вік, стать, рівень активності, цілі) [15].

Цей підхід дозволяє враховувати складні залежності між різними поживними речовинами, а також індивідуальні уподобання користувачів, що робить систему адаптивною та персоналізованою.

Багатокритеріальна оптимізація є ще одним ефективним підходом, який дозволяє враховувати кілька критеріїв одночасно. У випадку оптимізації харчових раціонів це може бути комбінація таких факторів, як калорійність, вартість, баланс макроелементів, вітамінний склад і багато інших.

Задача багатокритеріальної оптимізації полягає в знаходженні рішення, яке є оптимальним за всіма цими критеріями одночасно. Вона використовується, коли неможливо визначити єдиний критерій, який б відповідав всім вимогам користувача. Наприклад, на користь продукту може йти не лише його калорійність, а й вітамінно-мінеральний склад або можливість вегетаріанського або безглютенового раціону.

У рамках проекту з автоматизації харчового планування застосовуються різні алгоритми оптимізації для досягнення точних і збалансованих результатів. Наприклад, для початкових етапів планування використовуються алгоритми лінійного програмування, оскільки вони дозволяють швидко створювати загальний раціон, що відповідає вимогам щодо макроелементів та калорій. Після цього, для більш детальної оптимізації, включаються методи на основі генетичних алгоритмів і глибокого навчання для персоналізованих рекомендацій. Це дозволяє створювати не лише збалансовані, але й адаптовані до індивідуальних уподобань плани харчування [16].

Для візуалізації результатів оптимізації раціону можна використовувати графіки і діаграми, які наочно показують співвідношення між різними поживними речовинами (наприклад, білками, жирами та вуглеводами). Ось кілька прикладів таких візуалізацій:

- графік макроелементів у раціоні. Це може бути кругова діаграма, що показує пропорції білків, жирів і вуглеводів у кожному раціоні, рисунок 2.2;
- графік калорійності. Стовпчикові діаграми, що порівнюють калорійність різних продуктів чи раціонів.
- графік споживання мікроелементів. Лінійна або стовпчикова діаграма, яка ілюструє вміст вітамінів і мінералів у раціоні, наприклад, рівень вітаміну С, кальцію, заліза та інших важливих мікронутрієнтів.

## Macronutrient Distribution in Sample Diet

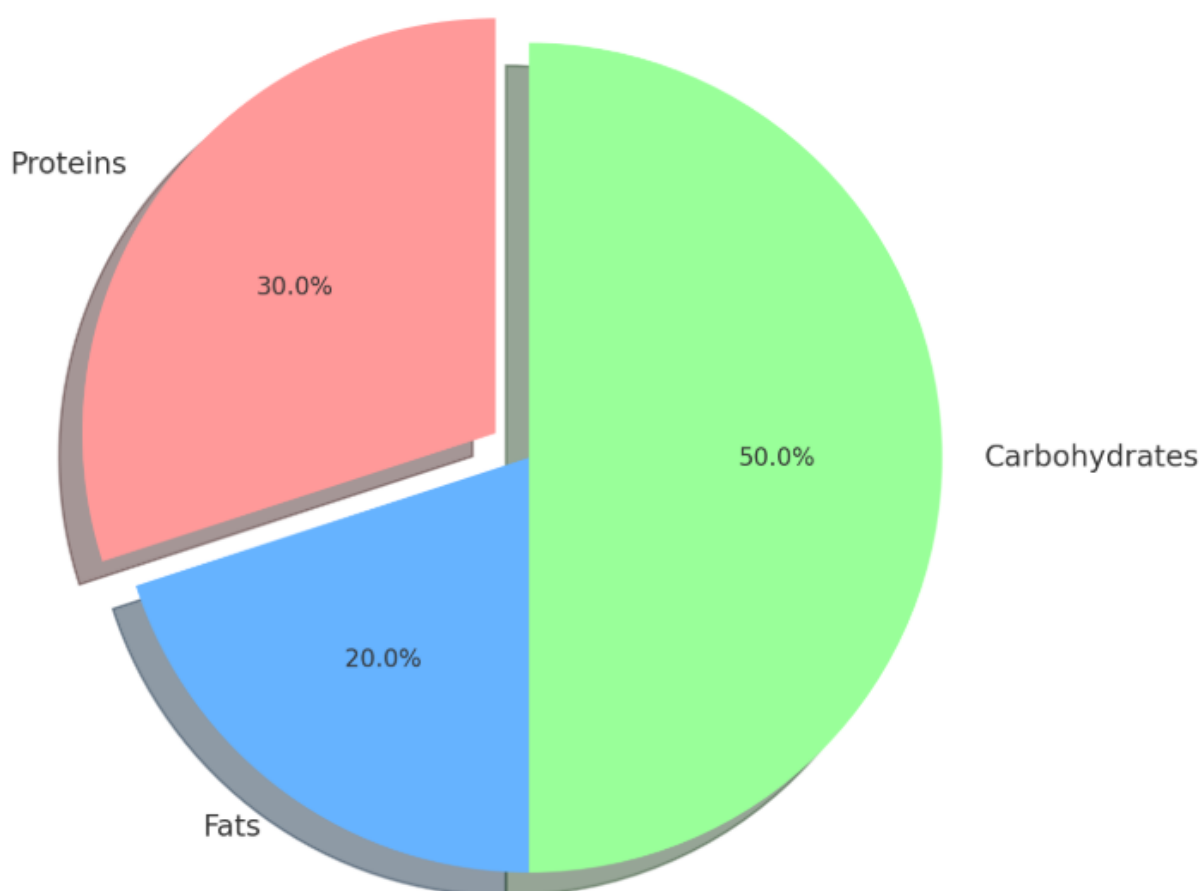


Рис. 2.2. Кругова діаграма, що показує пропорції білків, жирів та вуглеводів

### 2.3. Огляд методології дослідження

Методологія дослідження, що стосується автоматизації планування харчування на основі індивідуальних потреб, є багатогранною і включає кілька основних етапів: визначення проблеми, формулювання гіпотез, збір і обробка даних, розробка моделей та алгоритмів, а також оцінка результатів. Кожен з цих етапів є критично важливим для досягнення точності та ефективності в автоматизації планування харчування [17].

На першому етапі важливо чітко визначити проблему, яка стоїть перед дослідженням. У випадку автоматизації харчового планування, це може бути створення алгоритму, який здатний генерувати індивідуалізовані раціони, що відповідають фізіологічним потребам користувача, враховують його здоров'я,

фізичну активність та інші обмеження (наприклад, алергії або дієтичні вимоги). Завданням є побудова такої системи, яка б автоматично адаптувала харчовий план в залежності від змін в цих параметрах.

Огляд наявних рішень і наукових робіт з цієї теми допомагає виявити основні підходи до оптимізації харчових раціонів, вибір найбільш підходящих методів та інструментів для розробки власної моделі. В цьому етапі також відбувається вивчення таких питань, як визначення основних параметрів індивідуалізації харчового плану (калорії, білки, жири, вуглеводи, вітаміни, мінерали тощо), а також обмеження на основі існуючих дієтичних рекомендацій [18].

НАССР (Hazard Analysis and Critical Control Points) – це розроблена програма забезпечення виробництва безпечних харчових продуктів, яка фокусується на запобіганні небезпек шляхом застосування контролю від сировини до готової продукції, схема зображена на рисунку 2.3. [19]

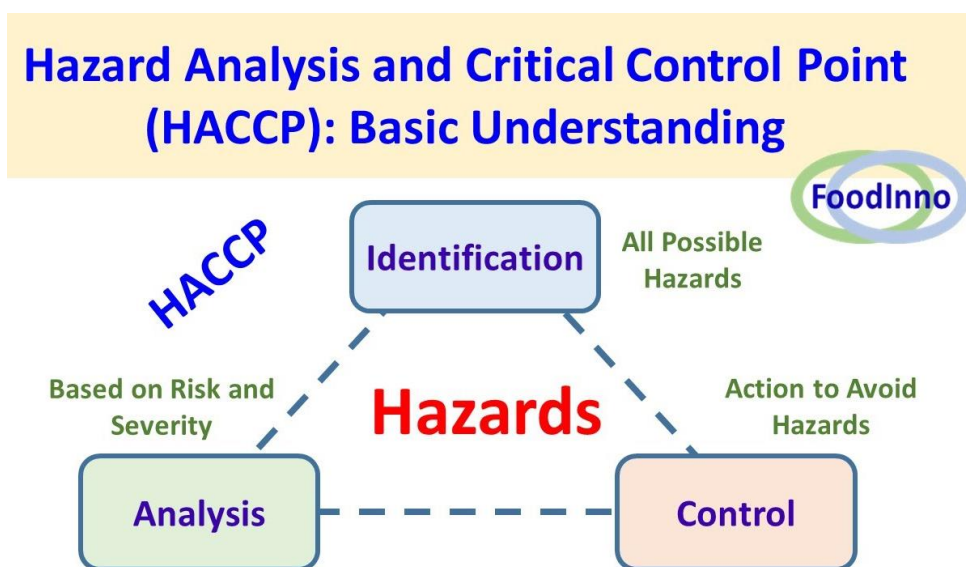


Рис. 2.3. План НАССР

Один із найважливіших етапів дослідження полягає у зборі і підготовці даних, необхідних для розробки алгоритмів оптимізації харчового плану. Це може включати збір інформації про користувачів (вік, стать, вага, зріст, рівень фізичної активності), дані про продукти харчування (калорійність, вміст

макро- та мікроелементів), а також інформацію про існуючі обмеження чи алергії. Дані повинні бути очищені, стандартизовані та приведені до єдиного формату, що дозволяє їх зручне використання для моделювання.

Основним етапом є розробка алгоритмів, що дозволяють автоматично генерувати індивідуалізовані харчові плани. Це може включати використання методів лінійного та нелінійного програмування, генетичних алгоритмів, методів машинного навчання та штучного інтелекту. Важливою складовою є формулювання функцій оптимізації, які забезпечують ефективне планування харчування, враховуючи всі необхідні обмеження та параметри.

Після розробки моделей необхідно провести їх валідацію та тестування. Це передбачає перевірку точності розрахунків, а також оцінку ефективності запропонованих алгоритмів у реальних умовах. Оцінка ефективності алгоритмів здійснюється на основі таких критеріїв, як точність, швидкість обчислень, здатність адаптуватися до змінних параметрів користувача, а також ступінь задоволення користувача результатами (якщо це можливо).

Одним з основних принципів методології є ітераційний підхід до розвитку моделей і алгоритмів. Оскільки автоматизація планування харчування є складним завданням, що враховує безліч змінних, алгоритми можуть потребувати постійного удосконалення та коригування на основі зворотного зв'язку від користувачів або нових даних. Цей процес дозволяє досягати більш високої точності та ефективності в розробці індивідуалізованих харчових планів.

Останнім етапом є оцінка результатів автоматизованого планування харчування, а також формулювання рекомендацій щодо подальшого удосконалення системи. Це включає аналіз ефективності розроблених моделей та їх адаптивності до реальних умов використання. Тестування системи в умовах реальних користувачів дозволяє оцінити її здатність до адаптації та підтримки здорового харчування у різних контекстах.

Таким чином, методологія дослідження оптимізації харчових раціонів включає чітку стратегію збору, обробки та аналізу даних, розробку

математичних моделей і алгоритмів, а також постійне вдосконалення системи на основі результатів тестування та зворотного зв'язку від користувачів.

## **2.4. Підготовка даних для автоматизації харчового планування**

Підготовка даних є важливим етапом автоматизації харчового планування, оскільки точність результату залежить від якості даних. Це включає збір і очищення інформації про користувачів, їхні потреби та уподобання, а також про харчові продукти.

### *2.4.1. Збір даних про користувача*

Було зібрано основні дані про користувачів, що включають демографічну інформацію (вік, стать, вага, зріст), рівень фізичної активності, медичні показники (хронічні захворювання, алергії), дієтичні переваги та цілі харчування (схуднення, набір маси тощо). Ця інформація дозволяє розрахувати індивідуальні енергетичні потреби та побудувати персоналізований харчовий план.

### *2.4.2. Очищення та стандартизація інформації*

Було очищено та стандартизовано зібрані дані, щоб уникнути помилок та забезпечити їхню сумісність з алгоритмами планування. Відсутні значення були заповнені або виключені, дублікати записів були видалені, а одиниці виміру стандартизовані. Дані були нормалізовані для подальшого використання в математичних моделях, що забезпечило точність розрахунків і ефективність алгоритмів.

## **2.5. Побудова моделей автоматизації**

Побудова моделей автоматизації харчового планування включає визначення параметрів, що визначають склад плану, а також налаштування

алгоритмів для його ефективного формування. Ці етапи забезпечують точність, персоналізацію та відповідність індивідуальним потребам користувача.

### *2.5.1 Параметри моделі індивідуалізації харчових планів*

Ключовими параметрами моделі є:

Енергетичні потреби. Для визначення кількості калорій, необхідних для підтримки поточної ваги або досягнення певної мети (схуднення, набір маси), розраховується базальна метаболічна швидкість (BMR) та враховується рівень фізичної активності користувача.

Розподіл макронутрієнтів. Враховуються індивідуальні потреби у білках, жирах і вуглеводах залежно від фізичних характеристик та цілей користувача. Наприклад, для активних людей збільшується кількість білків, для тих, хто хоче схуднути, — зменшується кількість вуглеводів.

Мікронутрієнти. Потреби в мікронутрієнтах (вітаміни, мінерали) також враховуються для досягнення оптимального функціонування організму. Це може включати особливі рекомендації залежно від медичних показників користувача.

Дієтичні обмеження та уподобання. Враховуються алергії, особливості харчування (вегетаріанство, безглютенова дієта) та інші обмеження, що можуть впливати на вибір продуктів.

Часовий фактор. У моделі визначається період, на який складається план (наприклад, добовий чи тижневий), а також кількість прийомів їжі і їхній склад протягом цього періоду.

Гідратація. У моделі враховується необхідна кількість рідини, яку потрібно споживати протягом дня, залежно від ваги, рівня фізичної активності та інших індивідуальних характеристик.

Форма подачі інформації. Важливо визначити, як буде поданий харчовий план — у вигляді списків продуктів, таблиць або рецептів, щоб забезпечити зручність для користувача, рисунок 2.4.

Наші програми, які рекомендовані для досягнення вашої мети:

The image shows two nutrition plan cards. The first card is for a 2000 kcal diet with 4 meals per day. It features a dropdown menu set to '3 м'ясом', a price of 365 грн per day, and a total price of 365 грн. The second card is for a 2400 kcal diet with 5 meals per day. It features a dropdown menu set to 'Шеф меню', a price of 605 грн per day, and a total price of 605 грн. Both cards have a green 'ДОДАТИ В ЗАМОВЛЕННЯ' button.

Рис. 2.4. Візуалізація подачі інформації про раціони які підходять під індивідуальні потреби користувача

### 2.5.2. Налаштування алгоритмів автоматизації

Алгоритми автоматизації харчового планування включають кілька основних етапів налаштування:

- оптимізація енергетичного балансу. Алгоритм має визначати правильне співвідношення калорій та макроелементів, щоб досягти поставленої мети користувача. Це включає точний розрахунок для кожного з макронутрієнтів на основі індивідуальних потреб;

- генерація продуктів для харчового плану. Алгоритм вибирає продукти, що відповідають вимогам щодо калорійності та харчової цінності, враховуючи дієтичні обмеження користувача;

- моделювання варіативності раціону. Алгоритм дозволяє створити варіативні плани харчування, щоб забезпечити різноманітність і збалансованість раціону на тривалий період;

- урахування обмежень часу і бюджету. Механізми алгоритму забезпечують коригування плану залежно від часу, який є у користувача для приготування їжі, а також його бюджету на продукти;

- індивідуальна адаптація. Алгоритм здатний автоматично коригувати план на основі змін у фізичній активності або стані здоров'я користувача;
- перевірка відповідності харчового плану. Після формування плану алгоритм здійснює перевірку, щоб переконатися, що всі параметри виконані коректно, і харчовий план відповідає заданим вимогам.

## 2.6. Валідація та тестування інструментів

Для забезпечення правильності роботи інструментів автоматизації харчового планування важливим є проведення валідації та тестування, які допомагають перевірити точність розрахунків і ефективність їхнього використання в реальних умовах.

### 2.6.1. Методи перевірки точності розрахунків

Для перевірки точності розрахунків енергетичних потреб користувача застосовуються стандартні формули, зокрема формула Сан-Жеора. Її можна використовувати для визначення базальної метаболічної швидкості (BMR) та для розрахунку загальної потреби в калоріях, враховуючи рівень фізичної активності користувача. Порівняння результатів автоматизованих розрахунків з результатами, отриманими за допомогою цієї формули, дає змогу оцінити точність інструменту. Формула Сан-Жеора була описана в пункті 2.1. [21]

Калорійність продукту визначається шляхом множення маси порції на кількість калорій на одиницю ваги продукту. Наприклад, для вареного рису, якщо в 100 г міститься 130 калорій, то для порції 200 г кількість калорій складе:

$$\text{Калорії} = 200 \times \frac{130}{100} = 260 \text{ калорій} \quad (2.3)$$

Для тестування точності застосовуються контрольні дані з відомими значеннями енергетичних потреб і харчових характеристик. Це дозволяє

перевірити, чи відповідають результати автоматизованих розрахунків реально очікуваним значенням.

Виявлення похибок є важливою частиною процесу перевірки точності. Для цього результати автоматизованих розрахунків порівнюються з реальними даними, і виявляються відхилення, що дозволяє коригувати параметри алгоритмів.

### *2.6.2. Оцінка ефективності використання*

Оцінка ефективності полягає в порівнянні реальних результатів (наприклад, зниження або набір ваги) з метою, яку ставив користувач. Якщо харчовий план, створений за допомогою автоматизованого інструменту, відповідає зазначеній меті, інструмент можна вважати ефективним.

Один із важливих критеріїв ефективності — це зручність користування інструментом. Оцінка зручності включає аналіз інтерфейсу, часу, необхідного для створення харчового плану, а також досвіду користувачів щодо взаємодії з інструментом [22].

Ступінь адаптації інструменту до конкретних потреб користувача є ще одним важливим критерієм. Алгоритм має здатність коригувати харчові плани, враховуючи індивідуальні дані користувача, що забезпечує точність та коректність результатів.

### *2.6.3. Валідація на основі контрольних даних*

Валідація інструментів автоматизації харчового планування здійснюється шляхом використання контрольних даних, що дозволяє оцінити коректність розрахунків і відповідність створених планів до фактичних потреб користувача:

Для валідації моделі створюються контрольні сценарії, які враховують відомі параметри користувачів (зокрема, вага, вік, рівень фізичної активності), а також визначають чіткі цілі та критерії успішності для оцінки ефективності розроблених планів.

Після генерації плану порівнюються результати, отримані від автоматизованого інструменту, з відомими еталонними значеннями або планами, складеними вручну. Це дозволяє оцінити точність та коректність розрахунків.

Після порівняння результатів автоматизованого плану з еталонними даними проводиться аналіз точності. Виявлення будь-яких відхилень допомагає коригувати інструмент, щоб підвищити його точність.

### **Висновки до розділу**

У другому розділі розглянуто основні моделі та методи автоматизації планування харчування, а також методологію побудови інструментів для індивідуалізації харчових планів. Детально описано алгоритми оптимізації харчових раціонів, використання формули Сан-Жеора для розрахунку енергетичних потреб, а також підготовку даних, включаючи збір, очищення та стандартизацію інформації.

Особлива увага приділена побудові моделей автоматизації, що включає налаштування параметрів для індивідуалізації планів харчування та налаштування алгоритмів автоматизації для точного підбору продуктів і страв відповідно до потреб користувача. У процесі розробки таких інструментів було враховано важливість коректного визначення параметрів, що впливають на точність планування.

У валідації та тестуванні інструментів акцент зроблено на методах перевірки точності розрахунків, оцінці ефективності використання та тестуванні результатів на контрольних даних.

Загалом, у розділі підкреслено важливість комплексного підходу до автоматизації харчового планування, де кожен етап — від підготовки даних до валідації результатів — сприяє точності і ефективності створених харчових планів, що відповідають індивідуальним потребам користувачів.

## РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ ХАРЧУВАННЯ

### 3.1. Загальна стратегія реалізації рішення

Даний розділ магістерської роботи присвячений детальному опису підходів до реалізації програмного забезпечення, яке забезпечує автоматизацію процесів створення індивідуальних раціонів харчування. Розроблена система базується на інтеграції сучасних веб-технологій, включаючи WordPress, Next.js та платформу Vercel, і покликана об'єднати зручність використання з високою продуктивністю.

Мета розробки системи — створити веб-додаток, який дозволяє автоматично формувати раціон харчування на основі даних користувача, таких як стать, вік, вага, фізична активність, індивідуальні цілі та вподобання. Для цього необхідно забезпечити злагоджену роботу кількох основних компонентів: бази даних, інтерфейсу користувача, бекенд-сервісів для обробки даних і алгоритмів генерації раціонів.

Загальна стратегія реалізації базується на принципах модульності, масштабованості, інтеграції та продуктивності. Детальніший розгляд кожного з цих аспектів наведено нижче.

Архітектура системи:

Архітектура системи розроблена за принципом поділу на фронтенд і бекенд для забезпечення максимальної гнучкості та простоти обслуговування.

Фронтенд реалізовано на основі Next.js — сучасного фреймворку для розробки React-додатків, який підтримує серверну генерацію сторінок (Server-Side Rendering, SSR) і статичне відтворення (Static Site Generation, SSG). Next.js обрано через його інтеграцію з Vercel, що дозволяє досягти високої швидкості завантаження сторінок і зручності розгортання.

Бекенд побудований на основі WordPress із використанням плагіна WPGraphQL для доступу до даних. WordPress обрано через його розвинені

можливості управління контентом та вбудовані інструменти для роботи з базою даних.

Платформа хостингу — Vercel, яка забезпечує автоматизоване розгортання, безперервну інтеграцію та високу продуктивність веб-додатків.

База даних WordPress містить усі необхідні дані, зокрема:

- інформацію про продукти харчування (калорійність, співвідношення макроелементів, категорії);
- додаткові метадані, такі як рекомендації щодо поєднання продуктів;
- дані про користувачів та їхні індивідуальні параметри. [23]

Центральним компонентом є WPGraphQL API, який забезпечує доступ до бази даних і зв'язок між фронтендом та бекендом. Нижче на рисунку 3.1 наведено графічний інтерфейс доступу до бази даних.



	Боул з куркою, кіноа та овочами	-	In stock	110,00 ₴	Основні страви
	Панакота	-	In stock	60,00 ₴	Десерти
	Салат Нісуаз	-	In stock	100,00 ₴	Основні страви, Салати
	Салат Цезар	-	In stock	100,00 ₴	Основні страви, Салати
	Лаваш з куркою та овочами	-	In stock	70,00 ₴	Вуличний стиль
	Сирники з ягідним джемом, сметаною	-	In stock	80,00 ₴	Десерти
	Млинці з вишнею, крем-сиром	-	In stock	80,00 ₴	Десерти

Рис. 3.1. Візуалізація таблиці товарів у WordPress

Основні етапи реалізації:

Етап 1. Проектування бази даних WordPress

На першому етапі було спроектовано структуру бази даних для збереження всіх необхідних даних. Основними сутностями стали:

- продукти харчування, які містять дані про назву, калорійність, співвідношення білків, жирів і вуглеводів;

- категорії продуктів, що дозволяють структурувати дані за типом (фрукти, овочі, м'ясо, напої тощо) ;

#### Етап 2. Розробка GraphQL-запитів для імпорту даних

WPGraphQL дозволив створити оптимізовані запити для імпорту даних з бази WordPress. Для забезпечення продуктивності було реалізовано:

- фільтри для вибірки лише необхідних даних (наприклад, продукти з певним типом макроелементів);

- пагінацію для роботи з великими обсягами даних без втрати швидкості;

- механізми обробки помилок у випадку некоректних даних. [24]

#### Етап 3. Розробка алгоритмів генерації раціонів

Алгоритми, що базуються на формулі Міффліна-Сан Жеора, враховують базовий обмін речовин (BMR) та добову потребу в калоріях. Для цього розроблено окремий модуль, який:

- приймає дані про користувача (вік, стать, активність тощо);

- визначає необхідну кількість калорій для досягнення цілей (зниження ваги, підтримання, набір);

- формує збалансований раціон із продуктів бази даних, орієнтуючись на оптимальне співвідношення макроелементів (40/30/30).

#### Етап 4. Інтеграція фронтенда з бекендом

За допомогою Next.js та WPGraphQL була реалізована інтеграція, що забезпечує динамічне оновлення даних на сторінках без необхідності перезавантаження. Інтерфейс користувача дозволяє:

- вводити дані користувача у форму;

- миттєво отримувати сформований раціон;

- змінювати параметри та оновлювати результати в реальному часі.

#### Етап 5. Хостинг і оптимізація

Останнім етапом стало розгортання веб-додатку на Vercel. У процесі розгортання було виконано:

- оптимізацію продуктивності сторінок за допомогою кешування і мінімізації запитів;
- налаштування адаптивного дизайну для забезпечення зручності роботи на різних пристроях (мобільних, планшетах, ПК) ;
- тестування продуктивності при високому навантаженні.

Принципи реалізації:

При розробці системи дотримувались таких принципів:

- 1) модульність. Кожен компонент (імпорт даних, обробка, генерація раціону, інтерфейс користувача) реалізований як окрема одиниця, що спрощує підтримку і вдосконалення;
- 2) масштабованість. Архітектура дозволяє легко додавати нові функції, наприклад, інтеграцію з іншими API або розширення бази продуктів; [25]
- 3) простота використання. Інтерфейс орієнтований на кінцевого користувача, а всі технічні деталі приховані "під капотом";
- 4) висока продуктивність. Завдяки використанню Next.js і Vercel досягнуто швидке завантаження сторінок і стабільна робота системи;

Загальна стратегія реалізації була розроблена з урахуванням вимог до сучасних веб-додатків, що передбачає інтеграцію зручного інтерфейсу, стабільної бази даних і продуктивного бекенду. Такий підхід забезпечив створення платформи, яка ефективно вирішує завдання автоматизації формування харчових планів.

### **3.2. Реалізація програми імпорту даних**

Цей розділ містить детальний опис реалізації програми імпорту даних, яка є невід'ємною частиною системи автоматизації планування раціонів харчування. Метою цього компонента є забезпечення стабільного і

автоматизованого способу збору та збереження даних із різних джерел, таких як таблиці продуктів, зовнішні API та файли у форматах CSV чи Excel.

У підрозділі 3.2.1 розглядаються контекст, область застосування та загальна стратегія створення програми імпорту. Далі буде надано структурований опис архітектури, основних функцій і технічної реалізації інструменту.

### *3.2.1. Контекст, область застосування та підхід до імпорту*

Програма імпорту є ключовим компонентом системи автоматизації планування харчування, оскільки забезпечує інтеграцію зовнішніх джерел даних у централізовану базу для подальшого використання в алгоритмах оптимізації та розрахунках. Основна мета цього етапу полягає у створенні ефективного механізму, який дозволяє системі отримувати актуальну інформацію про продукти, їхню харчову цінність, категорії, рекомендації та інші параметри, необхідні для формування персоналізованих раціонів.

У сучасному світі дані про продукти харчування доступні з різних джерел, таких як:

- локальні файли (CSV, Excel);
- сторонні API;
- бази даних клієнтів, які вже мають список продуктів і характеристик.

Для інтеграції цих джерел потрібно враховувати їхній формат, структуру, а також потенційну необхідність попередньої обробки даних. [26] Зокрема, деякі API можуть мати обмеження щодо частоти запитів, а файли від клієнтів можуть містити помилки, пропуски або несумісні формати.

Окрім цього, важливо врахувати такі аспекти:

Уніфікація даних. Оскільки різні джерела надають інформацію у власних форматах, необхідно забезпечити її стандартизацію та приведення до єдиного вигляду перед збереженням у базі даних, щоб уникнути помилок і забезпечити коректну обробку.

Актуалізація. Інформація про продукти змінюється з часом (оновлення складу, зміна енергетичної цінності), тому система імпорту повинна мати можливість регулярного оновлення даних.

Масштабованість. Обсяг даних може значно збільшуватися, тому обрана технологія та архітектура повинні підтримувати можливість роботи з великими обсягами інформації.

Помилки та валідація. Дані можуть містити некоректну інформацію (наприклад, неправильні значення калорій або грамів). Необхідно враховувати алгоритми перевірки, щоб уникнути негативного впливу на результати планування харчування.

Програма імпорту використовується у декількох ключових напрямках:

Інтеграція з базами даних продуктів харчування. Наприклад, імпорт інформації про калорійність, вміст білків, жирів і вуглеводів для розрахунку індивідуальних потреб клієнтів.

Розширення каталогу продуктів. Це дозволяє додавати нові інгредієнти або рецепти на основі зовнішніх джерел.

Аналіз ринкових тенденцій. Автоматизований збір даних про популярність певних продуктів або зміну їх характеристик.

Підготовка до локалізації. Можливість адаптації інформації під різні регіони, враховуючи локальні особливості харчування.

Програма імпорту розроблена з урахуванням модульної архітектури, що дозволяє легко адаптувати її до нових джерел даних. Основними компонентами є:

- 1) модуль збору даних:
  - інтерфейси для роботи з API сторонніх джерел (наприклад, REST API);
  - модулі для читання файлів різних форматів (CSV, Excel, JSON);
  - функції для взаємодії з базами даних клієнтів через SQL-запити або ORM.
- 2) модуль обробки даних:

- алгоритми трансформації даних у єдиний формат;
- видалення дублікатів, заповнення пропусків і валідація;
- обробка одиниць вимірювання (наприклад, перерахунок калорійності на 100 г).

3) модуль збереження:

- інтеграція з WordPress через WPGraphQL для збереження інформації в базі даних;
- взаємодія з Next.js для попередньої підготовки відображення даних у веб-інтерфейсі.

Для реалізації програми імпорту використовуються такі інструменти:

- Node.js для серверної частини, що відповідає за обробку запитів до API та читання файлів;
- GraphQL для передачі даних до WordPress;
- Next.js для забезпечення фронтенд-відображення результатів імпорту;
- Vercel для розгортання веб-інтерфейсу та забезпечення стабільного доступу.

Процес імпорту включає перевірку таких показників:

Формат даних (наприклад, числові значення калорій, текстові назви продуктів).

Логічна цілісність (неможливість збереження продукту з негативною калорійністю).

Відсутність дублікатів у каталозі.

Реалізація програми імпорту:

- 1) забезпечує стабільну інтеграцію даних із різних джерел;
- 2) автоматизує обробку та уніфікацію інформації для подальшого використання в алгоритмах;
- 3) спрощує додавання нових даних та зменшити людський фактор при внесенні інформації.

## Основні аспекти програми імпорту даних

Категорія	Опис	Інструменти/Методи
Джерела даних	Визначення та інтеграція даних із зовнішніх джерел, таких як локальні файли, API або бази даних клієнтів.	CSV, Excel, JSON, REST API, SQL-запити
Типи джерел	- Локальні файли: каталоги продуктів, отримані клієнтами. - API сторонніх сервісів.	OpenAPI, інтерфейси для читання файлів
Обробка даних	Трансформація, видалення дублікатів, валідація, нормалізація даних.	Алгоритми перевірки, бібліотеки для маніпуляції даними
Зберігання даних	Передача даних до централізованої бази (WordPress) для подальшого використання в плануванні.	WPGraphQL, SQL
Актуалізація даних	Регулярне оновлення даних у базі для підтримання їхньої актуальності.	Автоматизовані запити до API
Валідація	Перевірка якості та цілісності даних: - Формат числових значень. - Відсутність пропусків. - Логічна відповідність.	Бібліотеки валідації даних
Масштабованість	Підтримка великих обсягів інформації при зростанні бази даних і нових джерел даних.	Багатопоточність, оптимізація запитів
Технологічний стек	Набір інструментів і технологій для реалізації всіх етапів імпорту даних.	Node.js, GraphQL, Next.js, Vercel
Очікувані результати	Інтеграція даних із різних джерел. Автоматизація обробки інформації.	Ефективний інтерфейс і стабільна база даних

### 3.2.2 Інтеграція WordPress із Next.js

Відповідно до загальної стратегії, описаної у попередньому розділі, інтеграція WordPress із Next.js є критично важливим етапом реалізації програми імпорту та подальшого відображення даних у веб-застосунку. У цьому підрозділі буде детально розглянуто технічну реалізацію цього процесу, включаючи архітектуру взаємодії, обрані технології, інструменти, а також виклики, які виникли під час розробки.

Інтеграція реалізована як взаємодія між двома основними компонентами:

1) сервер WordPress — виступає як бекенд для зберігання даних продуктів і замовлень. Його головними функціями є:

- зберігання структурованої інформації;
- надання API через WPGraphQL для запитів до бази даних;
- обробка мутацій для створення, оновлення та видалення записів.

2) застосунок на Next.js — слугує фронтом для відображення даних та взаємодії користувачів із системою. Основні завдання:

- виконання GraphQL-запитів до WordPress;
- динамічний рендеринг сторінок із використанням серверного рендерингу (SSR);
- взаємодія з API для оновлення даних у режимі реального часу.

Як показано на рисунку 3.2, інтеграція включає такі ключові етапи:

1) користувач на фронтенді викликає дію (наприклад, перегляд продукту);

2) Next.js надсилає GraphQL-запит до сервера WordPress через WPGraphQL;

3) WordPress обробляє запит і повертає дані у форматі JSON;

4) Next.js отримує відповідь, обробляє її, і відображає результати на сторінці.

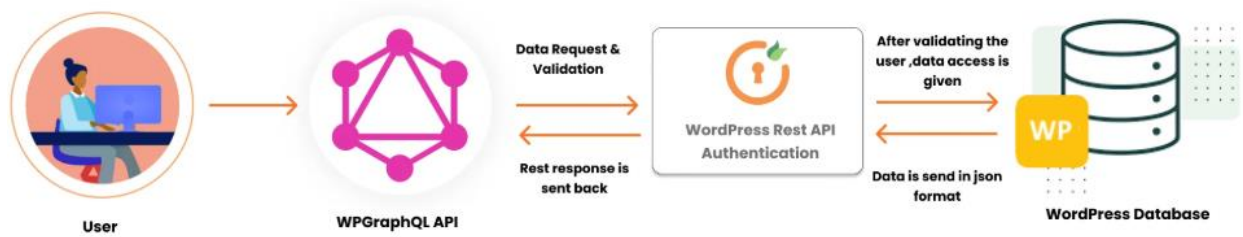


Рис. 3.2. Схема потоку даних, що ілюструє процес передачі даних між користувачем і сервером

Обрані технології:

WPGraphQL дозволяє доступ до бази даних WordPress через GraphQL-запити. Це забезпечує гнучкість і зручність у виборі та фільтрації даних. WPGraphQL підтримує мутації, що дає можливість створювати та оновлювати записи. [27]

У Next.js використовується Apollo Client як основна бібліотека для виконання GraphQL-запитів. Його функціонал включає кешування, обробку помилок та спрощену роботу з підписками в реальному часі.

Для обробки специфічних запитів або виконання мутацій серверні маршрути Next.js використовуються як проксі між WordPress та клієнтом.

Плагін WPGraphQL був встановлений і налаштований у WordPress. Основні кроки:

- 1) активація WPGraphQL через панель адміністратора WordPress;
- 2) конфігурація схем для відображення необхідних типів даних (постів, продуктів, замовлень);
- 3) забезпечення доступу до GraphQL API через /graphql ендпоінт;
- 4) налаштування політик доступу для захисту приватних даних і обмеження прав користувачів;
- 5) тестування запитів GraphQL за допомогою вбудованого інструменту WPGraphiQL.

Створено функції для виконання GraphQL-запитів:

```

const headers = { "Content-Type": "application/json" };
const query = `
  query AllProducts {
    products(
      first: 100
    ) {
      nodes {
        ... on SimpleProduct {
          slug
          name
          image {
            link
            uri
            sourceUrl
          }
          price(format: RAW)
          databaseId
          description(format: RAW)
          regularPrice(format: RAW)
        }
      }
    }
  }
`;
const res = await fetch(API_URL, {
  headers,
  method: "POST",
  body: JSON.stringify({ query }),
});

const {
  data: {
    products: { nodes },
  },
} = await res.json();

```

Рис. 3.3. Функція отримання списку продуктів

```

const createOrderMutation = `
  mutation CreateOrder($input: CreateOrderInput!) {
    createOrder(input: $input) {
      order {
        id
        status
        total
      }
      clientMutationId
    }
  }
`;

const createOrder = async (orderData: any) => {

  const res = await fetch("https://eat.for.fit.test.te.ua/graphql", {
    method: "POST",
    headers,
    body: JSON.stringify({
      query: createOrderMutation,
      variables,
    }),
  });

  const {
    data: {
      createOrder: { order, clientMutationId },
    },
  } = await res.json();

  return order;
};

```

Рис. 3.4. Обробка мутацій для створення замовлення

Next.js використовує SSR для попередньої генерації сторінок із даними, отриманими через WPGraphQL:

```
export async function getServerSideProps() {
  const { data } = await client.query({ query: GET_PRODUCTS });
  return {
    props: { products: data.products.edges },
  };
}
```

Валідація даних реалізована на рівні GraphQL-схеми WordPress, а також на фронтенді через обробку помилок.

### 3.2.3. Налаштування WPGraphQL для доступу до даних

В даному підрозділі розглядається налаштування WPGraphQL, необхідного для надання доступу до даних WordPress через GraphQL API. WPGraphQL є розширенням для WordPress, яке дозволяє отримувати доступ до даних за допомогою GraphQL-запитів, надаючи можливість інтеграції WordPress із сучасними frontend-фреймворками, такими як Next.js.

На рисунку 3.5 наведено схему взаємодії між WordPress, WPGraphQL та зовнішніми запитами з фронтенду.

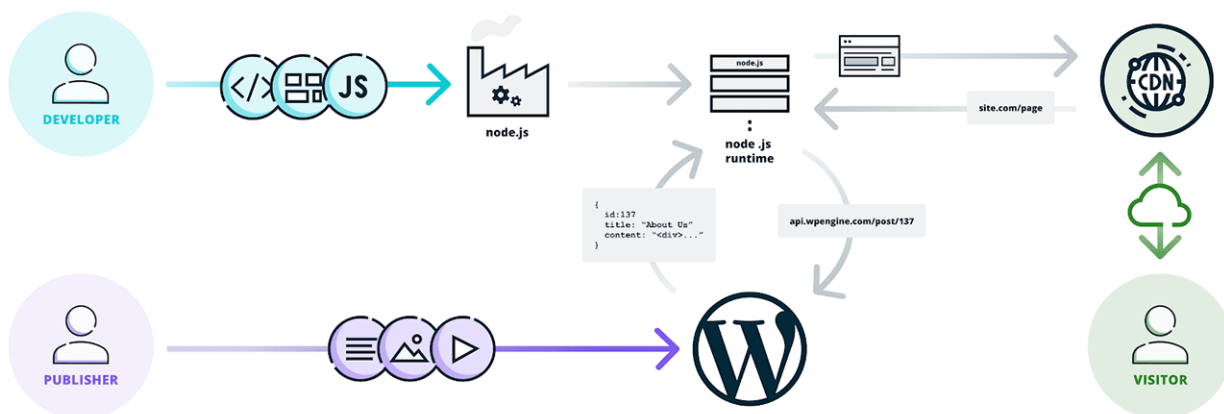


Рис. 3.5. Архітектура доступу до даних WordPress через WPGraphQL

Доступ до даних через GraphQL API можна контролювати за допомогою механізму ролей та дозволів WordPress:

Публічний доступ: Дозволяє отримувати дані, які доступні для всіх користувачів (наприклад, публічні пости).

Приватний доступ: Вимагає автентифікації користувача для отримання приватних даних (наприклад, чернеток, користувачів чи конфіденційних даних). [28]

Щоб налаштувати обмеження доступу, було використано плагіни для авторизації, такі як JWT Authentication та розширення REST API.

Однією з важливих переваг WPGraphQL є можливість розширення стандартного сховища даних WordPress шляхом додавання кастомних полів. Для цього використовують плагін Advanced Custom Fields (ACF).

Також для інтеграції з Next.js необхідно налаштувати CORS (Cross-Origin Resource Sharing), щоб дозволити обмін даними між доменами WordPress та фронтендом.

#### *3.2.4. Фіналізація імпорту даних*

Процес імпорту даних завершується їх обробкою та інтеграцією у фінальну структуру. На цьому етапі всі зібрані дані, що надійшли з різних джерел, об'єднуються в єдиний формат, що забезпечує безперебійну взаємодію з подальшими етапами автоматизації планування харчування.

Після імпорту даних, зібраних за допомогою WPGraphQL з WordPress, виконано їх перетворення в стандартний формат для подальшого використання в Next.js. Це включає в себе перевірку на відповідність усіх полів, що були отримані з API, а також забезпечення коректності структур даних, таких як категорії продуктів, їх харчова цінність, властивості, що визначають потреби користувачів.

Крім того, відбувається очищення непотрібних або неактуальних даних, що можуть призвести до помилок або некоректних результатів при плануванні

харчування. Усі дії з обробки даних автоматизовані, що забезпечує швидке і безпомилкове оновлення інформації.

Таким чином, після фіналізації імпорту дані готові до подальшого використання в системі, що дозволяє здійснювати автоматичне планування харчування на основі індивідуальних потреб користувачів.

### **3.3. Моделювання автоматизованого планування харчування**

Розділ містить опис процесу моделювання автоматизованого планування харчування, включаючи підхід до розробки алгоритмів для індивідуалізації харчових планів. Він охоплює контекст застосування, вибір методів оптимізації раціону та реалізацію основних структурних блоків у Next.js.

#### *3.3.1. Контекст, область застосування та підхід до моделювання*

Моделювання автоматизованого планування харчування для індивідуальних потреб користувачів є важливим етапом розробки системи, що використовує автоматичний аналіз даних для створення персоналізованих планів харчування. Основним завданням цього процесу є точний розрахунок енергетичних потреб користувача з урахуванням таких факторів, як вік, стать, вага, зріст, рівень фізичної активності та інші індивідуальні характеристики. В якості основи для цього розрахунку використовується формула Сан-Жеора, яка дозволяє визначити добову калорійність харчування для підтримки оптимальної фізичної форми. Ця формула враховує базовий рівень метаболізму (BMR) та додаткові корекції залежно від фізичної активності, що є важливим для створення точних індивідуальних планів харчування. [29]

Алгоритм моделювання на основі цієї формули розраховує кількість калорій, необхідних для кожного користувача, і складає раціон, що задовольняє ці потреби, з урахуванням оптимального співвідношення макроелементів (білків, жирів і вуглеводів). Страви та продукти, які включені

в план, підбираються таким чином, щоб вони не лише відповідали розрахованій калорійності, але й забезпечували організм необхідними вітамінами, мінералами та іншими корисними речовинами. Для кожного продукту в базі даних визначається його енергетична цінність і склад, що дає змогу формувати раціон з найбільш відповідних варіантів продуктів.

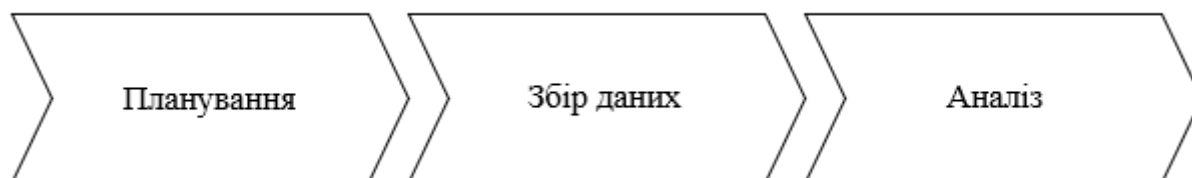


Рис. 3.6. Завдання процесу планування харчування

Процес розробки моделі включає кілька етапів. Першим кроком є збір даних користувача, що включає особисті характеристики та дані про рівень фізичної активності. Ці дані є основою для розрахунку добової калорійності та інших потреб у макро- та мікроелементах. Інформація, отримана від користувача, потім передається до алгоритмів, які оптимізують вибір страв і продуктів, орієнтуючись на встановлені цілі — чи то зниження ваги, підтримка поточної ваги, чи набір маси. Алгоритм моделює меню з урахуванням розподілу калорій по прийомах їжі протягом дня, забезпечуючи різноманітність та баланс.

Подальший етап — це інтеграція цього процесу з базою даних, що зберігає інформацію про продукти. Для реалізації цієї частини системи використовуються інструменти, такі як WPGraphQL, для зручного доступу до даних, збережених у WordPress. База даних містить інформацію про різноманітні продукти та їхні властивості, які використовуються для розрахунків. Алгоритм автоматично вибирає продукти з бази даних на основі розрахованої калорійності та потреб користувача. Також враховуються додаткові параметри, такі як обмеження дієти (наприклад, для вегетаріанців або людей з харчовими алергіями).

На рисунку 3.7 наведено графічний інтерфейс для зручного створення та оновлення записів у базі даних товарів.

Сирники з ягідним джемом, сметаною

Постійне посилання: <https://est4ft.f.ua/menu/syrnyky-z-yagidn-emom-i-smetanoю> Редагувати

Харчова цінність та позначки

Тип товару  
Звичайний

Кількість калорій  
Введіть кількість калорій на 100г продукту  
390

Кількість прийомів їжі на день  
Введіть кількість прийомів їжі на день  
1

Баланс калорій, білків, жирів та вуглеводів

Білки (г)  
29

Жири (г)  
20

Вуглеводи (г)  
24

Опублікувати

Переглянути зміни

Статус: Опубліковано Редагувати

Видимість: Публічно Редагувати

Опубліковано в: 11.09.2024 о 13:33  
Редагувати

SEO аналіз: ОК

Аналіз читабельності: Добре

Catalog visibility: Shop and search results  
Edit

Сору to a new draft  
Відмінити

Оновити

Product image

Натисніть на зображення, щоб розширити чи зменшити

Рис. 3.7. Графічний інтерфейс WordPress для створення продукту в базі

Моделювання також передбачає постійну адаптацію алгоритмів на основі отриманих результатів. Це включає тестування та коригування харчових планів в залежності від змін фізичного стану користувача або корекцій у рівні фізичної активності. Програма повинна мати можливість оновлювати план харчування в реальному часі відповідно до змін у характеристиках користувача, що дозволяє підтримувати максимальну ефективність дієти в процесі її застосування.

Завдяки такому підходу система автоматизованого планування харчування може надавати точні рекомендації та адаптуватися до індивідуальних потреб користувачів. Користувач отримує не лише калорійно збалансовані плани харчування, але й рекомендації, що відповідають його фізіологічним вимогам і вимогам його здоров'ю. [30]

### 3.3.2. Реалізація структурних блоків на Next.js

Реалізація структурних блоків на Next.js базується на чітко визначених компонентах, які взаємодіють між собою для забезпечення ефективної роботи

автоматизованого планування харчування. Основною метою є побудова системи, де кожен блок виконує чітко визначену функцію, забезпечуючи стабільність і масштабованість додатку.

Система складається з кількох основних компонентів: обробки вхідних даних користувача, інтерфейсу для взаємодії з користувачем, а також серверної частини для обчислення та генерації планів харчування. Дані, що вводяться користувачем, передаються на сервер для розрахунку добової калорійності та формування відповідного меню. Нижче на рисунку 3.8 показано детальну структуру проекту, розбиту на сторінки, компоненти, стилі, серверні функції та роботу з контекстом.

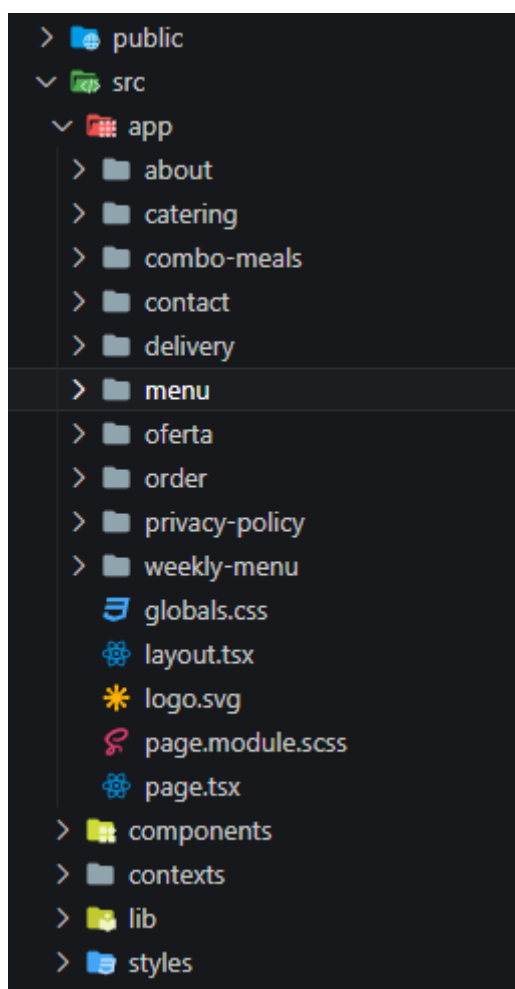


Рис. 3.8. Файлова структура проекту, розбита на модулі

Маршрутизація в Next.js забезпечує зручний доступ до різних частин програми, що дозволяє, зокрема, відобразити результати розрахунків або

переглядати деталі продуктів. Використання серверних API дозволяє ефективно обробляти запити від користувачів і повертати готові результати планування харчування.

Next.js використовується для реалізації серверного рендерингу, що дозволяє динамічно генерувати сторінки з продуктами та їхніми характеристиками. Кожен продукт в базі даних має енергетичну цінність та інші параметри, що використовуються для формування збалансованого харчування. Компоненти інтерфейсу користувача дозволяють переглядати меню, редагувати дані та взаємодіяти з результатами розрахунку.

### SSR (Server-side Rendering)

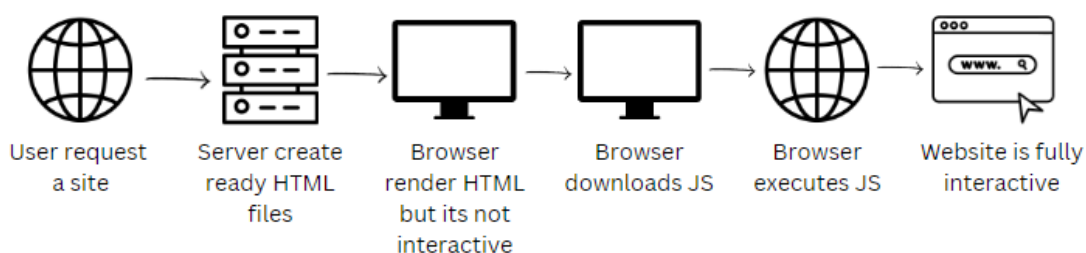


Рис. 3.9. Принцип роботи серверного рендерингу

Інтеграція з базою даних здійснюється через WPGraphQL, що забезпечує динамічний доступ до даних про продукти. Це дозволяє гнучко працювати з інформацією та оновлювати меню без необхідності змінювати серверну частину додатку.

#### 3.3.3. Вибір та використання алгоритмів для оптимізації раціону

Вибір та використання алгоритмів для оптимізації раціону зосереджуються на застосуванні ефективних методів для забезпечення збалансованого харчування відповідно до індивідуальних потреб користувача. У цьому контексті використовується кілька алгоритмів, кожен з яких має на меті розв'язання специфічних завдань у процесі побудови раціону. Одним з

основних підходів є використання алгоритмів для оптимізації добової калорійності, розподілу макро- та мікроелементів, а також вибору продуктів, які відповідають заданим критеріям.

Процес оптимізації включає кілька етапів, починаючи з оцінки введених користувачем даних, таких як вік, вага, рівень фізичної активності та цілі харчування (зниження ваги, підтримка ваги або набір маси). Алгоритм визначає необхідну кількість калорій і макроелементів, після чого на основі цього параметра формується раціон. Один із підходів, який застосовується, — це використання методу лінійного програмування для знаходження оптимального розподілу продуктів, яке задовольняє всі вимоги щодо калорійності, вітамінів та мінералів.

У реалізації застосовуються відомі алгоритми, які ефективно працюють з великими наборами даних. Наприклад, алгоритми на основі машинного навчання можуть бути використані для передбачення найкращих варіантів раціону на основі попереднього досвіду користувачів. Вони дозволяють адаптувати дієту під індивідуальні уподобання та обмеження, такі як алергії або дієтичні обмеження.

Для обчислення оптимального раціону використовуються методи, що враховують фактори, як-от сезонність продуктів або доступність певних продуктів у базі даних. Зазвичай в основу цього процесу покладено алгоритм, що працює з інформацією про продукти та їхні характеристики, що зберігаються у базі даних WordPress. З цієї бази на сервер передаються необхідні дані для виконання оптимізації раціону, після чого результат генерується у вигляді рекомендованого меню.

Основними алгоритмами, що застосовуються для досягнення цієї мети, є алгоритми на основі методу гілок та меж (branch and bound), принцип дії якого показано на рисунку 3.8, а також евристичні методи, що допомагають зменшити складність обчислень, коли необхідно працювати з великою кількістю змінних і обмежень. Використання таких алгоритмів забезпечує

швидке й ефективне формування персоналізованих раціонів навіть для складних вимог.

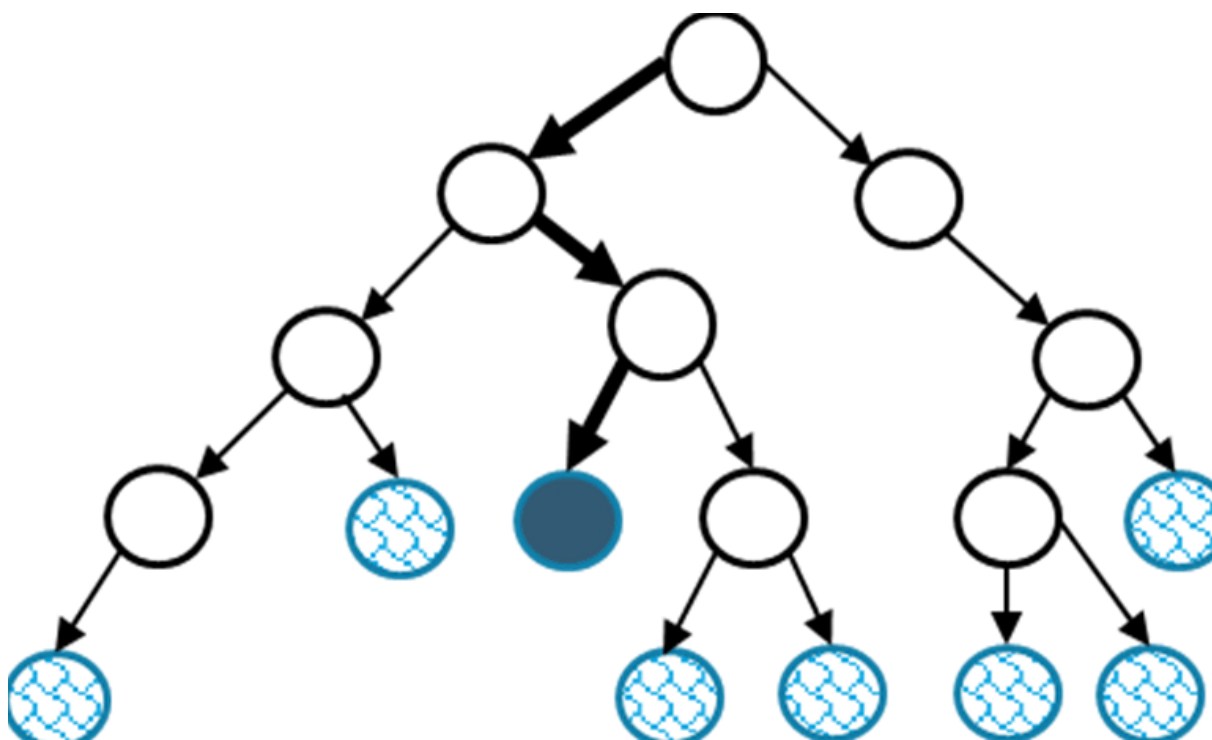


Рис. 3.10. Діаграма роботи алгоритму гілок та меж (branch and bound).

#### *3.3.4. Тестування алгоритмів і моделі*

Тестування алгоритмів та моделей здійснюється для того, щоб перевірити їх ефективність і точність в умовах реальних даних. Для тестування використовуються як тестові набори даних, так і реальні параметри, введені користувачем, щоб забезпечити перевірку на практиці. Основним етапом є запуск алгоритмів оптимізації раціону, що дозволяє порівняти результат з очікуваними значеннями. Для цього тестуються різні стратегії та варіанти розрахунків, зокрема, перевірка здатності алгоритму знаходити оптимальний раціон, що відповідає заданим параметрам (калорії, макроелементи, продукти).

Процес тестування включає кілька етапів. Перший етап — перевірка правильності виконання оптимізації на базі попередньо обраних наборів

даних. Це включає порівняння реальних результатів із бажаними, що дозволяє визначити точність алгоритмів. Після цього перевіряється здатність алгоритму адаптуватися до змін у параметрах користувача, таких як рівень активності або цілі харчування. Тестування проводиться як для одиничних, так і для складених сценаріїв, де алгоритм має враховувати кілька змінних одночасно.

Рисунок 3.11 демонструє блок-схему процесу тестування, де на кожному етапі зчитуються результати виконання оптимізації, проводиться аналіз помилок та визначається ефективність кожного з алгоритмів. Основною метою є забезпечення того, щоб раціон, згенерований алгоритмом, відповідав вимогам користувача, зокрема щодо калорійності та пропорції макроелементів.

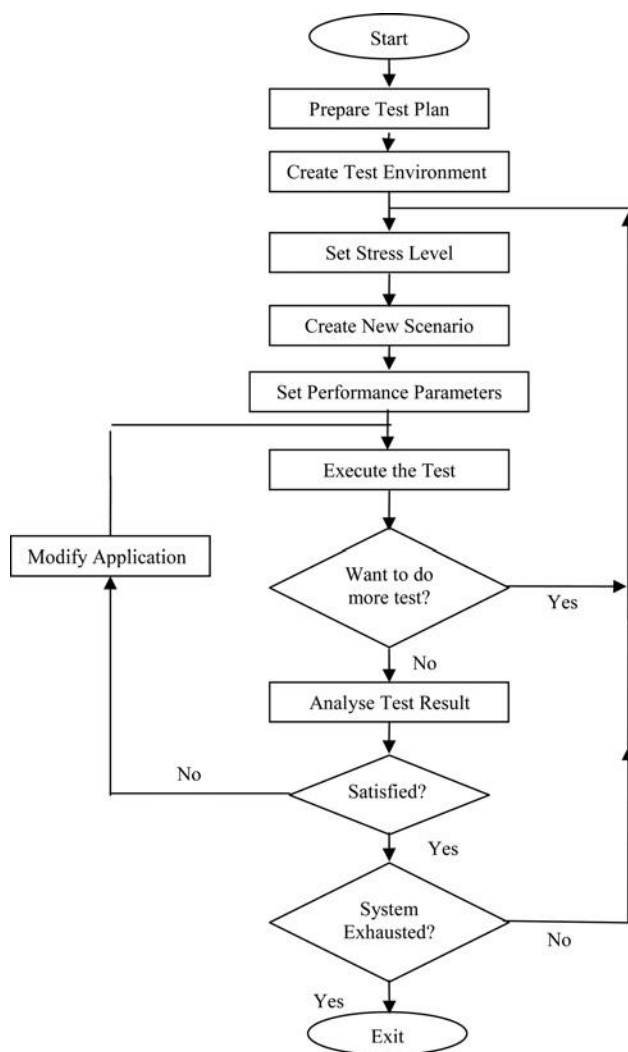


Рис. 3.11. Блок-схема процесу тестування

Однією з важливих частин тестування є валідація моделі за допомогою порівняння з реальними даними користувачів. Це дозволяє виявити можливі помилки в алгоритмах та налаштуваннях, що забезпечує більш точний результат в реальних умовах. Крім того, використовуються різні сценарії тестування для оцінки стабільності алгоритмів у разі зміни вихідних даних, таких як сезонність продуктів чи доступність певних інгредієнтів.

Остаточним етапом тестування є порівняння результатів з іншими існуючими рішеннями або алгоритмами, що дозволяє оцінити, наскільки ефективно запропоновані алгоритми оптимізації раціону працюють порівняно з іншими підходами на ринку.

### *3.3.5. Валідація результатів планування*

Після проведення тестування алгоритмів планування харчування наступним кроком є валідація результатів, яка дозволяє оцінити, наскільки ефективно алгоритм виконує завдання оптимізації раціону відповідно до заданих параметрів. Валідація вимагає використання реальних даних, щоб порівняти отримані результати з очікуваними значеннями, враховуючи різноманітні варіанти індивідуальних потреб користувача, такі як калорійність, співвідношення макроелементів та інші фактори, що можуть змінюватися залежно від конкретного запиту.

Процес валідації передбачає використання спеціальних тестових наборів даних, зібраних із реальних користувацьких запитів, щоб порівняти результати, отримані від оптимізації, з ідеальними варіантами раціонів. Це дозволяє перевірити точність моделі в умовах, наближених до реального використання. Зокрема, проводиться порівняння фактичного розподілу калорій та макроелементів у запропонованих раціонах із заданими значеннями, щоб оцінити, чи відповідають результати вимогам користувача.

Для оцінки ефективності планування використовується кілька метрик, серед яких ключовою є метрика середньої відносної помилки. Вона дозволяє оцінити середню різницю між очікуваними значеннями та результатами,

отриманими після оптимізації раціону. В результаті валідації отримують дані, що вказують на точність моделі та її здатність адаптуватися до змінюваних умов користувачів, таких як зміни в режимі харчування чи рівні фізичної активності.

### **3.4. Виклики в процесі автоматизації планування**

Під час автоматизації планування харчування виникло кілька важливих викликів, які потребували уваги на різних етапах розробки та впровадження системи. Однією з основних проблем стало управління різноманітними індивідуальними потребами користувачів, що включають змінювані параметри, такі як рівень фізичної активності, специфічні дієтичні обмеження та цілі, наприклад, набір або втрату ваги. Програма повинна була враховувати ці варіації і правильно адаптувати раціони для кожного користувача, що інколи призводило до необхідності уточнення алгоритмів оптимізації.

Ще одним викликом став процес інтеграції різних джерел даних. Система працює з великою кількістю параметрів, таких як калорійність продуктів, вміст макро- і мікроелементів, а також персональні налаштування користувача. Наявність таких великих обсягів змінних даних вимагала налаштування ефективних методів обробки та збереження інформації. Це виявило потребу в постійному вдосконаленні бази даних та алгоритмів для коректної взаємодії з різними джерелами та забезпечення їх узгодженості.

Не менш важливим було забезпечення ефективності роботи алгоритмів оптимізації, оскільки на кожному етапі планування необхідно обчислювати велику кількість варіантів раціонів із заданими параметрами. Виявилось, що для деяких складних запитів система може значно сповільнювати свою роботу через обмеження обчислювальних ресурсів, що потребувало додаткових оптимізацій на рівні коду. На рисунку 3.12 наведено приклад оптимізації.

```

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  return (
    <html lang="uk">
      <body className={raleway.className}>
        <div
          style={{
            overflowX: "hidden",
            position: "relative",
          }}
        >
          <CartContextProvider>
            <TopBar />
            <Header />
            {children}
            <Analytics />
            <Footer />
          </CartContextProvider>
        </div>
      </body>
    </html>
  );
}

```

Рис. 3.12. Оптимізація коду з використанням контекст провайдерів, що можуть працювати на серверній сторони

Іншою проблемою стало визначення правильних метрик для оцінки результатів планування. Оскільки індивідуальні вимоги користувачів можуть значно відрізнятися, було складно вибрати єдину метрику, яка б задовільняла всі варіанти застосування. Після кількох спроб визначити найбільш підходящу метрику, було обрано кілька показників, що дозволяють оцінювати, наскільки запропонований раціон відповідає заданим параметрам, і на основі цього проводити порівняння з іншими варіантами.

У цілому, процес автоматизації планування раціону вимагав постійного удосконалення алгоритмів, інтеграції даних та налаштування метрик для

оцінки ефективності. Це забезпечило не лише точність розрахунків, але й покращення загальної користувацької взаємодії з системою.

#### 3.4.1. Проблеми імпорту та обробки даних з WordPress

При розробці системи планування харчування на основі даних з WordPress важливим етапом є процес імпорту та обробки інформації, зокрема продуктів, що зберігаються в базі даних WordPress. Оскільки система харчування має інтеграцію з базою даних WordPress, було необхідно впровадити механізм отримання інформації про продукти, калорійність та інші параметри, а також обробити ці дані для подальшого використання в алгоритмах планування.

Однією з проблем, з якою зіткнувся під час імпорту, стала необхідність правильної обробки різних форматів даних, оскільки інформація про продукти може бути збережена у різних структурах залежно від плагінів і налаштувань WordPress. Наприклад, для отримання даних використовувалися WPGraphQL запити, які можуть повертати значення в різних форматах, включаючи списки, асоціативні масиви та інші структури, що ускладнювало їх подальшу обробку та зберігання у базі даних системи планування харчування.

Окрім цього, значну складність становить необхідність постійного оновлення даних, оскільки нові продукти можуть бути додані до WordPress, а також можуть змінюватися їх характеристики, такі як калорійність чи вміст макро- та мікроелементів. Задля синхронізації цієї інформації з системою планування було розроблено механізм періодичного оновлення, але навіть цей процес вимагав додаткових налаштувань для оптимізації роботи.

```
export const revalidate = 1800;
```

Рис. 3.13. Оновлення актуальних даних з WordPress, значення в мілісекундах, тобто 1800 = оновлення раз в кожні 30 хв

Ще однією проблемою стало зіставлення різних полів даних з WordPress із полями, необхідними для алгоритмів планування харчування. Наприклад, необхідно було точно визначити, які поля містять інформацію про калорії, білки, жири, вуглеводи, і забезпечити правильне відображення цих даних в алгоритмах оптимізації раціону.

Для вирішення цих проблем використовувалися спеціальні адаптери, які здійснюють перетворення даних між різними структурами. Цей процес значно полегшив інтеграцію з іншими компонентами системи, хоча на початкових етапах реалізації знадобилося кілька коригувань для забезпечення коректної роботи всіх функцій.

#### *3.4.2. Оптимізація продуктивності на Vercel*

Для забезпечення оптимальної роботи системи планування харчування на базі Next.js, хостованої на Vercel, важливо було провести оптимізацію продуктивності на різних етапах. Однією з головних задач стала оптимізація часу завантаження сторінок, оскільки високий час відгуку може негативно впливати на користувацький досвід, особливо при інтерактивних елементах, таких як калькулятор калорій або процес вибору продуктів для плану харчування.

Використовуючи серверний рендеринг, Next.js надає можливість значно скоротити час завантаження сторінки, виконуючи рендеринг на сервері і передаючи вже готовий HTML на клієнтську частину. Однак, оскільки велика частина даних для планування раціону надходить через WPGraphQL з WordPress, необхідно було врахувати ефективність запитів до бази даних. Для цього використовували кешування запитів та оптимізацію їх виконання.

На рисунку 3.14, враховуючи серверну оптимізацію та оптимізацію коду, представлено ключові показники продуктивності, які надає функція Speed Insights від Vercel, включаючи їхні значення та роль у оцінці швидкості та ефективності роботи веб-додатка.

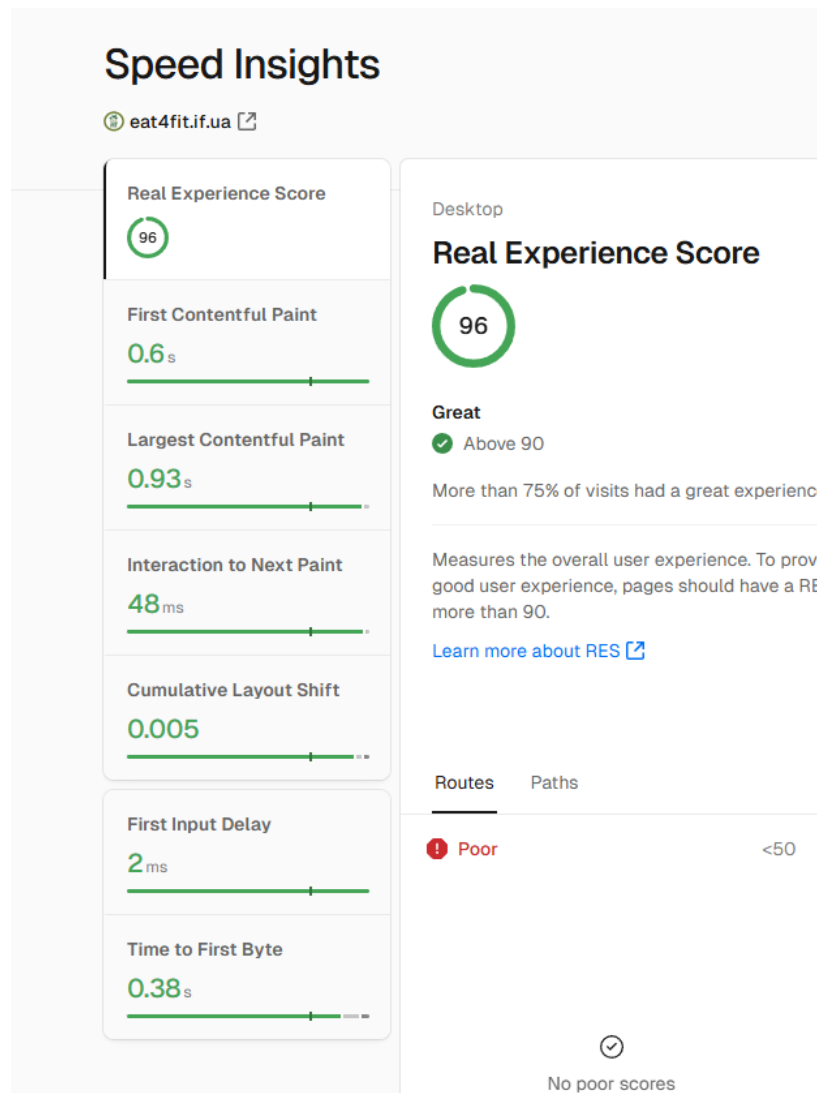


Рис. 3.14. Показники продуктивності, що надаються функцією Speed Insights від Vercel

Одним із важливих кроків стала реалізація кешування за допомогою Vercel Edge Functions, що дозволяє кешувати відповіді на часто запитувані дані, знижуючи навантаження на сервер і прискорюючи час відгуку. Також, була налаштована стратегія інкрементальних статичних оновлень (ISR), яка дозволяє частково оновлювати сторінки без необхідності перезавантажувати весь контент, що значно підвищує ефективність системи, особливо при роботі з великими базами даних продуктів.

Водночас, були вжиті заходи для оптимізації взаємодії з WPGraphQL. Для цього використано індексацію запитів, що дозволяє витягувати лише необхідні поля даних, мінімізуючи обсяг переданої інформації. Таким чином,

час завантаження і відгуку системи було знижено завдяки скороченню кількості непотрібних запитів.

Для подальшої оптимізації та мінімізації часу обробки було застосовано підхід lazy loading, що дозволяє відкладене завантаження компонентів до моменту їх необхідності. Це особливо важливо для великих елементів на сторінках, таких як таблиці з продуктами, що значно знижує час початкового завантаження сторінки і покращує загальну продуктивність.

```
const DynamicFrequentlyAsked = dynamic(  
  () =>  
    import("@/components/FrequentlyAsked/FrequentlyAsked").then(  
      (mod) => mod.FrequentlyAsked  
    ),  
  { ssr: false }  
);
```

Рис. 3.15. Динамічна загрузка компоненту (lazy loading), якого не видно поки користувач не прогортає сторінку нижче.

Загалом, застосування цих стратегій дозволило значно підвищити ефективність роботи сайту на Vercel, зокрема шляхом зменшення часу відгуку та забезпечення стабільної роботи навіть за високого навантаження.

### Висновки до розділу

Отже, в цьому розділі було детально розглянуто процес автоматизації планування харчування, зокрема, вибір алгоритмів, тестування моделей та валідацію результатів, а також обговорено виклики, що виникли під час реалізації проекту. В рамках розробки стратегії автоматизації планування було обрано підхід, що поєднує методи оптимізації раціону на основі індивідуальних потреб, використовуючи алгоритми для планування та

обробки даних з WordPress. Це дозволило створити ефективну систему, яка враховує індивідуальні вимоги користувачів та забезпечує їх задоволення.

Також були розглянуті проблеми, що виникли під час імпорту та обробки даних з WordPress, зокрема, ефективність запитів та проблеми кешування. Застосовані методи оптимізації продуктивності на платформі Vercel дозволили досягти покращення часу відгуку та забезпечити стабільну роботу системи, навіть при великому навантаженні. Зокрема, використання серверного рендерингу, кешування та стратегій інкрементальних оновлень забезпечили ефективне завантаження даних і покращили взаємодію з користувачем.

Таким чином, у цьому розділі було розроблено комплексне рішення для автоматизації планування харчування, яке може бути адаптоване до різних потреб користувачів та забезпечує високу продуктивність і точність у процесі планування індивідуального раціону.

## ВИСНОВКИ

Магістерська робота присвячена дослідженню та впровадженню алгоритмів автоматизації планування харчування на основі індивідуальних потреб користувачів.

У першому розділі проведено огляд основних аспектів планування харчування, визначено ключові проблеми в автоматизації цього процесу та запропоновано підходи для їх вирішення. Описано основні алгоритми, які використовуються для розрахунку індивідуальних харчових планів, а також інструменти для обробки даних з різних джерел, зокрема WordPress. Визначено вимоги до системи автоматизації та зазначено важливість інтеграції з платформами для управління даними.

У другому розділі здійснено аналіз існуючих методів для автоматизованого планування харчування, зокрема використання методів оптимізації раціону, алгоритмів для обробки даних, а також методів валідації результатів. Описано основні етапи впровадження алгоритмів планування, включаючи імпорт та обробку даних, тестування моделей та оцінку їх ефективності. Розглянуто ключові метрики для валідації результатів, такі як точність планування та ефективність роботи з великою кількістю даних.

У третьому розділі представлено практичну реалізацію розробленої методології. Визначено загальну стратегію автоматизації планування харчування, яка поєднує використання алгоритмів для створення харчових планів, обробки даних з WordPress та оптимізації продуктивності на платформі Vercel. Описано проблеми, що виникли під час імплементації, зокрема, труднощі при обробці даних та оптимізації запитів, а також шляхи їх вирішення. Було також проведено тестування та валідацію моделей, що підтвердило ефективність обраних алгоритмів.

Таким чином, у роботі вдалося розробити ефективну систему автоматизації планування харчування, що поєднує індивідуальний підхід до складання раціону з високою продуктивністю та надійністю. Запропоновані

алгоритми планування дозволяють враховувати різноманітні параметри, такі як калорійність, харчові потреби, переваги користувача та наявність продуктів, що забезпечує точність та адаптивність складання раціону. Система також здатна інтегруватися з платформами управління даними, такими як WordPress, що дозволяє обробляти великі обсяги інформації та забезпечувати її коректність і доступність.

Завдяки використанню передових методів обробки даних та валідації результатів, система здатна точно враховувати індивідуальні потреби користувачів, що робить її універсальним інструментом для різних областей застосування, від персоналізованого харчування до професійних програм для дієтологів і фітнес-тренерів. Створена система може бути адаптована для розв'язання практичних завдань у реальному часі, що дозволяє значно підвищити ефективність процесу планування харчування.

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Vovk L.V., Gavryliuk S.O. Automation Technologies for Meal Planning Overview and Prospects // *Visnyk of Kharkiv National University*. – 2018. – С. 45–56.
2. Makarenko I.Y., Vasylychenko O.V. Algorithms for Automated Meal Plan Selection // *Problems of Informatics*. – 2020. – Т. 15, No 2. – С. 102–115.
3. Aslanova T.V., Khrapova I.S. Approach to Calculating Individual Nutritional Needs Based on the San-Jore Method // *Food Science and Technology*. – 2017. – С. 121–130.
4. Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet Allocation // *Journal of Machine Learning Research*. – 2003. – С. 993–1022.
5. Mikolov T., Sutskever I., Chen K., Corrado G.S., Dean J. Distributed Representations of Words and Phrases and Their Compositionality // *Advances in Neural Information Processing Systems*. – 2013.
6. Lau J.H., Baldwin T. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation // *Proceedings of the 54th Annual Meeting of the ACL*. – 2016.
7. Griffiths T.L., Steyvers M. Finding Scientific Topics // *Proceedings of the National Academy of Sciences*. – 2004. – Т. 101 (Suppl 1). – С. 5228–5235.
8. Hofmann T. Probabilistic Latent Semantic Analysis // *Proceedings of UAI*. – 1999. – С. 289–296.
9. Chang S.K. *Handbook of Software Engineering & Knowledge Engineering*. – World Scientific, River Edge, NJ, London, 2001.
10. Ramage D., Rosen E., Chuang J., Manning C.D., McFarland D.A. Topic Modeling for the Social Sciences // *Advances in Neural Information Processing Systems*. – 2009.
11. Rosen M., Griffiths T., Smith P., Smyth P. The Author-Topic Model for Authors and Documents // *Proceedings of UAI*. – 2004.

12. Yin J., Wang J. A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering // Proceedings of KDD. – 2014. – C. 281–290.
13. Li J., Sun A., Han J., Li C. A Survey on Deep Learning for Named Entity Recognition // IEEE Transactions on Knowledge and Data Engineering. – 2020. – T. 32, No 6. – C. 1006–1020.
14. Gao J., Yan S., Xu F. Real-time Ingredient Recognition Using Computer Vision Algorithms // Journal of Health Informatics. – 2022. – C. 98–112.
15. Richards B., Ma G., Han I. Classification Algorithms for Personalized Meal Plans // Journal of Health Technology. – 2021. – C. 55–67.
16. Chen S., Li T., Zhang M. Intelligent Systems for Meal Management Methods and Applications // Journal of Informatics. – 2020. – C. 78–89.
17. Godoy L.R., García E.M., Peña A.L. Applying Machine Learning Algorithms to Analyze Dietary Habits // Proceedings of the 5th International Conference on Health Informatics. – 2019.
18. Bakman S., Greenberg I., Jones T. Using Deep Learning Algorithms to Create Personalized Meal Plans // AI Research. – 2021.
19. McDonald R., Roberts J., Johnson K. Algorithms for Determining Individual Nutritional Requirements // Food Research Journal. – 2020. – C. 120–135.
20. Wang L., Zhang Y., Zhao B. Topic Modeling in Health Informatics Applications and Methods // IEEE Transactions on Biomedical Engineering. – 2021.
21. Griffiths T.L., Steyvers M. Finding Topics in Large Text Corpora // Proceedings of the National Academy of Sciences. – 2004. – T. 101 (Suppl 1). – C. 5228–5235.
22. Blei D.M., Lafferty J.D. Dynamic Topic Models // Proceedings of the 23rd International Conference on Machine Learning. – 2006.
23. Zhang Y., Jin R., Zhou Z. Understanding the Bag-of-Words Model A Statistical Framework // IJCAI Proceedings. – 2010.
24. Zhao W.X., Li J., Yu J., Huang Z. A Survey on Topic Modeling Techniques and Applications // Journal of Information Science. – 2015.

25. Roberts M.E., Stewart B.M., Tingley D. STM An R Package for Structural Topic Models // Journal of Statistical Software. – 2014. – T. 57, No 1. – C. 1–38.
26. Wang C., Blei D.M. Collaborative Topic Modeling for Recommending Scientific Articles // Proceedings of KDD. – 2011. – C. 448–456.
27. Boyd-Graber J., Hu Y., Mimno D. Applications of Topic Models // Foundations and Trends in Information Retrieval. – 2014.
28. Hu Y., Boyd-Graber J., Satinoff B., Smith A. Interactive Topic Modeling // Machine Learning Journal. – 2014. – T. 90, No 1. – C. 15–40.
29. McAuliffe J.D., Blei D.M. Supervised Topic Models // Advances in Neural Information Processing Systems. – 2008.
30. Mimno D., Wallach H., Talley E., Leenders M., McCallum A. Optimizing Semantic Coherence in Topic Models // Proceedings of EMNLP. – 2011.