

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

Аліна САВЧЕНКО.

«_____» _____ 2024р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”

Тема: “ Веб-застосунок для персоналізованого планування харчування ”

Виконавець: _____ студентка групи УС-413Б Козирева Марина Андріївна

Керівник: _____ старший викладач Шевченко Олександр Петрович

Нормоконтролер: _____ старший викладач Шевченко Олександр Петрович

Київ – 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач випускової кафедри

Аліна САВЧЕНКО

« ____ » _____ 2024р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи (проекту)

Козиревої Марини Андріївни
(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Веб-застосунок для персоналізованого планування харчування»
затверджена наказом ректора від “05” квітня 2024р. за № 517/ст.
- 2. Термін виконання роботи:** 05.04.2024 – 07.06.2024р.
- 3. Вихідні дані до роботи:** розробка веб-застосунку для створення персонального планування харчування
- 4. Зміст пояснювальної записки:** вступ, дослідження та аналіз сучасних методів планування харчування, вибір та огляд технологій веб-розробки, визначення функціональних вимог і методів дослідження, проектування бази даних та користувацького інтерфейсу, розробка веб-додатку, огляд реалізованого проекту, висновки
- 5. Перелік обов'язкового графічного матеріалу:** слайди презентації Power Point

6. Календарний план-графік

| № п/п | Завдання | Термін виконання | Підпис керівника |
|-------|---|-----------------------------|------------------|
| 1 | Отримання завдання на дипломну роботу. Аналіз джерел та літератури за темою кваліфікаційної роботи. | 05.04.2024р. – 09.04.2024р. | |
| 2 | Розробка та затвердження плану дипломної роботи. | 10.04.2024р. | |
| 3 | Консультації з науковим керівником щодо структури роботи. | 11.04.2022р. | |
| 4 | Вибір технологій та інструментів для розробки веб-додатку. | 12.04.2024р. – 18.04.2024р. | |
| 5 | Проектування бази даних та архітектури додатку. | 19.04.2024р. – 24.04.2024р. | |
| 6 | Розробка користувацького інтерфейсу. | 25.04.2024р. – 30.04.2024р. | |
| 7 | Програмування функціональності додатку. | 01.05.2024р. – 09.05.2024р. | |
| 8 | Тестування додатку. | 10.05.2024р. – 12.05.2024р. | |
| 9 | Оформлення пояснювальної записки. | 13.05.2024р. – 26.05.2024р. | |
| 10 | Підготовка до презентації та захисту дипломної роботи. | 27.05.2024р. – 09.06.2024р. | |

7. Дата видачі завдання: «05» квітня 2024 р.

Керівник кваліфікаційної роботи : Олександр ШЕВЧЕНКО
(підпис керівника) (П.І.Б.)

Завдання прийняла до виконання : Марина КОЗИРЕВА
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Веб-застосунок для персоналізованого планування харчування»: 70 с., 44 рис., 14 літературних джерел.

Об'єктом дослідження є персональне планування харчування.

Предметом дослідження є методи та засоби індивідуалізації дієтичних рекомендацій у процесі планування харчування, що враховують особистісні дані, фізіологічні особливості та харчові переваги користувачів.

Метою дипломної роботи є розробка веб-додатку для персоналізованого планування харчування, який включає проектування користувацького інтерфейсу, інтеграцію бази даних для управління даними користувачів та створення ефективного механізму фільтрації страв відповідно до індивідуальних потреб користувачів.

Методи дослідження охоплюють детальний аналіз предметної області, вимог та потреб користувачів, теоретичний аналіз сучасних методик розробки веб-додатків, створення реляційної моделі баз даних, адаптацію сучасних практик UI/UX дизайну, а також розробку та комплексне тестування інтерактивного веб-сайту.

Отримані результати та їх новизна полягає у розробці алгоритму персоналізованого підбору дієтичних рекомендацій, що інтегрує комплексний аналіз особистісних даних користувачів, що робить дані рекомендації значно точнішими у порівнянні з існуючими підходами.

Результати бакалаврської роботи демонструють розробку ефективного веб-додатку для персоналізованого планування харчування, включаючи алгоритми для індивідуалізації дієти на основі особистих даних користувачів.

ВЕБ-ДОДАТОК, ПЕРСОНАЛІЗОВАНЕ ХАРЧУВАННЯ, МОВИ ПРОГРАМУВАННЯ, ІНТЕФЕЙС КОРИСТУВАЧА, ПЛАНУВАННЯ ХАРЧУВАННЯ, РОЗРОБКА САЙТУ

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ | 7 |
| ВСТУП..... | 8 |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 10 |
| 1.1. Сучасні підходи до здорового харчування..... | 10 |
| 1.2. Історія розвитку сервісів з персоналізованого харчування..... | 12 |
| 1.3. Існуючі платформи для планування харчування..... | 13 |
| 1.3.1. Apple Health..... | 13 |
| 1.3.2. FatSecret..... | 14 |
| 1.3.3. Plan to Eat | 15 |
| 1.3.4. Lose It! | 16 |
| 1.4. Розробка алгоритму підбору страв | 17 |
| 1.5. Аналіз потреб користувачів у веб-додатку..... | 21 |
| 1.6. Карта сайту майбутнього застосунку | 24 |
| 1.7. Висновок до розділу 1 | 26 |
| РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ..... | 28 |
| 2.1. Вибір мов програмування | 28 |
| 2.1.1. HTML..... | 28 |
| 2.1.2. CSS..... | 29 |
| 2.1.3. JavaScript | 30 |
| 2.1.4. PHP..... | 31 |
| 2.2. VS Code як середовища розробки..... | 31 |
| 2.3. Управління базами даних..... | 33 |
| 2.3.1. Вибір системи управління базами даних | 33 |
| 2.3.2. Конфігурація локального серверу з використанням MAMP..... | 35 |
| 2.3.3. Розробка схеми бази даних і використання реляційної моделі | 37 |
| 2.4. Проектування користувацького інтерфейсу..... | 39 |
| 2.4.1. Принципи створення інтуїтивно зрозумілого UI/UX | 39 |

| | |
|---|-----------|
| 2.4.2. Використання Figma для дизайну макетів | 40 |
| 2.4.3. Прототип проекту | 41 |
| 2.4.4. Проектування візуального стилю | 43 |
| 2.5. Висновок до розділу 2 | 46 |
| РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ..... | 48 |
| 3.1. Основні елементи управління сайтом. | 48 |
| 3.2. Реєстрація та аутентифікація користувачів у веб-застосунку | 50 |
| 3.3. Збір та обробка особистих даних користувачів..... | 52 |
| 3.4. Персоналізований підбір страв | 56 |
| 3.5. Розробка сторінки рецептів | 57 |
| 3.6. Реалізація календаря для розподілу страв на дні тижня | 61 |
| 3.7. Висновок до розділу 3 | 66 |
| ВИСНОВКИ | 67 |
| СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ | 69 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

| | |
|-------------------------|--|
| BMR | Basal Metabolic Rate – базовий обмін речовин |
| TDEE | Total Daily Energy Expenditure – загальна кількість витрат калорій за день |
| Customer Persona | Ідеальний портрет користувача або клієнта продукту |
| Sitemap | Карта сайту |
| HTML | HyperText Markup Language – мова гіпертекстової розмітки |
| CSS | Cascading Style Sheets - мова каскадних стилів |
| VS Code | Visual Studio Code - редактор коду для програмування. |
| СУБД | Система управління базами даних |
| SQL | Structured Query Language - мова для управління даними в СУБД |
| MAMP | Mac, Apache, MySQL, PHP - Програмний пакет для локального веб-розробника |
| OS | Операційна система |
| PHP | Hypertext Preprocessor - Скриптова мова програмування |
| UX | User Experience – користувацький досвід |
| UI | User Interface – користувацький інтерфейс |
| MD5 | Message-Digest Algorithm 5 - алгоритм для створення 128-бітного хешу з тексту |
| Cookies | Маленькі файли, збережені на комп'ютері, що відстежують інформацію про відвідування сайтів |
| Pop-up | Спливаюче вікно на сайті |

ВСТУП

У сучасному світі, де швидкість та ефективність визначають ритм життя, сучасна людина все частіше шукає способи оптимізувати свої повсякденні процеси. В контексті збільшення уваги до здоров'я та якості життя, особливо актуальним стає питання планування харчування. Індивідуалізоване планування харчування, що використовує досягнення цифрових технологій, набуває нового значення, оскільки дозволяє кожній людині підходити до формування свого раціону з урахуванням особистих фізіологічних особливостей та уподобань.

Актуальність дослідження визначається зростаючим попитом на рішення, що дозволяють індивідуально підійти до організації харчування в умовах стрімкого темпу сучасного життя. Застосування інтернет-технологій у цій сфері є відповіддю на потребу забезпечення балансу між раціональним харчуванням і динамікою щоденного графіка сучасної людини. Такий підхід дозволяє не тільки спростити процес вибору та керування раціоном, але й забезпечує можливість його точної індивідуалізації.

Об'єктом дослідження є процес планування харчування на основі персональних даних користувачів. Предметом дослідження виступають методи та засоби індивідуалізації дієтичних рекомендацій у процесі планування харчування, які охоплюють особистісні дані, фізіологічні особливості, а також харчові переваги користувачів.

Проект має на меті створення веб-додатку, який надасть користувачам інструменти для ефективного управління їхніми дієтичними планами. Завдання, що були поставлені для досягнення цієї мети, включають глибокий аналіз вимог споживачів, вибір оптимальних технологій та інструментів веб-розробки, створення користувацького інтерфейсу, що забезпечує простоту та зручність у використанні, а також розробка ефективного механізму фільтрації страв відповідно до індивідуальних потреб користувачів

Для досягнення цієї мети були визначені такі основні завдання:

- Проаналізувати вимоги споживачів до індивідуалізованих харчових планів;

- Вибрати технології та інструменти веб-розробки, що найкраще відповідають потребам проекту;
- Розробити веб-додаток, який забезпечить зручний інтерфейс та ефективне управління харчовими планами.

Задля реалізації поставлених завдань були використані технології HTML, CSS, JavaScript, PHP та SQL, які дозволяють створити динамічний і взаємодіючий веб-інтерфейс. Особлива увага приділялася не лише технічному аспекту розробки, але й взаємодії користувачів з системою, що забезпечується через впроваджений функціонал.

Наукова новизна отриманих результатів полягає в розробці інноваційного алгоритму персоналізованого підбору дієтичних рекомендацій, який інтегрує комплексний аналіз особистісних даних користувачів. Це дозволяє забезпечити значно більшу точність рекомендацій порівняно з існуючими системами.

Таким чином, у дипломному проекті зосереджується на використанні сучасних веб-технологій для створення веб-додатку персоналізації харчування. Розроблений веб-додаток не тільки вирішує практичні завдання індивідуалізації харчування, але й вносить вклад у розвиток технологій харчування та веб-дизайну, розширюючи можливості для подальших досліджень і вдосконалення в цій області.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Сучасні підходи до здорового харчування

Правильне харчування завжди було та залишається актуальною темою, особливо у наш час, коли все більше людей прагнуть до здорового способу життя. Правильне харчування - це підхід до споживання їжі, який забезпечує організм людини всіма необхідними поживними речовинами для підтримки здоров'я та оптимальної фізичної активності. Щоденно організм людини потребує білків, вуглеводів, жирів, води, вітамінів та мінералів у правильних пропорціях. Достатня кількість кожного компонента може забезпечити належний стан та функціонування всього організму, надаючи заряд сил та енергії.

За останні декілька років модною стала тенденція до підтримки здорового образу життя, що включає в себе раціональне харчування, фізичну активність та інші аспекти здорового життя. Цьому сприяють засоби масової інформації, такі як телебачення, радіо, газети, журнали, інтернет-портали та соціальні мережі, що переповнені рекомендаціями щодо здорового харчування, рецептами корисних страв та секретами швидкого скидання ваги.

Популярні телевізійні програми про здорове харчування, блоги та відео на YouTube, де відомі експерти та зірки діляться порадами щодо здорового способу життя, також мають великий вплив на свідомість громадськості. Інформаційні ресурси також активно популяризують рецепти здорової їжі, зеленого харчування, вегетаріанства та інших напрямків збалансованого харчування. У великій кількості порад важливо, щоб кожна людина, читаючи та слухаючи цю інформацію, яка стосується її здоров'я, відбирала тільки доказані дані.

| | | | | | | | |
|---------------------|----------------------|--|--|------------------------------|---------------|--------------|----------------|
| Кафедра КІТ | | | | НАУ 24 34 71 000 ПЗ | | | |
| <i>Виконав</i> | <i>Козирева М.А.</i> | | | АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ | <i>Літера</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| <i>Керівник</i> | <i>Шевченко О.П.</i> | | | | | 10 | 18 |
| <i>Консулт.</i> | | | | | УС-413 122 | | |
| <i>Норм. контр.</i> | <i>Шевченко О.П.</i> | | | | | | |

Успішне харчування ґрунтується на увазі до загального харчового раціону, а не окремих продуктів. Ніякий окремий поживний компонент не спроможний забезпечити гарним здоров'ям, тому рекомендується збалансоване та різноманітне харчування. Важливо враховувати індивідуальні особливості при складанні раціону, такі як хронічні захворювання, вік та алергічні реакції на їжу.

Розуміння того, що кожен організм є унікальним і вимагає індивідуального підходу до харчування, зросло серед людей. У майбутньому можна очікувати розвиток технологій та сервісів, які дозволять визначити індивідуальні потреби в поживних речовинах і скласти персоналізоване харчування на основі генетичного аналізу або інших методів. Цей підхід буде враховувати особливості кожного організму, його спосіб життя, алергії та інші індивідуальні характеристики.

Сучасні технології грають важливу роль у підтримці правильного харчування та здорового способу життя. Одним з таких технологічних рішень є мобільні додатки та онлайн-платформи, які надають користувачам доступ до інформації про харчування та дієту. Ці додатки допомагають вести щоденний журнал харчування, визначати калорійність страв, контролювати споживання поживних речовин та рекомендації щодо здорового харчування.

Також, розвиток сенсорних технологій дозволяє створювати різноманітні пристрої, які допомагають вимірювати рівень цукру в крові, калорійність страв, склад тіла та інші показники, що дозволяє користувачам моніторити свій стан здоров'я та адаптувати дієту відповідно до потреб організму.

Іншими сучасними технологіями є інтелектуальні кухонні прилади та пристрої, які допомагають приготувати здорові страви швидко та ефективно, контролюючи кількість жиру, солі та інших нежиттєважливих компонентів.

Сучасні технології в галузі здорового харчування відкривають широкі можливості для підтримки здорового способу життя. Вони допомагають збалансувати раціон, контролювати споживання поживних речовин та забезпечують доступ до інформації, необхідної для здорового харчування. Такі інновації сприяють популяризації здорового харчування та створенню індивідуальних підходів до підтримки організму у сучасному світі.

1.2. Історія розвитку сервісів з персоналізованого харчування

Персоналізоване харчування стало значущою складовою у здоров'ї та добробуті, як новаторське рішення для покращення життя людей. Історія цього напрямку розвитку була спрямована на пошук оптимального підходу до харчування кожної конкретної людини, враховуючи її унікальні потреби, стан здоров'я, фізичну активність, спосіб життя та смакові уподобання.

Поняття персоналізованого харчування спочатку активно використовувалось у медичних цілях для пацієнтів з різними захворюваннями та дієтологічними рекомендаціями. Різноманітні дослідження та практичний досвід лікарів дав поштовх розвитку цього напрямку.

З поширенням інформаційних технологій та доступності даних з'явилися інтерактивні сервіси для персоналізованого харчування. Перші з них були базовими, пропонуючи загальні поради та рекомендації.

За останні роки спостерігається швидкий розвиток технологій у галузі персоналізованого харчування. Популярність смарт-годинників та фітнес-трекерів зростає, що сприяє збору даних про фізичну активність, харчування та сон. Хоча точність цих пристроїв не завжди дорівнює професійним, вони допомагають людям стежити за своїм здоров'ям.

Веб-застосунки, мобільні додатки та інші інноваційні засоби дозволяють створювати досить точні та ефективні плани харчування для кожної особи. Згідно з даними з Deloitte Global Mobile Consumer Survey (2021), близько 28% респондентів із різних країн використовують смартфони та носимі пристрої для відстеження даних про споживання їжі, рівень фізичної активності, а також враховувати географічні та кліматичні умови. Такі сервіси надають користувачам індивідуалізовані рекомендації для підтримки здорового способу життя.

Дослідження, проведене Центром прикладних досліджень Вестмінстера (WARC), підтверджує ефективність трекерів активності та мобільних додатків для покращення фізичної активності. За даними дослідження, середній приріст фізичної активності становить близько 2000 кроків на день.

Майбутнє персоналізованого харчування вбачається у ще більшому використанні штучного інтелекту для аналізу даних, врахування генетичних особливостей та зв'язків між різними факторами. Також очікується зростання популярності використання віртуальної реальності та розширеної реальності для навчання та підтримки здорового харчування.

1.3. Існуючі платформи для планування харчування

Існують безліч платформ для планування харчування, вони представляють різноманітні підходи та функціональні можливості для користувачів у сфері контролю та організації свого раціону. Подібні додатки та веб-сайти, такі як Apple Health, FatSecret, Plan to Eat, та Lose It!, надають користувачам широкий спектр інструментів та функцій для досягнення їхніх харчувальних цілей та підтримки здорового способу життя.

1.3.1. Apple Health

Apple Health є його універсальним додатком з широким спектром функцій, які охоплюють не лише планування харчування, а й інші аспекти здоров'я та фітнесу. Додаток дозволяє збирати інформацію про фізичну активність, включаючи кількість кроків, дистанцію, витрачені калорії та серцевий ритм. Також, він відстежує сон та має можливість аналізувати цикл сну та якість відпочинку. Apple Health інтегрується з додатками, які мають спеціалізовані функції, такі як вимірювання рівня цукру в крові або тиск за допомогою відповідних пристроїв.

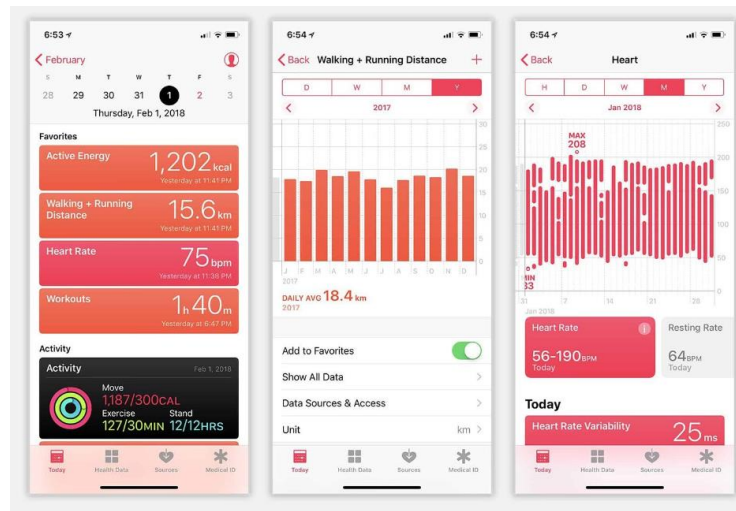


Рис. 1.1. Apple Health

Однією з головних переваг Apple Health є його зручність та доступність для користувачів пристроїв Apple. Він стає центральним місцем для зберігання та аналізу даних про здоров'я, що дозволяє користувачам отримувати повну картину свого фізичного стану та здоров'я. Інтеграція з іншими додатками та пристроями створює можливість для комплексного підходу до моніторингу здоров'я.

Незважаючи на переваги, деякі користувачі можуть відзначати певні недоліки Apple Health. Наприклад, для повного функціоналу потрібні пристрої Apple, що може бути обмеженням для користувачів інших марок. Також, інтеграція з деякими пристроями або додатками може бути не такою ефективною, як інші спеціалізовані платформи. Деякі користувачі можуть відчувати складність у використанні деяких функцій чи налаштуванні додатку під свої потреби.

1.3.2. FatSecret

FatSecret відрізняється також тим, що дозволяє користувачам встановлювати персоналізовані цілі харчування і фітнесу, щоб досягти конкретних результатів. Додаток також має можливість вести журнал фізичної активності та контролювати вагу, що допомагає стежити за загальним станом здоров'я. Більшість функцій додатку є безкоштовними. Користувачі з Premium-акаунтами мають індивідуальний план харчування, розроблений дієтологами.

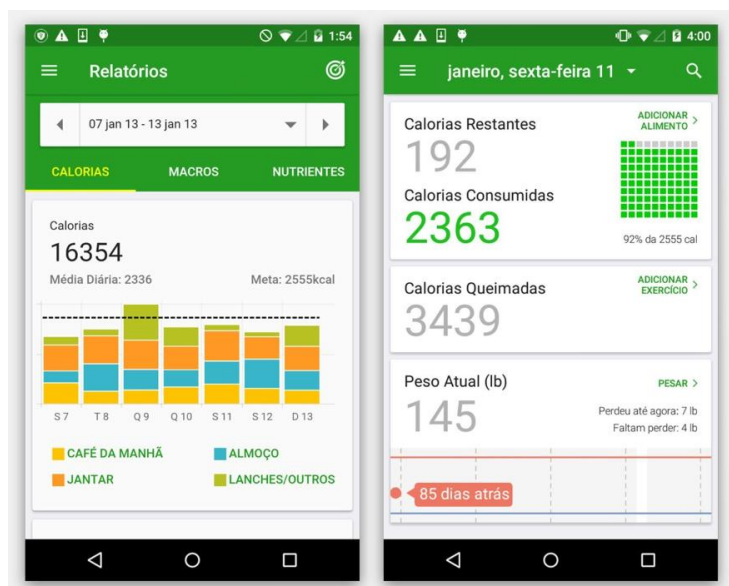


Рис. 1.2. FatSecret

Перевагами FatSecret є його широка база рецептів та харчових продуктів, що дозволяє знаходити ідеї для різноманітних страв та готувати їх з урахуванням калорійності та поживності. Також, додаток має зручний інтерфейс, який дозволяє легко знаходити необхідну інформацію та вести облік харчування без зайвих складнощів.

Проте, серед недоліків FatSecret може бути менша інтеграція з іншими пристроями та додатками, що може ускладнювати обмін даними або синхронізацію інформації. Також, інтерфейс додатку може виглядати менш сучасною порівняно з іншими платформами, що може бути менш привабливим для деяких користувачів.

1.3.3. Plan to Eat

Plan to Eat надає користувачам зручні інструменти для організації харчування та планування страв. Особливістю є можливості створення персоналізованих списків покупок на основі обраних рецептів, що допомагає організувати процес закупівель. Крім того, Plan to Eat дозволяє планувати меню на тиждень вперед, що спрощує контроль за харчуванням.

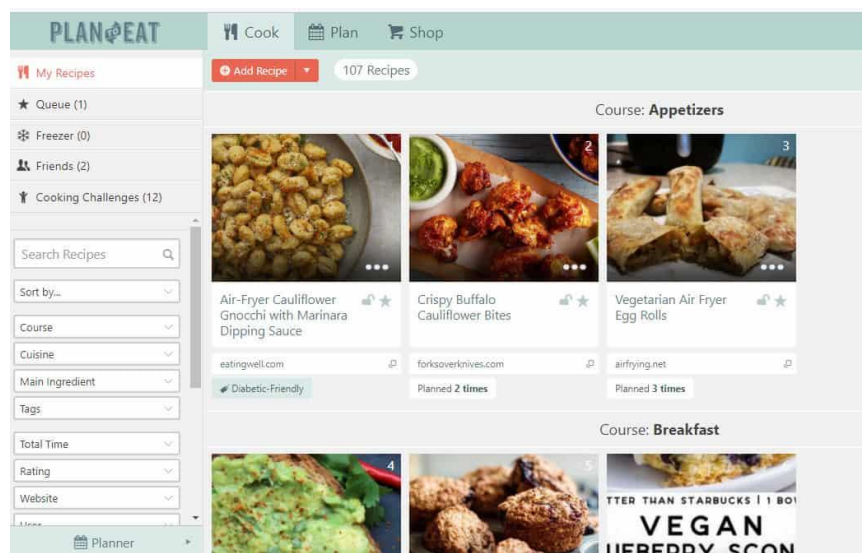


Рис. 1.3. Plan to Eat

Однією з переваг платформи є її спрощена інтерфейс та зручний процес навігації, що робить використання додатку приємним та зрозумілим для користувачів. Також, можливість спільного використання рецептів з родиною або друзями дозволяє створювати колективні списки покупок та планувати прийоми їжі разом.

Проте, деякі недоліки Plan to Eat включають обмежений функціонал у порівнянні з іншими платформами, наприклад, відсутність можливості ведення детального журналу харчування або розрахунку калорій. Також, деякі користувачі можуть відчувати нестачу деяких функцій, таких як автоматична синхронізація з іншими пристроями.

1.3.4. Lose It!

Особливістю додатку Lose It! є його функціонал, спрямований на контроль ваги та харчування. Користувачі можуть вести детальний журнал своєї їжі, встановлювати цілі калорій та вуглеводів, а також ведення тренувань. Додаток також відрізняється своєю активною спільнотою користувачів, яка надає підтримку та мотивацію для досягнення цілей.

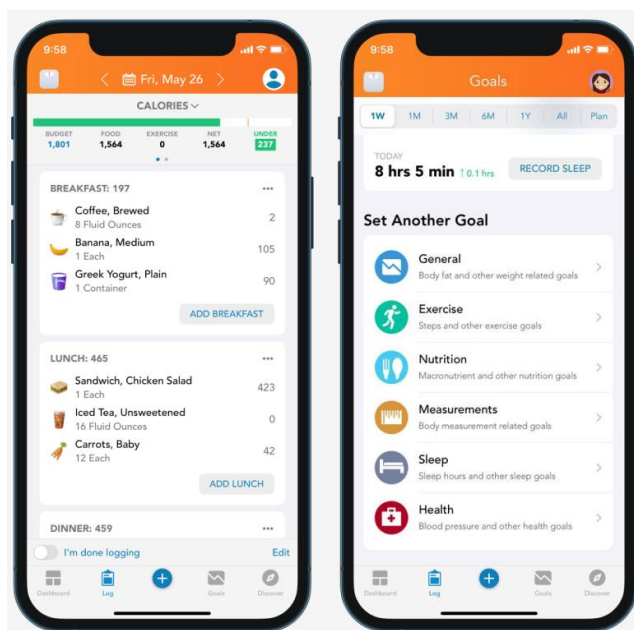


Рис. 1.4. Lose It!

Серед переваг Lose It! можна відзначити сучасний та зрозумілий інтерфейс, що робить використання додатку зручним для широкого кола користувачів. Також, наявність можливостей для ведення тренувань та фізичної активності дозволяє користувачам отримати комплексно підтримувати активний спосіб життя.

Проте, до недоліків Lose It! можна віднести обмежені функції для деяких категорій користувачів, наприклад, веганів, алергиків або осіб із специфічними харчовими вимогами.

1.4. Розробка алгоритму підбору страв

Для розробки алгоритму підбору страв, який враховує індивідуальні характеристики користувача, було проведено чотири основні етапи.

Перший етап полягає у зборі даних через анкетування, де користувачі вказують свої біометричні дані (вага, зріст і вік), рівень їх фізичної активності (сидяча робота, помірна активність чи регулярні тренування), цілі щодо ваги, а також інформацію про харчові переваги та непереносимості. Важливою частиною цього збору даних є визначення медичних обмежень, які мають значний вплив на підбір раціону харчування.

Для людей з діабетом рекомендується обмежити споживання цукрів до 5 грамів на порцію та вуглеводів до 20 грамів, оскільки вони швидко перетворюються на глюкозу. У разі гіпертонії користувачам необхідно зменшити споживання натрію, зокрема солі, до 0,5 грама на порцію. Для людей з серцево-судинними захворюваннями слід споживати страви, що сприяють зниженню рівня холестерину, з огляду на рекомендації щодо обмеження холестерину до 0,02 грама на порцію та жирів до 2 грамів. При хворобі нирок необхідно обмежити споживання білка до 10 грамів на порцію та солі до 0,2 грама.

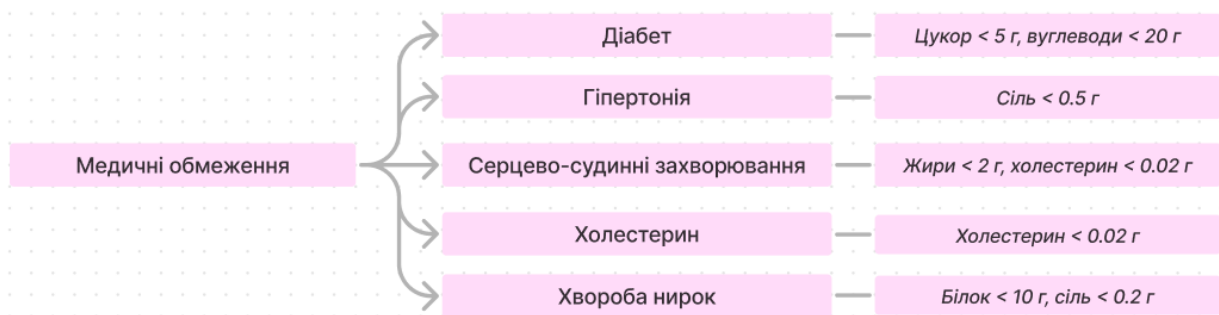


Рис. 1.5. Схема дієтичних обмежень за медичними станами

Ця інформація стане основою для наступних етапів, де буде розроблено логіку фільтрації страв з урахуванням зібраних даних.

Слід провести детальний аналіз щодо харчових непереносимостей користувачів. Це дозволить алгоритму точніше підібрати рецепти, які відповідають особливим дієтичним потребам користувачів. Для цього було виділено десять поширених категорій, що часто викликають алергію або непереносимість у багатьох людей. До цих категорій входять лактоза, глютен, горіхи, яйця, м'ясо, риба, морепродукти, солодоші, фруктоза та жирна їжа. Враховуючи цю інформацію, алгоритм буде виключати з рекомендованих страв ті, які містять інгредієнти, що можуть викликати алергічні реакції або які не входять до переліку переваг користувача.

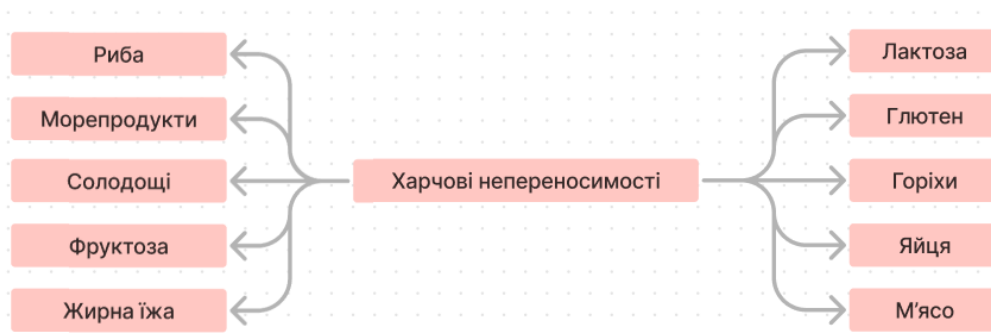


Рис. 1.6. Схема категорій харчових непереносимостей

Другим етапом є визначення енергетичних потреб — кількості енергії, необхідної людині для підтримки своїх життєвих функцій. Ця енергія вимірюється у кілокалоріях (ккал) і забезпечується через споживання їжі. Енергетична потреба змінюється залежно від віку, статі, ваги, зросту, рівня фізичної активності та цілей щодо ваги.

Базовий обмін речовин (BMR) — це кількість калорій, необхідних для підтримки базових життєвих функцій організму під час спокою, таких як дихання, контроль температури тіла, кровообіг, клітинний ріст тощо. Для розрахунку використовуються формули Mifflin-St Jeor або Harris-Benedict. Формула Mifflin-St Jeor, розроблена у 1990 році, вважається більш точною і часто використовується сучасними дієтологами та медичними спеціалістами. BMR розраховують за формулою для чоловіків (1) та жінок (2).

$$BMR = 10 \times m + 6.25 \times h - 5 \times a + 5, \quad (1)$$

$$BMR = 10 \times m + 6.25 \times h - 5 \times a - 161, \quad (2)$$

де m — вага, кг;

h — зріст, см;

a — вік, роки.

Розрахунок BMR дає приблизне значення калорій, які тіло споживає у спокійному стані. Для обчислення кількості калорій, необхідних для підтримки поточної ваги з урахуванням рівня активності, потрібно розрахувати TDEE (Total Daily Energy Expenditure), або сукупний добовий енерговитрат, який враховує всі види активності, від дихання та травлення до ходьби та спорту. Для розрахунку

TDEE використовують формулу (3).

$$TDEE = BMR \times C \quad (3)$$

де BMR – норма калорій у стані спокою, *ккал*;

C – множник (коефіцієнт) активності.

Множник активності визначається за рівнем фізичної активності:

- Сидячий (мало або без фізичної активності) – 1.2;
- Невелика активність (легкі вправи або спорт 1-3 дні на тиждень) – 1.375;
- Середня активність (вправи 3-5 днів на тиждень) – 1.55.

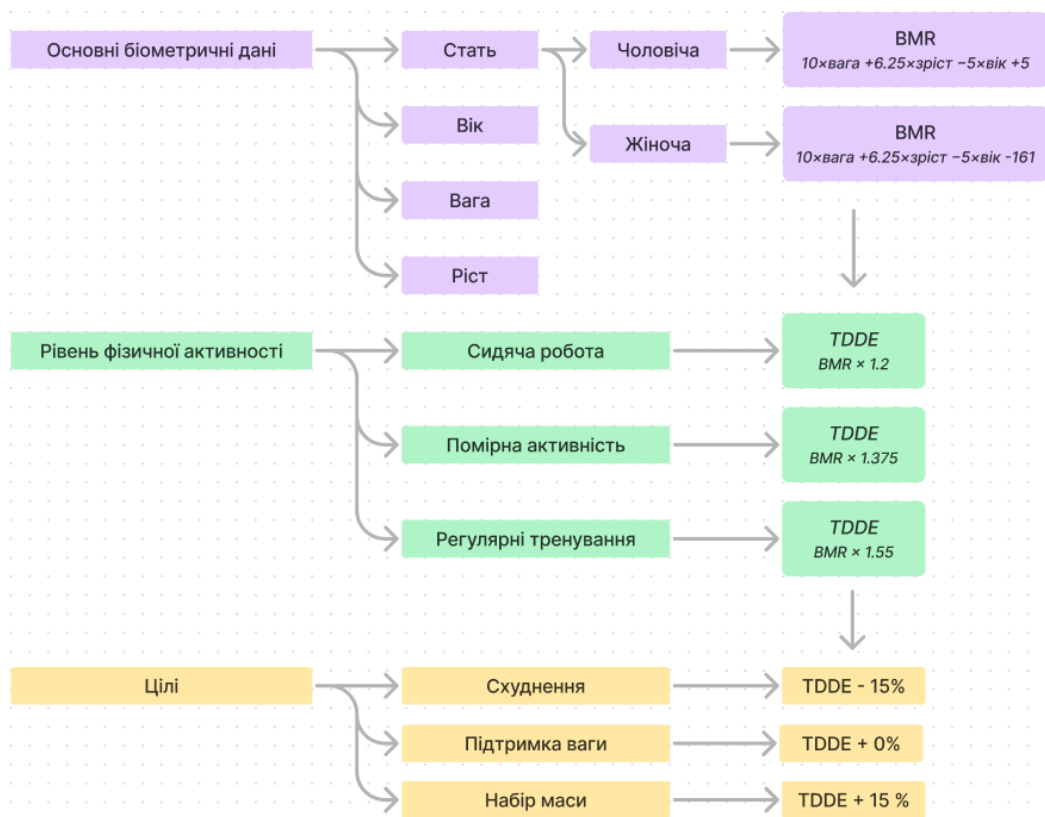


Рис. 1.7. Схема розрахунку енергетичної потреби

Підрахунок TDEE допоможе визначити загальну кількість калорій, які необхідно споживати щодня для підтримки поточної ваги. Якщо ж мета — схуднення або набір маси, потрібно відкоригувати калорійність відносно TDEE на 15%.

Третім етапом в розробці алгоритму є планування раціону, яке включає визначення загальної кількості калорій, необхідних користувачу протягом дня на

основі його сукупного добового енерговитрату (TDEE). Раціон поділяється на кілька прийомів їжі: зазвичай це три основні прийоми (сніданок, обід, вечеря) та два перекуси. Важливо забезпечити, що кожен прийом їжі містить збалансовану кількість білків, жирів та вуглеводів, які відповідають особистим дієтичним цілям користувача. Загальні дієтичні рекомендації для розподілу макронутрієнтів у добовому раціоні є наступними:

- Вуглеводи: 45% - 65% від загальної кількості калорій;
- Білки: 10% - 35% від загальної кількості калорій;
- Жири: 20% - 35% від загальної кількості калорій.

Оптимальне співвідношення макронутрієнтів важливе для підтримки метаболічного здоров'я, відчуття ситості та загального благополуччя.

Останній етап полягає у підборі страв, що починається з відфільтрування тих, які містять інгредієнти, що користувач не може або не хоче споживати, зокрема алергени або продукти, які викликають непереносимість. Наступний крок — забезпечення того, що кожна страва відповідає встановленому балансу макронутрієнтів, що дозволяє користувачу досягати своїх щоденних цілей, таких як певний відсоток калорій з білків, жирів та вуглеводів.

1.5. Аналіз потреб користувачів у веб-додатку

Перед тим як розпочати розробку веб-додатку для індивідуального планування раціону, важливо провести огляд та аналіз потреб цільової аудиторії. Один із ефективних методів для цього - створення Customer Person, або портрета користувача. Метод дозволяє визначити контекст, мотивацію та потреби користувачів, що допомагає зосередитися на найбільш важливих аспектах для їхнього задоволення. Результатом є більш реалістичне уявлення про те, що є значущим для користувачів та як ми можемо максимально відповісти їхнім потребам у майбутньому продукті.

Персона – це збірний образ живої людини, що містить типові характеристики нашого потенційного користувача. Відомо, що зазвичай створюють 3-5 персон, одна

з яких є домінуючою. Їхня основна роль полягає в об'єднанні основних потреб різних сегментів цільової аудиторії, тим самим створюючи чітку картину для команди розробників. Це дозволяє команді спрямовувати свої зусилля на створення продукту, який буде максимально відповідати потребам та очікуванням реальних користувачів.

У процесі дослідження було враховано ключові параметри користувачів, які використовують подібні додатки або веб-сайти. Особлива увага була зосереджена на віковій категорії, статі, професії, сімейному статусі, досвіді користування, цілях, мотивації та можливих бар'єрах. Аналіз цих факторів дозволив отримати глибше розуміння цільової аудиторії та їхніх потреб. За отриманими закономірностями було створено три портрети.

Марія - активна мама, що дбає про здоров'я своєї сім'ї та шукає ефективні рішення для планування харчування. Вона потребує підтримки у вигляді простих та швидких рецептів, організації закупівлі продуктів та зберігання улюблених страв. Планер харчування буде відмінним інструментом для систематизації цих процесів та покращення управління харчуванням сім'ї.

The image shows a user profile card for Maria. It includes a circular profile picture of a smiling woman with dark hair. To the right of the picture is a text box stating: "Раніше використовувала кулінарні книги та онлайн-рецепти для планування обідів, але часом не вистачало на пошук нових ідей". Below the picture, there are several fields: "Ім'я: **Марія**", "Вік: **35 років**", "Сімейний статус: **Заміжня, 2 дитини**", and "Професія: **Домогосподарка**". To the right of the profile information, there are three colored boxes: a white box with a red target icon and the text "Ціль: Оптимізувати планування їжі для всієї родини, знайти прості та смачні рецепти для швидкого приготування"; a green box with a thumbs-up icon and the text "Мотивація: Бажання забезпечити сім'ї здорове та смачне харчування, економія часу на приготуванні страв"; and an orange box with a barrier icon and the text "Бар'єри: Обмежений час на приготування, потреба в доступних та простих рецептах".

Рис. 1.8. Портрет користувача Марії

Наталія - молода жінка з активним спортивним життям, яка відчуває потребу в розрахунку калорій та оптимальному режимі харчування для досягнення своїх

спортивних цілей. Потребує зручного застосунку для планування їжі, швидкого підрахунку калорій та поживних речовин.

Користується спеціалізованими додатками для фітнесу та тренувань, але шукає інструмент для оптимального планування раціону

Ціль
Розрахунок калорій, забезпечення необхідною кількістю білків та вуглеводів для тренувань

Мотивація
Досягнення спортивних цілей, підтримка оптимального режиму харчування для покращення результатів тренувань

Бар'єри
Складність розрахунку калорій та поживних речовин, нестача часу на планування

Ім'я: **Наталія**
Вік: **28 років**
Сімейний статус: **Не заміжня**
Професія: **Фітнес-тренер, активний спортсмен**

Рис. 1.9. Портрет користувача Наталії

Ігор - веган, який зосереджений на своєму здоров'ї та дотримується етичних принципів харчування. Він активно шукає нові та різноманітні рецепти для збалансованого веганського раціону, але зазнає труднощів у знаходженні страв, які відповідають його потребам в поживних речовинах. Ігору необхідний кросплатформений додаток або веб-сайт, щоб зручно користуватися ним як на комп'ютері, так і на мобільному пристрої.

Користувався спеціалізованими кулінарними сайтами, книгами для веганів та був підписаний на різні сторінки у соціальних мережах. Необхідність спільного планера на комп'ютері та телефоні

Ціль
Знайти нові та цікаві рецепти, збалансований веганський раціон

Мотивація
Збереження здоров'я, підтримка етичного способу харчування

Бар'єри
Пошук нових та різноманітних веганських рецептів, відповідність раціону поживним речовинам

Ім'я: **Ігор**
Вік: **40 років**
Сімейний статус: **Не одружений**
Професія: **Інженер, веган**

Рис. 1.10. Портрет користувача Ігоря

Основні персони, які були розглянуті, – Марія, Наталія і Ігор, представляють різні групи користувачів зі своїми унікальними вимогами та очікуваннями від такого додатку. З урахуванням цих потреб можна сформулювати функціональні вимоги до нашого додатку, зокрема:

- База рецептів з можливістю фільтрації за типом харчування (веганське, спортивне, сімейне тощо), категорією, складністю приготування, тощо. Зберігання улюблених рецептів та можливість створення власних колекцій рецептів.
- Система нагадувань про приготування страв або покупки необхідних продуктів.
- Функція складання щоденного або тижневого раціону з можливістю вибору страв з бази рецептів. Розрахунок калорій та поживних речовин для кожного прийому їжі. Врахування індивідуальних вимог користувачів.
- Організація списку покупок.
- Можливість реєстрації та створення облікового запису для збереження персоналізованих налаштувань.
- Синхронізація даних між пристроями (комп'ютер, мобільний телефон) для зручного користування додатком у будь-який час та місце.

Такий підхід до аналізу потреб користувачів дозволяє зрозуміти, які конкретні інструменти та функції повинен надати веб-додаток для задоволення потреб цільової аудиторії та підвищення їхнього задоволення від користування продуктом.

1.6. Карта сайту майбутнього застосунку

Карта сайту або sitemap відіграє важливу роль у процесі проектування веб-сайтів. Цей інструмент надає можливість побачити майбутній проект в цілому і визначити оптимальне розташування контенту та логічну послідовність сторінок сайту, оцінити масштаб і складність проекту. Проаналізуємо, як відсутність коректної sitemap може вплинути на проект:

1. Необдумане планування може призвести до пропуску важливих сторінок, що може вплинути на вартість та терміни реалізації проєкту;
2. Втрати важливих навігаційних елементів, таких як кнопки заклику до дії та посилання, що з'єднують сторінки між собою, що ускладнить навігацію користувача;
3. Неясність загальної ієрархії сайту, що може вплинути на структуру та зручність навігації;
4. Неповнота зв'язків, що може призвести до неповного розуміння, які сторінки пов'язані між собою і яким чином.

Узагальному створення карти сайту починається зі збору і аналізу вихідної інформації про проєкт. Перш за все, визначаються основні цілі та завдання сайту, його цільова аудиторія та їх потреби. Потім визначається структура контенту, який буде присутній на сайті, та його ієрархія. Далі встановлюються зв'язки між сторінками, що допомагає визначити шляхи навігації для користувачів. Після визначення структури та зв'язків створюється сама карта сайту у вигляді діаграми. Діаграма візуалізує структуру та зв'язки між сторінками, допомагаючи зрозуміти загальний обсяг проєкту та логіку навігації.

Для побудови діаграми карти сайту можна використати платформу Figjam, яка є вбудованою в програму Figma. Figjam - це онлайн-інструмент, який надає можливість створювати різні типи діаграм та схем, включаючи карту сайту. Програма Figjam є дуже зручним інструментом для створення різних типів діаграм та схем, також є зручною для спільної роботи над проєктом, оскільки вона дозволяє багатьом користувачам одночасно працювати, коментувати та вносити зміни в реальному часі.

Для розробки карти сайту веб-застосунку для персоналізованого харчування виділимо сторінки сайту: головна, реєстрація, вхід, анкетування, профіль користувача, календар харчування, обрані рецепти, рецепт, список покупок. Шапка сайту (header) буде слугувати навігаційним меню для переміщення між сторінками сайту.

На рис. 1.11. наведено карту сайту для веб-ресурсу для створення персонального плану харчування, де зображено головну структуру, зв'язки та контент сторінок.

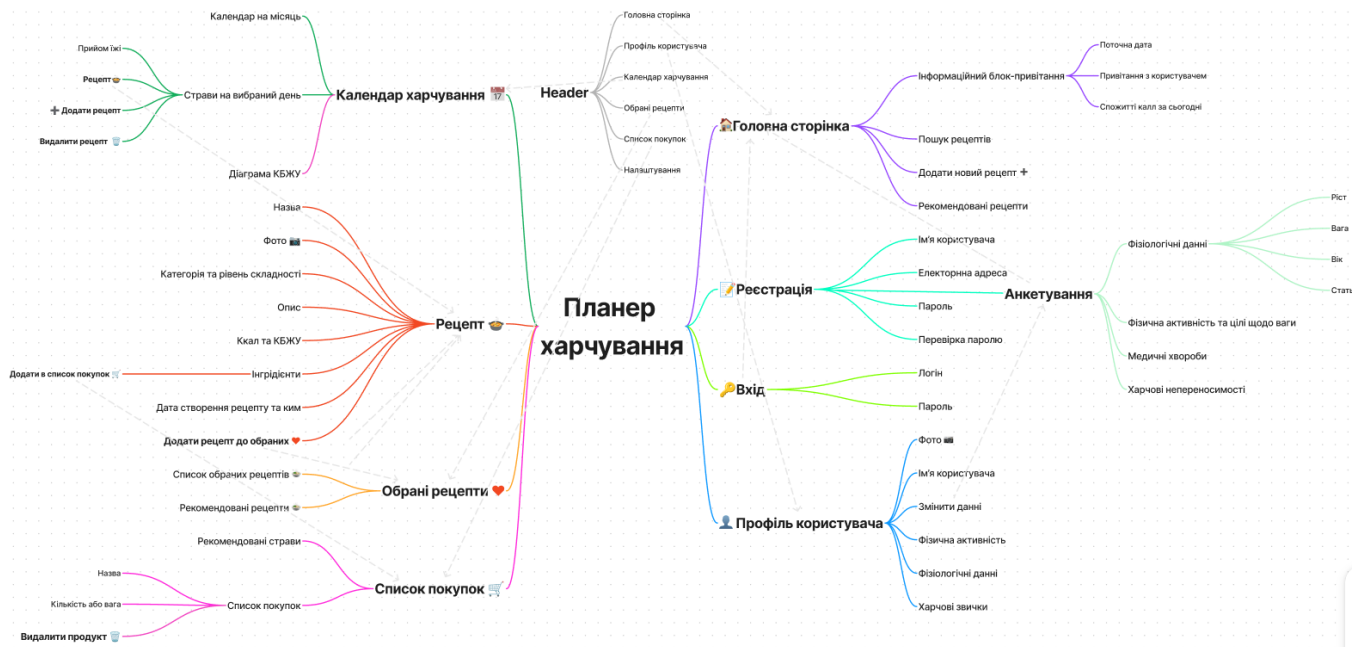


Рис. 1.11. Карта сайту

Створена діаграма дозволяє отримати чітку картину обсягу проекту, уникнути втрати важливих деталей та продумати ефективну, логічну та зручну навігацію для користувача та розробника. Отриману карту сайту можна використовувати на наступних етапах розробки веб-сайту, таких як прототипування та розробка.

1.7. Висновок до розділу 1

Перший розділ детально оглядає еволюцію підходів до здорового харчування та вплив сучасних технологій на формування індивідуалізованих харчових планів. Через аналіз існуючих платформ, таких як Apple Health, FatSecret, Plan to Eat, та Lose It!, було виявлено ключові потреби та вподобання споживачів, які включають персоналізацію на основі фізичної активності, харчових алергій, хронічних захворювань, вікових особливостей та особистих смаків.

Зі зростанням попиту на персоналізовані харчові рішення, важливість розробки веб-застосунків, які забезпечують гнучке управління дієтичними планами, стає все більш актуальною. Веб-застосунок для персоналізованого планування харчування включатиме створення інтуїтивно зрозумілого інтерфейсу, який підтримує візуалізацію даних та забезпечує легкість у використанні для користувачів.

Ключовим компонентом у розробці є проектування карти сайту, яке спрямоване на забезпечення логічної навігації та доступності всіх важливих функцій. Ефективне планування структури сайту є вирішальним для забезпечення зручності та ефективності веб-застосунку.

Таким чином, ретельний аналіз поточних трендів і вимог користувачів, а також інтеграція передових технологій у розробці веб-застосунку для персоналізованого харчування, покладає фундамент для успішного впровадження проекту, що має на меті підвищення якості життя та здоров'я користувачів.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

2.1. Вибір мов програмування

Вибір мов програмування є фундаментальним кроком у процесі розробки будь-якого програмного забезпечення. Він визначає основні інструменти та підходи, які будуть використовуватися в проєкті, та має значний вплив на його ефективність, гнучкість та легкість підтримки.

2.1.1. HTML

Кожна веб-сторінка, яку ми переглядаємо, є документом, написаним на HTML, що робить його однією з ключових технологій у галузі розробки. HTML, або HyperText Markup Language, є мовою гіпертекстових посилань, яка визначає структуру веб-сторінок і слугує каркасом для веб-сайтів. Використовуючи різноманітні елементи, HTML дозволяє створювати структуровані документи з текстами, зображеннями, посиланнями та формами. Завдяки тегам, таким як ``, HTML дозволяє детально описати вміст сторінки. Однак, для привабливого візуального оформлення, необхідний CSS, який додає стилістику і красу елементам HTML.

Особливе місце займає HTML5, найновіша версія HTML, яка розширила можливості мови, включаючи підтримку нативного відео та аудіо, а також розширені інтерактивні можливості, які раніше потребували зовнішніх плагінів. Оновлення значно розширило можливості веб-документів за рахунок введення нових елементів, таких як `<canvas>`, що дозволяє рисувати графіку безпосередньо

| | | | | | | | |
|---------------------|----------------------|--|--|--|---------------|--------------|----------------|
| Кафедра КІТ | | | | НАУ 24 34 71 000 ПЗ | | | |
| <i>Виконав</i> | <i>Козирева М.А.</i> | | | ПРОЕКТУВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ | <i>Літера</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| <i>Керівник</i> | <i>Шевченко О.П.</i> | | | | | 28 | 20 |
| <i>Консульт.</i> | | | | | УС-413 122 | | |
| <i>Норм. контр.</i> | <i>Шевченко О.П.</i> | | | | | | |

в браузері, або `<section>`, `<article>`, `<header>`, і `<footer>`, які сприяють кращій організації контенту. Крім того, підтримує веб-форми нового покоління, які забезпечують валідацію даних на стороні клієнта без залучення JavaScript. HTML5 версія значно поліпшила інтерактивність веб-сторінок.

Отже, HTML є фундаментальною мовою розмітки, що створює структурну основу для веб-сайтів, а введення HTML5 збільшило її функціональність, зробивши більш потужною.

2.1.2. CSS

CSS, або Cascading Style Sheets, слугує мовою каскадних таблиць стилів, яка дозволяє задавати візуальний вигляд HTML-документів. Вона надає розробникам засоби для керування розміщенням елементів, шрифтами, кольорами тексту та фону, значно полегшуючи управління презентацією веб-контенту. CSS використовується для відокремлення змісту веб-сторінок від їхнього стилю, що спрощує оновлення та обслуговування веб-сайтів.

Однією з ключових особливостей CSS є використання псевдо-класів, як-от `:hover` або `:focus`, що дозволяють змінювати стиль елементів при певних користувацьких взаємодіях, роблячи інтерфейс більш інтерактивним і зручним для користувачів. Медіа-запити у CSS забезпечують адаптивність веб-сайтів, дозволяючи сторінкам автоматично адаптуватися до різних умов відображення, зокрема до різних розмірів екранів мобільних пристроїв та десктопів.

В розробці веб-сайту для планування харчування використовувались сучасні CSS-технології, зокрема CSS Grid та Flexbox, що дозволяють ефективно керувати розташуванням елементів у макеті. CSS Grid ідеально підходить для створення складних двовимірних макетів, тоді як Flexbox оптимальний для одновимірних компоновань, що потребують гнучкого вирівнювання елементів.

В цілому, CSS є незамінним інструментом у сфері веб-розробки, що не тільки підвищує візуальну привабливість сторінок, але й розширює можливості для створення гнучких та ефективних веб-інтерфейсів.

2.1.3. JavaScript

JavaScript, мова програмування високого рівня, є ключовим інструментом для створення динамічних веб-сторінок, дозволяючи втілювати різноманітні ефекти, такі як анімації, вспливаючі форми та інтерактивні дії при натисненні на кнопки. Код не вимагає компіляції, оскільки браузері можуть виконувати його безпосередньо, що спрощує тестування та розгортання веб-додатків. Важливо зазначити, що, незважаючи на схожість назв, JavaScript не пов'язаний з Java.

JavaScript використовує скрипти або сценарії — набори інструкцій, що автоматично виконуються при завантаженні сторінки в браузері. Скрипти можуть бути вбудованими безпосередньо в HTML код сторінки або подані як окремі файлові сценарії.

JavaScript має кілька ключових особливостей, які роблять його незамінним у розробці веб-додатків. Він підтримує об'єктно-орієнтоване та функціональне програмування, працює на клієнтській стороні браузера, що забезпечує швидке реагування інтерфейсу без необхідності перезавантажувати веб-сторінку. Це асинхронна мова, яка може виконувати декілька завдань одночасно, завдяки чому веб-додатки працюють плавно та ефективно.

У розробці програми було використано AJAX-запити, які дозволяють веб-сторінкам асинхронно обмінюватися даними з сервером, вдосконалюючи взаємодію з користувачем шляхом оновлення тільки частини сторінки, замість повного її перезавантаження. Також були застосовані бібліотеки jQuery та jQuery UI, які спростили процес написання коду, додавши зручні функції для обробки подій, а також створення інтерактивних компонентів інтерфейсу, таких як випадаючі списки, календар, і таби.

Підсумовуючи, JavaScript є мовою з власними унікальними властивостями та синтаксисом, призначеною для вирішення специфічних задач у сфері веб-розробки. Особливо цінною вона стає при розробці інтерактивних веб-додатків, що вимагають динамічної взаємодії з користувачем.

2.1.4. PHP

PHP – це серверна мова, яка особливо підходить для розробки веб-додатків і може вбудовуватися прямо в HTML. PHP використовується для генерації динамічного контенту на сторінках, взаємодії з базами даних та обробки даних форм. Завдяки своїм можливостям, PHP забезпечує високу гнучкість та ефективність в обробці даних на серверному боці, роблячи його ідеальним вибором для розробки складних веб-додатків.

PHP відрізняється тим, що виконується на сервері, що дозволяє генерувати HTML на стороні сервера перед тим, як він буде відправлений до користувача. Це дозволяє забезпечити значний рівень безпеки, оскільки код PHP не передається до клієнта і таким чином залишається невидимим для кінцевого користувача.

Основною перевагою PHP є його здатність інтегруватися з багатьма популярними базами даних, такими як MySQL, Oracle та PostgreSQL, а також підтримка широкого спектра веб-технологій, що дозволяє розробникам створювати розширені веб-додатки з потужною функціональністю. PHP також підтримує велику кількість фреймворків, які можуть полегшити і прискорити процес розробки, надаючи вже готові рішення для багатьох завдань.

У розробці веб-додатку для планування харчування PHP використовується для створення серверних скриптів, що обробляють запити користувачів, забезпечують зв'язок з базою даних і відправляють відповіді назад на клієнтську сторону. Використання PHP дозволяє ефективно керувати сесіями користувачів, зберігати та відновлювати дані користувача, страв, планів харчування і взаємодіяти з іншими системами в реальному часі.

2.2. VS Code як середовища розробки

Visual Studio Code є інтегрованим середовищем розробки програмного забезпечення, розробленим компанією Microsoft. Це середовище дозволяє створювати консольні додатки, веб-сайти, веб-додатки, додатки з графічним

інтерфейсом та веб-сервіси для різних платформ. Його популярність зумовлена підтримкою широкого спектру мов програмування та можливостями розширеного функціоналу завдяки плагінам, доступним у магазині з розширень. Visual Studio Code активно використовується веб-розробниками, що працюють з HTML, CSS, JavaScript, PHP, а також підтримується багато інших мов програмування. Крім того, VS Code добре взаємодіє з різними фреймворками.

Важливо відрізнити VS Code від Visual Studio. Ці назви та інтерфейси схожі, але це різні програми. Visual Studio - це інтегроване середовище розробки (IDE), яке є досить масштабним та потужним. У той час, як VS Code - це легковагове середовище, простіше в освоєнні та популярне серед початківців програмістів, використовується там, де не потрібна повна потужність IDE.

Недоліки Visual Studio Code також варто розглянути. До них можна віднести обмежену функціональність порівняно з важкими IDE, особливо коли йдеться про великі проекти з складними налаштуваннями та інтеграцією з іншими інструментами. Деякі функціональності, які можуть бути важливі для деяких проектів, можуть бути менш розвиненими або не такими зручними.

Основними перевагами середовища розробки VS Code є безкоштовність, вбудовані інструменти для редагування та форматування коду, підтримка синтаксичної підсвітки, що дозволяє автоматично виправляти помилки, можливості рефакторингу коду, керування версіями та вбудований Git. Крім того, важливим фактором є те, що програма є кросплатформною.

Ефективне написання коду потребує не лише гарного розуміння мов програмування, а й навичок ефективного використання функціоналу редактора розробки для прискорення процесу та покращення якості коду.

Підходи та стратегії у VS Code для збільшення продуктивності роботи програміста:

- Розділ коду а модулі, робота з Git для сумісної роботи над проектом та контролю версіями;
- Налаштування середовища розробки, плагінів та розширень для оптимізації робочого процесу;

- Використання сніпсетів(кодових фрагменті) коду для швидкого написання повторюваних конструкцій;
- Для збільшення якості кода та усунення помилок використання відкладаючих інструментів та рефакторинга.

Visual Studio Code - це доволі функціональне та універсальне інтегроване середовище розробки, яке знаходить своє застосування у широкому колі завдань від веб-розробки до створення додатків з графічним інтерфейсом. Воно має свої переваги, такі як безкоштовність, кросплатформенність та розширена функціональність завдяки плагінам, але також має свої обмеження, особливо для великих та складних проєктів.

2.3. Управління базами даних

2.3.1. Вибір системи управління базами даних

При проектуванні веб-додатку одним із ключових технічних рішень є вибір системи управління базами даних (СУБД). Вона відіграє важливу роль у забезпеченні доступу, зберігання, організації та управління даними, які використовуються додатком. Також СУБД забезпечує резервне копіювання та гарантує безпеку інформації. Така система надає змогу працювати водночас декільком користувачам з однією базою даних.

При виборі СУБД важливо враховувати декілька критеріїв:

- Бюджетні обмеження проєкту, що вимагають вибору економічно ефективного рішення;
- Тип проєкту та обсяг зберігаємих даних, які впливають на вибір між легковаговими та високопродуктивними системами;
- Сумісність з використовуваними технологіями та мовами програмування, що забезпечує безпроблемну інтеграцію з іншими компонентами проєкту;
- Масштабованість, яка є важливою для підтримки зростання обсягу даних та кількості користувачів;

- Продуктивність обробки запитів та транзакцій для швидкої та надійної роботи системи;
- Безпека та конфіденційність, щоб гарантувати захист даних користувачів.

При виборі СУБД розробники надають перевагу найбільш популярним та перевірним системам, які себе вже зарекомендували на ринку, такими як MySQL, PostgreSQL, Microsoft SQL Server та SQLite.

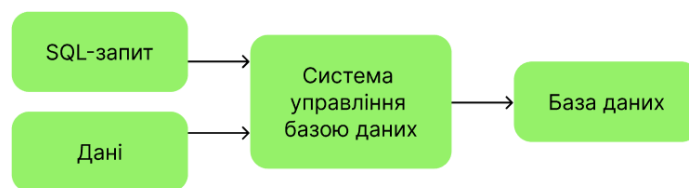


Рис. 2.1. Схема взаємодії SQL-запитів з СУБД

MySQL відома як широко використовувана відкрита реляційна база даних, славиться своєю надійністю та високою швидкістю обробки даних. Ця СУБД підходить як для масштабних проектів, так і для менших застосунків завдяки своїй гнучкості та ефективності.

PostgreSQL є потужною відкритою реляційною системою, яка пропонує розширені можливості порівняно з MySQL, включаючи підтримку складних типів даних. Відома своєю високою продуктивністю та масштабованістю, PostgreSQL є відмінним вибором для проектів, що вимагають складних запитів та великих обсягів даних.

Microsoft SQL Server вирізняється як комплексна СУБД, що вимагає більших витрат порівняно з відкритими альтернативами. Вона пропонує інтегровані бізнес-інструменти та графічний інтерфейс для адміністрування, що робить її особливо привабливою для використання у корпоративному секторі.

SQLite — це компактна та вбудована СУБД, яка забезпечує ефективне зберігання даних для додатків без потреби у складній інфраструктурі баз даних. Її легкість і простота роблять SQLite ідеальним рішенням для малих проектів або

мобільних застосунків, де потрібно швидко та без зайвих витрат забезпечити надійне зберігання даних.

Для проекту обрано СУБД MySQL, вибір якої зумовлений її високою продуктивністю, надійною підтримкою, гарною масштабованістю, та доступною вартістю. MySQL відмінно інтегрується з PHP, основною мовою програмування нашого проекту, що сприяє плавному розвитку та підтримці проекту. PhpMyAdmin, інструмент для управління MySQL, спрощує роботу з базою даних, оптимізуючи процеси розробки за рахунок зручного адміністрування.

PhpMyAdmin дозволяє адмініструвати сервер MySQL, навіть якщо він розташований на віддаленому комп'ютері, що важливо для команд, які працюють віддалено або в розподіленому середовищі. Цей інструмент надає гнучкість управління базами даних дистанційно, виконуючи різноманітні операції, включно з модифікацією, створенням та видаленням баз даних.

Однак, попри значні переваги, PhpMyAdmin має й обмеження. На локальному сервері цей інструмент зазвичай налаштовується з повними адміністративними правами, але у продуктивному середовищі доступ часто обмежується, щоб забезпечити безпеку та обмежити ризик несанкціонованого доступу до чутливої інформації. Це може ускладнити доступ до повного спектру функціональності для деяких користувачів. Також PhpMyAdmin вимагає роботи на веб-сервері та PHP, що може додавати додаткове навантаження на серверні ресурси.

Враховуючи ці аспекти, вибір MySQL та PhpMyAdmin для управління базами даних проекту є виправданим, оскільки переваги перевищують можливі недоліки.

2.3.2. Конфігурація локального серверу з використанням MAMP

Конфігурація локального серверу дозволяє розробникам тестувати та розгортати програми в контрольованому, ізольованому середовищі перед запуском у продуктивне середовище. MAMP – це інструмент для налаштування локального серверу, забезпечуючи простий і зручний спосіб запуску серверів Apache, MySQL та PHP на локальних машинах.

Абревіатура відображає складові програмного пакету, складається з наступних компонентів:

- М - Macintosh, що вказує на операційну систему Mac OS, хоча MAMP також доступний для Windows як WAMP та для Linux як LAMP;
- А - Apache, веб-сервер, який обробляє запити від браузера і відправляє їм відповідь у вигляді веб-сторінок;
- М - MySQL, система управління базами даних, яка зберігає і організовує дані, які використовуються веб-додатками;
- Р – PHP, мова програмування, зазвичай використовується для розробки серверної частини веб-додатків. PHP може також взаємодіяти з базами даних для витягу або зміни даних на запит веб-користувача.

При налаштуванні MAMP, процес розпочинається з встановлення програми на вибрану операційну систему. Завантажити файл потрібно з офіційного сайту та провести стандартну процедуру встановлення. Після інсталяції розробники мають можливість налаштувати сервери Apache, MySQL та PHP через інтуїтивно зрозумілий інтерфейс MAMP, що дозволяє змінювати налаштування портів, вибирати версії PHP та конфігурувати сервер баз даних.

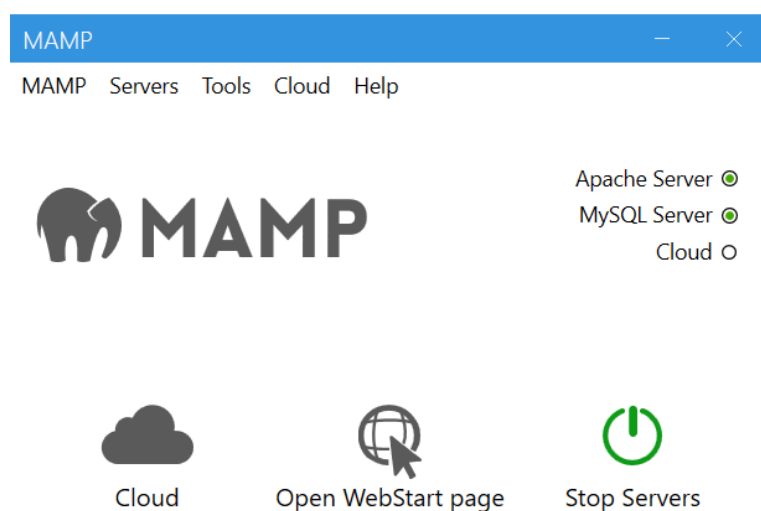


Рис. 2.2. Інтерфейс програми MAMP

Проекти розміщуються у спеціальній папці, зазвичай позначеної як «htdocs», звідки вони легко доступні через локальний веб-браузер. Розробники мають можливість запускати та тестувати веб-додатки локально, забезпечуючи можливість перевірки функціональності, відлагодження та продуктивності перед офіційним запуском.

МAMP надає все необхідне середовище для локальної розробки веб-додатків, забезпечуючи простий спосіб налаштування сервера, бази даних та інтерпретатора PHP.

2.3.3. Розробка схеми бази даних і використання реляційної моделі

Реляційна модель дозволяє зберігати дані у вигляді таблиць. Кожна таблиця складається з рядків (записів) і стовпців (атрибутів), які описують властивості об'єктів сутності. Реляційна модель забезпечує зв'язки між даними, що значно спрощує процеси введення, зберігання та обробки інформації. Модель є ідеальною для вирішення широкого спектру завдань, адже вона підтримує відносини між різними сутностями через використання первинних (Primary Key) та зовнішніх ключів (Foreign Key).

Система бази даних для веб-додатку для планування персоналізованого харчування містить шість основних сутностей: Users, Meals, DailyPlan, Shoppinglist, Ingredients і FavoriteMeal.

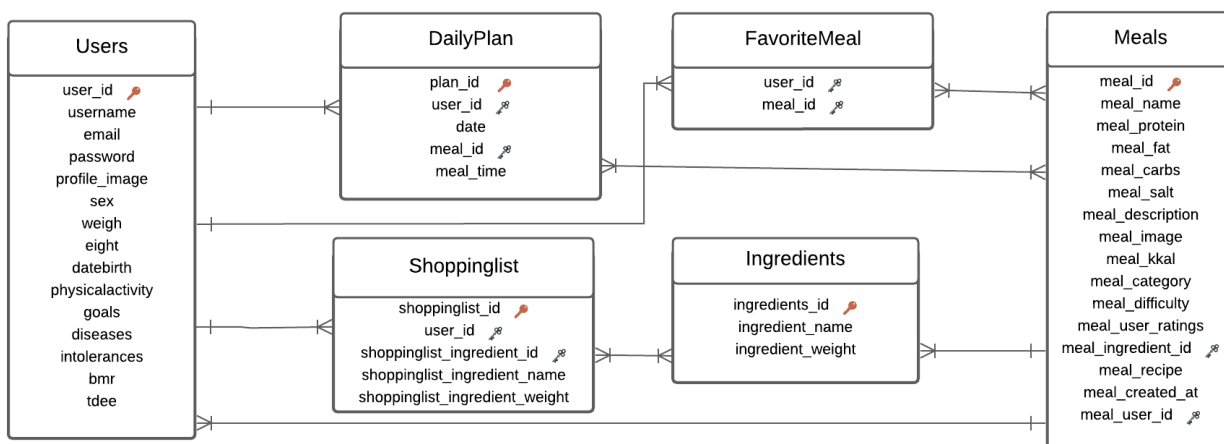


Рис. 2.3. E-R діаграма

User – таблиця, яка зберігає основну інформацію про користувачів, включаючи унікальний ідентифікатор користувача (user_id), ім'я (username), електронну адресу (email), зашифрований пароль (password), а також інші особисті дані, такі як стать (sex), вага (weigh), зріст (eight), дата народження (datebirth), рівень фізичної активності (physicalactivity), цілі (goals), хвороби (diseases), непереносимості (intolerances), базовий метаболічний рейт (bmr) і загальна денна витрата енергії (tdee).

Meals – містить інформацію про страви, включаючи їх ідентифікатори (meal_id), назви (meal_name), кількість білків (meal_protein), жирів (meal_fat), вуглеводів (meal_carbs), солі (meal_salt), описи (meal_description), зображення (meal_image), калорійність (meal_kkal), категорії (meal_category), складність приготування (meal_difficulty), оцінки користувачів (meal_user_ratings), ідентифікатор списку інгредієнтів (meal_ingredient_id), що дозволяє зв'язати інгредієнти із конкретними стравами, рецепти (meal_recipe) та дати створення (meal_created_at).

DailyPlan - таблиця для зберігання плану харчування користувачів на конкретний день, включаючи ідентифікатор плану (plan_id), ідентифікатор користувача (user_id), який вказує на власника плану, дату (date), ідентифікатор страви (meal_id) і час прийому їжі (meal_time).

Shoppinglist – таблиця, яка функціонує як перелік покупок, дозволяючи користувачам вести список потрібних для приготування страв інгредієнтів. Вона включає унікальний ідентифікатор списку (shoppinglist_id), ідентифікатор користувача (user_id), який вказує на власника списку, назву інгредієнта (shoppinglist_ingredient_name), та його вагу (shoppinglist_ingredient_weight).

Ingredients – це таблиця, що зберігає детальну інформацію про інгредієнти, які входять до складу страв. Вона включає ідентифікатор інгредієнта (ingredients_id), назву інгредієнта (ingredient_name) та вагу (ingredient_weight), забезпечуючи точність у складанні рецептів.

FavoriteMeal – таблиця, що містить улюблені страви користувача, для цього зберігає індифікатор користувача (user_id) та індифікатор страви (meal_id).

Ключові елементи реляційної моделі включають первинні ключі (Primary Keys), що гарантують унікальність записів у таблицях, та зовнішні ключі (Foreign Keys), які встановлюють зв'язки між таблицями, забезпечуючи цілісність даних. Наприклад, поля `user_id` у таблицях `Meals`, `DailyPlan`, `Shoppinglist`, і `FavoriteMeal` є зовнішніми ключами, які посилаються на таблицю `Users`, що дозволяє відслідковувати, який користувач додав певну страву, планує своє харчування, веде список покупок, чи зберігає улюблені страви.

Зв'язки між таблицями є фундаментальними для реалізації багатофункціонального веб-додатку, який підтримує персоналізоване харчування. Реляційна модель бази даних не тільки дозволяє ефективно зберігати великі обсяги інформації, але й забезпечує гнучкість у виконанні запитів і аналізі даних, що є критично важливим для динамічного і адаптивного сайту.

Використання реляційної моделі в даному веб-додатку сприяє оптимізації процесів збору, обробки та відображення даних, що зробило цю технологію ідеальним вибором для проекту. Розуміння залежностей між різними сутностями та їх атрибутами, також забезпечує високий рівень інтеграції та взаємодії компонентів системи.

2.4. Проектування користувацького інтерфейсу

2.4.1. Принципи створення інтуїтивно зрозумілого UI/UX

UI/UX дизайн є критичним аспектом будь-якої цифрової продукції, який забезпечує взаємодію користувача з програмним рішенням. Він орієнтований на зручність взаємодії, що може включати різноманітні платформи – від мобільних додатків до веб-сайтів. Хоч UI/UX дизайнери зараз вважаються представниками порівняно нової професії, коріння цих практик можна простежити далеко в історію. Яскравий приклад — Вільгельм Шиккард, який у 1623 році, розробляючи арифмомент, виконував основні завдання сучасного UX/UI дизайнера, проектуючи інтерфейс для зручної взаємодії людини з машиною..

User Experience, або UX, це комплекс вражень користувача, які виникають під час взаємодії з продуктом. Задачі UX дизайну включають аналіз потреб користувачів, тестування інтерфейсів на зручність, структурування інформації та розробку візуальної схеми взаємодії, що включає прототипування та тестування різних сценаріїв користування.

User Interface, або UI, визначає візуальні аспекти інтерфейсу, такі як вибір кольорів, шрифтів, розташування кнопок та інших елементів управління. UI дизайн спрямований на створення візуально привабливого та інтуїтивно зрозумілого інтерфейсу, що сприяє легкому доступу до функціоналу продукту.

Загалом, мета UI/UX дизайну полягає в тому, щоб зробити користувацький досвід якомога більш приємним та ефективним, мінімізувати можливі перешкоди та забезпечити легкий доступ до необхідних функцій, сприяючи таким чином задоволенню потреб користувачів. Важливість UI/UX дизайну не можна недооцінювати, адже він впливає на загальну якість користувацького досвіду та задоволеність продуктом.

2.4.2. Використання Figma для дизайну макетів

Використання Figma для дизайну макетів стало надзвичайно актуальним у сучасному світі цифрових продуктів, оскільки дозволяє дизайнерам швидко створювати та інтегрувати макети й прототипи. Що є критично важливим для розробки інтерфейсів, що відповідають як потребам користувачів, так і динамічним вимогам ринку. Figma, як хмарний інструмент для дизайну, забезпечує безпроблемну взаємодію команди над спільним проектом, дозволяючи всім членам команди вносити зміни та переглядати оновлення в реальному часі.

Програма пропонує інтуїтивно зрозумілий інтерфейс, що робить її доступною навіть для новачків у дизайні, при цьому не поступаючись глибиною функціоналу для досвідчених професіоналів. Основні функції включають векторне редагування, систему компонентів, що дозволяє повторно використовувати елементи дизайну по всьому проекту, і потужні інструменти для прототипування, які допомагають

створювати інтерактивні макети з анімаційними переходами.

Переваги Figma множинні. По-перше, вона підтримує колаборативну роботу в реальному часі, що є величезним плюсом для великих команд і розподілених проєктів. По-друге, оскільки Figma базується на хмарних технологіях, користувачі можуть працювати з будь-якого пристрою, що має підключення до інтернету, не хвилюючись про синхронізацію файлів чи версій проєктів. По-третє, інтеграція з іншими інструментами та сервісами, як Slack або JIRA, спрощує комунікацію та управління проєктами.

Однак, Figma не позбавлена недоліків. Наприклад, вона вимагає стабільного інтернет-з'єднання для оптимальної роботи, що може стати проблемою в областях з поганим покриттям. Ще одним мінусом є обмежена функціональність офлайн-режиму, що може ускладнити роботу під час подорожей або у віддалених локаціях. Хоча Figma пропонує безкоштовний план, повнофункціональне використання сервісу вимагає підписки, що може бути додатковим фінансовим навантаженням.

Таким чином, вибір Figma для дизайну макетів забезпечує потужний інструментарій, який допомагає визначати, тестувати та поліпшувати користувацький досвід, в той же час надаючи гнучкість і масштабованість для проєктів будь-якого розміру.

2.4.3. Прототип проєкту

Прототипування — це етап у процесі розробки веб-додатків та інших інтерфейсів, на якому створюється початкова модель продукту з метою тестування концепцій та інтерактивності перед розробкою кінцевого продукту. Це дозволяє дизайнерам та розробникам візуалізувати ідеї та експериментувати з лейаутами, потоками взаємодій та візуальними елементами.

Головна мета прототипування полягає в тому, щоб швидко створити версію продукту, яка може бути представлена користувачам та стейкхолдерам для збору зворотного зв'язку, який може бути використаний для покращення та доопрацювання продукту. Прототип може бути дуже простим, наприклад,

паперовим макетом, або більш складним цифровим інтерфейсом, який імітує взаємодію користувача з програмним забезпеченням.

В якості інструменту для створення прототипів було обрано програму Figma. На прикладі розробленого прототипу, можна побачити, як елементи інтерфейсу впорядковані для забезпечення зручності та зрозумілої навігації. Головна сторінка включає основні функціональні зони такі як пошук рецептів, рекомендації, панель управління харчуванням та доступ до персональних налаштувань. Центральна частина прототипу відображає інформацію про споживані калорії, що дозволяє користувачам легко відслідковувати свій щоденний прогрес.

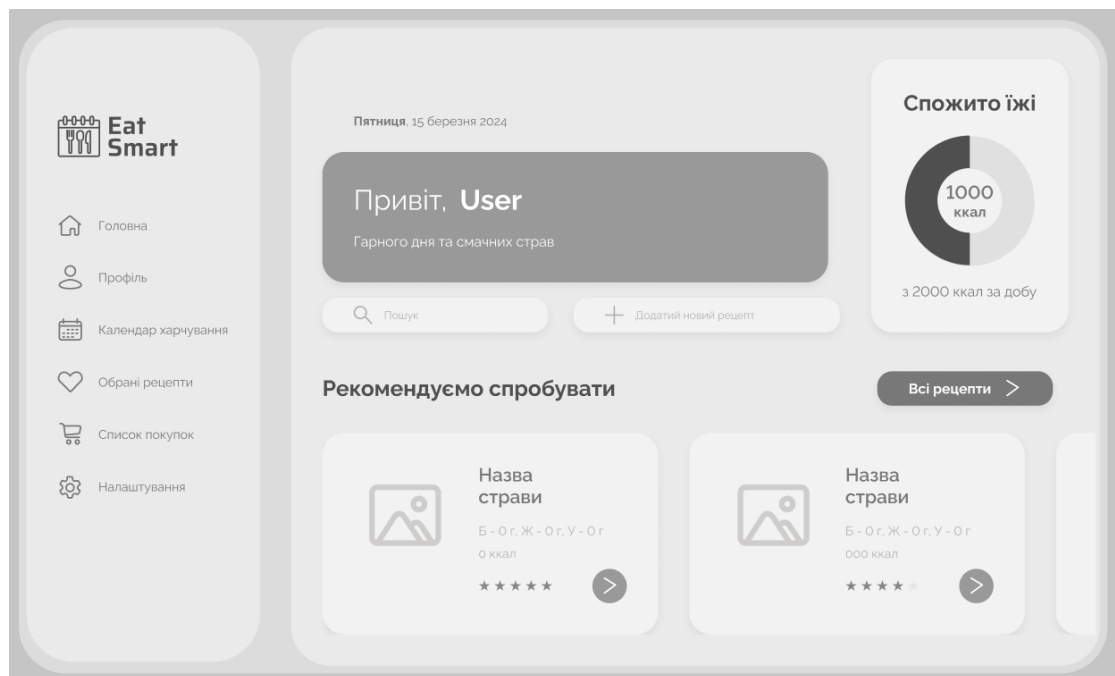


Рис. 2.4. Прототип головної сторінки

Прототипування допомагає виявити та виправити помилки на ранніх стадіях розробки, економить час та ресурси, які могли б бути витрачені на написання коду для функцій, що не відповідають потребам користувачів. Воно також сприяє кращому розумінню проектних вимог та оптимізації користувацького досвіду перед тим, як продукт вийде на ринок.

2.4.4. Проектування візуального стилю

У сучасному світі, де інформаційний шум досягає неймовірних розмірів, створення унікального та впізнаваного стилю стає ключовим фактором для виживання та успіху брендів і продуктів. Добре продуманий стиль допомагає користувачам швидше ідентифікувати продукт серед конкурентів, створюючи стійкі асоціації та емоційне залучення. Особливо стає важливим в цифровому просторі, де перше враження може вирішувати, чи затримається користувач на сайті чи додатку.

Витончений візуальний стиль сприяє збільшенню довіри до бренду. Користувачі, як правило, вважають веб-сайти, що виглядають професійно і зі смаком, більш надійними та безпечними для здійснення покупок або інших видів взаємодії.

Стиль у веб-дизайні — це візуальне відображення конкретного образу за допомогою спеціальних прийомів, які відповідають певному стилю. Стилi у веб-дизайні складаються з 4 основних складових: композиція, колір, типографіка та графіка. Веб-дизайн постійно розвивається, при цьому деякі стилі міцно закріплюються в трендах, а інші зникають.

Стиль

Неоморфізм є стилем, що базується на використанні внутрішніх та зовнішніх тіней, які створюють ефект м'яких, випуклих форм. Це надає об'єктам видимості об'ємності та реалістичності, додаючи інтерфейсу візуальної глибини без застосування чітких контурів, оскільки окреслення формується через м'яке тінювання. Назва стилю походить від поєднання слів «новий» та «скевоморфізм». Хоча неоморфізм і підкреслює футуристичність та унікальність, він може створювати складнощі з інтуїтивністю інтерфейсу, адже через відсутність чітких контурів важко відрізнити активні елементи управління від декоративних.

Матеріальний дизайн, створений командою розробників Google у 2014 році, це фірмовий стиль, який чітко визначає принципи його застосування. Він поєднує в собі елементи плоского дизайну та скевоморфізму, створюючи баланс між простотою та інтуїтивністю. Матеріальний дизайн, використовує яскраві кольори,

градієнти та багатошаровість для створення глибини. Головна ідея полягає у використанні метафори шарів паперу, які відкидають тіні один на одного, допомагаючи користувачам краще розуміти структуру інтерфейсу та принципи взаємодії з ним.

Інтеграція неоморфізму та матеріального дизайну, щоб створити збалансований і візуально привабливий інтерфейс. Використання неоморфізму дозволило наділити елементи управління, такі як кнопки та перемикачі, виглядом випуклості та об'ємності завдяки м'якому тінюванню, що надало інтерфейсу глибини та тактильності. Застосування принципів матеріального дизайну, з іншого боку, забезпечило більшу структурованість та зрозумілість інтерфейсу. Шляхом впровадження яскравих кольорів та градієнтів було досягнуто чіткості візуальної ієрархії, що допомогло підкреслити значимість ключових елементів управління. Крім того, використання шарів паперу з напівпрозорими кольорами, які відкидають тіні один на одного, надало додаткову глибину дизайну, полегшуючи користувачам орієнтування у взаємодіях з веб-застосунком.

Колористика

Колористика значно впливають на емоційний стан користувачів і можуть підсилювати чи послаблювати певні елементи інтерфейсу. Для проекту персоналізованого планера харчування була обрана колірна палітра у відтінках зеленого, що відповідає основній ідеї застосунку — підтримці здорового харчування.

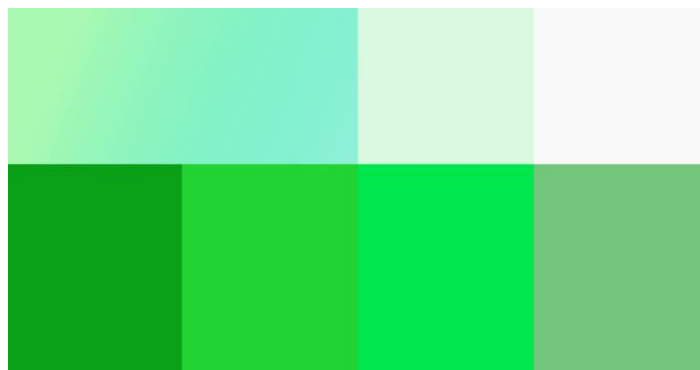


Рис. 2.5. Колористика

Використання насичених зелених відтінків асоціюються з здоров'ям, життєздатністю та енергією, роблячи їх ідеальними для акцентування важливих елементів навігації та інтерактивних кнопок. Водночас, більш м'які та світлі відтінки зеленого та білого створюють спокійний фон, що сприяє концентрації та не відволікає від основного контенту. Напівпрозорі зелені та білі відтінки створюють відчуття спокою та релаксації, ідеально підходячи для фонів та інших елементів, які не повинні відволікати користувача. Такі кольори можуть створити атмосферу гармонії, балансу, легкості та простору.

Фоновим кольором було обрано градієнт від салатого до блакитного, який створює візуально заспокійливий і охолоджувальний ефект. Такий перехід кольорів допомагає знижувати візуальне навантаження, оскільки м'які відтінки блакитного та зеленого легко сприймаються очима. Градієнт також символізує природне середовище, нагадуючи про зелені поля та чисте блакитне небо, що може позитивно вплинути на настрій користувачів, стимулюючи їх взаємодію з інтерфейсом на більш глибокому емоційному рівні.

Типографіка

Raleway був обраний за основний шрифт дизайну веб-додатку через його сучасний та стильний вигляд, що ідеально підходить для інтерфейсу спрямованого на мінімалізм та функціональність. Цей шрифт має геометричні пропорції та оптимізовану ширину символів, що забезпечує високу читабельність на екранах різних розмірів, від мобільних телефонів до десктопів.

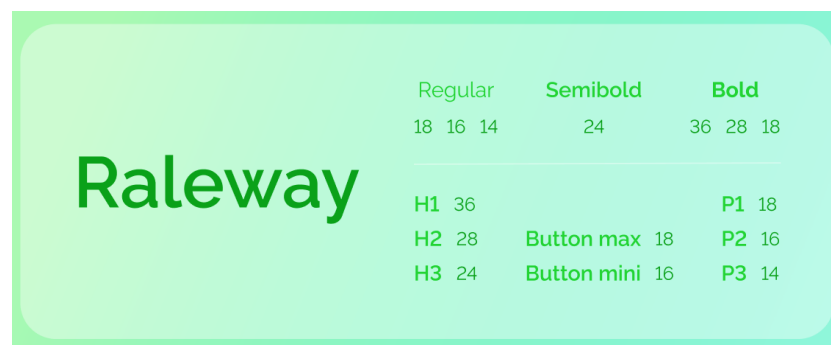


Рис. 2.6. Типографіка

Легкість читання *Raleway* пояснюється його відкритими формами букв та їх рівномірним розподілом, який знижує візуальну навантаження на очі користувачів під час читання. Шрифт також має характерні особливості, такі як злегка закруглені закінчення, які додають м'якості та гармонії тексту.

Використання різних товщин шрифту дозволяє ефективно використовувати типографіку для визначення ієрархії інформації. Наприклад, більш товсті та виділені заголовки привертають увагу користувача, водночас підкреслюючи ключові моменти контенту. Тонші версії шрифту використовуються для основного тексту, що сприяє легшому сприйняттю менш важливої інформації.

Визначальна роль візуального стилю в процесі створення інтерфейсу підкреслюється через детальний аналіз компонентів, таких як колористика, типографіка та застосування конкретних дизайнерських підходів, таких як неоморфізм і матеріальний дизайн, що дозволяє виділитися серед конкурентів та привернути більше користувачів до програмного продукту.

2.5. Висновок до розділу 2

Висновок другого розділу дипломної роботи акцентує на критичному значенні обережного вибору технологій та методів проектування для створення веб-додатку персоналізованого планування харчування. Ефективна розробка проекту була досягнута завдяки вдалому вибору таких мов програмування, як HTML, CSS, JavaScript, і PHP. Застосування Visual Studio Code як середовища розробки забезпечило комплексну підтримку потрібних мов та бібліотек.

Детальне проектування бази даних з використанням реляційної моделі гарантувало її надійність, безпеку та оперативність обробки даних. Використання MAMP допомогло налаштувати локальний сервер, забезпечуючи зручне тестування і діагностику помилок до запуску додатку в експлуатацію.

У розділі особлива увага приділена значенню UI/UX дизайну для створення зручного користувацького досвіду. Завдяки інструментам, таким як Figma, було розроблено інтуїтивно зрозумілі та естетично привабливі інтерфейси. Процес

прототипування та дизайну візуального стилю важливі для чіткої передачі ідей та функціональності користувачам.

Загалом, цей розділ наголошує на тому, як правильно підібрані технології та методики проектування впливають на успішність розробки веб-додатку, створюючи міцну основу для його подальшого розвитку і адаптації.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Основні елементи управління сайтом.

Підключення до бази даних є фундаментальним аспектом функціонування сайту. Через підключення до бази даних забезпечується взаємодія користувача з веб-додатком, зберігання та обробка інформації. На технічному рівні це включає використання серверних скриптів на PHP, які взаємодіють з базою даних MySQL через MAMP сервер. Скрипти обробляють запити користувача, реєструють нові дані, вносять зміни до існуючих даних та витягують необхідну інформацію для відображення на сторінках веб-додатку. Створення універсального файлу для з'єднання з базою даних, що включає головні параметри доступу, сприяє централізації управління підключеннями, що дозволяє уникнути повторення коду та забезпечує зручність у зміні конфігураційних параметрів.

```
.$user = 'root';
$pass = 'root';
$data = 'nutritionplanningdb';
$host = 'localhost';
$port = 8889;

$connection = new mysqli($host, $user, $pass, $data, $port);
if ($connection->connect_error) die ('Помилка підключення бази даних');
```

Рис. 3.1. Фрагмент коду підключення бази даних

Основні елементи управління сайтом включають вертикальне навігаційне меню, розміщене зліва від основного контенту на кожній сторінці. Меню містить логотип веб-додатку, який виступає як ключовий елемент брендингу продукту.

| | | | | | | | |
|---------------------|----------------------|--|--|---------------------------------|---------------|--------------|----------------|
| Кафедра КІТ | | | | НАУ 24 34 71 000 ПЗ | | | |
| <i>Виконав</i> | <i>Козирева М.А.</i> | | | ПРОГРАМНА РЕАЛІЗАЦІЯ | <i>Літера</i> | <i>Аркуш</i> | <i>Аркушів</i> |
| <i>Керівник</i> | <i>Шевченко О.П.</i> | | | | | 48 | 19 |
| <i>Консульт.</i> | | | | | УС-413 122 | | |
| <i>Норм. контр.</i> | <i>Шевченко О.П.</i> | | | | | | |

В ньому представлено шість основних розділів: Головна, Профіль, Календар харчування, Обрані рецепти, Список покупок та Налаштування. Для зручності та швидкого орієнтування користувачів у меню були додані іконки, які дозволяють визначати зміст сторінок без необхідності читання текстових підписів.

Головна сторінка веб-додатку слугує як візитна картка, яка формує перше враження користувачів про додаток, його можливості та інтерфейс. Під час відвідування, користувача вітає персоналізоване звернення, що включає його ім'я, якщо він уже зареєстрований у системі, та актуальну дату. Справа від основного контенту розміщена кругова діаграма, яка відображає спожиті калорії. На сторінці також реалізовані зручні інструменти для пошуку рецептів та можливість додавання нових страв. Крім того, присутній список рекомендованих страв, що оновлюється відповідно до переваг користувача. Цей список можна переглядати у формі прокручування, або ж перейти на сторінку з повним переліком доступних рецептів. Подальше дослідження програмного коду, що підтримує ці елементи, буде представлено у наступних розділах аналізу інших сторінок сайту.

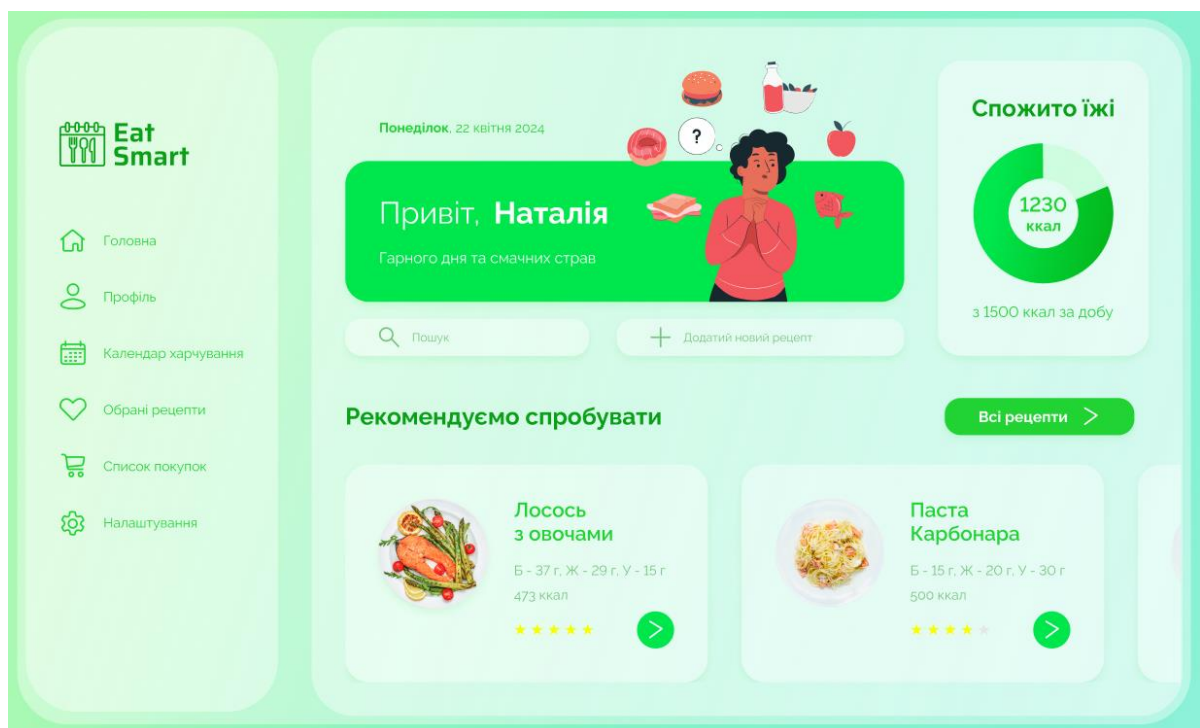


Рис. 3.2. Відображення головної сторінки

3.2. Реєстрація та аутентифікація користувачів у веб-застосунку

Процес реєстрації дозволяє користувачам створювати нові облікові записи в системі, вводячи свої основні дані, такі як ім'я, електронна адреса, пароль та його підтвердження. Дані проходять перевірку на співпадіння паролів, у разі неспівпадання користувача перенаправляють назад до форми реєстрації з відповідним повідомленням про помилку. Перед збереженням у базу даних, всі введені користувачем дані очищаються від зайвих пробілів і потенційно шкідливих символів.

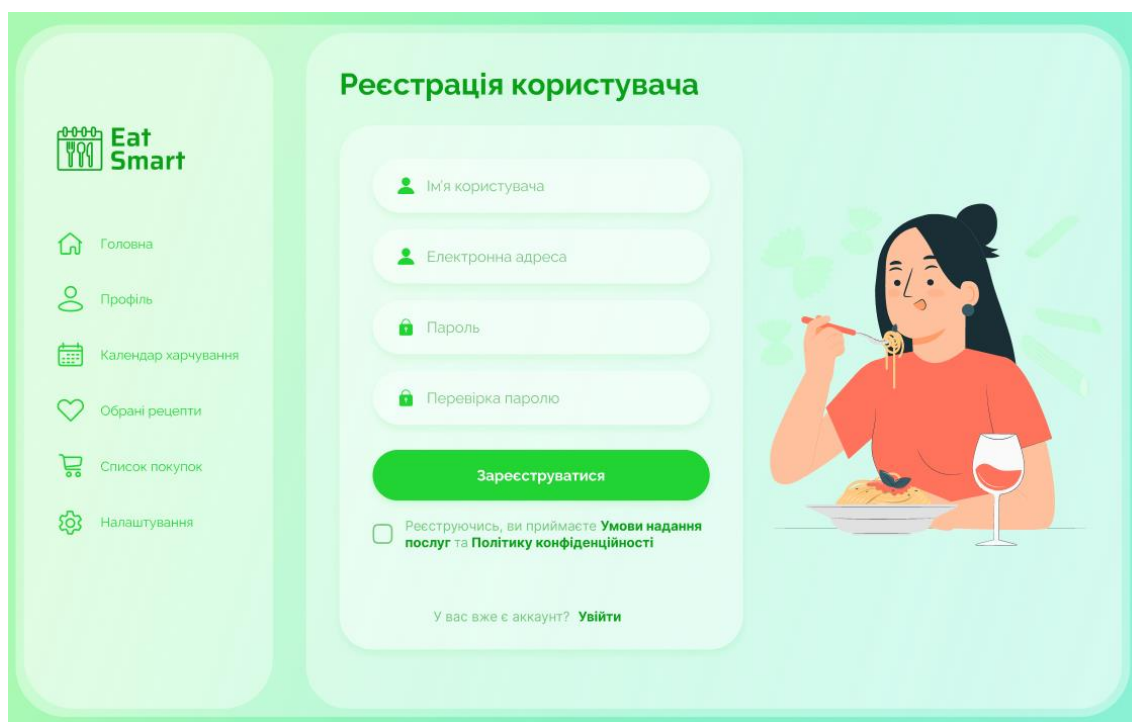


Рис. 3.3. Відображення сторінки реєстрації користувача

Після успішної реєстрації, інформація про користувача заноситься до бази даних. Пароль кодується з використанням MD5 хешування із сіллю, що підвищує безпеку зберігання. Відразу після додавання до бази, здійснюється автентифікація користувача через SELECT-запит до бази даних, який шукає записи з відповідною електронною адресою та зашифрованим паролем. Якщо запис знайдено, користувач автоматично входить в систему, ініціалізується сесія, і він перенаправляється на свою профільну сторінку.

```

session_start();
$username = filter_var(trim($_POST["username"]), FILTER_SANITIZE_STRING);
$email = filter_var(trim($_POST["email"]), FILTER_SANITIZE_STRING);
$password = filter_var(trim($_POST["pass"]), FILTER_SANITIZE_STRING);
$confirmPass = filter_var(trim($_POST["confirm-pass"]), FILTER_SANITIZE_STRING);
if ($password != $confirmPass) {
    echo '<script>alert("Паролі не співпадають!");</script>';
    header("Location: registration.php");
    exit();
}
$password = md5($password."t3d8w");
require_once 'db_connect.php';
$query = "INSERT INTO `user`(`username`, `email`, `pass`) VALUES ('$username', '$email', '$password')";
if ($mysql->query($query) === TRUE) {
    $query = "SELECT * FROM `user` WHERE `email`='$email' AND `pass`='$password'";
    $result = $mysql->query($query);
    if ($result->num_rows == 1) {
        $row = $result->fetch_assoc();
        $_SESSION['user_id'] = $row['id'];
        header("Location: profile.php?id=".$row['id']);
    } else {
        echo '<script>alert("Помилка автентифікації після реєстрації!");</script>';
        header("Location: login.php");
    }
} else {
    echo "Помилка виконання запита: " . $mysql->error;
}
$mysql->close();

```

Рис. 3.4. Фрагмент коду реєстрації користувача

Аутентифікація в веб-застосунку відбувається за допомогою перевірки електронної адреси та пароля на сторінці входу. Функція «Запам'ятати мене» дозволяє користувачам залишатися в системі навіть після закриття браузера. У разі забуття пароля користувачам надається опція доступу до сторінки відновлення пароля, забезпечуючи зручність та безпеку використання.

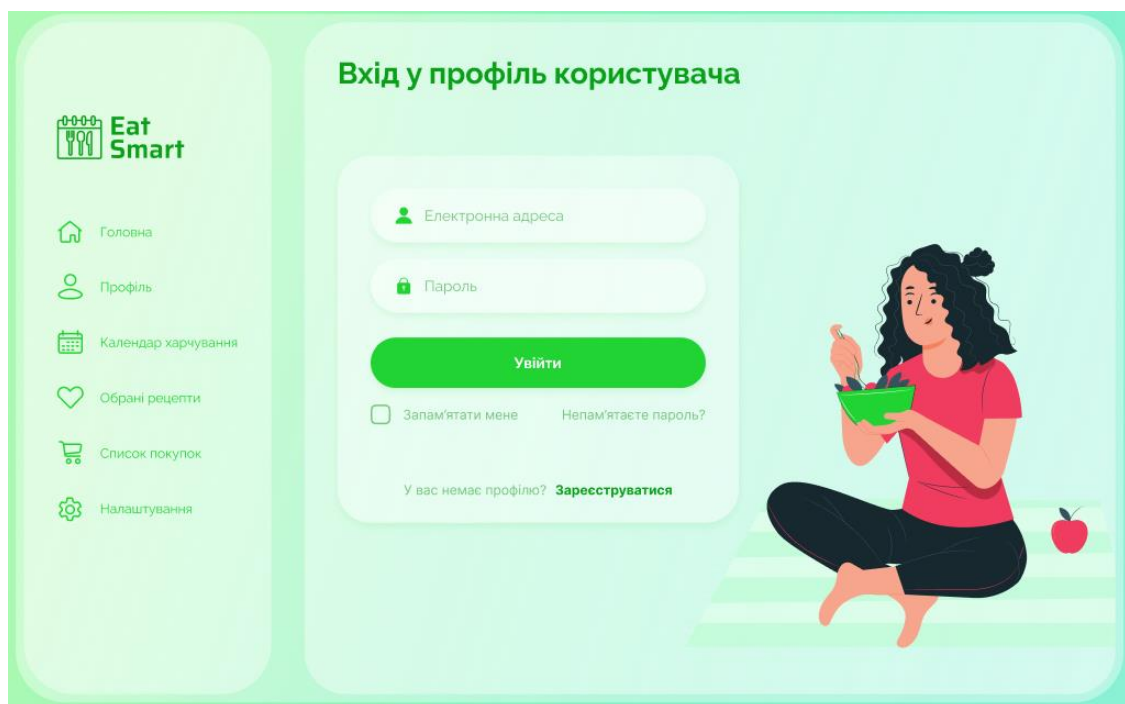


Рис. 3.5. Відображення сторінки аутентифікації користувача

Дані перевіряються за допомогою запиту до бази даних. Якщо вони не збігаються, користувач отримує повідомлення про помилку з можливістю спробувати ввійти знову. Якщо користувач забажав залишатися в системі та мати постійний доступ, ідентифікатор сесії або спеціальний токен зберігаються в cookies, що дозволяє користувачу залишатися в системі, забезпечуючи зручність без необхідності повторного входу після закриття браузера.

```
session_start();
$email = filter_var(trim($_POST["email"]), FILTER_SANITIZE_STRING);
$password = filter_var(trim($_POST["password"]), FILTER_SANITIZE_STRING);
$password = md5($password."t3d8w");
require_once 'db_connect.php';
$query = "SELECT * FROM `user` WHERE `email`='$email' AND `password`='$password'";
$result = $mysql->query($query);
if ($result->num_rows == 1) {
    $row = $result->fetch_assoc();
    $_SESSION['user_id'] = $row['id'];
    if (isset($_POST['remember']) && $_POST['remember'] == 'on') {
        setcookie('user_id', $row['id'], time() + (86400 * 30), "/");
    }
    header("Location: home.php?id=".$row['id']);
} else {
    echo '<script>alert("Неправильний email або пароль!");</script>';
    header("Location: login.php");
}
$mysql->close();
```

Рис. 3.6. Фрагмент коду аутентифікації користувача

Процес реєстрації та аутентифікації забезпечує не тільки зручність, але й важливий аспект безпеки, що є невід'ємною частиною сучасних веб-застосунків. Процедури, що були використані у розробці, допомагають зберегти конфіденційність інформації користувачів та забезпечити доступ до індивідуальних налаштувань харчування.

3.3. Збір та обробка особистих даних користувачів

Після успішної реєстрації, користувачі проходять через детальне анкетування, розділене на чотири ключові етапи. Єдиний стиль дизайну сторінок анкетування сприяє інтуїтивно зрозумілому введенню даних. На кожній сторінці зверху вміщено заголовок "Анкетування", під яким розміщений підзаголовок, який конкретизує ціль поточного етапу. Це орієнтує користувачів на вид інформації, яку вони повинні подати.

Індикативна шкала демонструє прогрес користувача через анкетування. На ілюстрації праворуч відображається персонаж жінки, думки якої змінюють символи залежно від поточного етапу, що додає візуального контексту та асоціативної підказки до запитаних даних. В нижній частині кожної сторінки розміщено виділені кнопки "Назад" та "Далі", вони дозволяють користувачам без зусиль переміщуватися між різними етапами анкетування.

На сторінці анкетування, присвяченій біометричним даним, користувачі мають можливість внести інформацію про свій ріст, вагу, дату народження та стать. Введення цих даних є критично важливим для розрахунку базового метаболічного рівня (BMR) та загальної добової потреби в калоріях.

Друга сторінка анкетування зосереджена на фізичній активності та цілей. Тут можна обрати рівень активності, який коливається від мінімальної активності до регулярних тренувань. Крім того, користувачі мають можливість визначити свої цілі, як-от схуднення, підтримка поточної ваги чи набір м'язової маси. Вибір конкретної мети дозволяє додатку точніше адаптувати рекомендації щодо калорій та нутрієнтів. Важливо, що в кожній категорії можливий вибір тільки одного параметра, що забезпечує однозначність та точність налаштувань.

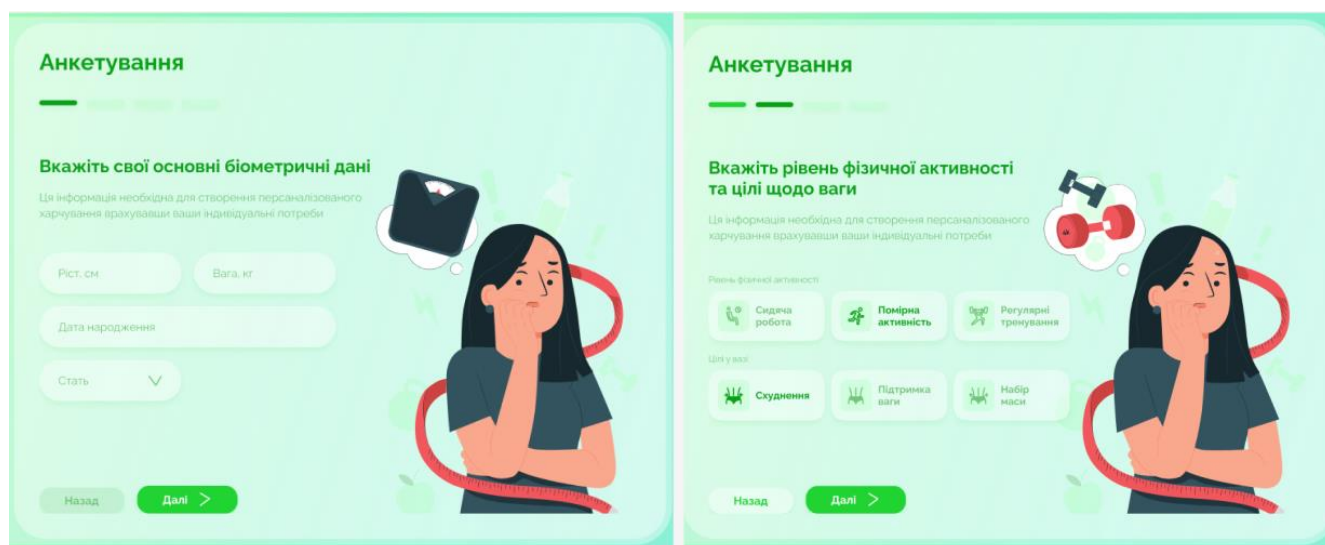


Рис. 3.7. Відображення перших двох етапів анкетування

На третій сторінці анкетування, користувачі мають можливість зазначити наявність певних медичних умов, що можуть вимагати спеціального дієтичного підходу. Серед таких станів можуть бути діабет, гіпертонія, високий рівень холестерину, серцево-судинні захворювання та інші захворювання, що впливають на рекомендації з харчування.

Четверта сторінка анкетування зосереджена на харчових непереносимостях та алергіях. Наприклад, може бути непереносимість глютену, лактози, алергія на горіхи та інші алергени. Система дозволяє вибрати кілька варіантів непереносимостей.

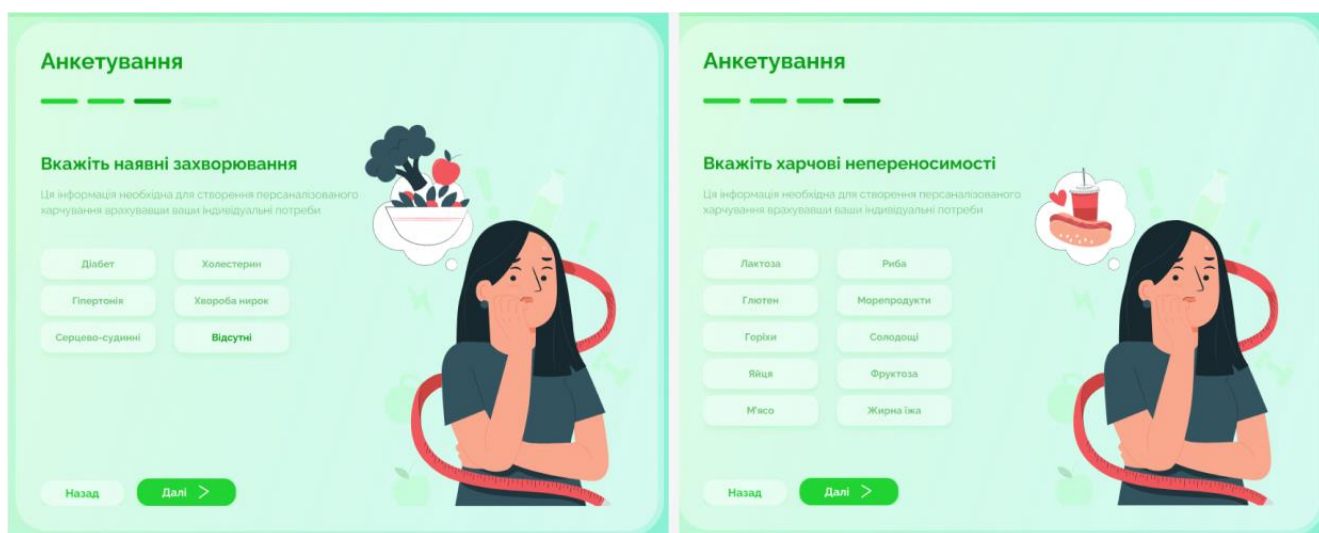


Рис. 3.8. Відображення останніх двох етапів анкетування

Отримуємо дані з форми та створюємо змінні. Для оптимізації зберігання дані про захворювання та непереносимості, подані у вигляді масивів, конвертуються в строки, розділені комами.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    if (isset($_POST['final'])) {  
        $birthdate = $_SESSION['birthdate'];  
        $weight = $_SESSION['weight'];  
        $height = $_SESSION['height'];  
        $sex = $_SESSION['sex'];  
        $physical_activity = $_SESSION['physical_activity'];  
        $goals = $_SESSION['goals'];  
        $diseases = implode(',', $_POST['diseases']);  
        $intolerances = implode(',', $_POST['intolerances']);  
    }  
}
```

Рис. 3.9. Фрагмент коду обробки даних з анкетування

Перетворення введеної дати народження користувача у вік відбувається через клас DateTime, що дозволяє системі точно визначати актуальний вік порівнюючи його з поточною датою. Цей вік є вирішальним для розрахунку базового обміну речовин (BMR), який проводиться за формулою Mifflin-St Jeor, залежно від статі. Розрахований BMR служить основою для визначення загальної добової потреби в енергії (TDEE), яка коригується в залежності від рівня фізичної активності та цілей користувача: на зменшення ваги знижується калорійність на 15%, для підтримки ваги корекції не потрібно, а для набору маси калорійність збільшується на 15%.

```

$birthDate = new DateTime($birthdate);
$today = new DateTime('today');
$age = $birthDate->diff($today)->y;
if ($sex == 'male') {
    $bmr = 10 * $weight + 6.25 * $height - 5 * $age + 5;
} else {
    $bmr = 10 * $weight + 6.25 * $height - 5 * $age - 161;
}
$activity_factors = ['sedentary' => 1.2, 'moderate' => 1.55, 'active' => 1.725];
$tdee = $bmr * $activity_factors[$physical_activity];
switch ($goals) {
    case 'weight_loss': $tdee *= 0.85;
        break;
    case 'weight_gain': $tdee *= 1.15;
        break;
}

```

Рис. 3.10. Фрагмент даних підрахунку віку, BMR та TDEE

Скрипти PHP обробляють ці дані, здійснюючи їх вставку до бази через SQL команди типу INSERT, зберігаючи таким чином відповіді та розраховані параметри користувачів. По завершенню операцій, скрипт закриває з'єднання з базою даних, гарантуючи таким чином цілісність і конфіденційність оброблюваних даних.

```

$sql = "INSERT INTO user (birthdate, weight, height, sex, physical_activity, goals,
bmr, tdee, diseases, intolerances) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?,?)";
if ($stmt = $mysqli->prepare($sql)) {
    $stmt->bind_param("sssssd", $birthdate, $weight, $height, $sex,
    $physical_activity, $goals, $bmr, $tdee);
    if ($stmt->execute()) {
        echo "Вашу анкету успішно збережено!";
    } else {
        echo "Помилка: " . $stmt->error;
    }
    $stmt->close();
}
$mysqli->close();

```

Рис. 3.11. Фрагмент коду додавання інформації до БД

3.4. Персоналізований підбір страв

Одним з найголовніших модулів програми є персоналізований підбір страв. Система ретельно враховує особливості користувача, адаптуючи рекомендації до його харчових обмежень, потенційних алергій та захворювань. Такий підхід гарантує, що страви не лише відповідають денному калорійному прийому, а й підтримують здоров'я користувачів.

```
session_start();
require_once 'db_connect.php';
$user_id = $_SESSION['user_id'];
$query = "SELECT intolerances, diseases, daily_calorie_intake FROM users WHERE user_id = $user_id";
$result = $conn->query($query);
$user_data = $result->fetch_assoc();
$intolerances = explode(',', $user_data['intolerances']);
$diseases = explode(',', $user_data['diseases']);
$daily_calories = $user_data['daily_calorie_intake'];
$condition = "WHERE meal_kkal <= $daily_calories";

foreach ($intolerances as $intolerance) {
    $condition .= " AND ingredients NOT LIKE '%$intolerance%'";
}
if (in_array('diabetes', $diseases)) { $condition .= " AND meal_carbs <= 20 AND meal_sugar <= 5";}
if (in_array('hypertension', $diseases)) { $condition .= " AND meal_salt <= 0.5";}
if (in_array('heart_disease', $diseases)) { $condition .= " AND meal_fat <= 2 AND meal_cholesterol <= 0.02";}
if (in_array('cholesterol', $diseases)) { $condition .= " AND meal_cholesterol <= 0.02";}
if (in_array('kidney_disease', $diseases)) { $condition .= " AND meal_protein <= 10 AND meal_salt <= 0.2";}
$sql = "SELECT * FROM meals $condition";
$meals_result = $conn->query($sql);
while($meal = $meals_result->fetch_assoc()) {
    include 'product_card.php';
}
$conn->close();
```

Рис. 3.12. Фрагмент коду фільтрації страв.

Фільтраційний скрипт інтегрований на багатьох сторінках веб-додатку. Він забезпечує з'єднання з базою даних, автоматично отримує відомості про дієтичні обмеження та захворювання користувача, і на основі цих даних формує запит SQL. Запит вибирає страви, що задовольняють умовам користувача та подає їх у вигляді карточок на сторінці. Кожна карточка містить основну інформацію про страву (зображення, назву, основні поживні речовини, калорійність та рейтинг від користувачів) та кнопку для переходу до детального перегляду рецепту. Прикладом відображення цього скрипту є блок "Рекомендації", який з'являється на різних сторінках сайту.

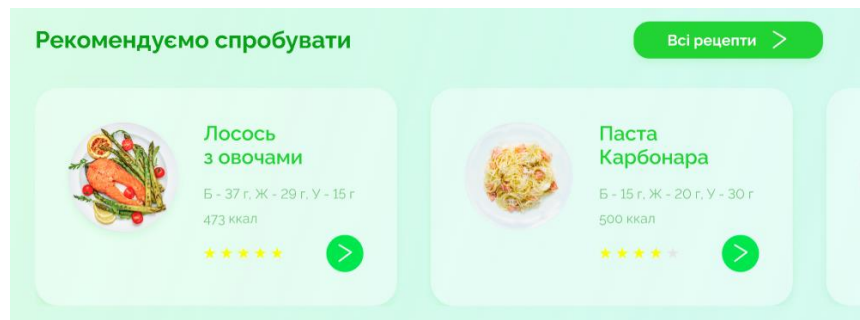


Рис. 3.13. Відображення рекомендованих страв.

Фільтраційний модуль важливий не лише для інформування користувачів про потенційно підходящі страви, а й для забезпечення впевненості у виборі їжі, що сприяє підтриманню здорового способу життя.

3.5. Розробка сторінки рецептів

Мета сторінки рецептів полягає у представленні конкретного рецепту, надаючи користувачам детальну інформацію про страву та додаткові можливості для її використання.

Дизайн сторінки рецептів виконаний у вигляді двоколонкової композиції. Права колонка візуально насичена: розміщено фотографію готової страви, детальний опис процесу приготування, інформація про автора рецепту та дату публікації, а також присутня кнопка для повернення на попередню сторінку для зручності навігації. Ліва колонка містить основну інформацію про страву: назву, рейтинг, який відображає популярність страви серед користувачів, опис страви, категорії до яких вона належить, поживні речовини та калорійність. Також є список інгредієнтів та функціональна кнопка, яка дозволяє додати страву до списку улюблених.

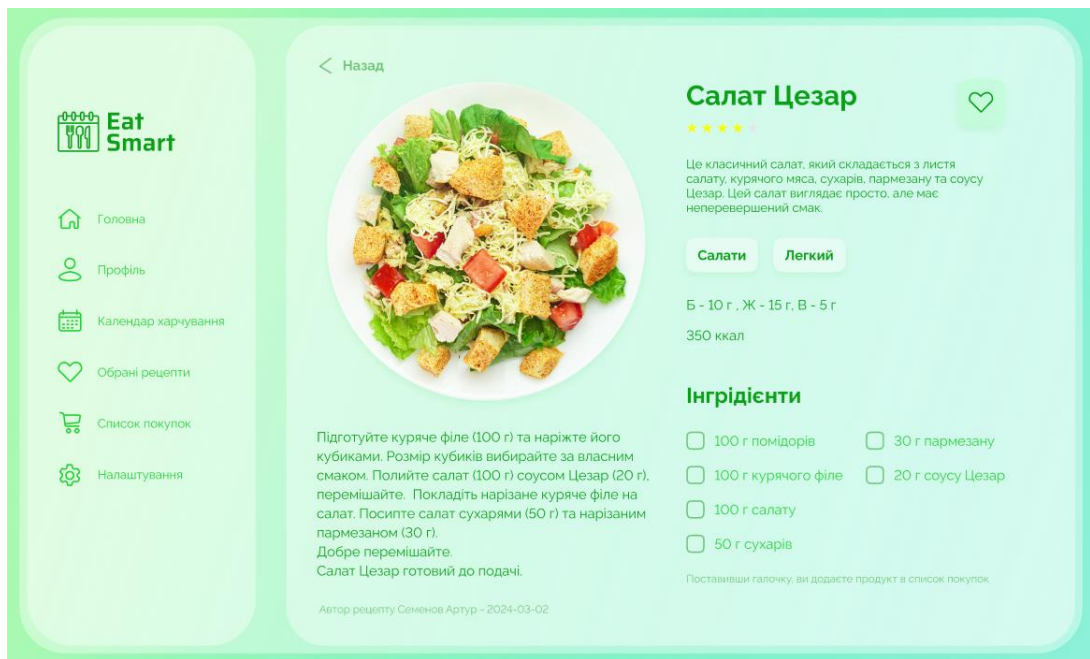


Рис 3.14. Відображення сторінки рецептів

Для програмної реалізації сторінки рецептів було створено спеціалізований файл-шаблон, що використовується для динамічного відображення різних рецептів. Універсальність цієї розробки полягає у використанні PHP для виведення даних із бази даних, що дозволяє автоматично адаптувати вміст сторінки під конкретний рецепт.

```

<div class="right">
  <button onclick="goBack()">Повернутися</button>
  
  <p1> <?php echo $row['meal_description']; ?></p1>
  <p3>Автор рецепту <?php echo $row['username']; ?> - <?php echo $row['meal_created_at']; ?></p3>
</div>
<div class="left">
  <h1> <?php echo $row['meal_name']; ?></h1>
  <div class="stars">
    <div class="star-rating <?php if ($row['meal_user_ratings'] >= 1) echo 'selected'; ?>">★</div>
    <div class="star-rating <?php if ($row['meal_user_ratings'] >= 2) echo 'selected'; ?>">★</div>
    <div class="star-rating <?php if ($row['meal_user_ratings'] >= 3) echo 'selected'; ?>">★</div>
    <div class="star-rating <?php if ($row['meal_user_ratings'] >= 4) echo 'selected'; ?>">★</div>
    <div class="star-rating <?php if ($row['meal_user_ratings'] >= 5) echo 'selected'; ?>">★</div>
  </div>
  <button id="addToFavoriteButton" onclick="addToFavorite(<?php echo $row['meal_id']; ?>)">
    </button>
  <p1 class="button"><?php echo $row['meal_category']; ?></p1>
  <p1>Б - <?php echo $row['meal_protein']; ?>, Ж - <?php echo $row['meal_fat']; ?>
  , <?php echo $row['meal_carbs']; ?>, К - <?php echo $row['meal_carbs']; ?></p1>
  <p1><?php echo $row['meal_kkal']; ?> ккал</p1>
  <h2>Інгредієнти</h2>
  <div class="ingredients">
    <form id="ingredientForm" action="process_form_ingredients.php" method="post">
      <?php while($row = $result->fetch_assoc()) {
        echo ' <input type="checkbox" name="ingredients[]" value="' . $row["ingredient_name"] .
        '" onchange="autoSubmit()"> ' . $row["ingredient_name"] . "<br>";
      } ?>
    </form>
  </div>
  <p3>Поставивши галочку, ви додаєте продукт в список покупок</p3>
</div>

```

Рис 3.15. Фрагмент коду сторінки рецепту

```

function goBack() {
    window.history.back();
}
function autoSubmit() {
    document.getElementById("ingredientForm").submit();
}
function addToFavorite(meal_id) {
var user_id = <?php echo $_SESSION['user_id']; ?>;
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        console.log(this.responseText);
        alert('Страву додано до улюблених!');
    }
};
xhttp.open("POST", "process_add_to_favorite.php", true);
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send("user_id=" + user_id + "&meal_id=" + meal_id);
}
}

```

Рис 3.16. Фрагмент коду функцій сторінки рецепту

На сторінці рецептів реалізовано кілька важливих функцій, що сприяють зручності користування та інтерактивності сайту. Важливим елементом є кнопка "Додати до улюблених", яка активує JavaScript-функцію `addToFavorite`. Ця функція відправляє AJAX-запит на сервер, що дозволяє користувачам додавати страви до своїх улюблених без необхідності перезавантаження сторінки. Операція збереження виконується шляхом вставки відповідних даних у таблицю `FavoriteMeal`.

```

if(isset($_POST['user_id']) && isset($_POST['meal_id'])) {
    require_once 'db_connect.php';
    $user_id = $_POST['user_id'];
    $meal_id = $_POST['meal_id'];
    $sql = "INSERT INTO FavoriteMeal (user_id, meal_id) VALUES ('$user_id', '$meal_id')";
    $conn->close();
}

```

Рис. 3.17. Фрагмент коду обробника додавання страви до списку улюблених

Крім того, список інгредієнтів для кожної страви відображається на сторінці з допомогою циклу, який виводить інформацію про назву та вагу кожного продукту. Для кожного елемента передбачено чекбокс, який дозволяє користувачам легко вибирати ті продукти, які вони хочуть додати до свого списку покупок. Вибір чекбокса автоматично ініціює відправку форми, яка обробляє запити на додавання вибраних продуктів до списку покупок.

```

if(isset($_POST['ingredients'])) {
    require_once 'db_connect.php';
    $user_id = $_SESSION['user_id'];
    foreach ($_POST['ingredients'] as $ingredient) {
        $ingredient_name = $ingredient['name'];
        $ingredient_weight = $ingredient['weight'];
        $sql = "INSERT INTO Shoppinglist (shoppinglist_ingredient_name,
            shoppinglist_ingredient_weight, user_id)
            VALUES ('$ingredient_name', '$ingredient_weight', '$user_id')";
    }
    $conn->close();
}

```

Рис. 3.18. Фрагмент коду обробника вибраних інгредієнтів з додаванням до списку покупок

Для відображення вибраних користувачем продуктів на сторінці списку покупок, веб-застосунок використовує дані, збережені в таблиці ShoppingList бази даних. Цей список покупок дозволяє користувачам зберігати, оновлювати, додавати та видаляти продукти, які вони планують придбати, тим самим спрощуючи процес закупівель харчовими продуктами.

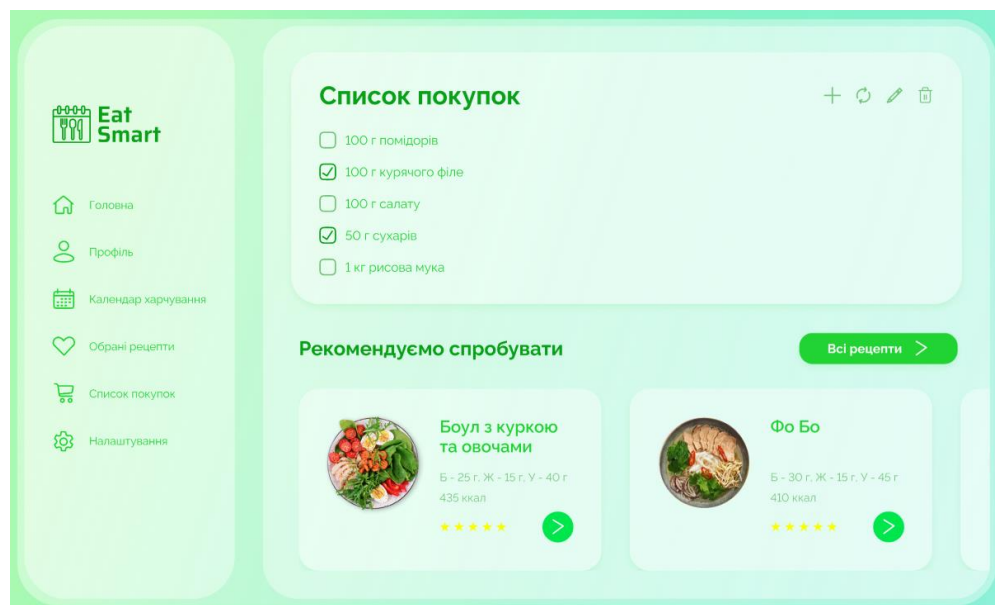


Рис. 3.19. Відображення сторінки списку продуктів

Завдяки використанню динамічних шаблонів та AJAX-запитів для обробки даних, а також інтеграції динамічних веб-технологій і інтерактивного контенту, сторінка рецептів стає основним інструментом для перегляду страв, забезпечуючи взаємодію без перезавантаження сторінки.

3.6. Реалізація календаря для розподілу страв на дні тижня

Календар харчування на веб-сайті є інтерактивним інструментом для планування дієти, дозволяючи користувачам візуально організовувати своє щоденне споживання їжі. Користувачі можуть легко додавати страви на конкретні дати, визначати час прийому їжі як сніданок, обід, вечеря чи перекус, а також переглядати свій денний та місячний раціон.

Як і інші сторінки, календар тісно інтегрований з базою даних, що дозволяє зберігати, видаляти, відтворювати та змінювати дані. Основні елементи сторінки включають інтерактивний календар для вибору днів, панель з інформацією про спожиті калорії та макронутрієнти обраних днів, а також інструменти для управління стравами, що надають можливість вносити зміни та додавати нові елементи до раціону.

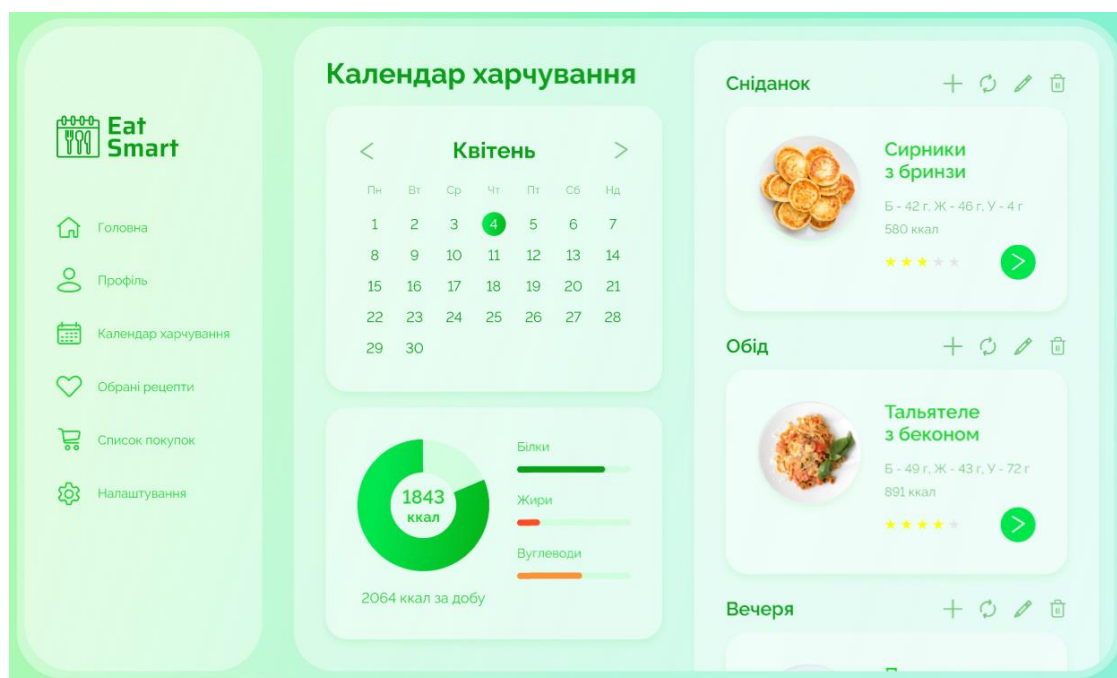


Рис. 3.20. Відображення сторінки календарю харчування

Основним елементом, який має бути на сторінці календаря харчування, є сам календар. Для його реалізації використовується бібліотека jQuery разом із jQuery UI, яка дозволяє створити інтерактивний вибір дати за допомогою віджета daterickер.

Коли користувач вибирає дату в календарі, активізується функція `onSelect`, яка отримує вибрану дату у форматі «місяць/день/рік». Це значення використовується для динамічного перенаправлення користувача на наступну сторінку, передаючи вибрану дату як параметр запиту в URL. Завдяки цьому користувачеві не потрібно натискати додаткові кнопки чи відправляти форму — перехід відбувається автоматично після вибору дати.

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script>
$( function() {
    $("#datepicker").datepicker({
        onSelect: function(dateText) {
            window.location.href = 'daily_calorie_tracker.php?date=' + dateText;
        }
    });
});
</script>
```

Рис. 3.21. Фрагмент коду функцій календарю

Розробка панелі з інформацією про спожиті калорії та макронутрієнти вимагає уважного підходу до програмування, адже необхідно врахувати багато нюансів. PHP скрипт спершу ініціює сесію та зв'язується з базою даних, використовуючи ідентифікатор користувача з сесії для виконання запиту на отримання денної потреби в калоріях.

На наступному кроці, скрипт формує SQL-запит, який визначає загальну кількість спожитих калорій, білків, жирів, і вуглеводів за вибраний день. З отриманих даних розраховуються відсотки споживання калорій та макронутрієнтів відносно денної норми користувача.

Після завершення роботи з базою даних і отримання необхідної інформації, скрипт передає ці дані на клієнтську сторону. Тут JavaScript використовується для динамічного оновлення візуальних елементів на сторінці: кругова діаграма анімує проценти споживання калорій, а горизонтальні індикатори відображають споживання білків, жирів та вуглеводів.

```

$user_id = $_SESSION['user_id'];
$date = isset($_GET['date']) ? $_GET['date'] : date('Y-m-d');

$query = "SELECT daily_calorie_intake FROM users WHERE user_id = $user_id";
$result = $conn->query($query);
if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $totalCalories = $row['daily_calorie_intake'];
}

$query = "SELECT SUM(meal_kkal) AS consumedCalories, SUM(meal_protein) AS totalProtein,
            SUM(meal_fat) AS totalFat, SUM(meal_carbs) AS totalCarbs FROM meals
            JOIN DailyPlan ON meals.meal_id = DailyPlan.meal_id
            WHERE DailyPlan.user_id = $user_id AND DailyPlan.date = '$date'";
$result = $conn->query($query);
if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $consumedCalories = $row['consumedCalories'];
    $totalProtein = $row['totalProtein'];
    $totalFat = $row['totalFat'];
    $totalCarbs = $row['totalCarbs'];
}
$conn->close();
$percentageConsumed = ($consumedCalories / $totalCalories) * 100;
$percentageLeft = 100 - $percentageConsumed;
$proteinPct = ($totalProtein / $totalCalories) * 100;
$fatPct = ($totalFat / $totalCalories) * 100;
$carbsPct = ($totalCarbs / $totalCalories) * 100;

```

Рис. 3.22. Фрагмент коду підрахунку даних для кругової діаграми спожитих калорій та індикаторів макронутрієнтів за вибраний день

JavaScript-функції `updateCaloriesChart` та `updateNutrientBar` відповідають за оновлення візуального стану діаграми та індикаторних смуг, змінюючи їх колір залежно від того, наскільки споживання входить у рекомендовані діапазони. Кольори індикаторів змінюються зелений (оптимальне споживання), проміжний оранжевий (на межі норми) та червоний (надмірне або недостатнє споживання), що допомагає користувачам візуально оцінити якість свого раціону.

```

function updateCaloriesChart(percentageConsumed) {
    var circle = document.querySelector('.circle');
    circle.style.strokeDasharray = `${percentageConsumed}, 100`;
    document.querySelector('.circle-label strong').innerText =
        `${Math.round(percentageConsumed)}% спожито`;
}

function updateNutrientBar(percentage, elementId, range) {
    var bar = document.getElementById(elementId);
    bar.style.width = `${percentage}%`;
    if (percentage >= range.minGreen && percentage <= range.maxGreen) {
        bar.style.backgroundColor = 'green';
    } else if ((percentage >= range.minOrange && percentage < range.minGreen) ||
        (percentage > range.maxGreen && percentage <= range.maxOrange)) {
        bar.style.backgroundColor = 'orange';
    } else {
        bar.style.backgroundColor = 'red';
    }
}

updateCaloriesChart(<?php echo $percentageConsumed; ?>);
updateNutrientBar(<?php echo $proteinPct; ?>, 'proteinBar', {minGreen: 10, maxGreen: 35, minOrange: 5, maxOrange: 55});
updateNutrientBar(<?php echo $fatPct; ?>, 'fatBar', {minGreen: 20, maxGreen: 35, minOrange: 10, maxOrange: 40});
updateNutrientBar(<?php echo $carbsPct; ?>, 'carbsBar', {minGreen: 45, maxGreen: 65, minOrange: 35, maxOrange: 75});

```

Рис. 3.23. Фрагмент коду функцій для виведення кругової діаграми спожитих калорій та індикаторів макронутрієнтів за вибраний день

Останній блок на сторінці календаря харчування — це інтерактивний список прийомів їжі, який динамічно оновлюється на основі обраної дати. Цей список включає опції додавання, оновлення, редагування та видалення прийому їжі, що забезпечує гнучкість у плануванні раціону користувача.



Рис. 3.24. Відображення елементів управління прийомами їжі

Для додавання страви до денного раціону користувач натискає на іконку з плюсом, що викликає вікно модального діалогу. Модальне вікно, відоме також як рор-уп, перекриває зміст основної сторінки, пропонуючи вибір зі списку улюблених страв користувача. Вікно містить всі необхідні опції для здійснення вибору та додавання страви безпосередньо у денний план харчування.

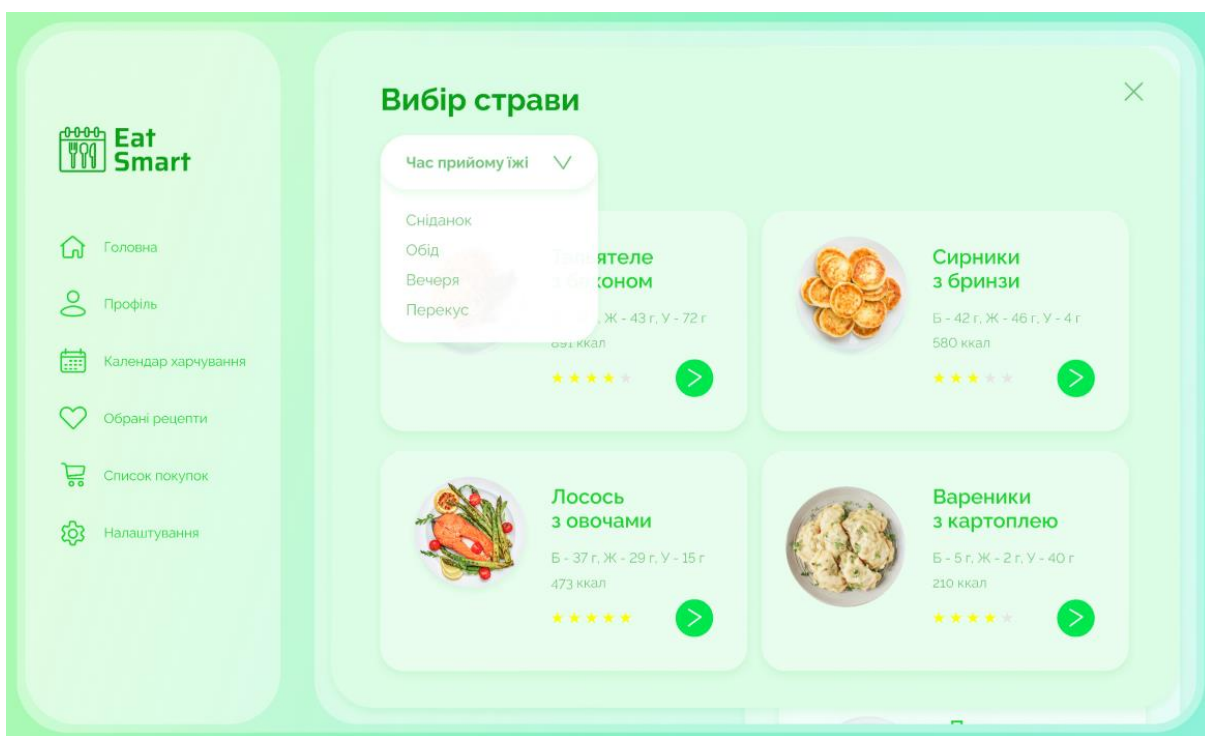


Рис. 3.25. Відображення рор-уп для вибору страви

Коли користувач натискає кнопку для відкриття рор-уп, вікно стає видимим, надаючи можливість вибрати страви для включення в денний раціон. Натискання на хрестик приховує рор-уп, дозволяючи користувачу без змін повернутися до попереднього інтерфейсу.

```
document.getElementById('openPopup').addEventListener('click', function()
|   document.getElementById('mealPopup').style.display = 'block';
});

document.querySelector('.close').addEventListener('click', function() {
|   document.getElementById('mealPopup').style.display = 'none';
});
```

Рис. 3.26. Фрагмент коду відкриття та закриття рор-уп

Функція `loadMealCards` відповідає за динамічне завантаження карточок улюблених страв користувача з файлу РНР, забезпечуючи адаптивне відображення страв, які користувач вже виокремив як улюблені.

```
document.addEventListener('DOMContentLoaded', function() {
  loadMealCards();
  document.getElementById('mealCardsContainer').addEventListener('click', function(e) {
    if (e.target.classList.contains('meal-card')) {
      const mealId = e.target.getAttribute('data-meal-id');
      const mealTime = document.getElementById('mealTimeSelect').value;
      if (!mealTime) {
        alert('Будь ласка, виберіть час прийому їжі'); return;
      }
      addToDailyPlan(mealId, mealTime);
    }
  });
});

function loadMealCards() {
  fetch('favorite_meal_card.php') .then(response => response.text())
  .then(html => document.getElementById('mealCardsContainer').innerHTML = html);
}

function addToDailyPlan(mealId, mealTime) {
  fetch('add_to_daily_plan.php', {
    method: 'POST',
    headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
    body: `meal_id=${mealId}&meal_time=${mealTime}`
  })
  .then(response => response.json())
  .then(data => {
    if (data.success) {
      console.log('Страву успішно додано!');
    } else {
      console.error('Не вдалось додати страву.');
```

Рис. 3.27. Фрагмент коду функцій рор-уп для додавання страв у раціон дня

Після завантаження карточок страв, користувачу потрібно натиснути по елементах карточок. Якщо користувач вибирає страву, але не вказав час прийому їжі, виводиться помилка, спонукаючи користувача зробити вибір. При виборі обидвох параметрів, інформація відсилається на сервер за допомогою AJAX-запиту для оновлення бази даних.

Функція `addToDailyPlan` використовує AJAX для відправки ідентифікатора страви та часу прийому до скрипту на сервері `php`-файлі, який обробляє додавання інформації до бази даних. Успішне додавання виводить в консоль повідомлення про успіх, у протилежному випадку виводиться помилка.

Сторінка календарю дозволяє користувачам зручно планувати своє харчування, маючи можливість вибору з улюблених страв, а також визначити час їх прийому.

3.7. Висновок до розділу 3

Висновок до третього розділу детально описує розробку програмних компонентів для веб-застосунку, призначеного для персоналізованого планування харчування. Розглянуто ключові сторінки, такі як головна, реєстрація, аутентифікація, профіль користувача, календар харчування, сторінка рецептів та анкетування. Інтеграція з базою даних грає важливу роль у збереженні та обробці користувацьких даних, а серверні скрипти PHP ефективно обробляють користувацькі запити.

Процес анкетування, який збирає критичну інформацію про користувачів, допомагає створити алгоритми для індивідуального підбору страв. Застосування HTML, CSS і JavaScript сприяє створенню естетично привабливих і зручних у навігації сторінок, що відображає високий рівень розуміння UI/UX дизайну. Всі сторінки веб-додатку спроектовані з мінімальною кількістю коду, що полегшує управління сайтом.

Виконана робота слугує грамотним поєднанням технічних знань та користувацьких вимог, що є ключовим у створенні сучасних веб-додатків.

ВИСНОВКИ

У дипломній роботі було створено веб-додаток для персонального планування харчування, спрямований на задоволення потреб кінцевих користувачів. Основною проблемою, яку вдалося розв'язати, стала необхідність забезпечення користувачів індивідуалізованими рекомендаціями з харчування, що враховують їхні особисті дані, фізіологічні особливості та харчові уподобання. Розробка веб-додатку здійснювалася з урахуванням глибокого аналізу історії, сучасних тенденцій та існуючих платформ планування харчування. Було розроблено алгоритм, який дозволяє адаптувати рекомендації стосовно харчування до особистих даних та уподобань користувачів, що значно підвищує точність та ефективність планування раціону.

Аналіз портретів користувачів допоміг визначити кілька основних потреб, на основі яких було сформовано функціональні вимоги і розроблено карту сайту. Це дозволило створити веб-додаток, який максимально відповідає очікуванням користувачів. Особлива увага була приділена системі управління базами даних та використанню реляційної моделі, що забезпечує структуроване та безпечне зберігання даних. Завдяки цьому, додаток здатний обробляти великі обсяги інформації і забезпечувати швидкий доступ до необхідних даних.

У проектуванні веб-інтерфейсу було проведено дослідження в області UI/UX дизайну, що дозволило створити індивідуальний візуальний стиль, який виділяється серед аналогічних додатків та підвищує впізнаваність продукту. Використання сучасних принципів UI/UX дизайну сприяло створенню зручного та інтуїтивно зрозумілого інтерфейсу, що значно полегшує взаємодію користувачів з системою.

Програмна реалізація була виконана з використанням HTML, CSS, JavaScript та PHP, що забезпечило динамічну та ефективну взаємодію з користувачами. Вибір цих технологій був обґрунтований їхньою широкою підтримкою та можливостями, що дозволяють створювати високоякісні веб-додатки. Протягом роботи над дипломним проектом були розглянуті та реалізовані різноманітні аспекти розробки,

від вибору та конфігурації розробницького середовища і систем управління базами даних до детальної реалізації серверних та клієнтських скриптів.

Важливим елементом проекту є інтеграція з базою даних, яка дозволяє зберігати, обробляти та відображати користувацькі дані в реальному часі. Це забезпечує користувачам можливість швидко і зручно керувати своїми дієтичними планами, отримувати актуальну інформацію та персоналізовані рекомендації.

Таким чином, дипломний проект не тільки досягнув своєї мети у створенні функціонального і привабливого додатку для планування харчування, але й продемонстрував вдале поєднання технічних знань і користувацьких вимог. Розроблений додаток дозволяє ефективно планувати харчування, враховуючи індивідуальні потреби користувачів, що сприяє підтримці здорового способу життя.

Новизна отриманих результатів полягає у створенні алгоритму персоналізованого підбору дієтичних рекомендацій, що інтегрує комплексний аналіз особистісних даних користувачів. Це дозволяє забезпечити значно більшу точність рекомендацій порівняно з існуючими системами, що робить цей підхід інноваційним і перспективним для подальших досліджень та вдосконалення.

Вірогідність отриманих результатів обґрунтована використанням сучасних методик розробки веб-додатків, аналізом потреб користувачів та застосуванням передових технологій у сфері програмування і дизайну. Результати даної роботи можуть бути використані як основа для подальшого розвитку персоналізованих систем планування харчування, а також для створення інших подібних додатків, що потребують індивідуалізації на основі користувацьких даних.

Рекомендації щодо практичного використання отриманих результатів включають впровадження розробленого веб-додатку у різних сферах, де важливе значення має персоналізація харчування, таких як дієтологія, фітнес, медичні установи та особисте користування. Використання цього додатку дозволить значно покращити якість життя користувачів, забезпечуючи їм зручний інструмент для планування та контролю свого раціону.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Як персоналізоване харчування трансформує наш підхід до їжі. URL: <https://www.healthline.com/nutrition/personalized-nutrition> (дата звернення: 23.05.2024).
2. What is digital health technology? What you should know. URL: <https://qubika.com/blog/digital-health-technology/> (дата звернення: 20.05.2024).
3. Найкращі добавки для правильного харчування. URL: <https://bigmag.ua/luchshee-prilozhenie-dlya-pravilnogo-pitaniya.html> (дата звернення: 20.05.2024).
4. What Is a Customer Persona? Definition, Uses, and Examples. URL: <https://www.delve.ai/blog/customer-personas> (дата звернення: 23.05.2024).
5. Індивідуальний план харчування: повний посібник з планування раціону на основі калорійності та макроелементів. URL: <https://gymbeam.ua/blog/uk/indyviduanlyi-plan-harchuvannia-rovni/> (дата звернення: 25.05.2024).
6. Мапа сайту – що, як, навіщо. Повний посібник з Sitemap. URL: <https://textum.com.ua/ru/blog/karta-sajta-cto-kak-zachem-polnoe-rukovodstvo-po-sitemap/> (дата звернення: 17.05.2024).
7. The difference between HTML, CSS, JavaScript and PHP languages. URL: <https://patriciasdesignsite.wordpress.com/2015/02/19/the-difference-between-html-css-javascript-and-php-languages/> (дата звернення: 25.05.2024).
8. У чому відмінність сайту від веб-додатка. URL: <https://foxminded.ua/chym-vidrizniaietsia-sait-vid-veb-dodatku/> (дата звернення: 25.05.2024).
9. Вступ до JavaScript. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/intro> (дата звернення: 13.05.2024).
10. Етапи створення веб-додатку. Основи PHP та MySQL. URL: <https://promoter.net.ua/articles/etapi-stvorennya-veb-dodatku-osnovi-php-ta-mysql.html> (дата звернення: 17.05.2024).

11. MAMP: The definitive guide. URL: <https://supporthost.com/mamp/> (дата звернення: 20.05.2024).
12. Мулеса О. Ю. Інформаційні системи та реляційні бази даних / О. Ю. Мулеса, Я. В. Варга, навч. посібник. Ужгород, 2023. 132 с.
13. Veronica Allen. The New 2022 UI/UX For Beginners And Experts: UX/UI Design for Automatic Designers / Veronica Allen. – N.p.: Amazon Digital Services LLC - Kdp, 2021. 92 s.
14. Гречко А. В. Web-дизайн, як поєднання психології та мультимедіа / А. В. Гречко, М. М. Колендовська; Радіоелектроніка та молодь в ХХІ столітті : матеріали 26-го Міжнародного молодіжного форуму, 24-25 листопада 2022 р. Харків : ХНУРЕ, 2022. № 3. С. 146–147.