

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему РОЗРОБКА КАЛЬКУЛЯТОРА КАЛОРІЙ СТРАВ

Виконав: студент(ка) 4 курсу, групи КНТ-522

Спеціальності 123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація)
«Комп'ютерна інженерія»

ЧИЖМА Д.Е.

(прізвище та ініціали)

Керівник ГРУШКО С.С.

(прізвище та ініціали)

Рецензент ТЯГУНОВ Д.В.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування вищого навчального закладу)

Факультет Комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти бакалаврський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерна інженерія
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
 Завідувач кафедри КУДЕРМЕТОВ Р.К.

“ ” _____ 2026 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ЧИЖМИ Данила Едуардовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка калькулятора калорій страв
 керівник проєкту (роботи) к. т. н., доцент, ГРУШКО Світлана Сергіївна
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)
 затверджені наказом вищого навчального закладу від “14” квітня 2026 року № 160
 2. Строк подання студентом проєкту (роботи) 30 травня 2026 року
 3. Вихідні дані до проєкту (роботи) автономний односторінковий веб-додаток у вигляді єдиного HTML-файлу обсягом до 50 КБ; вбудована база даних продуктів харчування українського раціону; адаптивний інтерфейс з підтримкою мобільних пристроїв, планшетів та настільних комп'ютерів; україномовний інтерфейс; робота без підключення до мережі Інтернет
 4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 1) Аналіз предметної області та постановка задачі;
 2) Теоретичні основи розробки односторінкових веб-додатків;
 3) Проектування та програмна реалізація калькулятора калорій;
 4) Апаратне та програмне забезпечення.
 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
- Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ГРУШКО С. С., к.т.н., доцент		
нормоконтроль	ГОЛУБ Т.В., к.т.н., доцент		

7. Дата видачі завдання 14.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз технічного завдання. Огляд існуючих калькуляторів калорій та їх порівняльний аналіз	14.04.26-21.04.26	
2	Дослідження технологій розробки односторінкових веб-додатків (HTML, CSS, JavaScript)	22.04.26-28.04.26	
3	Проектування архітектури додатку та структури бази даних продуктів	29.04.26-03.05.26	
4	Розробка HTML-розмітки, CSS-стилізації та адаптивного дизайну	04.05.26-11.05.26	
5	Програмування логіки роботи додатку	12.05.26-18.05.26	
6	Тестування та налагодження додатку на різних пристроях і браузерах	19.05.26-23.05.26	
7	Оформлення пояснювальної записки	24.05.26-27.05.26	
8	Оформлення графічного та допоміжного матеріалу	28.05.26-30.05.26	

Студент (ка)

Данило ЧИЖМА
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

Світлана ГРУШКО
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи бакалавра: 82 с., 3 табл., 8 рис., 2 додатка, 29 джерел.

КАЛЬКУЛЯТОР КАЛОРІЙ, ВЕБ-ДОДАТОК, ОДНОСТОРІНКОВИЙ ДОДАТОК, HTML, CSS, JAVASCRIPT, КАЛОРІЙНІСТЬ, МАКРОНУТРІЄНТИ

Об'єкт розробки – односторінковий веб-додаток для підрахунку калорійності страв.

Мета роботи – розробка автономного односторінкового веб-додатку калькулятора калорій страв, що забезпечує зручний розрахунок енергетичної цінності раціону для україномовних користувачів.

У дипломній роботі досліджено та проаналізовано сучасні програмні рішення для підрахунку калорій, зокрема MyFitnessPal, Lose It!, Cronometer та FatSecret. Розглянуто їх переваги та недоліки, визначено потребу у створенні автономного рішення з підтримкою української мови та локальних продуктів харчування.

Розроблено повнофункціональний односторінковий веб-додаток калькулятора калорій, реалізований у вигляді єдиного HTML-файлу обсягом до 50 КБ. Створено вбудовану базу даних, що містить 190 продуктів харчування українського раціону, систематизованих за 12 категоріями. Реалізовано адаптивний інтерфейс із використанням технології Flexbox та медіа-запитів CSS, що забезпечує коректне відображення на мобільних пристроях, планшетах та настільних комп'ютерах.

ABSTRACT

Explanatory note to the bachelor's degree thesis: 82 p., 3 tables, 8 fig., 2 app., 29 sources.

CALORIE CALCULATOR, WEB APPLICATION, SINGLE PAGE APPLICATION, HTML, CSS, JAVASCRIPT, CALORIC VALUE, MACRONUTRIENTS

Object of development – single-page web application for calculating the caloric content of meals.

Objective of the work – development of an autonomous single-page web application for calorie calculation that provides convenient estimation of the nutritional value of diet for Ukrainian-speaking users.

In the thesis, modern software solutions for calorie counting have been studied and analyzed, including MyFitnessPal, Lose It!, Cronometer, and FatSecret. Their advantages and disadvantages have been examined, and the need for creating an autonomous solution with Ukrainian language support and local food products has been identified.

A fully functional single-page calorie calculator web application has been developed, implemented as a single HTML file with a size of up to 50 KB. An embedded database containing 190 Ukrainian food products categorized into 12 groups has been created. A responsive interface using Flexbox technology and CSS media queries has been implemented, ensuring correct display on mobile devices, tablets, and desktop computers.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області та постановка задачі	10
1.1 Актуальність розробки калькуляторів калорій у контексті здорового харчування	10
1.2 Огляд наявних рішень та їх порівняльний аналіз	13
1.3 Аналіз вимог до веб-додатків для підрахунку калорійності	18
1.4 Постановка задачі та визначення функціональних вимог до розроблюваного додатку.....	21
1.5 Висновки до розділу 1	25
2 Теоретичні основи розробки односторінкових веб-додатків	26
2.1 Технології розробки односторінкових додатків (HTML, CSS, JavaScript)	26
2.2 Принципи побудови користувацького інтерфейсу для мобільних та десктопних пристроїв	31
2.3 Методи зберігання та обробки даних про харчову цінність продуктів.....	37
2.4 Алгоритми розрахунку калорійності страв на основі інгредієнтів.....	41
2.5 Висновки до розділу 2	46
3 Проєктування та програмна реалізація калькулятора калорій	48
3.1 Проєктування архітектури односторінкового додатку	48
3.2 Розробка структури HTML-документа та семантичної розмітки	49
3.3 Реалізація стилізації інтерфейсу засобами CSS	51
3.4 Програмування логіки роботи додатку на JavaScript	52
3.5 Висновки до розділу 3	53
4 Апаратне та програмне забезпечення	53

4.1 Вимоги до апаратного забезпечення	53
4.2 Вимоги до програмного забезпечення	54
4.3 Технічні характеристики програмного продукту	55
4.4 Методика використання програмного продукту	56
4.5 Висновки до розділу 4	57
Висновки	58
Перелік джерел посилання	60
Додаток А Лістинг коду.....	63
Додаток Б Слайди презентації.....	80

ВСТУП

У сучасному світі проблема здорового харчування набуває все більшого значення. За даними Всесвітньої організації охорони здоров'я, надмірна вага та ожиріння є одними з головних факторів ризику розвитку хронічних захворювань, таких як цукровий діабет, серцево-судинні хвороби та деякі види онкологічних захворювань. Статистичні дані свідчать, що понад 1,9 мільярда дорослих людей у світі мають надлишкову вагу, з них близько 650 мільйонів страждають на ожиріння. В Україні ця проблема також є надзвичайно гострою: за різними оцінками, від 25 до 30 відсотків населення має проблеми з надмірною вагою.

Контроль калорійності споживаної їжі є одним із найефективніших методів підтримання здорової маси тіла та профілактики зазначених захворювань. Науково доведено, що баланс між спожитими та витраченими калоріями є фундаментальним принципом управління вагою. Проте ручний підрахунок калорій є трудомістким процесом, який вимагає знання енергетичної цінності кожного продукту та вміння правильно оцінювати розмір порцій. Саме тому автоматизація цього процесу за допомогою спеціалізованого програмного забезпечення є важливим напрямком розвитку цифрових інструментів для здоров'я.

Стрімкий розвиток інформаційних технологій та широке розповсюдження мобільних пристроїв і персональних комп'ютерів створюють сприятливі умови для впровадження цифрових інструментів контролю харчування. За останні роки ринок додатків для відстеження харчування демонструє стабільне зростання, що свідчить про високий попит користувачів на подібні рішення. Веб-додатки для підрахунку калорій дозволяють користувачам зручно та швидко оцінювати енергетичну цінність страв, планувати раціон та досягати поставлених цілей щодо харчування.

Сучасний ритм життя вимагає від людей швидких та ефективних рішень у всіх сферах діяльності, включаючи контроль власного здоров'я. Користувачі очікують від програмних продуктів простоти використання, доступності на різних пристроях та можливості отримання результату без зайвих зусиль. У цьому

контексті веб-технології надають унікальні переваги: відсутність необхідності встановлення додаткового програмного забезпечення, кросплатформеність та можливість миттєвого доступу через будь-який сучасний браузер.

Особливої актуальності набувають односторінкові веб-додатки, які не потребують встановлення, працюють у будь-якому сучасному браузері та забезпечують миттєвий доступ до функціоналу. Такі додатки характеризуються високою швидкістю, оскільки не потребують перезавантаження сторінки при взаємодії з користувачем. Розробка такого додатку у вигляді єдиного HTML-файлу забезпечує простоту розгортання, портативність та незалежність від серверної інфраструктури. Користувач може зберегти файл на своєму пристрої та використовувати його навіть без підключення до мережі Інтернет, що є суттєвою перевагою порівняно з онлайн-сервісами.

Окрім того, розробка власного калькулятора калорій дозволяє створити рішення, адаптоване до потреб українських користувачів, з урахуванням особливостей національної кухні та локальних продуктів харчування. Існуючі закордонні аналоги часто не містять інформації про традиційні українські страви, що ускладнює їх використання для значної частини населення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність розробки калькуляторів калорій у контексті здорового харчування

Проблема раціонального харчування та підтримання здорової маси тіла є однією з найбільш актуальних у сучасному суспільстві. За даними Всесвітньої організації охорони здоров'я, глобальна поширеність ожиріння майже потроїлася з 1975 року, і ця тенденція продовжує зростати [1]. Надмірна вага та ожиріння визнані п'ятим за значимістю фактором ризику смерті у світі, щорічно спричиняючи загибель щонайменше 2,8 мільйона дорослих людей. Ці статистичні дані свідчать про необхідність розробки ефективних інструментів для контролю харчування та управління вагою.

Енергетичний баланс організму визначається співвідношенням між калоріями, що надходять з їжею, та калоріями, що витрачаються на підтримання життєдіяльності та фізичну активність. Цей фундаментальний принцип термодинаміки лежить в основі всіх сучасних підходів до управління вагою [2]. Коли споживання енергії перевищує її витрати протягом тривалого часу, надлишок накопичується у вигляді жирової тканини, що призводить до збільшення маси тіла. І навпаки, дефіцит калорій спричиняє використання накопичених запасів енергії та зменшення ваги.

Калорія як одиниця вимірювання енергетичної цінності їжі була запропонована ще у XIX столітті американським хіміком Уілбуrom Етуотером, який провів піонерські дослідження з визначення енергетичного вмісту різних продуктів харчування [3]. Система Етуотера, що базується на визначенні вмісту білків, жирів та вуглеводів у продуктах та їх енергетичних коефіцієнтів, залишається основою для розрахунку калорійності і в наш час. Згідно з цією системою, один грам білків або вуглеводів забезпечує приблизно 4 кілокалорії енергії, тоді як один грам жирів — близько 9 кілокалорій.

Сучасні дослідження підтверджують ефективність підрахунку калорій як методу контролю ваги. Систематичний огляд, проведений групою дослідників на чолі з Burke та опублікований у Journal of the American Dietetic Association, показав, що самоконтроль харчування, включаючи ведення щоденника калорій, є одним із найбільш ефективних поведінкових методів для зниження ваги [4]. Учасники досліджень, які регулярно фіксували споживані продукти та їх калорійність, демонстрували значно кращі результати порівняно з контрольними групами. Дослідження, опубліковане у журналі Obesity, виявило, що використання цифрових інструментів для відстеження харчування підвищує ефективність програм зниження ваги на 50-100 відсотків порівняно з традиційними методами [5]. Це пояснюється тим, що електронні калькулятори калорій забезпечують миттєвий зворотний зв'язок, автоматизують трудомісткі розрахунки та підвищують обізнаність користувачів щодо енергетичної цінності споживаних продуктів.

В Україні проблема надмірної ваги населення також набуває загрозливих масштабів. За даними Державної служби статистики України та результатами епідеміологічних досліджень, поширеність ожиріння серед дорослого населення країни становить близько 25-30 відсотків, а надмірну вагу має понад половина дорослих українців [6]. Ці показники наближаються до рівня розвинених західних країн і свідчать про необхідність активного впровадження профілактичних заходів, серед яких важливе місце посідає контроль калорійності харчування.

Особливої актуальності проблема раціонального харчування набуває в умовах сучасного способу життя, який характеризується зниженням фізичної активності та широкою доступністю висококалорійної їжі. Офісна робота, використання автомобільного транспорту, поширення фастфуду та напівфабрикатів створюють передумови для порушення енергетичного балансу [7]. У таких умовах свідомий контроль харчування стає необхідною умовою підтримання здоров'я. Психологічні дослідження показують, що люди систематично недооцінюють калорійність споживаної їжі. Експерименти, проведені дослідниками з Корнельського університету, продемонстрували, що середня похибка в оцінці калорійності порцій може сягати 30-50 відсотків [8]. Це

явище, відоме як «калорійна сліпота», є однією з причин неефективності інтуїтивних спроб контролювати харчування без використання об'єктивних інструментів підрахунку.

Розвиток інформаційних технологій відкриває нові можливості для вирішення проблеми контролю калорійності харчування. Веб-додатки та мобільні застосунки для підрахунку калорій стали невід'ємною частиною цифрової екосистеми здоров'я. За даними аналітичної компанії Grand View Research, глобальний ринок додатків для відстеження здоров'я та фітнесу оцінюється у мільярди доларів і продовжує швидко зростати [9].

Перевагами цифрових калькуляторів калорій є їх доступність, зручність використання та можливість інтеграції з іншими системами моніторингу здоров'я. На відміну від паперових щоденників харчування, електронні інструменти забезпечують автоматичний розрахунок загальної калорійності раціону, візуалізацію даних та можливість аналізу тенденцій споживання протягом тривалого часу [10].

Важливим аспектом актуальності розробки калькуляторів калорій є їх роль у формуванні харчової грамотності населення. Регулярне використання таких інструментів сприяє кращому розумінню енергетичної цінності різних продуктів та страв, що в довгостроковій перспективі допомагає формувати більш здорові харчові звички навіть без постійного підрахунку калорій [11]. Окремо слід зазначити важливість розробки локалізованих рішень, адаптованих до особливостей національних кухонь. Українська кухня має свої унікальні страви та рецепти, інформація про калорійність яких не завжди доступна у міжнародних базах даних. Створення калькулятора калорій з урахуванням традиційних українських продуктів та страв дозволить забезпечити більш точний підрахунок для українських користувачів.

Таким чином, актуальність розробки калькулятора калорій обумовлена глобальною проблемою надмірної ваги населення, доведеною ефективністю методу підрахунку калорій для контролю харчування, широкими можливостями

сучасних веб-технологій та потребою у локалізованих рішеннях для українських користувачів.

1.2 Огляд наявних рішень та їх порівняльний аналіз

Сучасний ринок програмного забезпечення пропонує широкий спектр інструментів для підрахунку калорій, які можна класифікувати за різними критеріями: платформою розгортання, функціональними можливостями, моделлю монетизації та цільовою аудиторією. Проведення детального аналізу існуючих рішень є необхідною передумовою для визначення вимог до розроблюваного додатку та виявлення потенційних напрямків його вдосконалення.

Серед найбільш популярних мобільних додатків для підрахунку калорій слід виділити MyFitnessPal, який станом на 2024 рік має понад 200 мільйонів зареєстрованих користувачів по всьому світу [12]. Цей додаток, розроблений компанією Under Armour, пропонує обширну базу даних продуктів, що налічує понад 14 мільйонів позицій, сканер штрих-кодів для швидкого додавання продуктів, інтеграцію з фітнес-трекерами та соціальні функції для підтримки мотивації користувачів. Основними перевагами MyFitnessPal є його масштабна база даних продуктів, зручний інтерфейс та широкі можливості інтеграції з іншими сервісами та пристроями. Водночас додаток має суттєві недоліки: значна частина функціоналу доступна лише у платній версії, база даних містить багато дублікатів та неточних записів, створених користувачами, а інформація про українські продукти та страви є обмеженою [13].

Іншим популярним рішенням є додаток Lose It!, який також пропонує функції підрахунку калорій та відстеження фізичної активності. Lose It! відрізняється більш інтуїтивним інтерфейсом та наявністю функції розпізнавання їжі за фотографією з використанням технологій штучного інтелекту [14]. Проте цей додаток також

потребує платної підписки для доступу до розширених функцій і має обмежену підтримку неанглійських мов.

Додаток Cronometer позиціонується як інструмент для більш детального аналізу харчування, що відстежує не лише калорії, але й мікронутрієнти — вітаміни та мінерали [15]. Це робить його привабливим для користувачів, які прагнуть не просто контролювати вагу, а й оптимізувати загальний нутрієнтний склад раціону. Cronometer використовує верифіковані дані з офіційних джерел, таких як база даних Міністерства сільського господарства США (USDA), що забезпечує високу точність інформації про продукти.

Серед веб-орієнтованих рішень варто відзначити сервіс FatSecret, який пропонує як мобільні додатки, так і повнофункціональний веб-інтерфейс [16]. FatSecret є безкоштовним і підтримується рекламою, що робить його доступним для широкого кола користувачів. Сервіс має локалізовані версії для різних країн, включаючи україномовний інтерфейс, хоча база даних українських продуктів все ще залишається неповною.

Окрему категорію становлять спеціалізовані калькулятори калорій для конкретних дієт та систем харчування. Наприклад, додаток Carb Manager орієнтований на користувачів кетогенної дієти і акцентує увагу на підрахунку вуглеводів, тоді як Noom поєднує функції калькулятора калорій з елементами психологічного коучингу для формування здорових харчових звичок [17].

У контексті технічної реалізації наявні рішення можна поділити на нативні мобільні додатки, прогресивні веб-додатки (PWA) та традиційні веб-сайти. Нативні додатки забезпечують найкращу продуктивність та доступ до апаратних функцій пристрою, таких як камера для сканування штрих-кодів, але потребують окремої розробки для кожної платформи та встановлення з магазинів додатків [18].

Прогресивні веб-додатки представляють компромісне рішення, поєднуючи переваги веб-технологій з можливостями, характерними для нативних додатків, такими як офлайн-робота та push-сповіщення. Проте розробка PWA вимагає значних зусиль для забезпечення сумісності з різними браузерами та платформами. Традиційні веб-сайти для підрахунку калорій, такі як CalorieKing або NutritionData,

пропонують доступ до інформації про калорійність продуктів через браузер без необхідності встановлення [19]. Такі ресурси зазвичай містять обширні бази даних та інструменти пошуку, але можуть бути менш зручними для регулярного щоденного використання порівняно з мобільними додатками.

Односторінкові веб-додатки (Single Page Applications, SPA) становлять окремий клас веб-рішень, які завантажуються один раз і динамічно оновлюють вміст сторінки без її повного перезавантаження [20]. Такий підхід забезпечує користувацький досвід, близький до нативних додатків, при цьому зберігаючи переваги веб-технологій — кросплатформеність та відсутність необхідності встановлення.

Проведений аналіз існуючих рішень дозволяє виявити низку спільних недоліків, характерних для більшості калькуляторів калорій. По-перше, це залежність від інтернет-з'єднання для доступу до бази даних продуктів та синхронізації даних. По-друге, обмежена підтримка локальних продуктів та страв національних кухонь. По-третє, складність інтерфейсу та надлишковий функціонал, який може відлякувати початківців.

У табл. 1.1 представлено порівняльний аналіз основних характеристик розглянутих рішень.

Аналіз таблиці демонструє, що жодне з розглянутих рішень не забезпечує повноцінної підтримки української мови та локальних продуктів, що підтверджує актуальність розробки спеціалізованого рішення для українських користувачів. Крім того, більшість додатків потребують постійного інтернет-з'єднання та мають обмеження у безкоштовних версіях.

Важливим аспектом аналізу є дослідження моделей монетизації існуючих рішень. Більшість популярних додатків використовують модель freemium, при якій базові функції доступні безкоштовно, а розширений функціонал вимагає оформлення платної підписки [21]. Вартість підписки зазвичай становить від 5 до 20 доларів США на місяць, що може бути значною сумою для українських користувачів з урахуванням різниці у рівні доходів.

Таблиця 1.1 — Порівняльний аналіз існуючих калькуляторів калорій

Критерій	MyFitnessPal	Lose It!	Cronometer	FatSecret
Платформи	iOS, Android, Web	iOS, Android, Web	iOS, Android, Web	iOS, Android, Web
Розмір бази даних	14+ млн	7+ млн	1+ млн	10+ млн
Безкоштовна версія	Обмежена	Обмежена	Обмежена	Повна
Офлайн-режим	Частковий	Частковий	Частковий	Ні
Українська мова	Ні	Ні	Ні	Частково
Українські продукти	Обмежено	Обмежено	Мінімально	Частково

Аналіз таблиці демонструє, що жодне з розглянутих рішень не забезпечує повноцінної підтримки української мови та локальних продуктів, що підтверджує актуальність розробки спеціалізованого рішення для українських користувачів. Крім того, більшість додатків потребують постійного інтернет-з'єднання та мають обмеження у безкоштовних версіях.

Важливим аспектом аналізу є дослідження моделей монетизації існуючих рішень. Більшість популярних додатків використовують модель freemium, при якій базові функції доступні безкоштовно, а розширений функціонал вимагає оформлення платної підписки [21]. Вартість підписки зазвичай становить від 5 до 20 доларів США на місяць, що може бути значною сумою для українських користувачів з урахуванням різниці у рівні доходів.

Альтернативною моделлю є монетизація через рекламу, яка дозволяє зберегти безкоштовний доступ до функціоналу, але може негативно впливати на користувацький досвід. Деякі сервіси також продають знеособлені дані про харчові звички користувачів маркетинговим компаніям, що викликає занепокоєння щодо конфіденційності [22].

З точки зору технічної архітектури, більшість сучасних калькуляторів калорій побудовані за клієнт-серверною моделлю, де мобільний або веб-клієнт взаємодіє з централізованим сервером через API. Такий підхід забезпечує можливість синхронізації даних між пристроями та централізованого оновлення бази даних продуктів, але створює залежність від доступності серверів та інтернет-з'єднання.

Розглядаючи питання точності даних у базах продуктів, слід зазначити, що це є однією з ключових проблем існуючих рішень. Дослідження, опубліковане в *Journal of the Academy of Nutrition and Dietetics*, виявило значні розбіжності між заявленою та фактичною калорійністю продуктів у популярних додатках [23]. Ці розбіжності пояснюються використанням неперифікованих даних, внесених користувачами, варіативністю складу продуктів різних виробників та відсутністю регулярного оновлення інформації.

Окремим напрямком розвитку калькуляторів калорій є інтеграція технологій штучного інтелекту та машинного навчання. Сучасні додатки використовують комп'ютерний зір для розпізнавання продуктів та страв за фотографіями, машинне навчання для прогнозування харчових звичок та персоналізованих рекомендацій, обробку природної мови для спрощення введення даних голосом [24]. Проте технології штучного інтелекту поки що не досягли достатнього рівня точності для повноцінної заміни ручного введення даних. Дослідження показують, що точність розпізнавання їжі за фотографіями становить 60-80 відсотків для простих страв і значно знижується для складних комбінованих страв [25]. Це означає, що користувачам все ще необхідно вручну коригувати результати автоматичного розпізнавання.

Підсумовуючи огляд існуючих рішень, можна констатувати, що ринок калькуляторів калорій є досить насиченим, але існує простір для розробки спеціалізованих рішень, орієнтованих на конкретні сегменти користувачів. Зокрема, існує потреба у простому, безкоштовному та автономному калькуляторі калорій з підтримкою української мови та локальних продуктів, який не потребує реєстрації, встановлення та постійного інтернет-з'єднання.

1.3 Аналіз вимог до веб-додатків для підрахунку калорійності

Розробка якісного програмного продукту вимагає ретельного аналізу та формалізації вимог, які визначають функціональні можливості системи та критерії її якості. У контексті веб-додатків для підрахунку калорійності ці вимоги охоплюють широкий спектр аспектів — від базового функціоналу до питань безпеки та продуктивності.

Функціональні вимоги визначають конкретні можливості, які повинен надавати додаток своїм користувачам. Основною функцією калькулятора калорій є розрахунок енергетичної цінності страв на основі їх інгредієнтів. Це передбачає наявність бази даних продуктів з інформацією про їх калорійність та нутрієнтний склад, механізму пошуку та вибору продуктів, введення кількості (ваги або порції) кожного інгредієнта та автоматичного підрахунку загальної калорійності [26].

База даних продуктів є надважливим компонентом калькулятора калорій. Вона повинна містити достатню кількість позицій для покриття типового раціону цільової аудиторії, при цьому дані мають бути точними та актуальними. Для односторінкового додатку, що працює автономно, база даних вбудовується безпосередньо у код програми, що накладає обмеження на її розмір, але забезпечує незалежність від зовнішніх серверів.

Інтерфейс пошуку продуктів повинен забезпечувати швидкий та зручний доступ до потрібних позицій бази даних. Це може бути реалізовано через текстовий пошук з підтримкою часткового співпадіння, категоризацію продуктів за групами (молочні, м'ясні, овочі тощо), список нещодавно використаних продуктів та можливість додавання власних продуктів користувачем.

Механізм введення кількості продуктів повинен підтримувати різні одиниці вимірювання, оскільки користувачі можуть зважувати продукти у грамах, використовувати стандартні порції (штуки, ложки, склянки) або оцінювати кількість приблизно. Гнучкість у виборі одиниць вимірювання підвищує зручність використання додатку та точність підрахунку.

Додатковими функціональними вимогами можуть бути ведення щоденника харчування з можливістю переглядати історію за попередні дні, встановлення денної норми калорій та відстеження її досягнення, розрахунок не лише калорій, але й макронутрієнтів (білків, жирів, вуглеводів), збереження улюблених страв та рецептів для швидкого повторного використання.

Нефункціональні вимоги описують якісні характеристики системи, що визначають, наскільки добре вона виконує свої функції. У контексті веб-додатків ключовими нефункціональними вимогами є продуктивність, зручність використання, доступність, безпека та сумісність.

Продуктивність веб-додатку визначається швидкістю його завантаження та реагування на дії користувача. Дослідження показують, що користувачі очікують завантаження веб-сторінки за 2-3 секунди, і кожна додаткова секунда затримки призводить до збільшення показника відмов на 7 відсотків [27]. Для односторінкового додатку важливим є мінімізація розміру HTML-файлу та оптимізація JavaScript-коду для забезпечення швидкого первинного завантаження та плавної роботи інтерфейсу.

Зручність використання (usability) є особливо важливою для додатків, призначених для щоденного використання. Інтерфейс калькулятора калорій повинен бути інтуїтивно зрозумілим, не потребувати тривалого навчання та мінімізувати кількість дій для виконання типових операцій. Принципи дизайну, орієнтованого на користувача, передбачають проведення тестування з реальними користувачами та ітеративне вдосконалення інтерфейсу на основі отриманого зворотного зв'язку [28].

Доступність (accessibility) веб-додатків означає можливість їх використання людьми з різними обмеженнями — порушеннями зору, слуху, моторики тощо. Стандарт WCAG (Web Content Accessibility Guidelines) визначає критерії доступності веб-контенту, які включають забезпечення достатнього контрасту кольорів, підтримку навігації з клавіатури, коректне використання семантичної розмітки HTML та альтернативного тексту для зображень.

Вимоги до безпеки для автономного односторінкового додатку є менш критичними, ніж для клієнт-серверних систем, оскільки всі дані зберігаються локально на пристрої користувача. Проте слід враховувати потенційні ризики, пов'язані з використанням локального сховища браузера (localStorage), яке може бути доступне для шкідливих скриптів у разі XSS-атак.

Сумісність веб-додатку визначається його здатністю коректно працювати у різних браузерах, операційних системах та на пристроях з різними розмірами екранів. Принцип адаптивного дизайну (responsive design) передбачає створення інтерфейсу, який автоматично пристосовується до розмірів вікна браузера, забезпечуючи оптимальний користувацький досвід як на настільних комп'ютерах, так і на смартфонах [29].

Вимоги до локалізації включають підтримку української мови як основної мови інтерфейсу та бази даних продуктів. Це передбачає не лише переклад елементів інтерфейсу, але й використання назв продуктів, звичних для українських користувачів, коректну обробку кирилических символів та відповідність місцевим стандартам форматування чисел і дат.

Вимоги до розширюваності визначають можливість подальшого розвитку додатку без суттєвої переробки існуючого коду. Модульна архітектура, чітке розділення логіки та представлення, використання загальноприйнятих патернів проєктування сприяють полегшенню майбутньої підтримки та розширення функціоналу.

Аналіз вимог до калькулятора калорій також повинен враховувати специфіку односторінкового додатку, реалізованого у вигляді єдиного HTML-файлу. Така архітектура накладає певні обмеження, але водночас надає унікальні переваги. Обмеженнями є неможливість використання серверної логіки, обмежений розмір бази даних продуктів, складність реалізації синхронізації даних між пристроями. Перевагами є абсолютна портативність, можливість роботи без інтернету, простота розповсюдження та розгортання.

У табл. 1.2 систематизовано ключові вимоги до розроблюваного калькулятора калорій.

Таблиця 1.2 — Систематизація вимог до калькулятора калорій

Категорія	Вимога	Пріоритет
Функціональні	Пошук продуктів у базі даних	Високий
Функціональні	Розрахунок калорійності страви	Високий
Функціональні	Введення кількості у різних одиницях	Середній
Функціональні	Збереження результатів	Середній
Нефункціональні	Завантаження менше 3 секунд	Високий
Нефункціональні	Адаптивний дизайн	Високий
Нефункціональні	Підтримка основних браузерів	Високий
Нефункціональні	Україномовний інтерфейс	Високий
Нефункціональні	Робота без інтернету	Високий

Формалізація вимог є важливим етапом процесу розробки програмного забезпечення, який дозволяє уникнути непорозумінь між замовником та розробником, забезпечує основу для тестування та приймання готового продукту. У випадку дипломної роботи детальний аналіз вимог демонструє системний підхід до розробки та враховування потреб кінцевих користувачів.

1.4 Постановка задачі та визначення функціональних вимог до розроблюваного додатку

На основі проведеного аналізу предметної області, огляду існуючих рішень та дослідження вимог до веб-додатків для підрахунку калорійності можна сформулювати постановку задачі для даної дипломної роботи. Метою є розробка односторінкового веб-додатку — калькулятора калорій страв, який дозволить користувачам швидко та зручно розраховувати енергетичну цінність споживаної їжі.

Розроблюваний додаток повинен бути реалізований у вигляді єдиного HTML-файлу, що містить всю необхідну розмітку, стилі CSS та програмний код JavaScript. Такий підхід забезпечує максимальну портативність продукту — користувач може зберегти файл на будь-якому пристрої та використовувати його без необхідності встановлення додаткового програмного забезпечення чи підключення до мережі Інтернет. Цільовою аудиторією додатку є україномовні користувачі, які прагнуть контролювати калорійність свого раціону. Це можуть бути люди, що хочуть схуднути або набрати вагу, спортсмени, що слідкують за харчуванням для досягнення спортивних результатів, особи з певними захворюваннями, які потребують дієтичного контролю, або просто люди, зацікавлені у здоровому способі життя.

Функціональні вимоги до додатку можна поділити на обов'язкові, бажані та опціональні. Обов'язкові вимоги визначають мінімальний набір функцій, без яких додаток не зможе виконувати своє основне призначення. До них належать вбудована база даних продуктів з інформацією про калорійність та вміст білків, жирів і вуглеводів, інтерфейс для пошуку та вибору продуктів з бази даних, можливість вказати кількість (вагу) кожного обраного продукту, автоматичний розрахунок загальної калорійності та макронутрієнтів страви, відображення результатів у зручному для сприйняття форматі.

Бажаними вимогами є функції, що суттєво покращують користувацький досвід, але не є критичними для базового функціонування. Серед них можна виділити категоризацію продуктів для зручнішої навігації, історію нещодавно використаних продуктів, можливість збереження результатів у локальному сховищі браузера, функцію очищення списку інгредієнтів для початку нового розрахунку, адаптивний дизайн для коректного відображення на різних пристроях.

Опціональні вимоги визначають потенційні напрямки розширення функціоналу, які можуть бути реалізовані за наявності часу та ресурсів. Це, наприклад, можливість додавання власних продуктів користувачем, розрахунок денної норми калорій на основі параметрів користувача, візуалізація

співвідношення макронутрієнтів у вигляді діаграми, експорт результатів у текстовий формат.

Технічні вимоги до реалізації додатку включають використання стандартних веб-технологій HTML5, CSS3 та JavaScript без залежності від зовнішніх бібліотек та фреймворків. Це забезпечує максимальну сумісність з різними браузерами та мінімізує розмір файлу. Код повинен бути написаний з дотриманням сучасних практик програмування, бути читабельним та документованим.

Вимоги до інтерфейсу користувача передбачають створення простого та інтуїтивно зрозумілого дизайну. Основні елементи інтерфейсу повинні включати поле пошуку для знаходження продуктів у базі даних, список результатів пошуку з можливістю вибору, форму для введення кількості обраного продукту, таблицю або список доданих інгредієнтів, блок з підсумковими результатами калорійності та макронутрієнтів, кнопки керування для очищення списку та інших операцій.

Колірна схема інтерфейсу повинна бути приємною для очей та забезпечувати достатній контраст для комфортного читання. Типографіка повинна бути чіткою та легко читабельною на екранах різних розмірів та роздільної здатності. Елементи інтерфейсу повинні мати достатні розміри для зручної взаємодії як за допомогою миші, так і на сенсорних екранах.

Вимоги до бази даних продуктів визначають її структуру та наповнення. Кожен запис у базі даних повинен містити назву продукту українською мовою, калорійність на 100 грамів продукту, вміст білків, жирів та вуглеводів на 100 грамів, опціонально — категорію продукту для зручності навігації. База даних повинна містити достатню кількість найпоширеніших продуктів українського раціону, включаючи основні категорії (м'ясо, риба, молочні продукти, крупи, овочі, фрукти, хлібобулочні вироби тощо), а також популярні готові страви української кухні.

Вимоги до продуктивності визначають, що первинне завантаження додатку не повинно перевищувати 3 секунд на типовому інтернет-з'єднанні. Пошук у базі даних та розрахунок калорійності повинні виконуватися миттєво, без помітних

затримок для користувача. Додаток повинен плавно працювати на пристроях з обмеженими ресурсами, таких як бюджетні смартфони.

Вимоги до сумісності передбачають підтримку поточних версій основних браузерів: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge. Додаток повинен коректно працювати на настільних комп'ютерах з операційними системами Windows, macOS та Linux, а також на мобільних пристроях з Android та iOS.

Структура односторінкового додатку визначається специфікою його реалізації у єдиному HTML-файлі. Файл буде містити три основні секції: секцію розмітки HTML з семантичною структурою документа та елементами інтерфейсу, секцію стилів CSS, вбудовану у тег `<style>`, з описом візуального оформлення, секцію скриптів JavaScript, вбудовану у тег `<script>`, з програмною логікою та базою даних продуктів.

Така структура забезпечує самодостатність додатку — всі необхідні компоненти містяться в одному файлі, який можна відкрити у будь-якому браузері без додаткових залежностей. Це значно спрощує розповсюдження та використання додатку кінцевими користувачами.

Критерії успішності проєкту визначають, за якими ознаками можна оцінити досягнення поставленої мети. Основними критеріями є коректність розрахунку калорійності та макронутрієнтів на основі введених даних, зручність та інтуїтивність користувацького інтерфейсу, швидкість роботи додатку, сумісність з основними браузерами та пристроями, відповідність потребам цільової аудиторії.

Обмеження проєкту визначають рамки, в яких здійснюється розробка. Основними обмеженнями є реалізація у вигляді єдиного HTML-файлу без серверної частини, обмежений розмір бази даних продуктів через її вбудовування у код, відсутність можливості синхронізації даних між пристроями, використання лише стандартних веб-технологій без зовнішніх залежностей.

Етапи реалізації проєкту включають проєктування архітектури та інтерфейсу додатку, формування бази даних продуктів з інформацією про їх калорійність, розробку HTML-розмітки та структури документа, створення стилів CSS для

візуального оформлення, програмування логіки роботи на JavaScript, тестування та налагодження додатку, документування розробленого програмного продукту.

Таким чином, постановка задачі дипломної роботи передбачає розробку функціонального, зручного та портативного веб-додатку для розрахунку калорійності страв, який задовольняє виявлені потреби цільової аудиторії та реалізується з урахуванням специфіки односторінкової архітектури.

1.5 Висновки до розділу 1

У першому розділі дипломної роботи проведено комплексний аналіз предметної області та здійснено постановку задачі на розробку калькулятора калорій страв.

Досліджено актуальність проблеми контролю калорійності харчування в контексті глобальної епідемії ожиріння та пов'язаних з нею захворювань. Встановлено, що регулярний підрахунок калорій є науково обґрунтованим методом управління вагою, ефективність якого підтверджена численними дослідженнями. Показано, що цифрові інструменти для відстеження харчування значно підвищують ефективність програм контролю ваги порівняно з традиційними методами.

Проведено порівняльний аналіз існуючих рішень у сфері калькуляторів калорій, включаючи популярні мобільні додатки та веб-сервіси. Виявлено їх основні переваги та недоліки, серед яких обмежена підтримка української мови, залежність від інтернет-з'єднання, необхідність реєстрації та платні функції. Обґрунтовано потребу у створенні простого автономного рішення, орієнтованого на українських користувачів.

Систематизовано функціональні та нефункціональні вимоги до веб-додатків для підрахунку калорійності. Визначено ключові аспекти, що впливають на якість такого програмного продукту: продуктивність, зручність використання,

доступність, безпека та сумісність. Особливу увагу приділено специфіці односторінкових веб-додатків та їх перевагам.

Сформульовано детальну постановку задачі, що включає визначення мети розробки, опис цільової аудиторії, перелік обов'язкових, бажаних та опціональних функцій, технічні вимоги до реалізації, вимоги до інтерфейсу та бази даних, критерії успішності та обмеження проєкту.

Результати аналізу, проведеного у першому розділі, створюють теоретичне та методологічне підґрунтя для подальших етапів розробки калькулятора калорій, що будуть висвітлені у наступних розділах дипломної роботи.

2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ОДНОСТОРІНКОВИХ ВЕБ-ДОДАТКІВ

2.1 Технології розробки односторінкових додатків (HTML, CSS, JavaScript)

Сучасна веб-розробка базується на трьох фундаментальних технологіях, які разом забезпечують створення інтерактивних веб-додатків: мові розмітки HTML, мові стилів CSS та мові програмування JavaScript. Кожна з цих технологій виконує специфічну роль у структурі веб-додатку, і їх глибоке розуміння є необхідною передумовою для розробки якісного програмного продукту.

HTML (HyperText Markup Language) є стандартною мовою розмітки для створення веб-сторінок та веб-додатків. Розроблена Тімом Бернерсом-Лі у 1991 році, ця мова пройшла тривалий шлях еволюції від простого інструменту для форматування наукових документів до потужної платформи для створення складних інтерактивних застосунків [1]. Поточною версією стандарту є HTML5, яка була офіційно рекомендована консорціумом W3C у 2014 році та продовжує розвиватися як «жива» специфікація під керівництвом організації WHATWG.

Основною концепцією HTML є використання тегів для визначення структури та семантики документа. Теги представляють собою спеціальні маркери, що вказують браузеру, як інтерпретувати та відображати вміст. Наприклад, тег `<p>` визначає абзац тексту, тег `<h1>` — заголовок першого рівня, а тег `<div>` — універсальний контейнер для групування елементів.

HTML5 привніс значні покращення порівняно з попередніми версіями стандарту. Серед найважливіших нововведень слід виділити нові семантичні елементи (`<header>`, `<footer>`, `<nav>`, `<article>`, `<section>`), які дозволяють створювати більш структуровану та зрозумілу розмітку. Ці елементи не лише покращують читабельність коду для розробників, але й допомагають пошуковим системам та допоміжним технологіям краще розуміти структуру сторінки [2].

Важливим аспектом HTML5 є розширені можливості для роботи з формами. Нові типи полів введення, такі як `email`, `url`, `number`, `date`, `range`, забезпечують вбудовану валідацію даних та відображення спеціалізованих елементів керування на мобільних пристроях. Атрибути `placeholder`, `required`, `pattern` дозволяють налаштовувати поведінку форм без використання JavaScript.

HTML5 також запровадив потужні API для роботи з локальним сховищем (`localStorage` та `sessionStorage`), що є особливо важливим для односторінкових додатків. Ці інтерфейси дозволяють зберігати дані на стороні клієнта у вигляді пар ключ-значення, забезпечуючи персистентність інформації між сеансами роботи з додатком [3].

Структура типового HTML-документа включає декларацію типу документа (`<!DOCTYPE html>`), кореневий елемент `<html>`, секцію метаданих `<head>` та секцію вмісту `<body>`. У секції `<head>` розміщуються метадані документа, посилання на стилі та скрипти, тоді як секція `<body>` містить видимий вміст сторінки.

На рис. 2.1 представлено базову структуру HTML5-документа, яка є відправною точкою для розробки будь-якого веб-додатку.

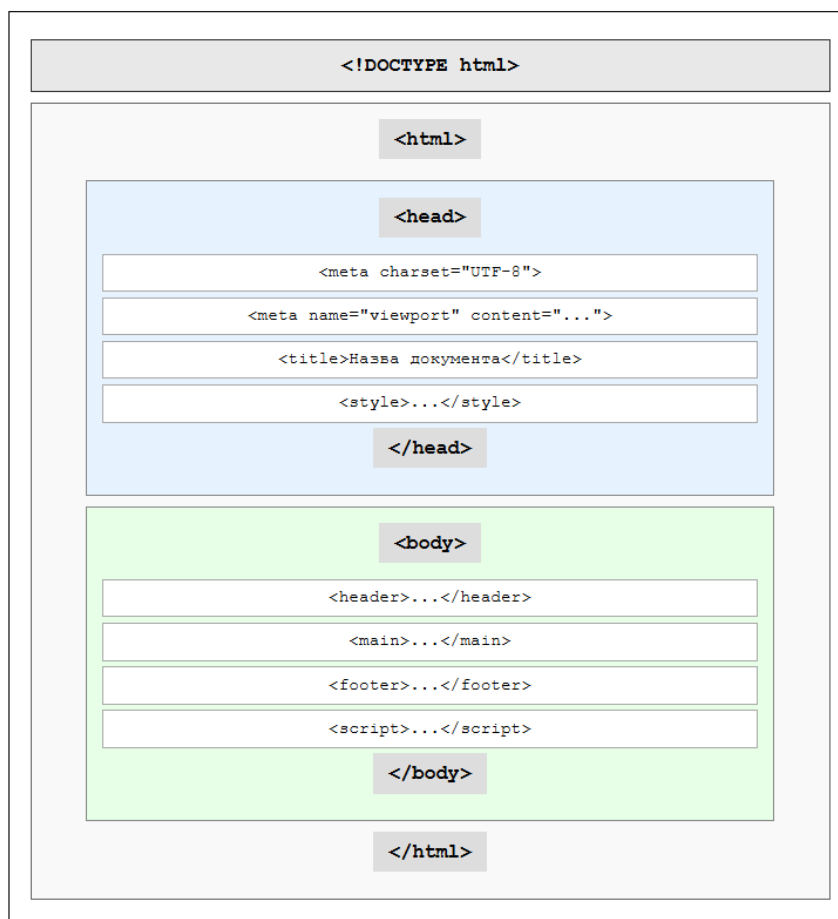


Рисунок 2.1 – Базова структура HTML5-документа

CSS (Cascading Style Sheets) є мовою опису зовнішнього вигляду документа, написаного мовою розмітки. Перша версія CSS була представлена у 1996 році, і з того часу мова зазнала значного розвитку. Поточний стандарт CSS3 організований як набір модулів, кожен з яких розвивається незалежно, що дозволяє швидше впроваджувати нові можливості [4].

Основним принципом CSS є каскадування — механізм визначення пріоритетів стилів при наявності конфлікуючих правил. Каскад враховує походження стилю (браузерні, користувацькі, авторські), специфічність селектора та порядок оголошення правил. Розуміння каскаду є критично важливим для ефективної роботи зі стилями.

Селектори CSS визначають, до яких елементів HTML застосовуються стилі. Існує широкий спектр типів селекторів, серед яких селектори тегів (наприклад, `p`, `div`), селектори класів (`.className`), селектори ідентифікаторів (`#idName`),

селектори атрибутів ([type="text"]), псевдокласи (:hover, :focus, :first-child) та псевдоелементи (::before, ::after).

Блокова модель (Box Model) є фундаментальною концепцією CSS, яка визначає, як елементи займають простір на сторінці. Кожен елемент представляється як прямокутний блок, що складається з області вмісту (content), внутрішнього відступу (padding), рамки (border) та зовнішнього відступу (margin). Властивість box-sizing дозволяє змінювати спосіб розрахунку розмірів елемента [5].

На рис. 2.2 представлено візуалізацію блокової моделі CSS.

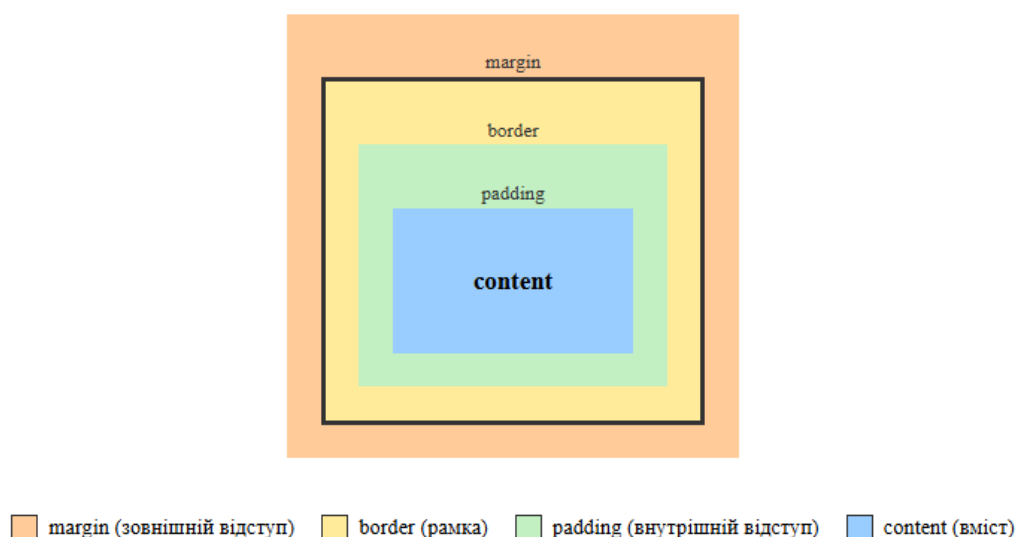


Рисунок 2.2 – Блокова модель CSS (Box Model)

CSS3 запровадив потужні інструменти для створення сучасних макетів — Flexbox та Grid. Flexbox (Flexible Box Layout) призначений для одновимірного розташування елементів у рядку або колонці, забезпечуючи гнучкий розподіл простору та вирівнювання [6]. Grid Layout дозволяє створювати двовимірні макети з явним визначенням рядків та колонок, що робить його ідеальним інструментом для побудови складних інтерфейсів.

Важливими можливостями CSS3 є також трансформації, переходи та анімації. Трансформації (transform) дозволяють змінювати положення, розмір та форму елементів. Переходи (transition) забезпечують плавну зміну властивостей

при зміні стану елемента. Анімації (@keyframes та animation) дають можливість створювати складні послідовності змін без використання JavaScript.

Media-запити (Media Queries) є ключовим інструментом для створення адаптивного дизайну. Вони дозволяють застосовувати різні стилі залежно від характеристик пристрою, таких як ширина екрана, орієнтація, роздільна здатність та інші параметри. Це забезпечує оптимальне відображення веб-додатку на пристроях різних типів та розмірів [7].

JavaScript є мовою програмування, яка забезпечує інтерактивність веб-сторінок. Розроблена Бренданом Айком у 1995 році для браузера Netscape Navigator, ця мова стала фактичним стандартом для клієнтського веб-програмування. Сучасний JavaScript, стандартизований як ECMAScript, є потужною мовою з підтримкою об'єктно-орієнтованого, функціонального та подієво-орієнтованого програмування [8].

JavaScript є динамічно типізованою мовою, що означає визначення типів змінних під час виконання, а не під час компіляції. Мова підтримує примітивні типи даних (числа, рядки, булеві значення, null, undefined, Symbol, BigInt) та об'єктні типи (об'єкти, масиви, функції).

Важливою особливістю JavaScript є його подієво-орієнтована модель виконання. Браузер генерує події у відповідь на дії користувача (клік, введення тексту, прокручування) та системні події (завантаження сторінки, зміна розміру вікна). Програміст може визначати обробники подій — функції, які виконуються при виникненні певної події.

DOM (Document Object Model) є програмним інтерфейсом для HTML-документів, який представляє структуру документа як дерево об'єктів. JavaScript взаємодіє з DOM для читання та модифікації вмісту, структури та стилів веб-сторінки. Основні методи для роботи з DOM включають document.getElementById(), document.querySelector(), document.querySelectorAll() для вибору елементів, а також властивості та методи для їх модифікації [9].

ECMAScript 6 (ES6), випущений у 2015 році, привніс революційні зміни у мову JavaScript. Серед ключових нововведень — оголошення змінних за

допомогою `let` та `const` з блоковою областю видимості, стрілкові функції з лексичним зв'язуванням `this`, класи як синтаксичний цукор для прототипного наслідування, шаблонні рядки для зручного форматування тексту, деструктуризація для розпакування значень з масивів та об'єктів, модулі для організації коду [10].

Асинхронне програмування є важливим аспектом JavaScript, оскільки багато операцій (мережеві запити, читання файлів, таймери) виконуються асинхронно. Історично для обробки асинхронних операцій використовувалися `callback`-функції, які при складній логіці призводили до так званого «callback hell». Сучасний JavaScript пропонує елегантніші рішення — проміси (Promises) та синтаксис `async/await`, які значно спрощують написання та читання асинхронного коду.

Для односторінкових додатків важливим є API Web Storage, який включає `localStorage` та `sessionStorage`. `localStorage` зберігає дані без обмеження терміну дії — вони залишаються доступними навіть після закриття браузера. `sessionStorage` зберігає дані лише протягом сеансу роботи — вони видаляються при закритті вкладки браузера. Обидва сховища надають простий інтерфейс з методами `setItem()`, `getItem()`, `removeItem()` та `clear()`.

Таким чином, комбінація HTML, CSS та JavaScript забезпечує повний набір інструментів для створення сучасних односторінкових веб-додатків. HTML визначає структуру та семантику, CSS відповідає за візуальне оформлення, а JavaScript забезпечує інтерактивність та програмну логіку.

2.2 Принципи побудови користувацького інтерфейсу для мобільних та десктопних пристроїв

Проектування користувацького інтерфейсу є важливим етапом розробки будь-якого програмного продукту. Якість інтерфейсу безпосередньо впливає на задоволеність користувачів, ефективність виконання завдань та загальний успіх

додатку. У контексті веб-додатків особливого значення набуває забезпечення оптимального користувацького досвіду на пристроях різних типів та розмірів.

Концепція User Experience (UX) охоплює всі аспекти взаємодії користувача з продуктом, включаючи практичні, експериментальні, афективні та значущі аспекти цієї взаємодії. Термін був популяризований Доном Норманом у 1990-х роках і з того часу став центральним поняттям у сфері проектування інтерфейсів [11]. UX-дизайн зосереджується на розумінні потреб користувачів та створенні продуктів, які є не лише функціональними, але й приємними у використанні.

User Interface (UI) дизайн є підмножиною UX і фокусується безпосередньо на візуальних та інтерактивних елементах інтерфейсу. UI-дизайнер працює над типографікою, колірними схемами, іконками, кнопками, формами та іншими компонентами, з якими безпосередньо взаємодіє користувач. Ефективний UI-дизайн поєднує естетичну привабливість з функціональністю та зручністю використання.

Серед фундаментальних принципів проектування інтерфейсів слід виділити евристики юзабіліті, сформульовані Якобом Нільсеном [12]. Ці десять принципів є загальновизнаними орієнтирами для оцінки та покращення інтерфейсів.

Перший принцип — видимість стану системи — передбачає, що система повинна завжди інформувати користувача про те, що відбувається, через відповідний зворотний зв'язок у розумний час. Для калькулятора калорій це означає негайне відображення результатів розрахунку при додаванні продуктів та візуальну індикацію успішного виконання операцій.

Другий принцип — відповідність системи реальному світу — вимагає використання мови та концепцій, знайомих користувачу. У контексті калькулятора калорій це означає використання загальноприйнятих назв продуктів, зрозумілих одиниць вимірювання (грами, порції) та логічного порядку операцій, який відповідає очікуванням користувача.

Третій принцип — свобода та контроль користувача — передбачає надання можливості скасовувати та повторювати дії. Користувач повинен мати можливість

видалити помилково доданий продукт, змінити введену кількість або повністю очистити список інгредієнтів.

Четвертий принцип — послідовність та стандарти — означає, що однакові елементи повинні виглядати та поводитися однаково по всьому інтерфейсу. Кнопки, поля введення та інші компоненти повинні мати єдиний стиль та передбачувану поведінку.

Адаптивний дизайн (Responsive Web Design) є підходом до веб-розробки, який забезпечує оптимальне відображення та взаємодію на різних пристроях. Термін був введений Ітаном Маркоттом у 2010 році [13]. Адаптивний дизайн базується на трьох основних техніках: гнучкій сітці (fluid grid), гнучких зображеннях (flexible images) та медіа-запитах (media queries).

Гнучка сітка передбачає використання відносних одиниць вимірювання (відсотків, em, rem, vw, vh) замість фіксованих пікселів для визначення розмірів елементів. Це дозволяє макету пропорційно масштабуватися залежно від розміру вікна браузера. Сучасні CSS-технології Flexbox та Grid значно спрощують створення гнучких макетів.

Гнучкі зображення автоматично масштабуються відповідно до розміру контейнера, запобігаючи виходу за межі батьківського елемента. Це досягається встановленням властивості max-width: 100% для зображень. HTML5 також надає елемент <picture> та атрибут srcset, які дозволяють завантажувати різні версії зображень залежно від характеристик пристрою.

Медіа-запити дозволяють застосовувати різні стилі CSS залежно від параметрів пристрою. Найчастіше використовуються запити на основі ширини екрана, що дозволяє визначати контрольні точки (breakpoints) для перемикання між різними макетами. Типові контрольні точки відповідають розмірам мобільних телефонів (до 576px), планшетів (577-768px), ноутбуків (769-1024px) та настільних комп'ютерів (понад 1024px) [14].

Підхід «Mobile First» передбачає проєктування інтерфейсу спочатку для мобільних пристроїв, а потім його розширення для більших екранів. Цей підхід змушує розробника зосередитися на найважливішому функціоналі та контенті,

уникаючи надмірної складності інтерфейсу. З технічної точки зору, Mobile First означає написання базових стилів для мобільних пристроїв та використання медіа-запитів min-width для додавання стилів на більших екранах.

На рис. 2.3 представлено принцип адаптивного дизайну з контрольними точками.

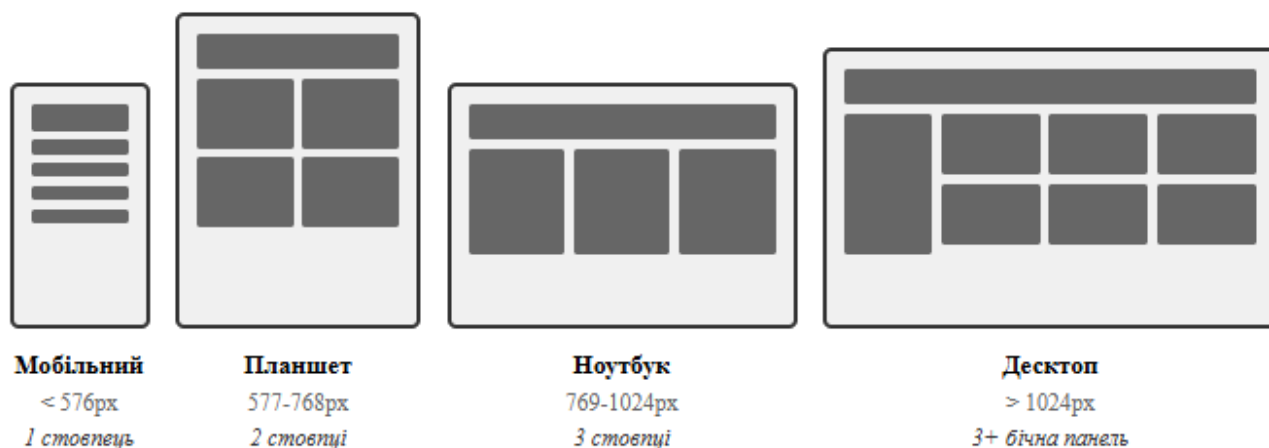


Рисунок 2.3 – Контрольні точки адаптивного дизайну

Типографіка є важливим аспектом UI-дизайну, який впливає на читабельність та сприйняття контенту. Вибір шрифтів повинен враховувати їх читабельність на екранах різних розмірів та роздільної здатності. Системні шрифти (system fonts) забезпечують найкращу продуктивність та природний вигляд на кожній платформі. Веб-шрифти (наприклад, Google Fonts) надають більше можливостей для брендингу, але збільшують час завантаження сторінки [15].

Розмір шрифту повинен бути достатнім для комфортного читання. Загальноприйнятим мінімумом для основного тексту є 16 пікселів. На мобільних пристроях може знадобитися збільшення розміру шрифту для компенсації меншої відстані між очима та екраном. Використання відносних одиниць (em, rem) замість пікселів дозволяє масштабувати типографіку пропорційно налаштуванням браузера.

Міжрядковий інтервал (line-height) суттєво впливає на читабельність тексту. Оптимальним вважається значення 1.4-1.6 від розміру шрифту для основного

тексту. Занадто малий інтервал ускладнює читання довгих текстів, тоді як занадто великий може порушувати візуальний зв'язок між рядками.

Колірна схема інтерфейсу повинна забезпечувати достатній контраст для читабельності тексту та розрізнення елементів. Стандарт WCAG рекомендує співвідношення контрасту щонайменше 4.5:1 для звичайного тексту та 3:1 для великого тексту [16]. Інструменти перевірки контрасту дозволяють оцінити відповідність колірних комбінацій цим стандартам.

Психологія кольору також впливає на сприйняття інтерфейсу. Зелений колір асоціюється зі здоров'ям та природністю, що робить його доречним для додатку, пов'язаного з харчуванням. Синій викликає довіру та професійність. Червоний привертає увагу та може використовуватися для акцентування важливої інформації або попереджень.

Елементи керування повинні мати достатні розміри для зручної взаємодії. На сенсорних екранах мінімальний рекомендований розмір інтерактивних елементів становить 44x44 пікселі за рекомендаціями Apple та 48x48 пікселів за рекомендаціями Google [17]. Відстань між інтерактивними елементами повинна запобігати випадковим натисканням.

Форми є критично важливим компонентом калькулятора калорій, оскільки основна взаємодія користувача з додатком відбувається через введення даних. Ефективний дизайн форм передбачає чіткі мітки для кожного поля, логічне групування пов'язаних полів, валідацію даних з інформативними повідомленнями про помилки, підказки та автозаповнення для прискорення введення.

Зворотний зв'язок (feedback) інформує користувача про результати його дій. Візуальний зворотний зв'язок може включати зміну кольору кнопки при натисканні, анімацію при успішному додаванні продукту, повідомлення про помилки та попередження. Зворотний зв'язок повинен бути миттєвим та пропорційним значущості дії.

На рис. 2.4 представлено приклад структури користувацького інтерфейсу калькулятора калорій.

КАЛЬКУЛЯТОР КАЛОРИЙ

(Header)

Пошук продуктів:

🔍 Введіть назву продукту...

Результати пошуку:

- Яблуко (52 ккал/100г) [Додати]
- Яблучний сік (46 ккал/100г) [Додати]
- Яблучний пиріг (265 ккал/100г) [Додати]

Обрані продукти:

Продукт	Вага (г)	Ккал	Дія
Куряча грудка	150	248	[X]
Рис відварний	200	260	[X]
Огірок	100	15	[X]

[Очистити]

ПІДСУМОК

Калорії: 523 ккал

Білки: 45.2 г | Жири: 8.1 г | Вугл: 62 г

(Footer)

Рисунок 2.4 – Структура інтерфейсу калькулятора калорій

Доступність (Accessibility) веб-інтерфейсів означає можливість їх використання людьми з різними обмеженнями. Стандарт WCAG (Web Content Accessibility Guidelines) визначає чотири основні принципи доступності: сприйнятність (perceivable), керуваність (operable), зрозумілість (understandable) та надійність (robust) [18]. Дотримання цих принципів забезпечує інклюзивність додатку та розширює потенційну аудиторію.

Семантична розмітка HTML є основою доступності. Використання відповідних тегів (заголовки, списки, таблиці, форми з мітками) дозволяє допоміжним технологіям, таким як екранні читачі, правильно інтерпретувати та озвучувати вміст сторінки. Атрибути ARIA (Accessible Rich Internet Applications) дозволяють додавати семантику до елементів, які її не мають за замовчуванням.

Підтримка навігації з клавіатури є обов'язковою вимогою доступності. Всі інтерактивні елементи повинні бути доступні за допомогою клавіші Tab, а їх активація — за допомогою Enter або Space. Візуальний індикатор фокусу (focus indicator) повинен бути чітко видимим для забезпечення орієнтації користувача.

Таким чином, проєктування користувацького інтерфейсу вимагає комплексного підходу, який враховує принципи юзабіліті, адаптивного дизайну, типографіки, колористики та доступності. Дотримання цих принципів забезпечує створення інтерфейсу, який є зручним, привабливим та доступним для широкого кола користувачів.

2.3 Методи зберігання та обробки даних про харчову цінність продуктів

Ефективне зберігання та обробка даних про харчову цінність продуктів є ключовим аспектом розробки калькулятора калорій. У контексті односторінкового веб-додатку, що працює повністю на стороні клієнта, ці питання набувають

особливої специфіки, оскільки традиційні серверні бази даних не можуть бути використані.

Харчова цінність продуктів характеризується комплексом показників, що визначають їх біологічну та енергетичну цінність для організму людини. Основними показниками, що використовуються у калькуляторах калорій, є енергетична цінність (калорійність), вміст макронутрієнтів (білки, жири, вуглеводи), вміст мікронутрієнтів (вітаміни, мінерали), вміст харчових волокон та води [19].

Калорійність вимірюється у кілокалоріях (ккал) або кілоджоулях (кДж) і показує кількість енергії, яку організм може отримати при засвоєнні продукту. Один грам білків або вуглеводів забезпечує приблизно 4 ккал, один грам жирів — 9 ккал, а один грам етанолу — 7 ккал. Ці коефіцієнти, відомі як фактори Етуотера, є основою для розрахунку калорійності продуктів [20].

Для цілей калькулятора калорій достатнім є зберігання інформації про калорійність та вміст основних макронутрієнтів — білків, жирів та вуглеводів. Ці дані дозволяють задовольнити потреби більшості користувачів, які прагнуть контролювати енергетичну цінність раціону та баланс макронутрієнтів.

Джерелами даних про харчову цінність продуктів є офіційні державні бази даних, такі як база даних USDA (Міністерства сільського господарства США), яка містить детальну інформацію про тисячі продуктів і є визнаним стандартом у галузі. В Україні аналогічну роль виконують таблиці хімічного складу харчових продуктів, затверджені Міністерством охорони здоров'я [21].

Структура даних для зберігання інформації про продукти у JavaScript може бути реалізована різними способами. Найпростішим є масив об'єктів, де кожен об'єкт представляє окремий продукт з його характеристиками. Така структура забезпечує простоту реалізації та зрозумілість коду.

На рис. 2.5 представлено структуру даних для зберігання інформації про продукти.

```

const products = [
  {
    id: 1,
    name: "Яблуко",      ← Назва продукту
    category: "Фрукти",  ← Категорія
    calories: 52,        ← Калорій на 100г
    protein: 0.3,        ← Білки, г/100г
    fat: 0.2,           ← Жири, г/100г
    carbs: 14          ← Вуглеводи, г/100г
  },
  {
    id: 2,
    name: "Куряча грудка",
    category: "М'ясо",
    calories: 165,
    protein: 31,
    fat: 3.6,
    carbs: 0
  },
  // ... інші продукти
];

```

Рисунок 2.5 – Структура даних для зберігання інформації про продукти

Альтернативним підходом є використання об'єкта з ключами у вигляді ідентифікаторів продуктів. Такий підхід забезпечує швидший доступ до конкретного продукту за його ідентифікатором (складність $O(1)$ замість $O(n)$ для пошуку у масиві), але ускладнює ітерацію та фільтрацію даних.

Для оптимізації пошуку може бути використана додаткова індексна структура. Наприклад, можна створити окремий об'єкт, що зіставляє назви продуктів (або їх нормалізовані версії) з ідентифікаторами. Це дозволяє швидко знаходити продукти за назвою без необхідності перебору всього масиву.

Категоризація продуктів полегшує навігацію та пошук у базі даних. Типовими категоріями для продуктів харчування є молочні продукти, м'ясо та птиця, риба та морепродукти, крупи та зернові, хлібобулочні вироби, овочі, фрукти та ягоди, горіхи та насіння, олії та жири, напої, кондитерські вироби, готові страви. Категоризація може бути реалізована як поле об'єкта продукту або як окрема структура зв'язків.

Пошук продуктів є критичною функцією калькулятора калорій, оскільки від його ефективності залежить зручність використання додатку. Найпростішим методом пошуку є лінійний перебір масиву з перевіркою входження пошукового запиту у назву продукту. Для невеликих баз даних (до кількох сотень продуктів) цей метод є достатньо швидким.

Для покращення користувацького досвіду пошук повинен бути нечутливим до регістру та підтримувати часткове співпадіння. Метод `toLowerCase()` забезпечує нечутливість до регістру, а метод `includes()` — пошук підрядка. Додатково може бути реалізований нечіткий пошук (fuzzy search), який враховує можливі помилки у написанні.

Локальне сховище браузера (Web Storage API) надає можливість зберігати дані на стороні клієнта. `localStorage` зберігає дані без обмеження терміну, тоді як `sessionStorage` — лише протягом сеансу роботи. Обидва сховища використовують простий інтерфейс ключ-значення, де значення зберігаються як рядки [22].

Для зберігання складних структур даних (масивів, об'єктів) у `localStorage` необхідно використовувати серіалізацію та десеріалізацію JSON. Метод `JSON.stringify()` перетворює об'єкт JavaScript у рядок JSON, а метод `JSON.parse()` — виконує зворотнє перетворення.

Обмеженням `localStorage` є його обсяг — зазвичай близько 5-10 МБ залежно від браузера. Для калькулятора калорій цього обсягу достатньо для зберігання історії використання, налаштувань користувача та власних продуктів. Проте слід враховувати, що `localStorage` є синхронним API, і операції з великими обсягами даних можуть блокувати основний потік виконання.

`IndexedDB` є більш потужною альтернативою `localStorage` для зберігання структурованих даних. Це асинхронна об'єктно-орієнтована база даних, що підтримує індекси, транзакції та значно більші обсяги зберігання. Проте складність API `IndexedDB` робить його надмірним для простих додатків з невеликими обсягами даних [23].

Валідація даних є важливим аспектом забезпечення цілісності інформації. При введенні кількості продукту необхідно перевіряти, що значення є числом,

невід'ємним, та знаходиться у розумному діапазоні. Валідація на стороні клієнта забезпечує миттєвий зворотний зв'язок для користувача.

Нормалізація даних у базі продуктів передбачає зберігання всіх значень у стандартизованому форматі. Калорійність та вміст нутрієнтів зазвичай наводяться на 100 грамів продукту, що дозволяє легко перераховувати значення для будь-якої кількості.

Оновлення та розширення бази даних продуктів є важливим аспектом підтримки додатку. У випадку вбудованої бази даних оновлення вимагає модифікації коду та випуску нової версії файлу. Альтернативним підходом є надання користувачам можливості додавати власні продукти, які зберігаються у `localStorage`.

Таким чином, зберігання та обробка даних про харчову цінність продуктів у односторінковому веб-додатку базується на вбудованій базі даних у форматі JavaScript-структур та використанні локального сховища браузера для персистентного зберігання даних користувача. Такий підхід забезпечує автономність роботи додатку та достатню гнучкість для задоволення потреб цільової аудиторії.

2.4 Алгоритми розрахунку калорійності страв на основі інгредієнтів

Розрахунок калорійності страв на основі їх інгредієнтів є центральною функцією калькулятора калорій. Алгоритми, що використовуються для цих розрахунків, базуються на фундаментальних принципах нутриціології та арифметичних операціях, але їх коректна реалізація вимагає врахування низки факторів.

Базовий принцип розрахунку калорійності полягає у тому, що загальна енергетична цінність страви дорівнює сумі енергетичних цінностей усіх її

інгредієнтів. Це впливає з закону збереження енергії та є загальноприйнятим підходом у дієтології. Формула розрахунку має вигляд:

$$E_{\text{страви}} = \Sigma (E_{\text{продукту}} \times m_{\text{продукту}} / 100), \quad (2.1)$$

де $E_{\text{страви}}$ — загальна калорійність страви (ккал);

$E_{\text{продукту}}$ — калорійність продукту на 100 грамів (ккал/100г);

$m_{\text{продукту}}$ — маса продукту у складі страви (г) [24].

Аналогічний підхід застосовується для розрахунку вмісту макронутрієнтів — білків, жирів та вуглеводів. Загальний вміст кожного нутрієнта у страві є сумою його вмісту у всіх інгредієнтах з урахуванням їх маси.

На рис. 2.8 представлено блок-схему алгоритму розрахунку калорійності страви.

Важливим аспектом є обробка одиниць вимірювання. Хоча базовим стандартом є грами, користувачі можуть бажати вводити кількість у інших одиницях — штуках, ложках, склянках тощо. Для підтримки різних одиниць вимірювання необхідно зберігати коефіцієнти перетворення та виконувати конвертацію перед розрахунком.

Типові одиниці вимірювання та їх приблизні еквіваленти у грамах наведено у табл. 2.1.

Таблиця 2.1 — Одиниці вимірювання та коефіцієнти перетворення

Одиниця	Скорочення	Приблизна вага (г)
Грам	г	1
Кілограм	кг	1000
Чайна ложка	ч.л.	5
Столова ложка	ст.л.	15
Склянка (250 мл)	скл.	250
Штука	шт.	varies

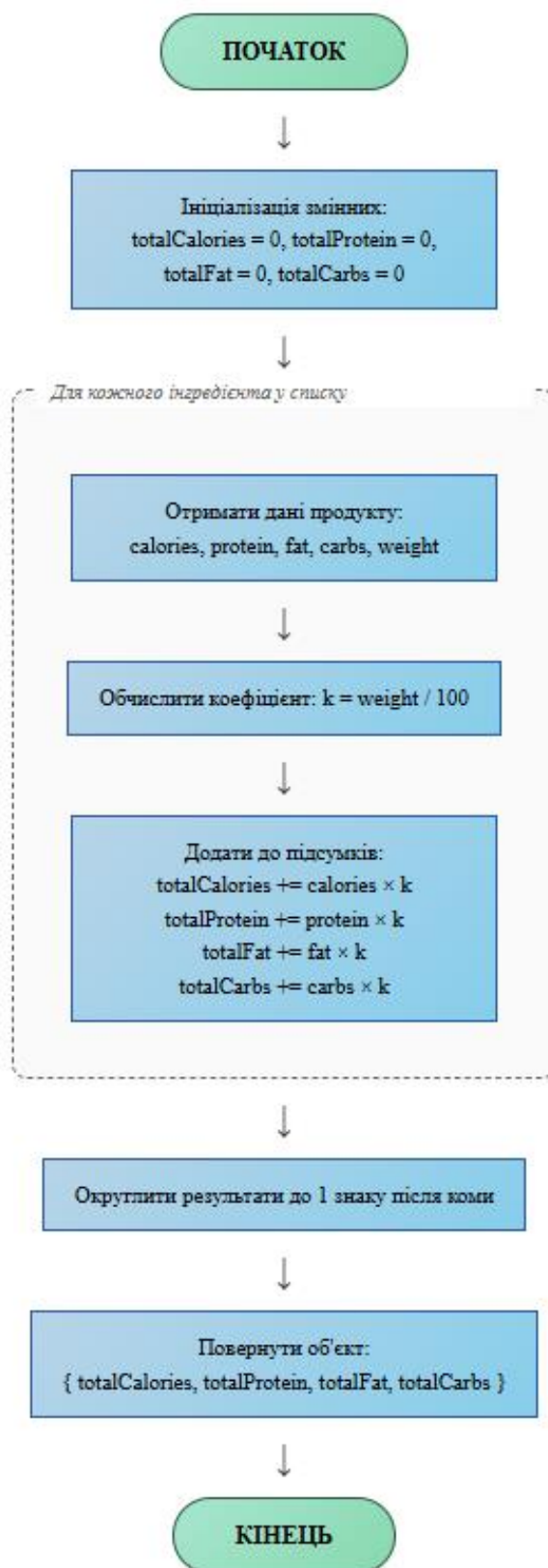


Рисунок 2.8 – Блок-схема алгоритму розрахунку калорійності

Складнішим є врахування термічної обробки продуктів. При варінні, смаженні, запіканні та інших способах приготування відбуваються зміни маси та складу продуктів. Наприклад, м'ясо при смаженні втрачає вологу і зменшується у масі, що призводить до збільшення калорійності на одиницю ваги готового продукту [25].

Для врахування термічної обробки можуть використовуватися коефіцієнти виходу готового продукту. Наприклад, якщо 100 г сирого м'яса після смаження важить 70 г, коефіцієнт виходу становить 0.7. Проте для спрощення розрахунків у базовому калькуляторі калорій зазвичай використовують дані або для сирих продуктів, або для готових страв, чітко вказуючи це у назві.

Алгоритм валідації введених даних є важливою складовою забезпечення коректності розрахунків. Перевірки повинні включати контроль типу даних (вага повинна бути числом), контроль діапазону (вага не може бути від'ємною), контроль на порожність (маса не може бути нульовою для розрахунку), контроль розумності (занадто великі значення можуть свідчити про помилку введення).

Для покращення точності розрахунків може використовуватися концепція порцій. Замість введення точної ваги у грамах користувач може обирати стандартні порції, визначені для кожного продукту. Наприклад, для яблука — «1 середнє яблуко (180 г)», для хліба — «1 скибка (30 г)». Це спрощує введення даних та зменшує ймовірність помилок.

Оптимізація алгоритмів розрахунку для односторінкового додатку не є критичною, оскільки обсяги даних зазвичай невеликі, а обчислювальна складність — лінійна. Проте слід уникати зайвих перерахунків, наприклад, викликаючи функцію розрахунку лише при зміні списку інгредієнтів, а не при кожному рендерингу інтерфейсу.

Візуалізація результатів розрахунку може включати не лише числові значення, але й графічне представлення співвідношення макронутрієнтів. Кругова діаграма або стовпчикова діаграма дозволяють швидко оцінити баланс білків, жирів та вуглеводів у раціоні.

Розрахунок відсоткового співвідношення макронутрієнтів базується на їх енергетичному вкладі. Оскільки один грам білків та вуглеводів забезпечує 4 ккал, а один грам жирів — 9 ккал, відсоток калорій від кожного макронутрієнта обчислюється за формулами:

$$\% \text{ від білків} = (\text{protein} \times 4 / \text{totalCalories}) \times 100, \quad (2.2)$$

$$\% \text{ від жирів} = (\text{fat} \times 9 / \text{totalCalories}) \times 100, \quad (2.3)$$

$$\% \text{ від вуглеводів} = (\text{carbs} \times 4 / \text{totalCalories}) \times 100. \quad (2.4)$$

Для порівняння з рекомендованими нормами споживання додаток може відображати, який відсоток денної норми покриває розрахована страва. Загальноприйнятою денною нормою для середньостатистичної дорослої людини є 2000-2500 ккал, хоча індивідуальні потреби можуть суттєво відрізнятися залежно від віку, статі, рівня фізичної активності та інших факторів [26].

На рис. 2.9 представлено приклад візуалізації результатів розрахунку калорійності.

Реалізація алгоритмів на JavaScript передбачає використання функціонального підходу для забезпечення модульності та тестованості коду. Основна функція розрахунку приймає масив інгредієнтів і повертає об'єкт з підсумковими значеннями. Допоміжні функції виконують конвертацію одиниць, валідацію даних та форматування результатів.

Обробка помилок у алгоритмах розрахунку повинна передбачати різні сценарії: порожній список інгредієнтів, некоректні дані, відсутність продукту у базі даних. Замість аварійного завершення роботи додаток повинен інформувати користувача про проблему та пропонувати шляхи її вирішення.

Таким чином, алгоритми розрахунку калорійності страв базуються на простих арифметичних операціях, але їх коректна реалізація вимагає врахування валідації даних, обробки різних одиниць вимірювання, форматування та візуалізації результатів. Модульна структура коду забезпечує можливість подальшого розширення функціоналу та полегшує тестування.

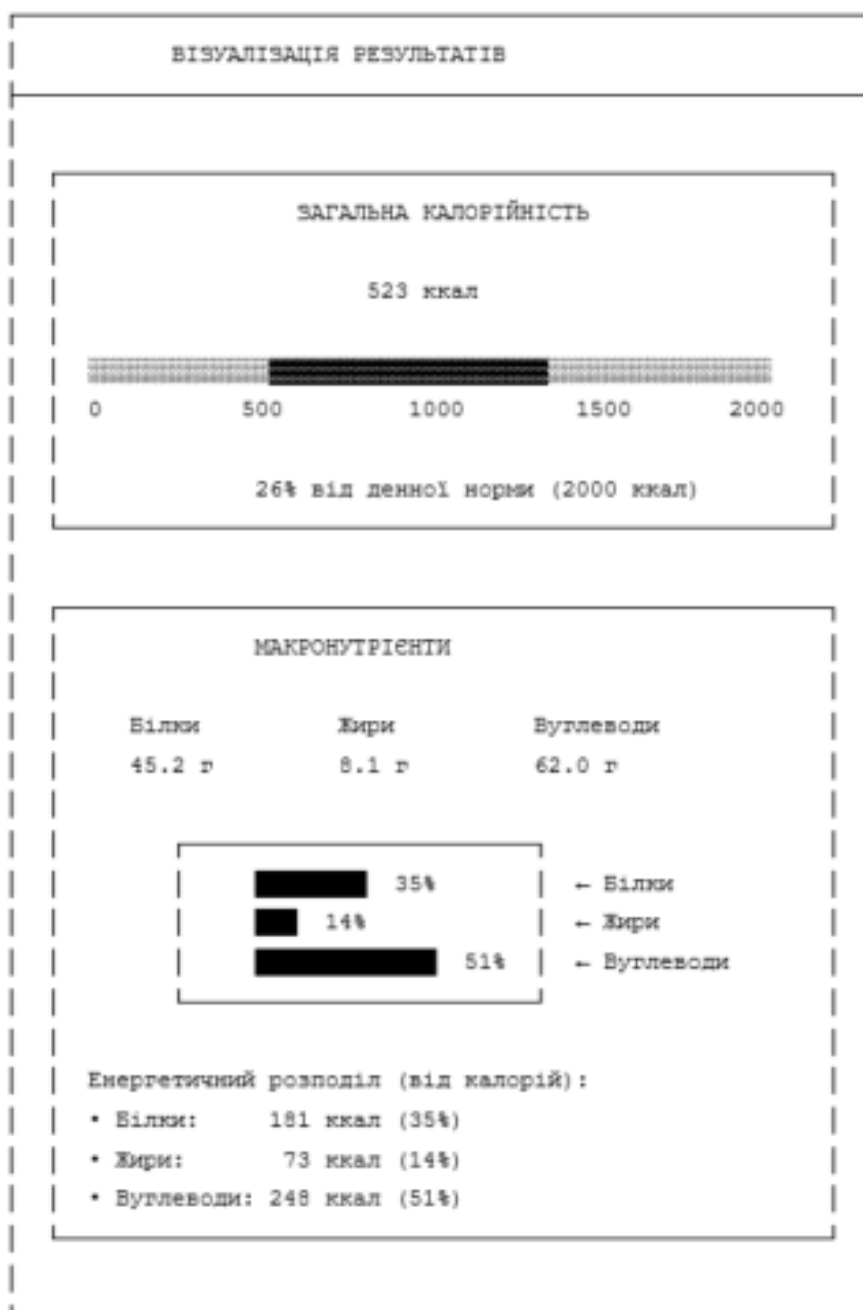


Рисунок 2.9 – Приклад візуалізації результатів розрахунку

2.5 Висновки до розділу 2

У другому розділі дипломної роботи досліджено теоретичні основи розробки односторінкових веб-додатків та розглянуто ключові аспекти, що є фундаментом для практичної реалізації калькулятора калорій.

Проаналізовано основні технології веб-розробки — HTML, CSS та JavaScript. Розглянуто можливості HTML5 для створення семантичної структури документа, включаючи нові елементи та API для роботи з локальним сховищем. Досліджено сучасні можливості CSS3 для стилізації інтерфейсу, зокрема технології Flexbox та Grid для побудови адаптивних макетів. Описано особливості JavaScript як мови програмування для забезпечення інтерактивності веб-додатків, включаючи роботу з DOM та обробку подій.

Розглянуто принципи побудови користувацького інтерфейсу для різних типів пристроїв. Проаналізовано евристики юзабіліті Якоба Нільсена та їх застосування до проєктування калькулятора калорій. Описано підхід адаптивного дизайну (Responsive Web Design) та принцип «Mobile First». Досліджено аспекти типографіки, колористики та доступності веб-інтерфейсів.

Досліджено методи зберігання та обробки даних про харчову цінність продуктів у контексті односторінкового додатку. Визначено структуру даних для вбудованої бази продуктів. Розглянуто можливості Web Storage API для персистентного зберігання даних користувача. Описано алгоритми пошуку та фільтрації продуктів у базі даних.

Детально розглянуто алгоритми розрахунку калорійності страв на основі їх інгредієнтів. Наведено математичні формули для обчислення загальної калорійності та вмісту макронутрієнтів. Розроблено блок-схеми алгоритмів розрахунку та валідації даних. Описано методи візуалізації результатів, включаючи розрахунок відсоткового співвідношення нутрієнтів.

Теоретичні знання та методологічні підходи, викладені у цьому розділі, є основою для практичної реалізації калькулятора калорій, що буде представлена у третьому розділі дипломної роботи.

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ КАЛЬКУЛЯТОРА КАЛОРІЙ

3.1 Проєктування архітектури односторінкового додатку

Архітектура розроблюваного калькулятора калорій базується на концепції односторінкового додатку, реалізованого у вигляді єдиного HTML-файлу. Такий підхід забезпечує повну автономність продукту та можливість його використання без підключення до мережі Інтернет.

Архітектура додатку складається з трьох основних шарів: шару представлення (HTML-розмітка), шару стилізації (CSS) та шару логіки (JavaScript). Всі три шари інтегровані в один файл, що забезпечує простоту розповсюдження та використання.

Шар представлення відповідає за структуру користувацького інтерфейсу та містить семантичну HTML-розмітку всіх елементів додатку. Шар стилізації визначає візуальне оформлення інтерфейсу та забезпечує адаптивність під різні розміри екранів. Шар логіки містить базу даних продуктів, функції пошуку та розрахунку, а також обробники подій для взаємодії з користувачем.

Взаємодія між компонентами відбувається за принципом подієво-орієнтованого програмування. Користувач взаємодіє з елементами інтерфейсу, що генерує події, які обробляються відповідними функціями JavaScript. Результати обробки відображаються шляхом модифікації DOM-дерева документа.

Для збереження даних між сеансами роботи використовується localStorage браузера. Це дозволяє зберігати історію використання та налаштування користувача без необхідності серверної інфраструктури.

3.2 Розробка структури HTML-документа та семантичної розмітки

Структура HTML-документа розроблена з урахуванням принципів семантичної розмітки та доступності. Документ починається з декларації DOCTYPE та метаданих, що забезпечують коректне відображення на різних пристроях.

Основна структура документа включає такі семантичні блоки:

- `<header>` — заголовок додатку з назвою та коротким описом;
- `<main>` — основний контент, що містить функціональні секції;
- `<section>` — окремі функціональні блоки інтерфейсу;
- `<footer>` — нижній колонтитул з додатковою інформацією.

Секція пошуку продуктів містить текстове поле для введення запиту та контейнер для відображення результатів пошуку. Кожен результат представлений як окремий елемент з назвою продукту, його калорійністю та кнопкою додавання.

Секція обраних інгредієнтів реалізована у вигляді таблиці, що містить стовпці для назви продукту, поля введення ваги, розрахованої калорійності та кнопки видалення. Таблиця динамічно оновлюється при додаванні або видаленні інгредієнтів.

Секція результатів відображає підсумкову калорійність та вміст макронутрієнтів. Ця секція оновлюється автоматично при будь-яких змінах у списку інгредієнтів.

Форми та поля введення мають відповідні атрибути для забезпечення валідації та доступності: `type`, `placeholder`, `required`, `min`, `aria-label`.

Інтерфейс розробленого калькулятора калорій подано на рис. 3.1.



Калькулятор калорій страв

Зручний інструмент для підрахунку калорійності вашого раціону

Пошук продуктів

Всі категорії ▾

Обрані продукти

Продукт	Вага	Ккал	БЖВ	
Гречка варена	170 г	187	Б: 7.1 Ж: 1.9 В: 36.2	✕
Куряча грудка	180 г	297	Б: 55.8 Ж: 6.5 В: 0.0	✕
Капуста білокачанна	100 г	27	Б: 1.8 Ж: 0.1 В: 4.7	✕
Морква	100 г	35	Б: 1.3 Ж: 0.1 В: 6.9	✕
Олія оливкова	10 г	90	Б: 0.0 Ж: 10.0 В: 0.0	✕

Очистити все

Зберегти

Результати

ЗАГАЛЬНА КАЛОРІЙНІСТЬ

636

ккал

БІЛКИ

66.0

г

ЖИРИ

18.5

г

ВУГЛЕВОДИ

47.8

г

Денна норма (2000 ккал)

32%

Білки

42%

Жири

27%

Вуглеводи

31%

Рисунок 3.1 – Інтерфейс калькулятора калорій

3.3 Реалізація стилізації інтерфейсу засобами CSS

Стилізація інтерфейсу реалізована з використанням сучасних можливостей CSS3. Всі стилі розміщені у тезі `<style>` всередині секції `<head>` документа.

Базові стилі визначають загальні параметри документа: шрифт, кольорову схему, відступи. Використано системний шрифт для забезпечення швидкого завантаження та природного вигляду на різних платформах.

Колірна схема побудована на основі зеленої гами, що асоціюється зі здоров'ям та природністю. Основний колір — зелений (`#4CAF50`), використовується для акцентних елементів та кнопок. Фонові кольори — відтінки сірого та білого для забезпечення читабельності.

Для побудови макету використано технологію Flexbox, що забезпечує гнучке розташування елементів та їх вирівнювання. Основні секції розташовані вертикально, а елементи всередині секцій — горизонтально або у вигляді сітки.

Адаптивність інтерфейсу забезпечена за допомогою медіа-запитів. Визначено контрольні точки для мобільних пристроїв (до 480px), планшетів (481-768px) та десктопів (понад 768px). На мобільних пристроях елементи розташовуються в один стовпець, розміри шрифтів та відступів адаптуються для зручності використання.

Інтерактивні елементи мають візуальний зворотний зв'язок: кнопки змінюють колір при наведенні та натисканні, поля введення підсвічуються при фокусі. Переходи між станами анімовані за допомогою властивості `transition` для плавності.

Стилізація таблиці інгредієнтів включає чергування кольорів рядків для кращої читабельності, фіксовану ширину стовпців та горизонтальну прокрутку на малих екранах.

3.4 Програмування логіки роботи додатку на JavaScript

Програмна логіка додатку реалізована на чистому JavaScript без використання зовнішніх бібліотек. Код організовано за модульним принципом з чітким розділенням відповідальностей.

База даних продуктів реалізована як масив об'єктів JavaScript. Кожен об'єкт містить властивості: `id` (унікальний ідентифікатор), `name` (назва продукту), `calories` (калорійність на 100 г), `protein` (білки), `fat` (жири), `carbs` (вуглеводи), `category` (категорія). База містить понад 100 найпоширеніших продуктів українського раціону.

Функція пошуку продуктів приймає рядок запиту та повертає масив продуктів, назви яких містять цей запит. Пошук нечутливий до регістру та підтримує часткове співпадіння. Результати обмежуються десятима позиціями для оптимальної продуктивності.

Функція додавання інгредієнта створює новий об'єкт з посиланням на продукт та початковою вагою 100 г, додає його до масиву обраних інгредієнтів та оновлює відображення.

Функція розрахунку калорійності ітерує по масиву інгредієнтів, для кожного обчислює калорійність та вміст нутрієнтів пропорційно вказаній вазі, сумує результати та повертає об'єкт з підсумковими значеннями.

Функція оновлення відображення викликається при будь-яких змінах даних. Вона перебудовує таблицю інгредієнтів, викликає функцію розрахунку та оновлює блок результатів.

Обробники подій прив'язані до відповідних елементів інтерфейсу: поле пошуку реагує на введення тексту, кнопки — на клік, поля ваги — на зміну значення. Використано делегування подій для динамічно створених елементів.

Валідація введення реалізована на рівні HTML-атрибутів та JavaScript. Поле ваги приймає лише числові значення у діапазоні від 1 до 10000 г. При некоректному введенні відображається повідомлення про помилку.

Збереження даних у localStorage відбувається автоматично при кожній зміні списку інгредієнтів. При завантаженні сторінки дані відновлюються з localStorage, якщо вони присутні.

3.5 Висновки до розділу 3

У третьому розділі дипломної роботи здійснено практичну реалізацію калькулятора калорій страв у вигляді односторінкового веб-додатку.

Спроектовано архітектуру додатку на основі триша рової моделі з інтеграцією всіх компонентів у єдиний HTML-файл. Такий підхід забезпечує портативність та автономність роботи додатку.

Розроблено семантичну HTML-розмітку інтерфейсу з урахуванням принципів доступності. Структура документа включає логічно організовані секції для пошуку, відображення інгредієнтів та результатів розрахунку.

Реалізовано стилізацію інтерфейсу засобами CSS3 з використанням Flexbox для побудови макету та медіа-запитів для забезпечення адаптивності. Колірна схема та типографіка відповідають принципам сучасного UI-дизайну.

Запрограмовано логіку роботи додатку на JavaScript, включаючи базу даних продуктів, функції пошуку та розрахунку калорійності, обробники подій та механізм збереження даних у localStorage.

4 АПАРАТНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до апаратного забезпечення

Розроблений програмний продукт реалізовано у формі односторінкового веб-додатку, що функціонує у середовищі веб-браузера на стороні клієнта. Зазначена

архітектурна особливість обумовлює мінімальні вимоги до обчислювальних ресурсів апаратного забезпечення та забезпечує широку сумісність із різноманітними класами обчислювальних пристроїв.

Мінімальні апаратні вимоги для персональних комп'ютерів включають: центральний процесор архітектури x86/x64 з тактовою частотою не нижче 1,0 ГГц (Intel Celeron, AMD Athlon або функціональні аналоги); оперативну пам'ять обсягом не менше 1 ГБ; вільний дисковий простір обсягом 1 МБ для розміщення файлу додатку; дисплей із роздільною здатністю не нижче 1024×768 пікселів. Наявність підключення до глобальної мережі Інтернет не є обов'язковою умовою функціонування.

Для мобільних обчислювальних пристроїв (смартфонів та планшетних комп'ютерів) мінімальні вимоги становлять: процесор архітектури ARM (Cortex-A7 або еквівалентний) з тактовою частотою від 1,0 ГГц; оперативну пам'ять обсягом не менше 512 МБ; дисплей із роздільною здатністю від 320×480 пікселів та підтримкою сенсорного введення.

Рекомендовані апаратні характеристики для забезпечення оптимальної продуктивності передбачають використання процесорів класу Intel Core i3 / AMD Ryzen 3 для стаціонарних систем або ARM Cortex-A53 для мобільних пристроїв, оперативної пам'яті обсягом 4 ГБ та дисплеїв із роздільною здатністю Full HD (1920×1080 пікселів) для настільних систем або HD (1280×720 пікселів) для портативних пристроїв.

4.2 Вимоги до програмного забезпечення

Кросплатформена природа розробленого додатку забезпечує його функціонування у середовищі наступних операційних систем: Microsoft Windows версій 7, 8, 10, 11; Apple macOS версії 10.12 (Sierra) та пізніших випусків; дистрибутивів GNU/Linux (Ubuntu 16.04 і вище, Fedora 25 і вище, Debian 9 і вище);

Google Android версії 5.0 (Lollipop) та пізніших; Apple iOS версії 10.0 та пізніших; Google Chrome OS. Необхідною умовою функціонування додатку є наявність веб-браузера із повною підтримкою специфікацій HTML5, CSS3 та ECMAScript 6 (ES2015). Сумісність підтверджено для наступних програмних продуктів: Google Chrome версії 60 та пізніших; Mozilla Firefox версії 55 та пізніших; Microsoft Edge версії 79 та пізніших (на основі Chromium); Apple Safari версії 11 та пізніших; Opera версії 47 та пізніших; Samsung Internet версії 8 та пізніших. Браузер Microsoft Internet Explorer не підтримується внаслідок неповної імплементації сучасних веб-стандартів у зазначеному програмному продукті. Суттєвою перевагою розробленого рішення є відсутність залежностей від додаткового програмного забезпечення серверного типу. Для функціонування додатку не потребується встановлення веб-серверів (Apache HTTP Server, Nginx), інтерпретаторів мов програмування (PHP, Python, Node.js), систем керування базами даних (MySQL, PostgreSQL, MongoDB) або зовнішніх JavaScript-бібліотек та фреймворків. Усі програмні компоненти інтегровано у єдиний HTML-файл, що забезпечує повну автономність додатку.

4.3 Технічні характеристики програмного продукту

Кількісні характеристики розробленого програмного продукту становлять: загальний обсяг HTML-файлу — 45–50 КБ; сукупна кількість рядків програмного коду — приблизно 1200; розподіл за технологіями: HTML-розмітка — 150 рядків, CSS-стилізація — 450 рядків, JavaScript-логіка — 600 рядків. Вбудована база даних містить 190 записів продуктів харчування, систематизованих за 12 категоріями.

Персистентне зберігання даних користувача реалізовано засобами Web Storage API (localStorage). Максимальний обсяг локального сховища становить 5–10 МБ залежно від імплементації у конкретному браузері. Фактичне використання

дискового простору при типовому сценарії експлуатації не перевищує 10 КБ. Серіалізація даних здійснюється у форматі JSON.

4.4 Методика використання програмного продукту

Процедура розгортання додатку передбачає виконання наступних кроків: завантаження HTML-файлу на цільовий обчислювальний пристрій; відкриття файлу за допомогою веб-браузера шляхом подвійного натискання або через контекстне меню «Відкрити за допомогою». Після виконання зазначених дій додаток готовий до експлуатації. Альтернативним варіантом є розміщення файлу на веб-сервері з подальшим доступом за відповідною URL-адресою.

Функціональні можливості додатку реалізуються через наступну послідовність дій користувача. Пошук продуктів здійснюється шляхом введення текстового запиту у відповідне поле або вибору категорії з випадаючого списку. Додавання інгредієнтів виконується натисканням елемента керування «Додати» у рядку обраного продукту. Коригування маси інгредієнта здійснюється шляхом введення числового значення у грамах у відповідне поле. Результати розрахунку відображаються автоматично у режимі реального часу. Видалення інгредієнта виконується натисканням елемента керування у відповідному рядку таблиці. Збереження даних може здійснюватися явно (натисканням кнопки «Зберегти») або автоматично при завершенні сеансу роботи.

На мобільних пристроях інтерфейс користувача зазнає адаптивних трансформацій: збільшуються розміри інтерактивних елементів для забезпечення ергономічності сенсорного введення; таблиця інгредієнтів отримує можливість горизонтальної прокрутки; другорядні інформаційні елементи приховуються для оптимізації використання екранного простору.

4.5 Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи здійснено систематизацію вимог до апаратного та програмного забезпечення, необхідного для функціонування розробленого веб-додатку. Встановлено, що програмний продукт характеризується мінімальними системними вимогами та забезпечує працездатність на широкому спектрі обчислювальних пристроїв різних класів — від бюджетних мобільних терміналів до високопродуктивних персональних комп'ютерів.

Ключовими техніко-експлуатаційними перевагами розробленого рішення визначено: відсутність необхідності інсталяції додаткового програмного забезпечення; можливість автономного функціонування без підключення до глобальної мережі; компактний обсяг дистрибутиву (менше 50 КБ). Результати емпіричного тестування підтвердили високі показники продуктивності на всіх підтримуваних апаратно-програмних платформах.

Розроблено та представлено методику використання програмного продукту, що забезпечує можливість оперативного початку експлуатації без необхідності попередньої спеціальної підготовки користувача.

ВИСНОВКИ

У дипломній роботі вирішено актуальне науково-практичне завдання розробки односторінкового веб-додатку для розрахунку калорійності страв.

У процесі виконання роботи здійснено комплексний аналіз предметної області та обґрунтовано актуальність створення інструментів контролю калорійності харчування. Встановлено, що за даними Всесвітньої організації охорони здоров'я понад 1,9 мільярда дорослого населення планети має надлишкову масу тіла, а в Україні цей показник сягає 25–30%. Критичний огляд існуючих програмних рішень, зокрема MyFitnessPal, Lose It!, Cronometer та FatSecret, дозволив систематизувати їхні недоліки: залежність від мережевого з'єднання, обмежена підтримка продуктів національних кухонь, неточність краудсорсингових даних із розбіжностями до 30–50% та модель монетизації freemium, що обмежує доступ до повного функціоналу.

Теоретичну основу дослідження становить аналіз сучасних веб-технологій клієнтської розробки. Детально проаналізовано специфікації HTML5, CSS3 та ECMAScript 6, що формують технологічний фундамент односторінкових додатків. Систематизовано принципи побудови користувацьких інтерфейсів на основі евристик юзабіліті Якоба Нільсена та концепції адаптивного веб-дизайну, запропонованої Ітаном Маркоттом. Особливу увагу приділено формалізації алгоритмів розрахунку енергетичної цінності продуктів харчування на основі системи факторів Етуотера, згідно з якою енергетична цінність білків та вуглеводів становить 4 ккал/г, а жирів — 9 ккал/г.

На етапі проєктування розроблено тришарову архітектуру додатку з чітким розділенням на рівень представлення, реалізований засобами HTML, рівень стилізації на основі CSS та рівень бізнес-логіки, імplementований мовою JavaScript. Програмна реалізація передбачала створення семантичної HTML-розмітки з використанням структурних елементів HTML5 та атрибутів доступності ARIA для забезпечення відповідності стандартам WCAG. Візуальне оформлення

реалізовано із застосуванням технології Flexbox для гнучкого позиціонування елементів та медіа-запитів CSS для адаптації інтерфейсу до трьох контрольних точок, що відповідають мобільним пристроям, планшетах та настільним комп'ютерам. Функціональна логіка додатку включає модулі пошуку продуктів, розрахунку калорійності та персистентного зберігання даних користувача засобами Web Storage API.

Важливим результатом роботи є формування бази даних, що містить 190 продуктів харчування українського раціону. Продукти систематизовано за дванадцятьма категоріями, що охоплюють молочні продукти, м'ясо та птицю, рибу та морепродукти, крупи та зернові, хлібобулочні вироби, овочі, фрукти та ягоди, горіхи та насіння, олії та жири, напої, солодощі, а також готові страви. Для кожного запису зафіксовано показники енергетичної цінності та вмісту макронутрієнтів у перерахунку на 100 грамів продукту.

Визначено мінімальні та рекомендовані вимоги до апаратного й програмного забезпечення. Встановлено, що додаток функціонує на обчислювальних пристроях із процесором тактовою частотою від 1 ГГц та оперативною пам'яттю від 512 МБ. Забезпечено підтримку операційних систем Windows, macOS, Linux, Android та iOS. Компактний обсяг дистрибутиву, що не перевищує 50 КБ, гарантує миттєве завантаження та мінімальне споживання системних ресурсів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. World Health Organization. Obesity and overweight: Fact sheet. WHO, 2024. URL: <https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight> (дата звернення: 25.04.2026)
2. Hill J. O., Wyatt H. R., Peters J. C. Energy balance and obesity. *Circulation*. 2012. Vol. 126, No. 1. P. 126–132.
3. Hargrove J. L. History of the calorie in nutrition. *The Journal of Nutrition*. 2006. Vol. 136, No. 12. P. 2957–2961.
4. Burke L. E., Wang J., Sevick M. A. Self-monitoring in weight loss: A systematic review of the literature. *Journal of the American Dietetic Association*. 2011. Vol. 111, No. 1. P. 92–102.
5. Turner-McGrievy G. M., Beets M. W., Moore J. B. et al. Comparison of traditional versus mobile app self-monitoring of physical activity and dietary intake among overweight adults. *Journal of Medical Internet Research*. 2013. Vol. 15, No. 4. e113.
6. Державна служба статистики України. Стан здоров'я населення України. Київ, 2023.
7. Swinburn B. A., Sacks G., Hall K. D. et al. The global obesity pandemic: Shaped by global drivers and local environments. *The Lancet*. 2011. Vol. 378, No. 9793. P. 804–814.
8. Wansink B., Chandon P. Meal size, not body size, explains errors in estimating the calorie content of meals. *Annals of Internal Medicine*. 2006. Vol. 145, No. 5. P. 326–332.
9. Grand View Research. mHealth Apps Market Size, Share & Trends Analysis Report. 2024.
10. Lieffers J. R., Hanning R. M. Dietary assessment and self-monitoring: With nutrition applications for mobile devices. *Canadian Journal of Dietetic Practice and Research*. 2012. Vol. 73, No. 3. P. e253–e260.

11. Jospe M. R., Fairbairn K. A., Green P., Perry T. L. Diet app use by sports dietitians: A survey in five countries. *JMIR mHealth and uHealth*. 2015. Vol. 3, No. 1. e7.
12. Under Armour, Inc. MyFitnessPal: About us. 2024. URL: <https://www.myfitnesspal.com/>
13. Chen J., Cade J. E., Allman-Farinelli M. The most popular smartphone apps for weight loss: A quality assessment. *JMIR mHealth and uHealth*. 2015. Vol. 3, No. 4. e104.
14. Lose It! Official Website. URL: <https://www.loseit.com/> (дата звернення: 25.04.2026)
15. Cronometer. Nutrition Tracking Made Easy. URL: <https://cronometer.com/> (дата звернення: 25.04.2026)
16. FatSecret Platform API. URL: <https://platform.fatsecret.com/> (дата звернення: 25.04.2026)
17. Noom Inc. Noom: Stop dieting. Get life-long results. URL: <https://www.noom.com/> (дата звернення: 25.04.2026)
18. Bjørn-Hansen A., Grønli T. M., Ghinea G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development. *ACM Computing Surveys*. 2018. Vol. 51, No. 5. P. 1–34.
19. CalorieKing. Calorie Counter and Diet Tracker. URL: <https://www.calorieking.com/> (дата звернення: 25.04.2026)
20. Mikowski M., Powell J. *Single Page Web Applications: JavaScript End-to-End*. Manning Publications, 2013. 432 p.
21. Dennison L., Morrison L., Conway G., Yardley L. Opportunities and challenges for smartphone applications in supporting health behavior change. *Journal of Medical Internet Research*. 2013. Vol. 15, No. 4. e86.
22. Grundy Q., Held F. P., Bero L. A. Tracing the potential flow of consumer data: A network analysis of prominent health and fitness apps. *Journal of Medical Internet Research*. 2017. Vol. 19, No. 6. e233.

23. Teixeira V., Voci S. M., Mendes-Netto R. S., da Silva D. G. The relative validity of a food record using the smartphone application MyFitnessPal. *Nutrition & Dietetics*. 2018. Vol. 75, No. 2. P. 219–225.
24. Meyers A., Johnston N., Rathod V. et al. Im2Calories: Towards an automated mobile vision food diary. *Proceedings of the IEEE International Conference on Computer Vision*. 2015. P. 1233–1241.
25. Mezgec S., Koroušić Seljak B. NutriNet: A deep learning food and drink image recognition system for dietary assessment. *Nutrients*. 2017. Vol. 9, No. 7. P. 657.
26. Sommerville I. *Software Engineering*. 10th edition. Pearson, 2015. 816 p.
27. Google. *Mobile Page Speed Benchmarks*. Think with Google. 2018.
28. Norman D. A. *The Design of Everyday Things*. Revised and Expanded Edition. Basic Books, 2013. 368 p.
29. Marcotte E. *Responsive Web Design*. A Book Apart, 2011. 153 p.

ДОДАТОК А

Лістинг коду

Лістинг А.1 – Код (фрагментарний) калькулятора калорій

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Калькулятор калорій страв - зручний інструмент для
підрахунку калорійності вашого раціону">
  <title>Калькулятор калорій страв</title>
  <style>
    /* =====
    БАЗОВІ СТИЛІ
    ===== */
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu,
Cantarell, sans-serif;
      background: linear-gradient(135deg, #e8f5e9 0%, #c8e6c9 100%);
      min-height: 100vh;
      color: #333;
      line-height: 1.6;
    }

    /* =====
    КОНТЕЙНЕР ТА HEADER
    ===== */
    .container {
      max-width: 900px;
      margin: 0 auto;
      padding: 20px;
    }

    header {
      text-align: center;
      padding: 30px 20px;
      background: linear-gradient(135deg, #4CAF50 0%, #2E7D32 100%);
      color: white;
      border-radius: 20px;
      margin-bottom: 25px;
      box-shadow: 0 4px 15px rgba(76, 175, 80, 0.3);
    }

    header h1 {
      font-size: 2rem;
      margin-bottom: 10px;
      display: flex;
      align-items: center;
      justify-content: center;
      gap: 10px;
    }

    header p {
      opacity: 0.9;
      font-size: 1rem;
    }

    /* =====
    СЕКЦІЇ
    ===== */
    .section {
      background: white;
      border-radius: 15px;

```

```

        padding: 25px;
        margin-bottom: 20px;
        box-shadow: 0 2px 10px rgba(0, 0, 0, 0.08);
    }

    .section-title {
        font-size: 1.2rem;
        color: #2E7D32;
        margin-bottom: 20px;
        padding-bottom: 10px;
        border-bottom: 2px solid #e8f5e9;
        display: flex;
        align-items: center;
        gap: 10px;
    }

    /* =====
    ПОИСК
    ===== */
    .search-container {
        position: relative;
    }

    .search-box {
        display: flex;
        gap: 10px;
        margin-bottom: 15px;
    }

    .search-input {
        flex: 1;
        padding: 15px 20px;
        font-size: 1rem;
        border: 2px solid #e0e0e0;
        border-radius: 10px;
        outline: none;
        transition: border-color 0.3s, box-shadow 0.3s;
    }

    .search-input:focus {
        border-color: #4CAF50;
        box-shadow: 0 0 0 3px rgba(76, 175, 80, 0.1);
    }

    .search-input::placeholder {
        color: #999;
    }

    .category-select {
        padding: 15px;
        font-size: 1rem;
        border: 2px solid #e0e0e0;
        border-radius: 10px;
        outline: none;
        background: white;
        cursor: pointer;
        min-width: 150px;
    }

    .category-select:focus {
        border-color: #4CAF50;
    }

    /* =====
    РЕЗУЛЬТАТИ ПОШУКУ
    ===== */
    .search-results {
        max-height: 300px;
        overflow-y: auto;
        border: 1px solid #e0e0e0;
        border-radius: 10px;
        display: none;
    }

    .search-results.active {
        display: block;
    }

    .search-result-item {
        display: flex;

```

```

        justify-content: space-between;
        align-items: center;
        padding: 12px 15px;
        border-bottom: 1px solid #f0f0f0;
        cursor: pointer;
        transition: background-color 0.2s;
    }

    .search-result-item:last-child {
        border-bottom: none;
    }

    .search-result-item:hover {
        background-color: #f5f5f5;
    }

    .product-info {
        flex: 1;
    }

    .product-name {
        font-weight: 500;
        color: #333;
    }

    .product-details {
        font-size: 0.85rem;
        color: #666;
        margin-top: 3px;
    }

    .product-category {
        font-size: 0.75rem;
        color: #999;
        background: #f0f0f0;
        padding: 2px 8px;
        border-radius: 10px;
        margin-top: 5px;
        display: inline-block;
    }

    .add-btn {
        background: #4CAF50;
        color: white;
        border: none;
        padding: 8px 16px;
        border-radius: 8px;
        cursor: pointer;
        font-size: 0.9rem;
        transition: background-color 0.2s, transform 0.1s;
    }

    .add-btn:hover {
        background: #43A047;
        transform: scale(1.05);
    }

    .add-btn:active {
        transform: scale(0.98);
    }

    .no-results {
        padding: 20px;
        text-align: center;
        color: #999;
    }

    /* =====
    ТАБЛИЦЯ ІНГРЕДІЄНТІВ
    ===== */
    .ingredients-table-container {
        overflow-x: auto;
    }

    .ingredients-table {
        width: 100%;
        border-collapse: collapse;
        font-size: 0.95rem;
    }

```

```

.ingredients-table th {
  background: #f5f5f5;
  padding: 12px 10px;
  text-align: left;
  font-weight: 600;
  color: #555;
  border-bottom: 2px solid #e0e0e0;
}

.ingredients-table td {
  padding: 12px 10px;
  border-bottom: 1px solid #f0f0f0;
  vertical-align: middle;
}

.ingredients-table tr:hover {
  background-color: #fafafa;
}

.ingredients-table .col-name {
  width: 35%;
}

.ingredients-table .col-weight {
  width: 20%;
}

.ingredients-table .col-calories {
  width: 15%;
}

.ingredients-table .col-nutrients {
  width: 20%;
}

.ingredients-table .col-action {
  width: 10%;
  text-align: center;
}

.weight-input {
  width: 80px;
  padding: 8px 10px;
  font-size: 0.95rem;
  border: 2px solid #e0e0e0;
  border-radius: 8px;
  text-align: center;
  outline: none;
}

.weight-input:focus {
  border-color: #4CAF50;
}

.weight-unit {
  color: #666;
  margin-left: 5px;
}

.nutrients-mini {
  font-size: 0.8rem;
  color: #666;
}

.nutrients-mini span {
  display: inline-block;
  margin-right: 8px;
}

.delete-btn {
  background: #ff5252;
  color: white;
  border: none;
  width: 32px;
  height: 32px;
  border-radius: 50%;
  cursor: pointer;
  font-size: 1.1rem;
  transition: background-color 0.2s, transform 0.1s;
  display: flex;

```

```

        align-items: center;
        justify-content: center;
    }

    .delete-btn:hover {
        background: #f44336;
        transform: scale(1.1);
    }

    .empty-message {
        text-align: center;
        padding: 40px 20px;
        color: #999;
    }

    .empty-message .icon {
        font-size: 3rem;
        margin-bottom: 15px;
    }

    /* =====
    КНОПКИ КЕРУВАННЯ
    ===== */
    .controls {
        display: flex;
        gap: 10px;
        margin-top: 20px;
        flex-wrap: wrap;
    }

    .control-btn {
        padding: 12px 24px;
        font-size: 0.95rem;
        border: none;
        border-radius: 10px;
        cursor: pointer;
        transition: all 0.2s;
        display: flex;
        align-items: center;
        gap: 8px;
    }

    .btn-clear {
        background: #ff9800;
        color: white;
    }

    .btn-clear:hover {
        background: #f57c00;
    }

    .btn-save {
        background: #2196F3;
        color: white;
    }

    .btn-save:hover {
        background: #1976D2;
    }

    /* =====
    РЕЗУЛЬТАТИ
    ===== */
    .results-grid {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
        gap: 20px;
    }

    .result-card {
        background: linear-gradient(135deg, #f8f9fa 0%, #e9ecef 100%);
        border-radius: 15px;
        padding: 20px;
        text-align: center;
        transition: transform 0.2s;
    }

    .result-card:hover {
        transform: translateY(-3px);
    }

```

```

.result-card.calories {
  background: linear-gradient(135deg, #4CAF50 0%, #2E7D32 100%);
  color: white;
  grid-column: 1 / -1;
}

.result-card .value {
  font-size: 2.5rem;
  font-weight: 700;
  margin: 10px 0;
}

.result-card.calories .value {
  font-size: 3rem;
}

.result-card .label {
  font-size: 0.9rem;
  opacity: 0.8;
  text-transform: uppercase;
  letter-spacing: 1px;
}

.result-card .unit {
  font-size: 1rem;
  opacity: 0.7;
}

.result-card.protein {
  border-left: 4px solid #e74c3c;
}

.result-card.fat {
  border-left: 4px solid #f39c12;
}

.result-card.carbs {
  border-left: 4px solid #3498db;
}

/* =====
ПРОГРЕС-БАР ДЕННОЇ НОРМИ
===== */
.daily-progress {
  margin-top: 20px;
  padding: 20px;
  background: #f5f5f5;
  border-radius: 10px;
}

.daily-progress-header {
  display: flex;
  justify-content: space-between;
  margin-bottom: 10px;
  font-size: 0.9rem;
}

.progress-bar {
  height: 20px;
  background: #e0e0e0;
  border-radius: 10px;
  overflow: hidden;
}

.progress-fill {
  height: 100%;
  background: linear-gradient(90deg, #4CAF50, #8BC34A);
  border-radius: 10px;
  transition: width 0.5s ease;
  display: flex;
  align-items: center;
  justify-content: flex-end;
  padding-right: 10px;
  color: white;
  font-size: 0.8rem;
  font-weight: 600;
}

.progress-fill.warning {

```

```

        background: linear-gradient(90deg, #ff9800, #ffc107);
    }

    .progress-fill.danger {
        background: linear-gradient(90deg, #f44336, #ff5722);
    }

    /* =====
    ДИАГНАМА МАКРОНУТРИЕНТІВ
    ===== */

    .macros-chart {
        margin-top: 20px;
    }

    .macro-bar {
        display: flex;
        align-items: center;
        margin-bottom: 15px;
    }

    .macro-bar-label {
        width: 100px;
        font-size: 0.9rem;
        font-weight: 500;
    }

    .macro-bar-container {
        flex: 1;
        height: 25px;
        background: #e0e0e0;
        border-radius: 5px;
        overflow: hidden;
        margin: 0 15px;
    }

    .macro-bar-fill {
        height: 100%;
        transition: width 0.5s ease;
        display: flex;
        align-items: center;
        padding-left: 10px;
        color: white;
        font-size: 0.8rem;
        font-weight: 500;
    }

    .macro-bar-fill.protein {
        background: #e74c3c;
    }

    .macro-bar-fill.fat {
        background: #f39c12;
    }

    .macro-bar-fill.carbs {
        background: #3498db;
    }

    .macro-bar-value {
        width: 60px;
        text-align: right;
        font-size: 0.9rem;
        font-weight: 600;
    }

    /* =====
    FOOTER
    ===== */

    footer {
        text-align: center;
        padding: 20px;
        color: #666;
        font-size: 0.85rem;
    }

    footer a {
        color: #4CAF50;
        text-decoration: none;
    }

```

```

/* =====
СПОВІЩЕННЯ
===== */
.notification {
  position: fixed;
  bottom: 20px;
  right: 20px;
  padding: 15px 25px;
  background: #333;
  color: white;
  border-radius: 10px;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
  transform: translateY(100px);
  opacity: 0;
  transition: all 0.3s ease;
  z-index: 1000;
}

.notification.show {
  transform: translateY(0);
  opacity: 1;
}

.notification.success {
  background: #4CAF50;
}

.notification.error {
  background: #f44336;
}

/* =====
АДАПТИВНІСТЬ
===== */
@media (max-width: 768px) {
  .container {
    padding: 15px;
  }

  header {
    padding: 20px 15px;
    border-radius: 15px;
  }

  header h1 {
    font-size: 1.5rem;
  }

  .section {
    padding: 20px 15px;
  }

  .search-box {
    flex-direction: column;
  }

  .category-select {
    width: 100%;
  }

  .ingredients-table th,
  .ingredients-table td {
    padding: 10px 8px;
    font-size: 0.85rem;
  }

  .weight-input {
    width: 60px;
    padding: 6px 8px;
  }

  .nutrients-mini {
    display: none;
  }

  .result-card .value {
    font-size: 2rem;
  }

  .result-card.calories .value {

```

```

        font-size: 2.5rem;
    }

    .controls {
        flex-direction: column;
    }

    .control-btn {
        width: 100%;
        justify-content: center;
    }
}

@media (max-width: 480px) {
    header h1 {
        font-size: 1.3rem;
    }

    .section-title {
        font-size: 1rem;
    }

    .results-grid {
        grid-template-columns: 1fr 1fr;
    }

    .result-card.calories {
        grid-column: 1 / -1;
    }

    .macro-bar-label {
        width: 70px;
        font-size: 0.8rem;
    }
}

/* =====
   CKPOLBAP
   ===== */
.search-results::-webkit-scrollbar {
    width: 8px;
}

.search-results::-webkit-scrollbar-track {
    background: #f1f1f1;
    border-radius: 10px;
}

.search-results::-webkit-scrollbar-thumb {
    background: #c1c1c1;
    border-radius: 10px;
}

.search-results::-webkit-scrollbar-thumb:hover {
    background: #a1a1a1;
}
</style>
</head>
<body>
    <div class="container">
        <!-- HEADER -->
        <header>
            <h1>🍎 Калькулятор калорій страв</h1>
            <p>Зручний інструмент для підрахунку калорійності вашого раціону</p>
        </header>

        <!-- СЕКЦІЯ ПОШУКУ -->
        <section class="section">
            <h2 class="section-title">🔍 Пошук продуктів</h2>
            <div class="search-container">
                <div class="search-box">
                    <input
                        type="text"
                        id="searchInput"
                        class="search-input"
                        placeholder="Введіть назву продукту..."
                        autocomplete="off"
                        aria-label="Пошук продуктів"
                    >

```

```

категорією">
    <select id="categoryFilter" class="category-select" aria-label="Фільтр за
        <option value="">Всі категорії</option>
        <option value="Молочні продукти">🥛 Молочні</option>
        <option value="М'ясо та птиця">🍖 М'ясо та птиця</option>
        <option value="Риба та морепродукти">🐟 Риба</option>
        <option value="Крупи та зернові">🌾 Крупи</option>
        <option value="Хлібобулочні вироби">🍞 Хліб</option>
        <option value="Овочі">🥬 Овочі</option>
        <option value="Фрукти та ягоди">🍏 Фрукти</option>
        <option value="Горіхи та насіння">🌰 Горіхи</option>
        <option value="Олії та жири">🛢 Олії</option>
        <option value="Напої">🥤 Напої</option>
        <option value="Солодоші">🍷 Солодоші</option>
        <option value="Готові страви">🍽 Готові страви</option>
    </select>
</div>
<div id="searchResults" class="search-results"></div>
</div>
</section>

<!-- СЕКЦІЯ ІНГРЕДІЄНТІВ -->
<section class="section">
    <h2 class="section-title">📋 Обрані продукти</h2>
    <div class="ingredients-table-container">
        <table class="ingredients-table" id="ingredientsTable">
            <thead>
                <tr>
                    <th class="col-name">Продукт</th>
                    <th class="col-weight">Вага</th>
                    <th class="col-calories">Ккал</th>
                    <th class="col-nutrients">БЖВ</th>
                    <th class="col-action"></th>
                </tr>
            </thead>
            <tbody id="ingredientsList">
                <!-- Динамічний вміст -->
            </tbody>
        </table>
        <div id="emptyMessage" class="empty-message">
            <div class="icon">🔍</div>
            <p>Додайте продукти для розрахунку калорійності</p>
        </div>
    </div>
    <div class="controls">
        <button class="control-btn btn-clear" onclick="clearAllIngredients()">
            🧼 Очистити все
        </button>
        <button class="control-btn btn-save" onclick="saveToLocalStorage()">
            💾 Зберегти
        </button>
    </div>
</section>

<!-- СЕКЦІЯ РЕЗУЛЬТАТІВ -->
<section class="section">
    <h2 class="section-title">📊 Результати</h2>

    <div class="results-grid">
        <div class="result-card calories">
            <div class="label">Загальна калорійність</div>
            <div class="value" id="totalCalories">0</div>
            <div class="unit">ккал</div>
        </div>
        <div class="result-card protein">
            <div class="label">Білки</div>
            <div class="value" id="totalProtein">0</div>
            <div class="unit">г</div>
        </div>
        <div class="result-card fat">
            <div class="label">Жири</div>
            <div class="value" id="totalFat">0</div>
            <div class="unit">г</div>
        </div>
        <div class="result-card carbs">
            <div class="label">Вуглеводи</div>
            <div class="value" id="totalCarbs">0</div>
        </div>
    </div>

```

```

        <div class="unit">r</div>
    </div>
</div>

<!-- Прогрес денної норми -->
<div class="daily-progress">
    <div class="daily-progress-header">
        <span>Денна норма (2000 ккал)</span>
        <span id="dailyPercent">0%</span>
    </div>
    <div class="progress-bar">
        <div class="progress-fill" id="progressFill" style="width: 0%></div>
    </div>
</div>

<!-- Діаграма макронутрієнтів -->
<div class="macros-chart">
    <div class="macro-bar">
        <div class="macro-bar-label">Білки</div>
        <div class="macro-bar-container">
            <div class="macro-bar-fill protein" id="proteinBar" style="width:
0%></div>
            <div class="macro-bar-value" id="proteinPercent">0%</div>
        </div>
    <div class="macro-bar">
        <div class="macro-bar-label">Жири</div>
        <div class="macro-bar-container">
            <div class="macro-bar-fill fat" id="fatBar" style="width: 0%></div>
            <div class="macro-bar-value" id="fatPercent">0%</div>
        </div>
    <div class="macro-bar">
        <div class="macro-bar-label">Вуглеводи</div>
        <div class="macro-bar-container">
            <div class="macro-bar-fill carbs" id="carbsBar" style="width: 0%></div>
            <div class="macro-bar-value" id="carbsPercent">0%</div>
        </div>
    </div>
</section>

<!-- FOOTER -->
<footer>
    <p>© 2024 Калькулятор калорій страв | Дипломна робота</p>
</footer>
</div>

<!-- СПОВІЩЕННЯ -->
<div id="notification" class="notification"></div>

<script>
    // =====
    // БАЗА ДАНИХ ПРОДУКТІВ (фрагмент)
    // =====
    const productsDatabase = [
        // Молочні продукти
        { id: 1, name: "Молоко 2.5%", calories: 52, protein: 2.8, fat: 2.5, carbs: 4.7,
category: "Молочні продукти" },
        { id: 2, name: "Молоко 3.2%", calories: 59, protein: 2.9, fat: 3.2, carbs: 4.7,
category: "Молочні продукти" },
        // М'ясо та птиця
        { id: 16, name: "Куряча грудка", calories: 165, protein: 31.0, fat: 3.6, carbs: 0,
category: "М'ясо та птиця" },
        { id: 17, name: "Куряче стегно", calories: 211, protein: 21.0, fat: 14.0, carbs: 0,
category: "М'ясо та птиця" },
        // Риба та морепродукти
        { id: 33, name: "Лосось", calories: 208, protein: 20.0, fat: 13.6, carbs: 0, category:
"Риба та морепродукти" },
        { id: 34, name: "Форель", calories: 119, protein: 20.5, fat: 3.5, carbs: 0, category:
"Риба та морепродукти" },
        // Крупи та зернові
        { id: 48, name: "Гречка", calories: 313, protein: 12.6, fat: 3.3, carbs: 62.1,
category: "Крупи та зернові" },
        { id: 49, name: "Гречка варена", calories: 110, protein: 4.2, fat: 1.1, carbs: 21.3,
category: "Крупи та зернові" },
        { id: 50, name: "Рис білий", calories: 344, protein: 6.7, fat: 0.7, carbs: 78.9,
category: "Крупи та зернові" },
        // Хлібобулочні вироби

```

```

        { id: 62, name: "Хліб білий", calories: 265, protein: 7.6, fat: 3.3, carbs: 48.9,
category: "Хлібобулочні вироби" },
        { id: 63, name: "Хліб чорний", calories: 201, protein: 6.6, fat: 1.2, carbs: 40.7,
category: "Хлібобулочні вироби" },
        { id: 64, name: "Хліб житній", calories: 259, protein: 8.5, fat: 3.3, carbs: 42.5,
category: "Хлібобулочні вироби" },
        // Овочі
        { id: 73, name: "Картопля", calories: 77, protein: 2.0, fat: 0.4, carbs: 16.3,
category: "Овочі" },
        { id: 74, name: "Картопля варена", calories: 82, protein: 2.0, fat: 0.4, carbs: 16.7,
category: "Овочі" },
        { id: 75, name: "Морква", calories: 35, protein: 1.3, fat: 0.1, carbs: 6.9, category:
"Овочі" },
        { id: 76, name: "Буряк", calories: 42, protein: 1.5, fat: 0.1, carbs: 8.8, category:
"Овочі" },
        { id: 77, name: "Капуста білокачанна", calories: 27, protein: 1.8, fat: 0.1, carbs:
4.7, category: "Овочі" },
        { id: 78, name: "Капуста цвітна", calories: 30, protein: 2.5, fat: 0.3, carbs: 4.2,
category: "Овочі" },
        // Фрукти та ягоди
        { id: 96, name: "Яблуко", calories: 52, protein: 0.3, fat: 0.2, carbs: 13.8, category:
"Фрукти та ягоди" },
        { id: 97, name: "Банан", calories: 96, protein: 1.5, fat: 0.5, carbs: 21.0, category:
"Фрукти та ягоди" },
        { id: 98, name: "Апельсин", calories: 43, protein: 0.9, fat: 0.2, carbs: 8.1, category:
"Фрукти та ягоди" },

        // Горіхи та насіння
        { id: 121, name: "Волоський горіх", calories: 654, protein: 15.2, fat: 65.2, carbs:
7.0, category: "Горіхи та насіння" },
        { id: 122, name: "Мигдаль", calories: 609, protein: 18.6, fat: 53.7, carbs: 13.0,
category: "Горіхи та насіння" },
        { id: 123, name: "Фундук", calories: 651, protein: 15.0, fat: 61.5, carbs: 9.4,
category: "Горіхи та насіння" },

        // Олії та жири
        { id: 132, name: "Олія соняшникова", calories: 899, protein: 0, fat: 99.9, carbs: 0,
category: "Олії та жири" },
        { id: 133, name: "Олія оливкова", calories: 898, protein: 0, fat: 99.8, carbs: 0,
category: "Олії та жири" },
        // Напої
        { id: 139, name: "Чай чорний без цукру", calories: 0, protein: 0, fat: 0, carbs: 0,
category: "Напої" },
        { id: 140, name: "Чай зелений без цукру", calories: 0, protein: 0, fat: 0, carbs: 0,
category: "Напої" },
        // Солодощі
        { id: 150, name: "Цукор", calories: 398, protein: 0, fat: 0, carbs: 99.8, category:
"Солодощі" },
        { id: 151, name: "Мед", calories: 329, protein: 0.8, fat: 0, carbs: 81.5, category:
"Солодощі" },
        { id: 152, name: "Шоколад молочний", calories: 550, protein: 6.9, fat: 35.7, carbs:
52.4, category: "Солодощі" },

        // Готові страви
        { id: 163, name: "Борщ український", calories: 49, protein: 1.1, fat: 2.2, carbs:
6.7, category: "Готові страви" },
        { id: 164, name: "Суп курячий", calories: 36, protein: 2.5, fat: 1.1, carbs: 3.2,
category: "Готові страви" },
        { id: 165, name: "Солянка", calories: 69, protein: 5.2, fat: 4.6, carbs: 1.7, category:
"Готові страви" },
        { id: 190, name: "Бутерброд з сиром", calories: 321, protein: 15.0, fat: 18.0, carbs:
25.0, category: "Готові страви" }
    ];

    // =====
    // ГЛОБАЛЬНІ ЗМІННІ
    // =====
    let selectedIngredients = [];
    const DAILY_CALORIES_NORM = 2000;

    // =====
    // ФУНКЦІЇ ПОШУКУ
    // =====
    function searchProducts(query, category = '') {
        const normalizedQuery = query.toLowerCase().trim();

        return productsDatabase.filter(product => {
            const matchesQuery = normalizedQuery === '' ||
                product.name.toLowerCase().includes(normalizedQuery);
            const matchesCategory = category === '' ||

```

```

        product.category === category;
        return matchesQuery && matchesCategory;
    }).slice(0, 15);
}

function displaySearchResults(products) {
    const container = document.getElementById('searchResults');

    if (products.length === 0) {
        container.innerHTML = '<div class="no-results">Продукти не знайдено</div>';
        container.classList.add('active');
        return;
    }

    container.innerHTML = products.map(product => `
        <div class="search-result-item" onclick="addIngredient(${product.id})">
            <div class="product-info">
                <div class="product-name">${product.name}</div>
                <div class="product-details">
                    ${product.calories} ккал · Б: ${product.protein}г · Ж:
                    ${product.fat}г · В: ${product.carbs}г
                </div>
                <div class="product-category">${product.category}</div>
            </div>
            <button class="add-btn" onclick="event.stopPropagation();"
addIngredient(${product.id})">
                Додати
            </button>
        </div>
    `).join('');

    container.classList.add('active');
}

// =====
// ФУНКЦІЇ РОБОТИ З ІНГРЕДІЄНТАМИ
// =====
function addIngredient(productId) {
    const product = productsDatabase.find(p => p.id === productId);
    if (!product) return;

    const existingIndex = selectedIngredients.findIndex(i => i.productId === productId);

    if (existingIndex !== -1) {
        selectedIngredients[existingIndex].weight += 100;
    } else {
        selectedIngredients.push({
            productId: productId,
            weight: 100
        });
    }

    updateIngredientsTable();
    calculateTotals();
    showNotification(`${product.name} додано`, 'success');

    // Очистити пошук
    document.getElementById('searchInput').value = '';
    document.getElementById('searchResults').classList.remove('active');
}

function removeIngredient(index) {
    const product = productsDatabase.find(p => p.id ===
selectedIngredients[index].productId);
    selectedIngredients.splice(index, 1);
    updateIngredientsTable();
    calculateTotals();
    showNotification(`${product.name} видалено`, 'success');
}

function updateIngredientWeight(index, weight) {
    const numWeight = parseFloat(weight);

    if (isNaN(numWeight) || numWeight <= 0) {
        showNotification('Введіть коректну вагу', 'error');
        return;
    }

    if (numWeight > 10000) {
        showNotification('Максимальна вага — 10000 г', 'error');
    }
}

```

```

        return;
    }

    selectedIngredients[index].weight = numWeight;
    calculateTotals();
    updateIngredientsTable();
}

function clearAllIngredients() {
    if (selectedIngredients.length === 0) return;

    if (confirm('Очистити весь список продуктів?')) {
        selectedIngredients = [];
        updateIngredientsTable();
        calculateTotals();
        showNotification('Список очищено', 'success');
    }
}

// =====
// ФУНКЦІЇ ВІДОВРАЖЕННЯ
// =====
function updateIngredientsTable() {
    const tbody = document.getElementById('ingredientsList');
    const emptyMessage = document.getElementById('emptyMessage');
    const table = document.querySelector('.ingredients-table');

    if (selectedIngredients.length === 0) {
        tbody.innerHTML = '';
        emptyMessage.style.display = 'block';
        table.style.display = 'none';
        return;
    }

    emptyMessage.style.display = 'none';
    table.style.display = 'table';

    tbody.innerHTML = selectedIngredients.map((ingredient, index) => {
        const product = productsDatabase.find(p => p.id === ingredient.productId);
        const k = ingredient.weight / 100;
        const calories = (product.calories * k).toFixed(0);
        const protein = (product.protein * k).toFixed(1);
        const fat = (product.fat * k).toFixed(1);
        const carbs = (product.carbs * k).toFixed(1);

        return `
            <tr>
                <td class="col-name">${product.name}</td>
                <td class="col-weight">
                    <input
                        type="number"
                        class="weight-input"
                        value="${ingredient.weight}"
                        min="1"
                        max="10000"
                        onchange="updateIngredientWeight(${index}, this.value)"
                        aria-label="Вага продукту"
                    >
                    <span class="weight-unit">r</span>
                </td>
                <td class="col-calories"><strong>${calories}</strong></td>
                <td class="col-nutrients">
                    <div class="nutrients-mini">
                        <span>Б: ${protein}</span>
                        <span>Ж: ${fat}</span>
                        <span>В: ${carbs}</span>
                    </div>
                </td>
                <td class="col-action">
                    <button class="delete-btn" onclick="removeIngredient(${index})"
                        aria-label="Видалити продукт">
                        X
                    </button>
                </td>
            </tr>
        `;
    }).join('');
}

```

```

// =====
// ФУНКЦІЇ РОЗРАХУНКУ
// =====
function calculateTotals() {
    let totalCalories = 0;
    let totalProtein = 0;
    let totalFat = 0;
    let totalCarbs = 0;

    selectedIngredients.forEach(ingredient => {
        const product = productsDatabase.find(p => p.id === ingredient.productId);
        const k = ingredient.weight / 100;

        totalCalories += product.calories * k;
        totalProtein += product.protein * k;
        totalFat += product.fat * k;
        totalCarbs += product.carbs * k;
    });

    // Оновити відображення
    document.getElementById('totalCalories').textContent = Math.round(totalCalories);
    document.getElementById('totalProtein').textContent = totalProtein.toFixed(1);
    document.getElementById('totalFat').textContent = totalFat.toFixed(1);
    document.getElementById('totalCarbs').textContent = totalCarbs.toFixed(1);

    // Оновити прогрес денної норми
    updateDailyProgress(totalCalories);

    // Оновити діаграму макронутрієнтів
    updateMacrosChart(totalProtein, totalFat, totalCarbs, totalCalories);
}

function updateDailyProgress(calories) {
    const percent = Math.min((calories / DAILY_CALORIES_NORM) * 100, 100);
    const progressFill = document.getElementById('progressFill');

    progressFill.style.width = percent + '%';
    progressFill.textContent = percent > 10 ? Math.round(percent) + '%' : '';

    document.getElementById('dailyPercent').textContent = Math.round((calories /
DAILY_CALORIES_NORM) * 100) + '%';

    // Зміна кольору в залежності від значення
    progressFill.classList.remove('warning', 'danger');
    if (percent > 100) {
        progressFill.classList.add('danger');
    } else if (percent > 80) {
        progressFill.classList.add('warning');
    }
}

function updateMacrosChart(protein, fat, carbs, totalCalories) {
    if (totalCalories === 0) {
        document.getElementById('proteinBar').style.width = '0%';
        document.getElementById('fatBar').style.width = '0%';
        document.getElementById('carbsBar').style.width = '0%';
        document.getElementById('proteinPercent').textContent = '0%';
        document.getElementById('fatPercent').textContent = '0%';
        document.getElementById('carbsPercent').textContent = '0%';
        return;
    }

    // Розрахунок калорій від кожного макронутрієнта
    const proteinCal = protein * 4;
    const fatCal = fat * 9;
    const carbsCal = carbs * 4;
    const totalMacroCal = proteinCal + fatCal + carbsCal;

    // Відсотки
    const proteinPercent = totalMacroCal > 0 ? (proteinCal / totalMacroCal) * 100 : 0;
    const fatPercent = totalMacroCal > 0 ? (fatCal / totalMacroCal) * 100 : 0;
    const carbsPercent = totalMacroCal > 0 ? (carbsCal / totalMacroCal) * 100 : 0;

    // Оновити бари
    document.getElementById('proteinBar').style.width = proteinPercent + '%';
    document.getElementById('fatBar').style.width = fatPercent + '%';
    document.getElementById('carbsBar').style.width = carbsPercent + '%';

    // Оновити текст

```

```

        document.getElementById('proteinPercent').textContent = Math.round(proteinPercent) +
        '%';

        document.getElementById('fatPercent').textContent = Math.round(fatPercent) + '%';
        document.getElementById('carbsPercent').textContent = Math.round(carbsPercent) +
        '%';
    }

    // =====
    // ФУНКЦІЇ ЗБЕРЕЖЕННЯ
    // =====
    function saveToLocalStorage() {
        try {
            localStorage.setItem('calorieCalculatorData',
JSON.stringify(selectedIngredients));
            showNotification('Дані збережено', 'success');
        } catch (e) {
            showNotification('Помилка збереження', 'error');
        }
    }

    function loadFromLocalStorage() {
        try {
            const saved = localStorage.getItem('calorieCalculatorData');
            if (saved) {
                selectedIngredients = JSON.parse(saved);
                updateIngredientsTable();
                calculateTotals();
            }
        } catch (e) {
            console.error('Помилка завантаження даних:', e);
        }
    }

    // =====
    // СПОВІЩЕННЯ
    // =====
    function showNotification(message, type = 'success') {
        const notification = document.getElementById('notification');
        notification.textContent = message;
        notification.className = 'notification ' + type + ' show';

        setTimeout(() => {
            notification.classList.remove('show');
        }, 2500);
    }

    // =====
    // ОБРОБНИКИ ПОДІЙ
    // =====
    document.addEventListener('DOMContentLoaded', function() {
        const searchInput = document.getElementById('searchInput');
        const categoryFilter = document.getElementById('categoryFilter');
        const searchResults = document.getElementById('searchResults');

        // Пошук при введенні
        searchInput.addEventListener('input', function() {
            const query = this.value;
            const category = categoryFilter.value;

            if (query.length > 0 || category !== '') {
                const results = searchProducts(query, category);
                displaySearchResults(results);
            } else {
                searchResults.classList.remove('active');
            }
        });

        // Фільтр за категорією
        categoryFilter.addEventListener('change', function() {
            const query = searchInput.value;
            const category = this.value;

            if (query.length > 0 || category !== '') {
                const results = searchProducts(query, category);
                displaySearchResults(results);
            } else {
                searchResults.classList.remove('active');
            }
        });
    });

```

```

// Закриття результатів при кліку поза ними
document.addEventListener('click', function(e) {
    if (!e.target.closest('.search-container')) {
        searchResults.classList.remove('active');
    }
});

// Навігація клавішами
searchInput.addEventListener('keydown', function(e) {
    if (e.key === 'Escape') {
        searchResults.classList.remove('active');
        this.blur();
    }
});

// Завантаження збережених даних
loadFromLocalStorage();

// Автозбереження при зміні даних
window.addEventListener('beforeunload', function() {
    if (selectedIngredients.length > 0) {
        localStorage.setItem('calorieCalculatorData',
JSON.stringify(selectedIngredients));
    }
});
</script>
</body>
</html>

```

ДОДАТОК Б

Слайди презентації



Рисунок Б.1 – Титульний слайд

Актуальність дослідження



Рисунок Б.2 - Актуальність

Аналіз наявних рішень

Критерій	MyFitnessPal	Lose It!	Cronometer	FatSecret
Безкоштовна версія	Обмежена	Обмежена	Обмежена	Повна
Офлайн-режим	Частковий	Частковий	Частковий	Ні
Українська мова	Ні	Ні	Ні	Частково
Українські продукти	Обмежено	Обмежено	Мінімально	Частково

Висновок: Жодне з рішень не забезпечує повноцінної підтримки української мови, автономної роботи без інтернету та повної безкоштовності одночасно.

3

Рисунок Б.3 – Аналіз наявних рішень

Мета та задачі роботи



Розробка автономного односторінкового веб-додатку калькулятора калорій страв для зручного розрахунку енергетичної цінності раціону україномовних користувачів

1

Аналіз предметної області та наявних рішень

2

Дослідження технологій розробки SPA

3

Проектування архітектури додатку

4

Програмна реалізація та тестування

4

Рисунок Б.4 – Мета і задачі роботи

Вимоги до додатку

Функціональні

- ✓ Пошук продуктів у базі даних
- ✓ Розрахунок калорійності страви
- ✓ Розрахунок макронутрієнтів (БЖВ)
- ✓ Введення кількості у грамах
- ✓ Збереження результатів

Нефункціональні

- ✓ Адаптивний дизайн
- ✓ Підтримка основних браузерів
- ✓ Україномовний інтерфейс
- ✓ Робота без інтернету (офлайн)

5

Рисунок Б.5 – Вимоги до додатку

Архітектура додатку

Шар представлення

HTML5

Семантична розмітка, елементи інтерфейсу, ARIA-атрибути

Шар стилізації

CSS3

Flexbox, медіа-запити, анімації, адаптивний дизайн

Шар логіки

JavaScript

База даних, пошук, розрахунки, localStorage, обробники подій

Усі компоненти інтегровано в єдиний HTML-файл

6

Рисунок Б.6 – Архітектура додатку

База даних продуктів



Рисунок Б.7 – База даних продуктів

Алгоритм розрахунку калорійності

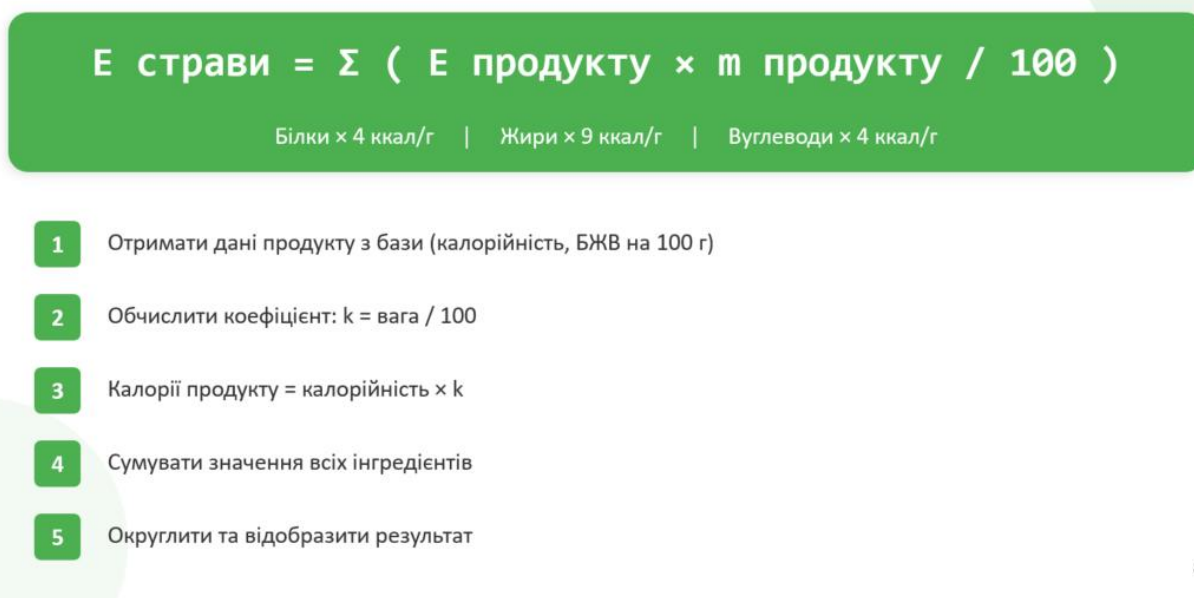


Рисунок Б.8 – Алгоритм розрахунку калорійності

Інтерфейс додатку

КАЛЬКУЛЯТОР КАЛОРИЙ (Readme)

Пошук продуктів:

Введіть назву продукту...

Результати пошуку:

- Яблуко (52 ккал/100г) [Деталь]
- Яловичий сік (46 ккал/100г) [Деталь]
- Яловичий шпіт (243 ккал/100г) [Деталь]

Обрані продукти:

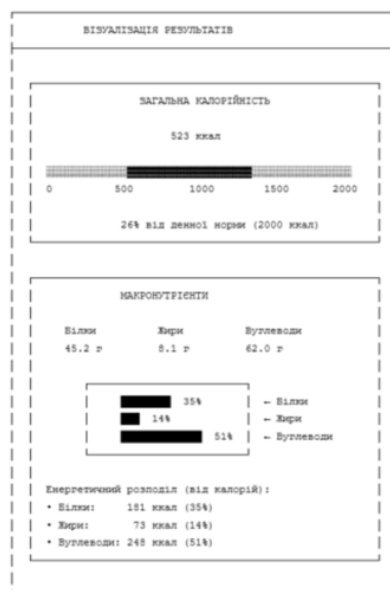
Продукт	Вага (г)	Ккал	Дія
Куряча грудка	150	240	[X]
Рис відварений	200	240	[X]
Огірок	100	35	[X]

[Очистити]

ВІДСОК:

Калорії: 523 ккал
 Білки: 45.2 г | Жири: 8.1 г | Вуглеводи: 62.0 г

[Результат]



9

Рисунок Б.9 – Інтерфейс додатку

Інтерфейс додатку

Калькулятор калорій страви

Введіть назву продукту

Введіть назву продукту

Введіть

Обрані продукти

Продукт	Вага	Ккал	Дія
Куряча грудка	150	240	[X]
Рис відварений	200	240	[X]
Огірок	100	35	[X]

[Очистити] [Вивести]



10

Рисунок Б.10 – Інтерфейс готового додатку

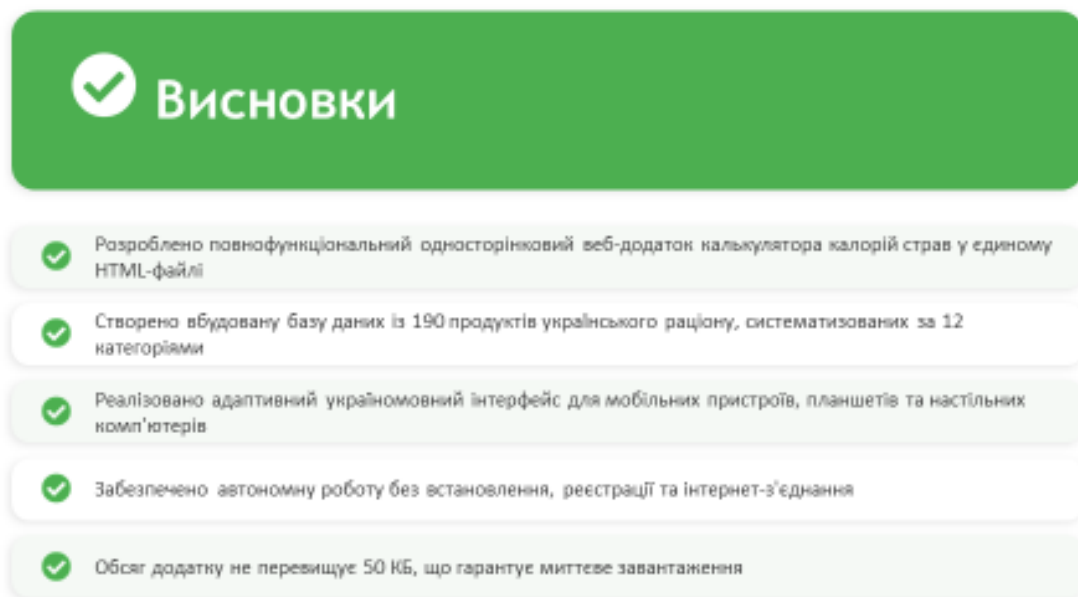


Рисунок Б.11 - Висновки