

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-07

Дипломний проєкт

здобувача освіти денної форми навчання

КГ.07.17.000.ДП

**ЯВОРСЬКОГО ІГОРЯ
ОЛЕГОВИЧА**

м. Одеса
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-07

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

**Розробка веб орієнтованої експертної системи
для індивідуального підбору дієт харчування**

Проектний матеріал складається з пояснювальної записки на 82 сторінках та графічного (презентаційного) матеріалу на 13 аркушах (слайдах).

Дипломник  (Яворський І. О.)

Керівник  (Жадан А. С.)

Консультанти:

з економічного розділу  (Іванченков В. С.)

з розділу охорони праці та техніки безпеки  (Чорновол Н. І.)

з нормоконтролю  (Петрашова В. І.)

старший консультант  (Кривченко Ю. В.)

До захисту допущений


Голова циклової комісії  (Кривченко Ю. В.)

Завідувач відділення  (Скорнякова О. В.)

Захист «17» 06 2024 р.

Протокол ЕК № 1

Оцінка ЕК 5 (відмінно) 90%

Секретар ЕК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка і веб-дизайн»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І. В.

“ 15 ” 01 2024 року

ЗАВДАННЯ
на дипломний проєкт

Здобувачеві освіти Яворському Ігорю Олеговичу
1. Тема проєкту Розробка веб-орієнтованої експертної системи
для індивідуального підбору дієт

Затверджена наказом по коледжу від “ 02 ” листопада 2023 р., наказ № 244-А2-ОД

2 Термін здачі закінченого проєкту _____

3. Вихідні дані до проєкту _____

1. Передбачити сучасний графічний дизайн користувача (GUI) веб-системи

2. Застосовувати дружній користувацький досвід (UX) веб-системи

3. Застосовувати Front-End фреймворк React

4. Визначити алгоритм підбору дієт

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

1. Аналіз предметної області. 2. Технології та засоби розробки (проєктування).

3. Проєктування веб-дизайну. 4. Проєктування архітектури веб-застосунку.

5. Розробка веб-застосунку. 6. Тестування створеного веб-застосунку.

7. Економічний розрахунок. 8. Аспекти охорона праці та техніки безпеки

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

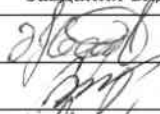


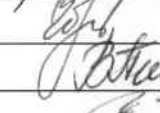


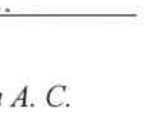


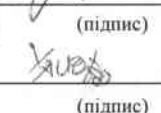
Презентація Power Point – 13 слайдів

(Етапи створення веб-застосунку; стек технологій веб-застосунку;

Структура веб-застосунку; Розробка дизайну веб-застосунку; Кольори веб-застосунку

Розробка веб-застосунку; Скриншоти веб-застосунку)

6. Консультанти по проєкту, із зазначенням розділів проєкту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Жадан А. С.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання

15 січня 2024 р.

Керівник

Жадан А. С.



(підпис)

Завдання прийняв до виконання

Яворський І. О.



(підпис)

КАЛЕНДАРНИЙ ПЛАН

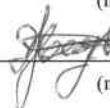
№ з/р	Назва етапів дипломного проєкту (роботи)	Термін виконання етапів дипломного проєкту (роботи)	Відмітка про виконання
1	Формування вступу	29.04.24	Виконано
2	Аналіз предметної області	10.05.24	Виконано
3	Підбір технічної літератури	19.05.24	Виконано
4	Вибір технологій та засобів розробки (проектування)	20.05.24	Виконано
5	Проектування веб-дизайну веб-застосунку	22.05.24	Виконано
6	Проектування архітектури веб-застосунку	24.05.24	Виконано
7	Реалізація веб-застосунку	27.05.24	Виконано
8	Тестування створеного веб-застосунку	29.05.24	Виконано
9	Оформлення пояснювальної записки	31.05.24	Виконано
10	Оформлення графічної (презентаційної) частини	01.06.24	Виконано
11	Економічний розрахунок	02.06.24	Виконано
12	Опис охорони праці та техніки безпеки	09.06.24	Виконано
13	Аналіз результатів проектування	13.06.24	Виконано
14	Підготовка доповіді для захисту	16.06.24	Виконано

Дипломник



(підпис)

Керівник



(підпис)

ЗМІСТ

ВСТУП	7
1 ОСНОВНИЙ РОЗДІЛ	9
1.1 Аналіз предметної області	9
1.1.1 Огляд існуючих рішень	9
1.1.2 Технології та засоби розробки	15
1.2 Проєктування веб-системи	19
1.2.1 Технічне завдання на розробку	19
1.2.2 Проєктування дизайну веб-застосунку	20
1.2.3 Проєктування архітектури веб-застосунку	24
1.3 Реалізація веб-системи	25
1.3.1 Файлова структура веб-застосунку	25
1.3.2 Налаштування роутингу	30
1.3.3. Створення шаблонізаторів	32
1.3.4 Створення представлень	34
1.3.5 Створення анімацій	39
1.3.6 Оптимізаці веб-застосунку	40
1.4 Тестування веб-застосунку	41
1.4.1 Функціональний огляд головної сторінки “Home”	42
1.4.2 Функціональний огляд сторінки “Diets”	44
1.4.3 Функціональний огляд сторінки “Calorie Calculator”	45
1.4.4 Функціональний огляд сторінки “About Us”	46
1.4.5 Функціональний огляд темного режиму у веб-застосунку	47
1.4.6 Функціональний огляд мобільної версії веб-застосунку	49
2 ЕКОНОМІЧНИЙ РОЗДІЛ	50
3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	57
3.1 Вступ	57
3.2 Аналіз умов праці й забезпечення безпеки при виконанні основних видів робіт на об’єкті дипломного проєктування	57

					КГ 07. 17 000. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		5

3.2.1	Небезпечні та шкідливі виробничі фактори	57
3.2.2	Заходи щодо забезпечення безпеки праці	58
3.3	Гігієнічні вимоги до виробничого середовища:	59
3.3.1	Приміщення	59
3.3.2	Мікроклімат	59
3.3.3	Освітлення	59
3.3.4	Шум	60
3.3.5	Організація робочого місця	60
3.3.6	Електробезпека	60
3.4	Пожежна безпека	61
3.5	Оптимальні рішення щодо забезпечення безпечного виконання робіт:	62
ВИСНОВКИ		63
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ		64
ДОДАТОК А. Програмний код основної логіки веб-застосунку		66
ДОДАТОК Б. Слайди мультимедійної презентації		76

ВСТУП

У сучасному світі питання про турботу про здоров'я та раціональне харчування є надзвичайно актуальними. Створення веб-застосунку для підбору дієт може виявитися важливим кроком у покращенні стану здоров'я населення. Такий застосунок може допомогти людям знаходити оптимальні дієтичні підходи, що сприятимуть їхньому оздоровленню та збереженню здорового способу життя.

Зростання обізнаності про принципи здорового харчування не гарантує уникнення проблем нездорової харчової поведінки, особливо серед молоді, що залишається актуальною проблемою. Нераціональні вибори в харчуванні можуть призвести до серйозних захворювань, таких як ожиріння, серцево-судинні захворювання та діабет.

Створення веб-застосунку для підбору дієт може стати ефективним рішенням цієї проблеми, оскільки він надає доступ до достовірної та актуальної інформації про різні дієтичні підходи. Крім того, він допомагає користувачам усвідомлено зробити вибір на користь більш здорового харчування, сприяючи покращенню їхнього стану здоров'я.

Мета цієї роботи полягає в розробці зручного та ефективного інструменту для користувачів, щоб допомогти їм знайти оптимальну дієту для досягнення своїх конкретних цілей, таких як зниження ваги, збереження здоров'я, покращення спортивних результатів тощо. Розроблений додаток має за мету сприяти покращенню життя користувачів шляхом надання персоналізованих рекомендацій та підтримки в досягненні їхніх цілей в області здоров'я та фізичної активності.

Під час дослідження основних проблем даної предметної області та аналізу аналогів та засобів розробки, був використаний системний підхід. Для розробки веб-застосунку використано мову програмування JavaScript та фреймворк React. Це підходи, які дозволяють забезпечити ефективну та функціональну реалізацію проекту.

					КГ 07. 17 000. 00 ДП ПЗ	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

Результатом розробки є веб-застосунок, який може бути корисним для людей, що шукають достовірну та актуальну інформацію про різні дієтичні підходи. Крім того, він буде корисним для тих, хто бажає схуднути, нормалізувати вагу, підвищити енергію або покращити настрій.

					<i>КГ 07. 17 000. 00 ДП ПЗ</i>	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз предметної області

1.1.1 Огляд існуючих рішень

Розглянемо наступні аналоги до застосунку, що розроблюються – “Calorie Calculator”, “Calorie calculator – Calculate your calorie needs” та “Daily Calorie Intake Calculator”.

“Calorie Calculator” являє собою веб-сервіс, що дозволяє користувачам розрахувати свою добову норму калорій. Він аналізує індивідуальні параметри, такі як вік, стать, зріст, вага і рівень активності, щоб надати точні рекомендації щодо кількості калорій, необхідних для здорового харчування та досягнення поставлених цілей [1].

На рис. 1.1. зображено “Calorie Calculator”.

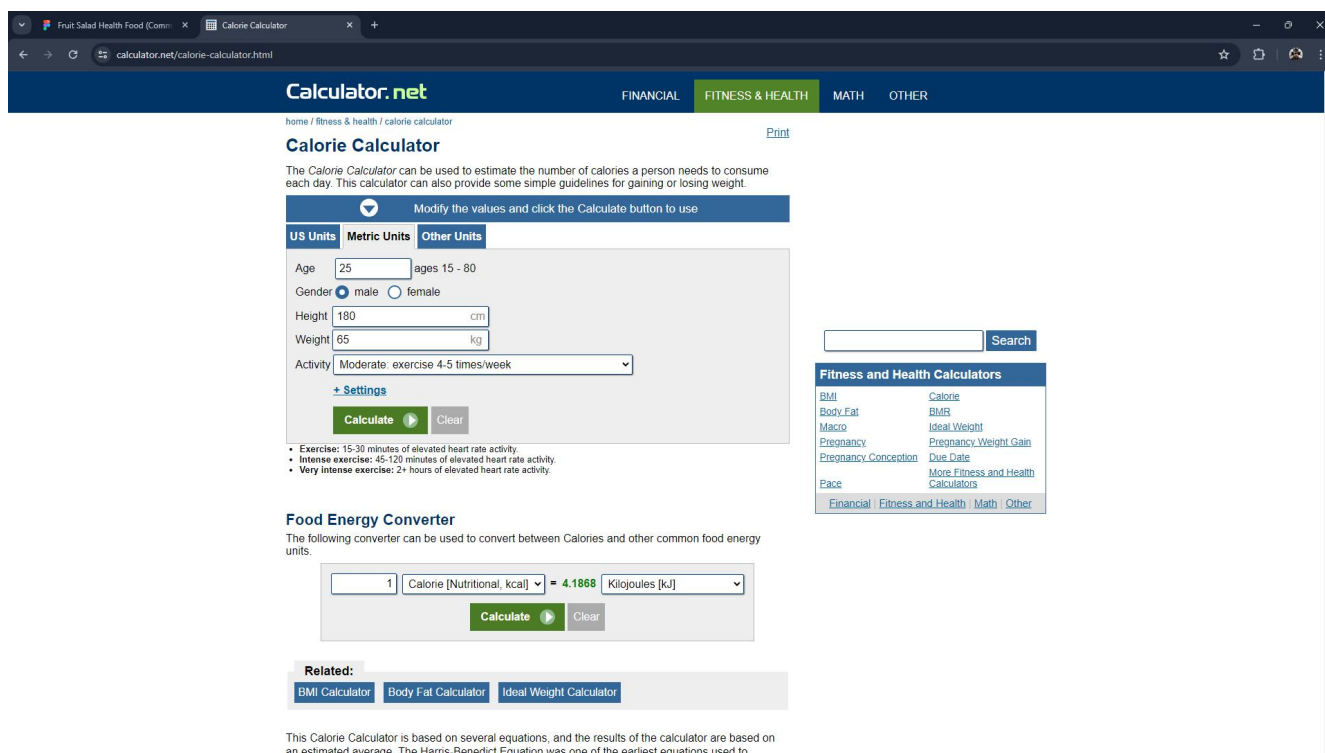


Рисунок 1.1. “Calorie Calculator”

Переваги:

1. Враховує вік людини, стать, зріст, вагу, рівень активності та її цілі. Наприклад, програма може коригувати рекомендації залежно від того, чи

прагне користувач схуднути, підтримувати вагу або набрати масу. Такий підхід дозволяє забезпечити персоналізований підхід до кожного користувача.

2. Показує, скільки калорій людині потрібно споживати щодня, щоб досягти її цілей. Програма враховує індивідуальні параметри, такі як вік, стать, зріст, вага та рівень активності, щоб надати точні рекомендації. Завдяки цьому користувач може ефективно планувати своє харчування і досягати бажаних результатів.
3. Сервіс є безкоштовним для всіх користувачів. Це означає, що будь-хто може скористатися його функціями без жодних фінансових зобов'язань. Відсутність плати за користування робить його доступним для широкої аудиторії. Такий підхід сприяє популяризації сервісу.
4. Онлайн 24/7 на будь-якому пристрої. Це означає, що користувачі можуть отримати доступ до сервісу в будь-який час і з будь-якого місця. Незалежно від того, чи це смартфон, планшет або комп'ютер, сервіс завжди доступний для зручності користувачів.

Недоліки:

1. Відсутність персонального супроводу. Це означає, що користувачі не отримують індивідуальної підтримки від фахівців. Усі рекомендації надаються автоматично на основі введених даних. Це може бути зручно для самостійного використання, але не передбачає консультацій чи додаткової допомоги.
2. Сервіс не пропонує дієти. Це означає, що користувачі не отримують конкретних дієтичних планів чи рекомендацій щодо харчування. Всі поради стосуються лише загальних принципів здорового харчування, без детальних інструкцій. Такий підхід може бути менш корисним для тих, хто шукає конкретні дієтичні плани.
3. Наявність реклами. Це означає, що під час користування сервісом користувачі бачитимуть рекламні оголошення. Це може включати банери, відео або інші форми рекламного контенту. Хоча це дозволяє сервісу

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						10
Ізм.	Лист	№ докум.	Підпис	Дата		

залишатися безкоштовним, реклама може відволікати і знижувати зручність використання.

- Немає темного режиму, що означає відсутність можливості перемикатися на темний режим інтерфейсу. Це може бути незручним для користувачів, які віддають перевагу темному фону для зменшення навантаження на очі під час тривалого використання. Відсутність темного режиму може знизити комфорт роботи із сервісом у нічний час або в умовах низького освітлення.

“Calorie calculator – Calculate your calorie needs” представляє собою веб-сервіс, який дозволяє користувачам розрахувати свою добову норму калорій і придбати необхідні харчові добавки [2]. Він надає інформацію про кількість калорій, необхідних для досягнення певних цілей, таких як схуднення, підтримка ваги чи нарощування маси. Крім того, користувачі можуть замовити харчові добавки, які підтримують їхні дієтичні потреби.

На рис. 1.2. зображено “Calorie calculator – Calculate your calorie needs”.

The screenshot shows a web browser window with the URL ch.esn.com/en/pages/kalorienrechner. The page features the ESN logo and navigation links: Products, Blog, About Us, Order tracking. The main heading is "Calorie calculator". Below it, there are several input fields and buttons:

- Gender:** FEMALE (selected), MALE
- Age:** [input field]
- Height (cm):** [input field]
- Body Weight (kg):** [input field]
- Body Fat (%):** [input field] with an "Examples" button
- Everyday Activity (except sports):** little active (only sitting)
- Sport per week:** no sports
- Goal:** WEIGHT LOSS (selected), KEEP WEIGHT, WEIGHT GAIN

A "Calculate" button is located at the bottom right of the form. To the right of the form, a summary box displays "Your target calories" and lists macronutrient targets: Proteins, Carbs, and Fat, each with a unit of grams (- g). At the bottom of the page, the text reads "ESN calorie calculator - your way to a targeted diet".

Рисунок 1.2. Calorie calculator – Calculate your calorie needs

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						11
Ізм.	Лист	№ докум.	Підпис	Дата		

Переваги:

1. Розрахунок добової норми калорій враховує не лише основний метаболічний тариф, а й активність людини протягом дня. Це дозволяє забезпечити точніші рекомендації з харчування, враховуючи фізичну активність і потреби організму в енергії.
2. Розрахунок добової норми калорій враховує вік людини, її стать, зріст, вагу, рівень активності, а також відсоток підшкірного жиру та поставлені цілі щодо здоров'я і фітнесу. Цей комплексний підхід дозволяє надати користувачам більш точні і персоналізовані рекомендації щодо харчування та досягнення їхніх цілей.
3. Сервіс доступний онлайн цілодобово, без вихідних, на будь-якому пристрої з доступом до Інтернету. Це означає, що користувачі можуть скористатися ним у будь-який зручний для них час і місце, використовуючи комп'ютер, планшет або смартфон.

Недоліки:

1. Цей сервіс не може замінити консультацію фахівця, такого як лікар чи дієтолог. Важливо звертатися до кваліфікованого спеціаліста для отримання індивідуальної консультації та рекомендацій щодо здорового харчування та фізичної активності.
2. Сервіс має платні функції, які можуть бути доступні за додаткову плату. Це може включати розширені можливості або додатковий контент, який не входить до стандартного безкоштовного пакету послуг.
3. Сервіс не має рецептів здорової їжі у своєму асортименті. Користувачі не зможуть знайти тут конкретні рецепти чи поради з приготування здорової їжі.
4. Сервіс має наявність реклами, які можуть бути різних форматів, таких як банери, відео або тексти. Реклама може відображатися на сторінках сервісу або під час користування його функціями.
5. Сервіс не має можливості темного режиму, що означає, що користувачі не можуть переключати інтерфейс на темну тему. Такий режим корисний для

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						12
Ізм.	Лист	№ докум.	Підпис	Дата		

зменшення навантаження на очі в умовах низького освітлення або в нічний час.

“Daily Calorie Intake Calculator” являє собою веб-сервіс, що дозволяє користувачам розрахувати свою добову норму калорій. Він аналізує введені користувачем дані, такі як вік, стать, зріст, вага, рівень фізичної активності, і видає рекомендації щодо кількості калорій, які необхідно споживати щодня для підтримки здорового способу життя або досягнення конкретних цілей, таких як схуднення чи набір маси [3].

На рис. 1.3. зображено Daily Calorie Intake Calculator.

The screenshot shows a web browser window with the URL 'calories.info/calorie-intake-calculator'. The page features a search bar at the top right. The main heading is 'Daily Calorie Intake Calculator' with a subtext: 'Feel free to enter your information below in the Daily Calorie Intake calculator to receive your personal current daily calorie intake, and what your body needs to fuel itself during the day with your routine!'. Below this are four numbered input boxes: 1. 'What is your sex?' with 'Male' and 'Female' buttons; 2. 'How old are you?' with '21' in the input field and 'Years' next to it; 3. 'How tall are you?' with '170' in the input field, 'cm' next to it, and radio buttons for 'cm' and 'ft/in'; 4. 'How much do you weigh?' with '60' in the input field, 'kg' next to it, and radio buttons for 'kg' and 'lb'. At the bottom, a green banner displays 'Your Personalized Calorie Counter: Use for Free!' with a star rating and a 'Create My Plan' button.

Рисунок 1.3. “Daily Calorie Intake Calculator”

Переваги:

1. Цей веб-сервіс розраховує оптимальне співвідношення маси тіла та добову норму калорій. Він враховує параметри, такі як вік, стать, зріст, рівень фізичної активності та цілі користувача, щоб надати персоналізовані рекомендації щодо харчування і підтримки оптимального стану здоров'я.
2. Сервіс завжди доступний онлайн і працює цілодобово без перерв на будь-якому пристрої з доступом до Інтернету.

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						13
Ізм.	Лист	№ докум.	Підпис	Дата		

3. Сервіс є повністю безкоштовним для всіх користувачів, без необхідності оплати чи придбання платних підписок.

Недоліки:

1. Цей сервіс, незважаючи на свої переваги, не може повністю замінити консультацію фахівця, такого як лікар чи дієтолог. Він пропонує користувачам інформацію та рекомендації здорового способу життя, але для індивідуального підходу та лікування варто звертатися до кваліфікованого медичного працівника.
2. Сервіс не містить рецептів здорової їжі. Він спрямований на розрахунок добової норми калорій та не надає конкретних страв або рецептів. Для отримання рекомендацій з харчування рекомендується звертатися до фахівців у галузі дієтології чи здорового харчування.
3. Цей сервіс має наявність реклами, які можуть бути різних форматів, таких як банери, відео або тексти. Реклама може відображатися на сторінках сервісу або під час користування його функціями. Це дозволяє сервісу залишатися безкоштовним для користувачів.
4. У сервісі відсутня можливість вибору темної теми інтерфейсу, яка зазвичай корисна для зменшення втоми очей у темний час доби або в умовах низького освітлення.

Розглянемо порівняльну таблицю аналогів. Порівняльна таблиця представлена на таблиці 1.1.

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
						14
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 1.1. Порівняльна характеристика аналогів

Характеристика	Calorie Calculator	Calorie calculator – Calculate your calorie needs	Daily Calorie Intake Calculator
Наявність реклами	Так	Так	Так
Розрахунок калорій	Так	Так	Так
Наявність рецептів	Ні	Ні	Так
Платні функції	Ні	Так	Ні
Гарний UI-UX дизайн	Ні	Ні	Ні
Темний режим	Ні	Ні	Ні

За допомогою таблиці можна зробити висновок, що жоден з аналогів не має гарного UI-UX дизайну, темної теми та містять настирливу рекламу, тому розробка мого застосунка є доцільною.

1.1.2 Технології та засоби розробки

У моєму проєкті для розробки графічного інтерфейсу та клієнтського додатку (Front-End) будуть використовуватись чотири основні технології: HTML, CSS, JavaScript та React. Розглянемо ці технології більш детально.

Особливості HTML:

1. Мова розмітки веб-сторінок, яка описує структуру та вміст сторінки [4].
2. Використовується для створення каркасу веб-сторінки, визначаючи заголовки, абзаци, зображення, посилання та інші елементи.
3. Забезпечує базову структуру, на якій будується веб-сторінка.

Особливості CSS:

1. Мова стилів, яка описує візуальне оформлення веб-сторінки (колір, шрифти, розміри тощо) [5].
2. Використовується для контролю зовнішнього вигляду веб-сторінки, роблячи її більш привабливою та зручною для користувачів.
3. Дозволяє контролювати зовнішнє оформлення веб-сторінки відповідно до бренду та цілей.

Особливості JavaScript:

1. Мова програмування, що використовується для додавання інтерактивності до веб-сторінок [6].
2. Дозволяє зробити веб-сторінки більш динамічними та функціональними, додаючи анімацію, валідацію форм, динамічні завантаження контенту та інші можливості.
3. Збільшує залучення користувачів та робить веб-сторінки більш зручними.

Особливості React:

1. Бібліотека JavaScript, що використовується для створення динамічних веб-інтерфейсів [7].
2. Спрощує розробку складних веб-інтерфейсів, роблячи його більш модульним та масштабованим.
3. Дозволяє створювати веб-інтерфейси, які швидко реагують на дії користувачів, забезпечуючи кращий користувацький досвід.

Цей вибір технологій зумовлений популярністю та передовими позиціями цих технологій у сфері веб-розробки. Їх широке використання та активний розвиток роблять їх привабливими для впровадження у мій проєкт.

Розглянемо статистику від сервісу “[lambdatest](#)”[8]. На рис. 1.4. Зображено топ технологій.

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
						16
Ізм.	Лист	№ докум.	Підпис	Дата		

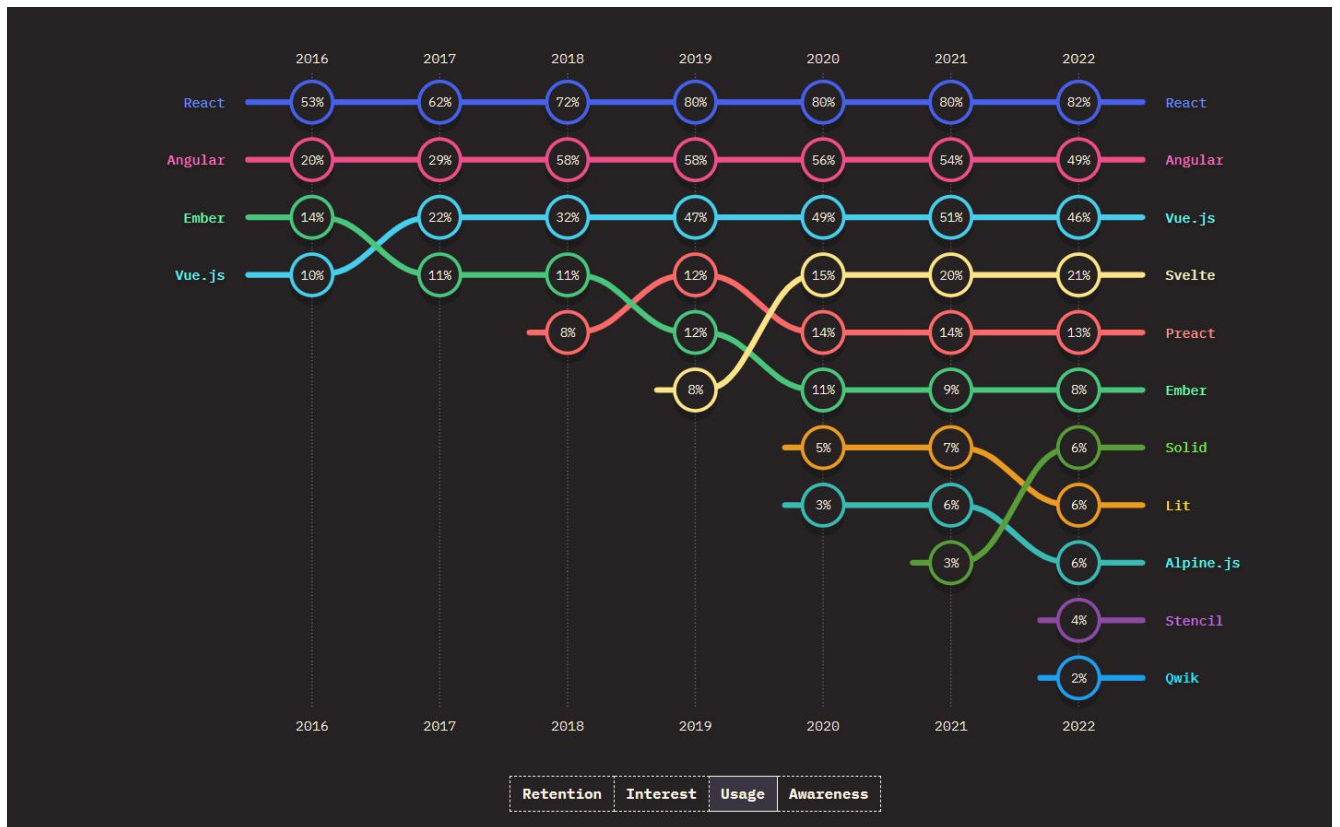


Рисунок 1.4. Топ технологій

Порівняємо популярні технології:

1. React: бібліотека для побудови інтерфейсів від Facebook, що використовує віртуальний DOM для високої продуктивності. Потребує сторонніх бібліотек для маршрутизації та управління станом, має середній поріг входу.
2. Angular: повноцінний фреймворк від Google з вбудованими рішеннями для маршрутизації, форм, та управління станом, що використовує TypeScript. Має високий поріг входу через велику кількість концепцій.
3. Vue: прогресивний фреймворк, який може використовуватись як бібліотека або як повноцінний фреймворк з офіційними інструментами для маршрутизації та управління станом. Відрізняється низьким порогом входу та легкістю вивчення.

Розглянемо порівняльну таблицю популярних технологій. Порівняльна таблиця представлена на таблиці 1.2.

Таблиця 1.2. Порівняльна характеристика популярних технологій

Критерій	React	Angular	Vue.js
Складність вивчення	Помірна, потрібно знати JS	Середня, вимагає вивчення TS	Легка, має просту API
Продуктивність	Висока, через віртуальний DOM	Висока, є побудована обертка над DOM	Висока, швидка робота з DOM
Компонентна структура	Дуже сильна, компоненти підтримуються широким спектром функцій	Сильна, вбудована підтримка компонентів	Дуже сильна, рекомендується використовувати композицію компонентів
Реактивність	Потужна реактивність через стейти та контексти	Потужна реактивність через RxJS та вбудовані сервіси	Потужна реактивність через Vue-об'єкти та компоненти
Величина спільноти	Велика, активне співтовариство та багато зовнішніх бібліотек	Велика, має підтримку від Google та активну спільноту	Велика, швидко зростає та має активну спільноту
SEO підтримка	Потребує додаткових зусиль, але можливе	Добра, можливе використання SSR та Prerendering	Добра, можливе використання SSR
Масштабованість	Дуже добре масштабується за рахунок структури	Добре масштабується, але може бути складніше у великих проєктах	Дуже добре масштабується за рахунок простоти та модульності

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 07 001. 00 ДП ПЗ

Арк.

18

1.2 Проектування веб-системи

1.2.1 Технічне завдання на розробку

Нижче наведені необхідні вимоги до застосунку, що розробляється:

Наявність привабливого графічного інтерфейсу користувача (GUI):

Графічний інтерфейс повинен бути: сучасним, відповідати останнім трендам та технологіям [16]. Також мати естетичний дизайн, що привертає увагу користувачів і сприяє комфортному використанню програмного продукту. При створенні дизайну важливо враховувати: сучасні тенденції у дизайні, такі як чітка структура та правильний розмір елементів, а також гармонійність кольорів та використання приємних кольорових схем, які забезпечують комфорт інтерфейсу для користувачів.

Інтуїтивно зрозумілий користувацький досвід (UX):

Ефективна навігація в інтерфейсі повинна: бути простою та інтуїтивно зрозумілою для користувачів, забезпечувати швидкий доступ до основних функцій та розділів програми та використовувати зрозумілі піктограми, текстові посилання та інші елементи для зручної навігації [17].

Інформаційні розділи:

Блог або розділ новин з трендами у світі здорового харчування має: представляти актуальні тенденції та новини у сфері здорового харчування та надавати інформацію про сучасні дієти.

Адаптивний дизайн:

Повна підтримка адаптивного дизайну для зручного відображення на різних пристроях (планшети, мобільні телефони, комп'ютери) включає в себе: гнучке розташування елементів інтерфейсу з урахуванням розмірів екрану та орієнтації пристрою, оптимізацію контенту для швидкого завантаження на різних пристроях та використання адаптивних зображень та шрифтів для забезпечення чіткості та зручності відображення.

Калькулятор калорій:

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
						19
Ізм.	Лист	№ докум.	Підпис	Дата		

Розробка калькулятора калорій для користувачів для визначення денної норми калорій передбачає: введення основних параметрів користувача, таких як вік, стать, вага, зріст і рівень активності, врахування коефіцієнтів, що визначають необхідну кількість калорій для підтримання поточного ваги, зниження або збільшення ваги та виведення результатів у зручному форматі.

Оптимізація швидкості роботи веб-системи:

Оптимізація швидкості роботи веб-системи є ключовим аспектом у забезпеченні гарного користувацького досвіду: ліниве завантаження компонентів - цей підхід дозволяє завантажувати компоненти або ресурси лише тоді, коли вони потрібні для відображення на сторінці або в додатку. Це допомагає зменшити час завантаження і поліпшити продуктивність та оптимізація коду - це означає вдосконалення коду програми, щоб він працював ефективніше і швидше. Це може включати усунення зайвих або неефективних частин коду, використання оптимізованих алгоритмів та структур даних.

1.2.2 Проєктування дизайну веб-застосунку

Коли замовляють веб-сайт, замовник очікує отримати прототипи для подальшого узгодження. Це важливий етап, де визначається дизайн та функціональність перед реалізацією. При розробці прототипів враховують технічні вимоги та предметну область, а макети основних елементів, як футер чи навігаційне меню, розробляються у Figma, або інших графічних редакторах, таких як Sketch, Adobe XD, InVision, і Marvel.

Розглянемо порівняльну таблицю аналогів для розробки дизайну. Порівняльна таблиця представлена на таблиці 1.3.

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						20
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 1.3. Порівняльна таблиця аналогів для розробки дизайну

Критерій	Figma	Sketch	Adobe XD	InVision	Marvel
Ціна	До \$45/місяць	\$120/рік	До \$23/місяць	До \$5/місяць	До \$42/місяць
Платформи	Веб, Windows, macOS, Linux	macOS	Windows, macOS	Веб, Windows, macOS	Веб
Макетування	Інтерактивні і векторні макети	Векторні макети (macOS)	Автоматичне вирівнювання	Макетування та прототипування	Просте макетування
Прототипування	Інтерактивні прототипи	Інтеграція з InVision	Мікровзаємодії та переходи	Детальні анімації	Швидке прототипування
Спільна робота	В реальному часі	Обмежена	В реальному часі	Потужні інструменти	Коментування
Відстеження змін	Відстеження всіх змін	Вбудоване	Немає	Обмежена функціональність	Немає

Використання Figma для створення прототипів.

Figma є потужним інструментом для створення прототипів, який дозволяє дизайнерам працювати в реальному часі, обмінюватися ідеями і отримувати зворотний зв'язок від замовника [10].

Основні переваги використання Figma включають:

1. Колаборація в реальному часі: Дизайнери та замовники можуть одночасно працювати над прототипом, вносячи зміни та коментарі в режимі реального часу. Це значно прискорює процес узгодження і дозволяє миттєво реагувати на запити замовника.
2. Гнучкість та масштабованість: Figma дозволяє створювати прототипи різного рівня деталізації, від простих каркасів до високодеталізованих інтерактивних макетів. Це забезпечує гнучкість у плануванні та адаптації до змінних вимог проєкту.
3. Інтерактивність: З допомогою Figma можна створювати інтерактивні прототипи, які демонструють, як буде працювати навігація, переходи між сторінками і взаємодія з користувачем. Це допомагає замовнику краще зрозуміти кінцевий продукт і внести корективи на ранніх стадіях.

Елементи дизайну інтерфейсу користувача (UI):

Розглянемо дизайн хедеру веб-застосунку, що розробляється. Хедер є одним із ключових елементів інтерфейсу, адже він задає тон всьому веб-сайту та забезпечує користувачам швидкий доступ до основних розділів та функцій.

Основні елементи хедеру:

1. Логотип: логотип компанії або бренду зазвичай розташовується в лівому верхньому куті хедеру. Він виконує функцію ідентифікації веб-сайту і є посиланням на головну сторінку.
2. Навігаційне меню: основне меню веб-системи, яке включає посилання на ключові розділи. Навігаційне меню повинно бути зручним та інтуїтивно зрозумілим, щоб користувачі могли легко знаходити потрібну інформацію.
3. Перемикач темної теми: перемикач для зміни режиму відображення веб-сайту на темну або світлу тему. Ця функція стає все популярнішою, оскільки дозволяє користувачам налаштовувати інтерфейс під свої вподобання та зменшувати навантаження на очі.

На рис. 1.5. зображено дизайн хедеру веб-застосунку, що розробляється.

TASTY HEALTH

Home

Diets

Calorie Calculator

About Us



Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 07 001. 00 ДП ПЗ

Арк.

22

Рисунок 1.5. Дизайн хедеру веб-застосунка, що розробляється

Розглянемо дизайн футера веб-застосунка, котрий розробляється. Футер є важливим елементом інтерфейсу, адже він завершує структуру сторінки та надає користувачам доступ до додаткової інформації та функцій.

Основні елементи футера:

1. Інформація про компанію: футер зазвичай містить основну інформацію про компанію або бренд, таку як назва, короткий опис та місцезнаходження. Це допомагає користувачам зрозуміти, хто стоїть за веб-сайтом.
2. Контактні дані: секція з контактною інформацією, яка може включати номер телефону, електронну пошту та фізичну адресу. Також можуть бути додані посилання на форми зворотного зв'язку.
3. Посилання на політику конфіденційності та умови використання: важливі документи, які пояснюють користувачам їхні права та обов'язки під час використання веб-сайту. Посилання на ці документи зазвичай розміщуються у футері для легкого доступу.
4. Навігаційні посилання: футер може включати додаткові навігаційні посилання на важливі розділи веб-сайту, такі як "Про нас", "Послуги", "Контакти", "Блог" тощо.
5. Соціальні мережі: іконки з посиланнями на профілі компанії у соціальних мережах дозволяють користувачам швидко знайти і підписатися на ваші сторінки в Facebook, Twitter, Instagram тощо.

На рис. 1.6. зображено дизайн футера веб-застосунка, що розробляється.

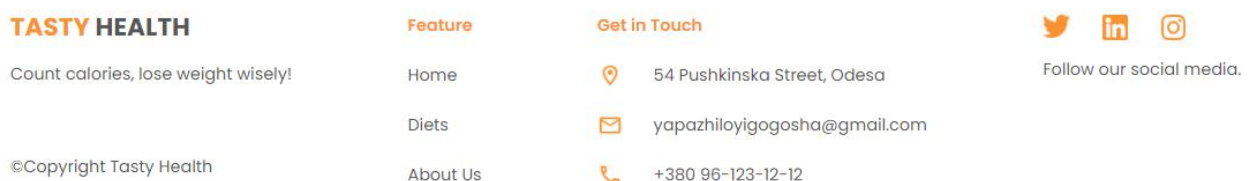


Рисунок 1.5. Дизайн футера веб-застосунка, що розробляється

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		23

1.2.3 Проєктування архітектури веб-застосунку

Застосунок, що розробляється, буде проєктуватися на архітектурі SPA [9]. Дана архітектура є важливою для створення сучасних веб-додатків, оскільки вона дозволяє забезпечити більш швидку та плавну взаємодію з користувачем, без зайвих перезавантажень сторінок.

Переваги SPA [9]:

SPA динамічно переписує поточну веб-сторінку з новими даними з веб-сервера, на відміну від звичайного методу, коли веб-браузер завантажує абсолютно нову сторінку.

1. SPA пропонують гнучкий і динамічний користувацький досвід, завантажуючи всі необхідні ресурси під час першого завантаження сторінки. Під час взаємодії користувачів з додатком контент оновлюється динамічно, що усуває необхідність перезавантаження сторінок і зменшує навантаження на сервер.
2. Покращена продуктивність додатків, узгодженість, скорочення часу на розробку та зниження витрат на інфраструктуру.

Недоліки SPA [9]:

1. У контексті пошукової оптимізації (SEO) можуть виникнути такі труднощі: Оскільки контент генерується динамічно за допомогою JavaScript, пошукові системи можуть зіткнутися з труднощами при індексації та ранжируванні сторінок SPA, особливо якщо не використовувати правильні налаштування SEO.
2. Оскільки весь код і ресурси завантажуються один раз під час першого завантаження сторінки, це може призвести до збільшення часу завантаження великих додатків, особливо на повільних інтернет-з'єднаннях.

Мапа SPA застосунку, який розробляється зображена на рис. 1.5.

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						24
Ізм.	Лист	№ докум.	Підпис	Дата		

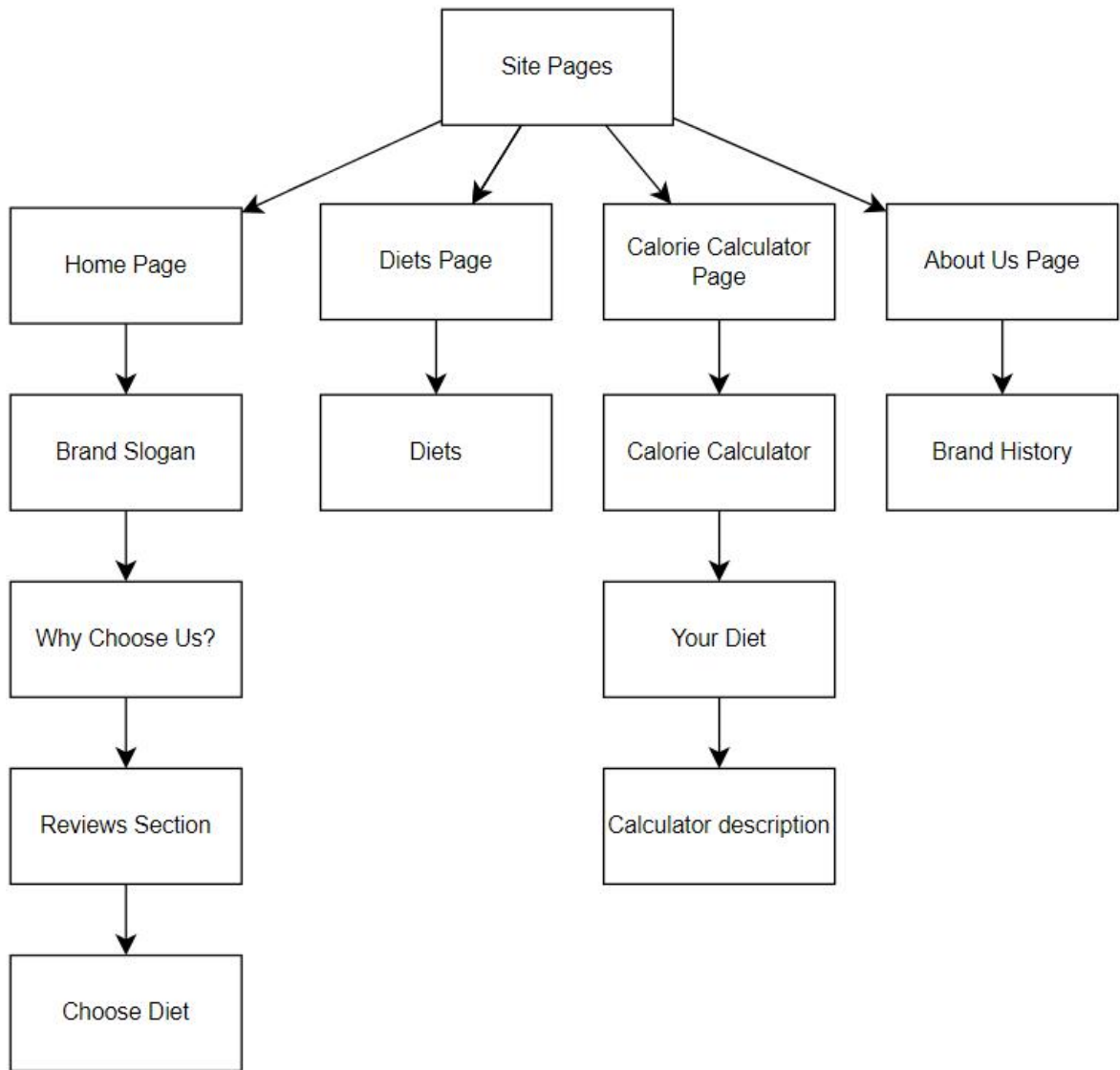


Рисунок 1.5. Мапа застосунку

1.3 Реалізація веб-системи

1.3.1 Файлова структура веб-застосунку

Коренева папка веб-застосунку:

Коренева папка веб-застосунку, що розробляється, наведений на рис. 1.6. На цьому зображенні представлена структура папок і файлів, які використовуються для організації проєкту.

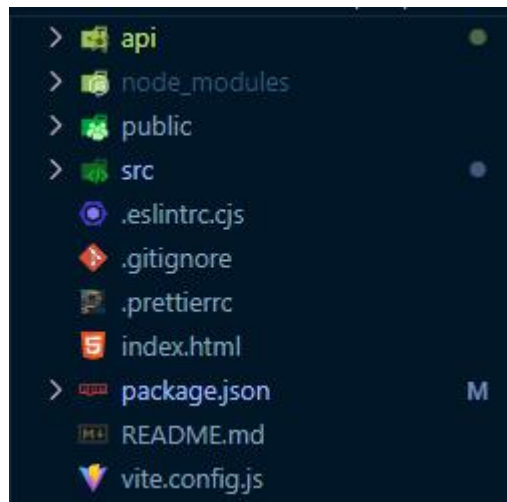


Рисунок 1.6. Кореневі папки веб-застосунку

Папка “./api” у проєкті призначений для зберігання коду, що відповідає за взаємодію застосунку із сервером. Це може включати файли для виклику API, обробки запитів і відповідей сервера, роботи з базою даних або інші компоненти, пов'язані з взаємодією з сервером у веб-застосунку.

Папка “./public” призначений для зберігання статичних файлів, таких як зображення, шрифти та favicon. Вона використовується для доступу до цих файлів напряму з клієнтського браузера у веб-застосунку.

Папка “./src” використовується для зберігання вихідного коду програми. У цій папці зазвичай розміщуються файли з основними скриптами, компонентами, стилізацією та іншими ресурсами, що стосуються розробки програмного продукту.

Файл “README.md” містить опис проєкту. Зазвичай в ньому розміщують інформацію про загальну мету проєкту, його функціональність, вимоги до запуску та використання, а також інші корисні вказівки для розробників або користувачів.

Файл “package.json” містить метадані проєкту, такі як залежності, сценарії для запуску, тестування та інші операції, а також конфігурацію інструментів розробки, наприклад, Vite у вашому випадку. Цей файл є важливим для керування проєктом та його налаштування.

Папка “public”:

Папка “public” веб-застосунку, що розробляється зображений на рис. 1.7.

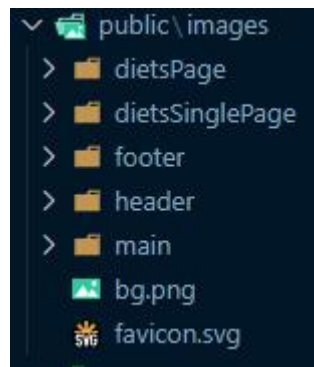


Рисунок 1.7. Папка “ public”

Папка “images” у папці “public” зазвичай використовується для зберігання статичних зображень, які використовуються у React-застосунку. Ці зображення включають логотипи, іконки, фотографії та інші графічні елементи, необхідні для відображення інтерфейсу користувача. Розміщення цих файлів у папці “./public/images” забезпечує зручне та організоване управління графічними ресурсами.

Папка “src”:

Папка “src” веб-застосунку, що розробляється зображений на рис. 1.8.

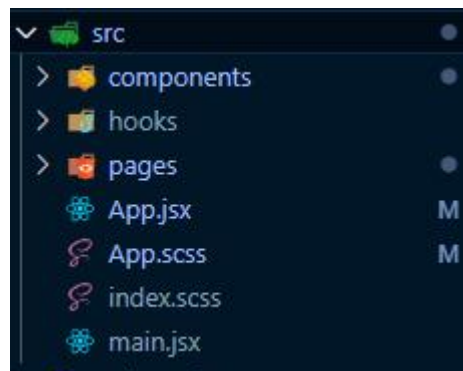


Рисунок 1.8. Папка “src”

Файли "app.jsx" та "main.jsx" у папці "src" React застосунку відіграють ключову роль у структурі та функціонуванні додатку.

Файл “app.jsx” зазвичай містить основний компонент застосунку, який служить кореневим компонентом для всієї програми. У ньому визначаються всі основні елементи інтерфейсу, а також інтеграція інших компонентів, таких як маршрутизація, управління станом, та інші важливі функціональні можливості.

Код файла “app.jsx” у веб-застосунку, що розробляється приведений нижче.

```
import { createBrowserRouter, RouterProvider } from "react-router-dom";
import MainPage, {
  KcalCalcPage,
  DietsPage,
  AboutUsPage,
  NotFoundPage,
  DietsSinglePage,
} from "./pages";
import "./App.scss";
const router = createBrowserRouter([
  {
    path: "/",
    element: <MainPage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/kcal-calc",
    element: <KcalCalcPage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/diets",
    element: <DietsPage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/diets/:id",
    element: <DietsSinglePage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/about-us",
    element: <AboutUsPage />,
    errorElement: <NotFoundPage />,
  },
],
]);
const App = () => {
  return <RouterProvider router={router} />;
};
export default App;
```

Файл “main.jsx”, з іншого боку, служить точкою входу для додатку. У цьому файлі відбувається рендеринг основного компонента “App” до кореневого DOM-елемента. Саме в “main.jsx” відбувається імпорт всіх необхідних бібліотек та ресурсів, таких як React, ReactDOM, та глобальні стилі, а також ініціалізація застосунку через метод ReactDOM.render або createRoot для новіших версій React.

Код файла “main.jsx” у веб-застосунку, що розробляється приведений нижче.

```

import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";
import "./index.scss";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
);

```

Папка "components" містить React-компоненти, які є будівельними блоками користувацького інтерфейсу веб-застосунку. Ці компоненти використовуються для створення різноманітних елементів інтерфейсу, таких як кнопки, форми, картки тощо.

Папка "components" веб-застосунку, що розробляється зображений на рис. 1.9.

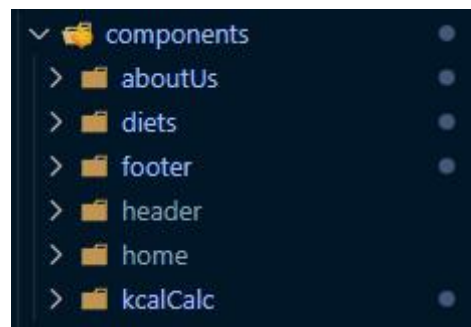


Рисунок 1.9. Папка "components"

Папка "hooks": містить React-хуки, які дозволяють керувати станом та життєвим циклом компонентів без написання громіздкого коду, а також хуки зазвичай використовуються для повторно використовуваної логіки, такої як управління формою, підписка на дані або виконання побічних ефектів.

Папка "hooks" веб-застосунку, що розробляється зображений на рис. 1.10.

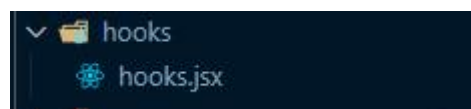


Рисунок 1.10. Папка "hooks"

Папка "pages": містить React-компоненти, які представляють собою окремі сторінки вашої програми, а також кожен компонент сторінки зазвичай

відповідає за маршрутизацію, завантаження даних та відображення контенту своєї сторінки.

Папка pages веб-застосунку, що розробляється зображений на рис. 1.11.

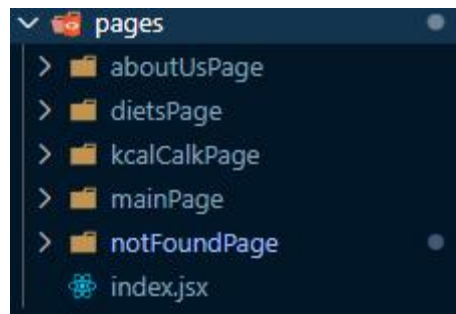


Рисунок 1.11. Папка pages

1.3.2 Налаштування роутингу

Роутинг являє собою складний процес визначення та управління шляхами обробки запитів користувачів [11]. Роутинг використовується для того, щоб визначити, як додаток повинен реагувати на різні URL-адреси, які введені користувачем. Він визначає, який компонент або сторінка має бути відображений для кожного URL-запиту, дозволяючи користувачам легко навігувати по додатку та отримувати необхідну інформацію.

У React застосунках роутинг може бути реалізований за допомогою різних бібліотек і фреймворків таких як React Router, Wouter, та Next.js. Вони дозволяють організувати навігацію між різними сторінками або компонентами в односторінкових додатках (SPA).

Розглянемо порівняльну таблицю аналогів для роутингу. Порівняльна таблиця представлена на таблиці 1.4.

Таблиця 1.4. Порівняльна таблиця аналогів для роутингу

Критерій	React Router	Wouter	Next.js
Зручність API	Потужний, але складний для новачків	Простий і зрозумілий API	Вбудований роутинг, зручний для швидкого старту
Інтеграція з React	Офіційна бібліотека для React	Незалежна бібліотека з хуками	Вбудований роутинг
Складність налаштування	Складний, вимагає конфігурації	Проста і швидка налаштування	Стандартизований підхід
Підтримка функціональності	Вкладені, динамічні, захищені маршрути	Базовий функціонал	Вбудована підтримка роутингу та інших функцій

Для веб-застосунку, що розробляється я вирішив використовувати саме React Router. Цей фреймворк надає зручний та потужний механізм для визначення маршрутів в додатку, що дозволяє ефективно керувати відображенням компонентів на основі URL-адреси користувача [12].

Плюси React Router:

1. Декларативний підхід до роутингу: React Router дозволяє визначати маршрути у декларативному стилі, що спрощує розуміння та підтримку коду. Наприклад, можна визначити маршрути за допомогою компонентів, таких як `<Route>` і `<Switch>`.
2. Вбудована підтримка навігації: React Router надає засоби для навігації між різними сторінками застосунку. Можна використовувати компоненти, такі як `<Link>` для створення посилань, які автоматично оновлюють URL-адресу без перезавантаження сторінки.

3. Параметри маршрутизації: React Router підтримує передачу параметрів у маршрути, що дозволяє створювати динамічні URL-адреси. Наприклад, можна зробити URL-адресу /users/:id, де :id - це динамічний параметр, який може бути використаний для завантаження конкретного користувача.

Код налаштування роутингу у веб-застосунку, що розробляється приведений нижче.

```
const router = createBrowserRouter([\n  {\n    path: "/",\n    element: <MainPage />,\n    errorElement: <NotFoundPage />,\n  },\n  {\n    path: "/kcal-calc",\n    element: <KcalCalcPage />,\n    errorElement: <NotFoundPage />,\n  },\n  {\n    path: "/diets",\n    element: <DietsPage />,\n    errorElement: <NotFoundPage />,\n  },\n  {\n    path: "/diets/:id",\n    element: <DietsSinglePage />,\n    errorElement: <NotFoundPage />,\n  },\n  {\n    path: "/about-us",\n    element: <AboutUsPage />,\n    errorElement: <NotFoundPage />,\n  },\n]);
```

1.3.3 Створення шаблонізаторів

Шаблонізатори в React – це механізми або підходи, які дозволяють створювати динамічні компоненти, що генерують HTML-код на основі даних та логіки [13]. В React це досягається через використання JSX та компонентів, що дозволяють описувати UI у вигляді декларативних шаблонів, які автоматично оновлюються при зміні стану або пропсів.

Шаблонізатор хедеру веб-застосунку зображений на рис. 1.12.

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
						32
Ізм.	Лист	№ докум.	Підпис	Дата		

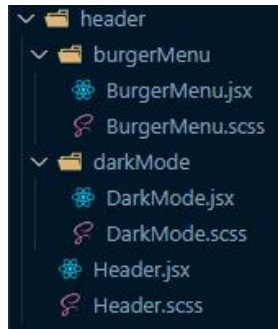


Рисунок 1.12. Шаблонізатор хедеру веб-застосунку

Файл "Header" – це основний файл, що відповідає за відображення хедеру, забезпечуючи його правильну роботу та відображення на всіх сторінках веб-сайту.

У папці "darkMode" знаходиться компонент, спеціально призначений для реалізації перемикача темної теми в хедері, що дозволяє користувачам легко змінювати вигляд веб-сайту відповідно до своїх уподобань.

У папці "burgerMenu" розміщений компонент бургер-меню, який з'являється в хедері на екранах з роздільною здатністю менше 992 пікселів, забезпечуючи зручну навігацію на мобільних пристроях і маленьких екранах.

Розглянемо шаблонізатор хедеру:

```
const Header = () => {
  return (
    <motion.header
      className="header"
      initial={{ opacity: 0 }}
      animate={{ opacity: 1 }}
      transition={{ duration: 0.3 }}>
      <div className="container">
        <div className="header__menu menu">
          <Link to="/" className="menu__logo"><span>Tasty</span>
Health</Link>
          <nav className="menu__body">
            <ul className="menu__list">
              <li className="menu__item">
                <NavLink to="/" className="menu__link"> </NavLink>
              </li>
              <li className="menu__item">
                <NavLink to="/diets"
className="menu__link">Diets</NavLink>
              </li>
              <li className="menu__item">
                <NavLink to="/kcal-calc" className="menu__link">Calorie
Calculator</NavLink>
              </li>
              <li className="menu__item">

```

```

        <NavLink to="/about-us" className="menu__link">About
Us</NavLink>
        </li>
    </ul>
</nav>
<div className="menu__buttons">
    <DarkMode />
    <BurgerMenu />
</div>
</div>
</div>
</motion.header>
);
};

```

Шаблонізатор футеру веб-застосунку зображений на рис. 1.13.

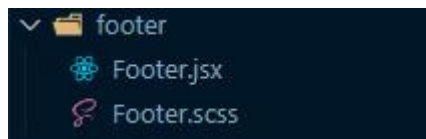


Рисунок 1.13. Шаблонізатор футеру веб-застосунку

Файл "Footer" – це основний файл, що відповідає за відображення футеру, забезпечуючи його правильну роботу та відображення на всіх сторінках веб-сайту.

1.3.4 Створення представлень

У React компонентах, представлення відповідає за візуальну складову інтерфейсу користувача, описуючи його структуру та зовнішній вигляд [14]. Представлення описується за допомогою JSX або JavaScript, що дозволяє React динамічно оновлювати користувацький інтерфейс на основі змін стану або пропсів.

Створимо представлення для головної сторінки:

Представлення для головної сторінки веб-застосунку зображені на рис. 1.14.

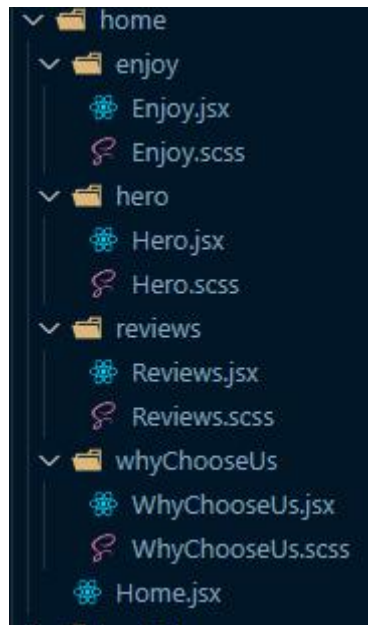


Рисунок 1.14. Представлення для головної сторінки веб-застосунку

У папці "home" знаходяться представлення для головної сторінки веб-застосунку, які включають в себе компоненти, шаблони та стилізацію, необхідні для відображення основного вмісту та інтерфейсу сторінки.

Папка "hero" в структурі проекту містить компоненти, які спеціалізуються на відображенні першої секції головної сторінки веб-застосунку. Ці компоненти відповідають за візуальне оформлення та функціонал.

Папка "whyChooseUs" в структурі проекту містить компоненти, які спеціалізуються на відображенні секції "Чому обирають нас" на головній сторінці веб-застосунку.

Папка "reviews" у структурі проекту містить компоненти, які відповідають за відображення секції відгуків на головній сторінці веб-застосунку. Ці компоненти забезпечують візуальне оформлення та функціонал для відображення відгуків користувачів про веб-застосунок.

Папка "enjoy" у структурі проекту містить компоненти, які відповідають за відображення секції "Насолоджуйтеся" на головній сторінці веб-застосунку. Ці компоненти відповідають за візуальне оформлення та функціонал.

Файл "Home" у структурі проекту відповідає за відображення головної сторінки веб-застосунку.

Створимо представлення для сторінки з дієтами:

Представлення для сторінки з дієтами зображені на рис. 1.15.

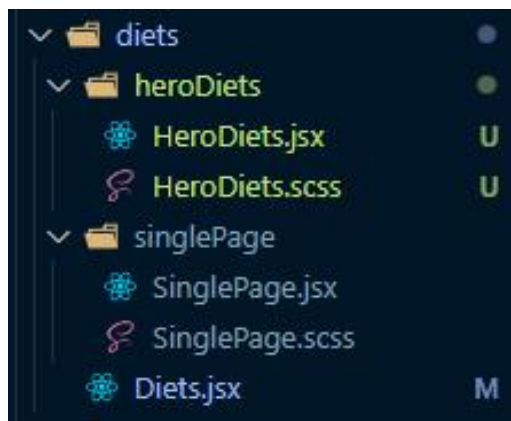


Рисунок 1.15. Представлення для головної сторінки веб-застосунку

У папці «diets» знаходяться всі потрібні представлення для сторінки з дієтами.

Папка «heroDiets» містить компонент, який відповідає за представлення секції з дієтами.

Папка «singlePage» містить компонент, який відповідає за представлення сторінки з одною дієтою.

Файл «Diets» відповідає за відображення сторінки з дієтами.

Створимо представлення для сторінки “Про нас”:

Представлення для сторінки “Про нас” зображені на рис. 1.16.

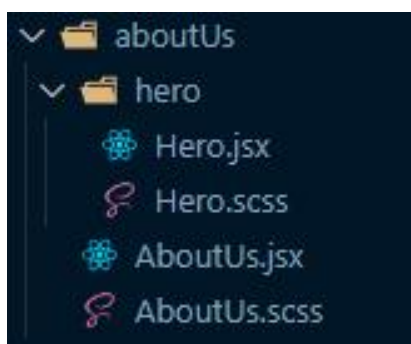


Рисунок 1.16. Представлення для сторінки “Про нас”

У папці «aboutUs» знаходяться всі потрібні представлення для сторінки “Про нас”.

Папка «heroAbout» містить компонент, який відповідає за представлення секції “Про нас”.

Файл «AboutUs» відповідає за відображення сторінки “Про нас.

Створимо представлення для сторінки з калькулятором калорій:

Представлення для сторінки з калькулятором калорій зображені на рис.

1.17.

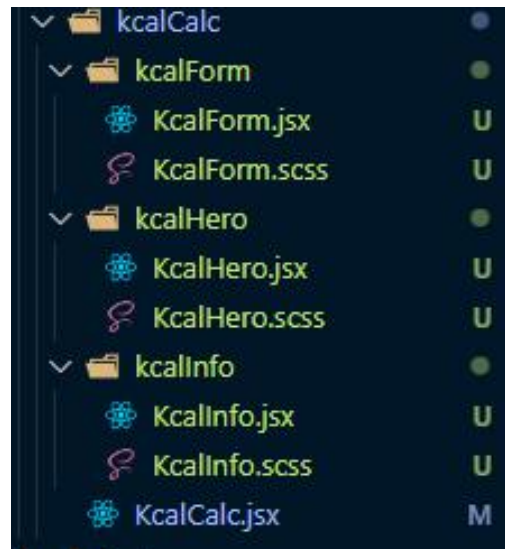


Рисунок 1.17. Представлення для головної сторінки веб-застосунку

У папці «kcalCalc» знаходяться всі потрібні представлення для сторінки з калькулятором калорій.

Папка «kcalHero» містить компонент, який відповідає за представлення секції з калькулятором калорій.

Папка «kcalForm» відповідає за компонент, який містить представлення форми з калькулятором калорій.

Папка «kcalInfo» відповідає за компонент, який містить представлення з інформацією про калькулятор калорій.

Файл «KcalCalc» відповідає за відображення сторінки з калькулятором калорій.

Розглянемо функцію для підрахунка калорій біль розгорнуто:

Код функції для підрахунка калорій:

```
const formik = useFormik({
```

```

initialValues: {
  weight: "",
  height: "",
  age: "",
},
validationSchema: validationSchema,
onSubmit: values => {
  const { weight, height, age } = values;
  const weightNum = parseFloat(weight);
  const heightNum = parseFloat(height);
  const ageNum = parseInt(age, 10);
  const activityNum = parseFloat(activity);
  let bmr;
  if (gender === "male") {
    bmr = 10 * weightNum + 6.25 * heightNum - 5 * ageNum + 5;
  } else {
    bmr = 10 * weightNum + 6.25 * heightNum - 5 * ageNum - 161;
  }
  let totalCalories = bmr * activityNum;
  if (goal === "lose") {
    totalCalories -= 300;
  } else if (goal === "gain") {
    totalCalories += 300;
  }
  setCalories(totalCalories.toFixed(0));
},
});

```

Ініціалізація значень форми:

`initialValues`: об'єкт, що містить початкові значення полів форми: `weight` (вага), `height` (зріст) та `age` (вік), які спочатку є порожніми рядками.

Схема валідації:

`validationSchema`: схема валідації для перевірки введених значень.

Обробка відправлення форми:

`onSubmit`: функція, що викликається при відправленні форми. Вона приймає об'єкт `values`, що містить значення всіх полів форми.

Всередині `onSubmit`:

1. Значення полів форми (`weight`, `height`, `age`) вилучаються з об'єкта `values`.
2. Ці значення перетворюються в числові формати: `weightNum` (вага), `heightNum` (зріст) та `ageNum` (вік).
3. Значення `activity` також перетворюється в число (`activityNum`).
4. Розраховується основний обмін речовин (BMR) в залежності від статі (`gender`):

Для чоловіків: $BMR = 10 * weightNum + 6.25 * heightNum - 5 * ageNum + 5$

Для жінок: $BMR = 10 * weightNum + 6.25 * heightNum - 5 * ageNum - 161$

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
						38
Ізм.	Лист	№ докум.	Підпис	Дата		

1. Загальна калорійність розраховується як добуток BMR і activityNum:
totalCalories = bmr * activityNum
2. В залежності від мети (goal), загальна калорійність коригується:
Якщо мета - схуднення (lose), калорійність зменшується на 300.
Якщо мета - набір ваги (gain), калорійність збільшується на 300.
3. Підсумкова кількість калорій встановлюється за допомогою функції setCalories і округлюється до цілого числа.

1.3.5 Створення анімацій

У моєму проєкті будуть анімації, які додадуть інтерактивності та покращать користувацький досвід. Для реалізації анімацій я планую використовувати бібліотеку Framer Motion, яка забезпечує простий та гнучкий спосіб створення плавних та складних анімацій. Завдяки цій бібліотеці я зможу легко анімувати компоненти React, додаючи ефекти переходів, змін стану та руху елементів.

Розглянемо анімації компонента “Enjoy” нижче:

```
<motion.section
  className="enjoy"
  initial={{ y: 150 }}
  whileInView={{ y: 0 }}
  transition={{ duration: 0.3 }}
  viewport={{ once: true }}
>
  <div className="container">
    <div className="enjoy__content">
      <p className="enjoy__label">DIETS THAT MAKES YOU FALL IN LOVE</p>
      <h2 className="enjoy__title title">
        <span>ENJOY</span> THE BEST DIETS
      </h2>
      {loading ? <div className="loader"></div> : renderMenu()}
    </div>
  </div>
</motion.section>
```

Цей код робить наступне:

1. <motion.section>: Це тег section, який обгорнутий в компонент motion з бібліотеки Framer Motion. Це дозволяє застосовувати анімації до цього елементу.

2. `initial={{ y: 150 }}`: Встановлює початковий стан анімації, де секція зміщена вниз на 150 пікселів по осі y.
3. `whileInView={{ y: 0 }}`: Визначає кінцевий стан анімації, який буде досягнуто, коли елемент стане видимим на екрані (секція підніметься на 0 пікселів по осі y).
4. `transition={{ duration: 0.3 }}`: Налаштовує тривалість анімації на 0.3 секунди.
5. `viewport={{ once: true }}`: Гарантує, що анімація буде програна тільки один раз при першій появі елементу в зоні видимості.

Всі інші анімації секцій створені по аналогії.

Важливою частиною проєкту буде оптимізація продуктивності анімацій, щоб забезпечити плавну роботу додатку навіть на пристроях з обмеженими ресурсами. Для цього я планую використовувати техніку, таку як `lazy-loading` компонентів.

1.3.6 Оптимізаці веб-застосунку.

В проєкті для оптимізації буде використовуватися `React.lazy`, що дозволяє здійснювати динамічне завантаження компонентів [15]. Це означає, що компоненти будуть завантажуватися лише тоді, коли вони дійсно необхідні, що зменшує час початкового завантаження додатку і покращує його продуктивність.

Нижче приклад ледачого компоненту з мого веб-застосунку:

```
const LazyKcalCalcPage = React.lazy(  
  () => import("./kcalCalcPage/KcalCalcPage"),  
);
```

Розглянемо код більш детально:

1. `React.lazy`: Дозволяє ліниво завантажувати компонент, відкладаючи його завантаження до моменту, коли він дійсно знадобиться.
2. Динамічний імпорт: Функція `() => import(«./kcalCalcPage/KcalCalcPage») повертає проміс, який завантажує модуль KcalCalcPage.`
3. Використання змінної: `LazyKcalCalcPage` зберігає компонент, що ліниво завантажується.

Нижче приклад ледачого компоненту з мого веб-застосунку:

```
export const KcalCalcPage = () => {
  return (
    <Suspense fallback={<div className="loader"></div>}>
      <LazyKcalCalcPage />
    </Suspense>
  );
};
```

Розглянемо код більш детально:

1. **Suspense**: компонент React, який показує запасний контент доки ледачий компонент завантажується.
2. `fallback={<div className=«loader»></div>}`: запасний контент, який відображається, поки `LazyKcalCalcPage` завантажується.
3. `<LazyKcalCalcPage />`: Компонент, що завантажується ліниво, який буде завантажений і відображений всередині `Suspense`.

1.4 Тестування веб-застосунку

Виконаємо ручне тестування створеного веб-застосунку, перевіривши основні функції ключових веб-сторінок.

Ручне тестування включає процес перевірки програмного забезпечення вручну для виявлення дефектів без використання автоматизованих інструментів.

Цей метод тестування є критично важливим етапом життєвого циклу майже будь-якого програмного продукту, оскільки дозволяє виявити помилки, які можуть бути пропущені автоматизованими тестами, а також помилки, що впливають на користувацький досвід.

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						41
Ізм.	Лист	№ докум.	Підпис	Дата		

1.4.1 Функціональний огляд головної сторінки “Home”

Розглянемо наступну головну сторінку веб-застосунку – “Home”.

Веб-сторінка "Home" надає користувачам кілька важливих функціоналів:

1. На головній сторінці веб-застосунку користувачі зустрічаються з вітальним повідомленням. Це повідомлення створює позитивне перше враження у користувачів та налаштовує їх на подальшу взаємодію з сайтом. Такий підхід допомагає покращити користувацький досвід та збільшити відвідуваність сайту.
2. На сторінці веб-застосунку є розділ з відгуками клієнтів. Тут користувачі можуть ознайомитися з досвідом інших людей, яке сприяє створенню довіри до сервісу та дозволяє їм отримати уявлення про якість та надійність продукту або послуги.
3. Також на сторінці веб-застосунку присутній селектор дієт за цілями. Це функціональний елемент, який дозволяє користувачам швидко знайти необхідну інформацію про різні дієти та вибрати оптимальний варіант відповідно до їхніх потреб та цілей. Такий підхід сприяє зручності використання застосунку і підвищує задоволення користувачів від його використання.

На рис. 1.18. зображено головну сторінку веб-застосунку.

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
						42
Ізм.	Лист	№ докум.	Підпис	Дата		

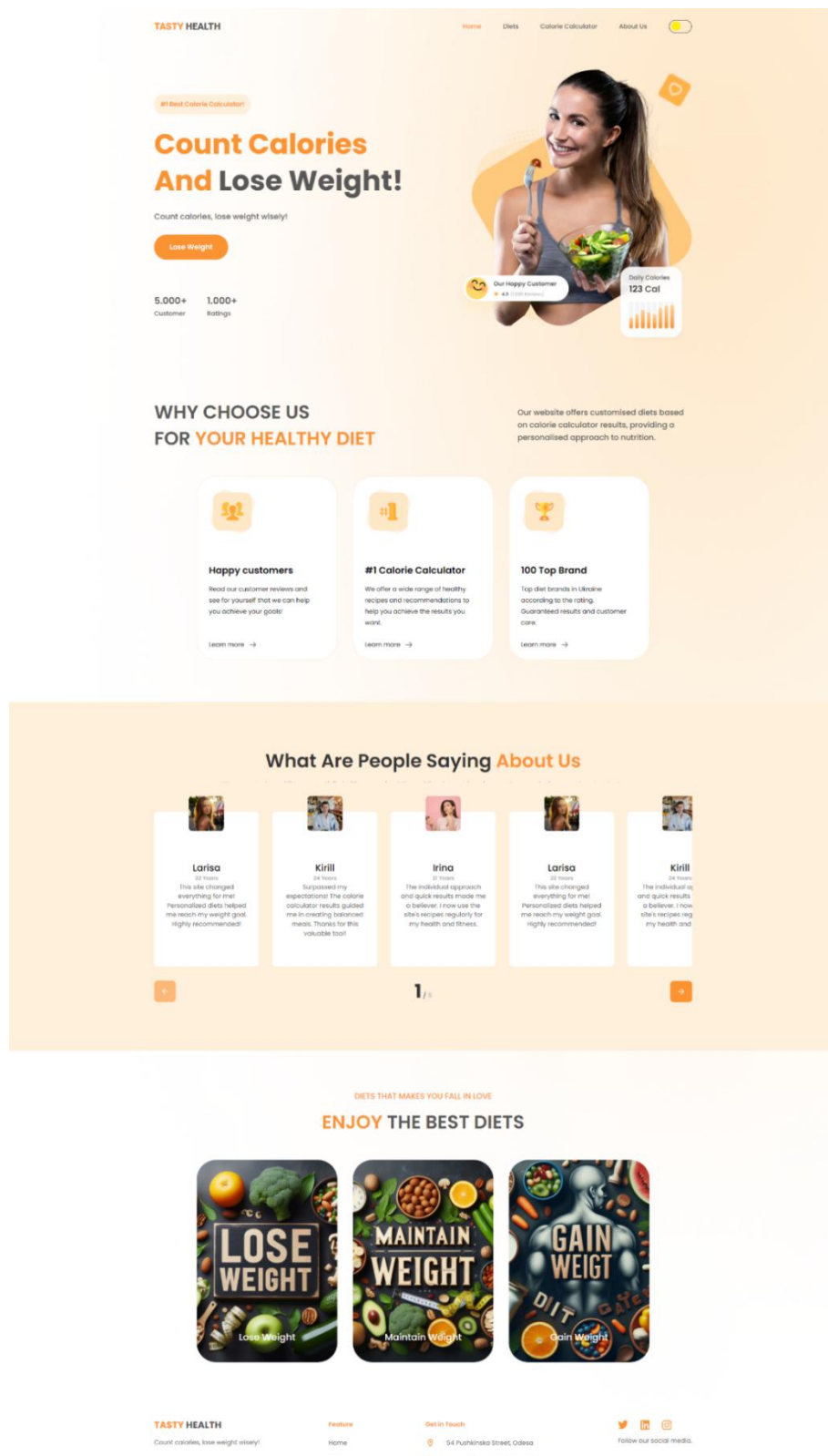


Рисунок 1.18. Головна сторінка “Home”

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 07 001. 00 ДП ПЗ

Арк.

43

1.4.2 Функціональний огляд сторінки “Diets”

Розглянемо наступну сторінку веб-застосунку – “Diets”.

Веб-сторінка “Diets” надає користувачам наступний функціонал:

Користувачі можуть ознайомитися з широким вибором дієт, що підходять для різних цілей та потреб, будь то схуднення, набір м'язової маси або підтримка загального здоров'я. Кожна дієта детально описана, включаючи рекомендовані продукти, добові норми споживання та поради щодо харчування.

На рис. 1.19. зображено сторінку “Diets”.

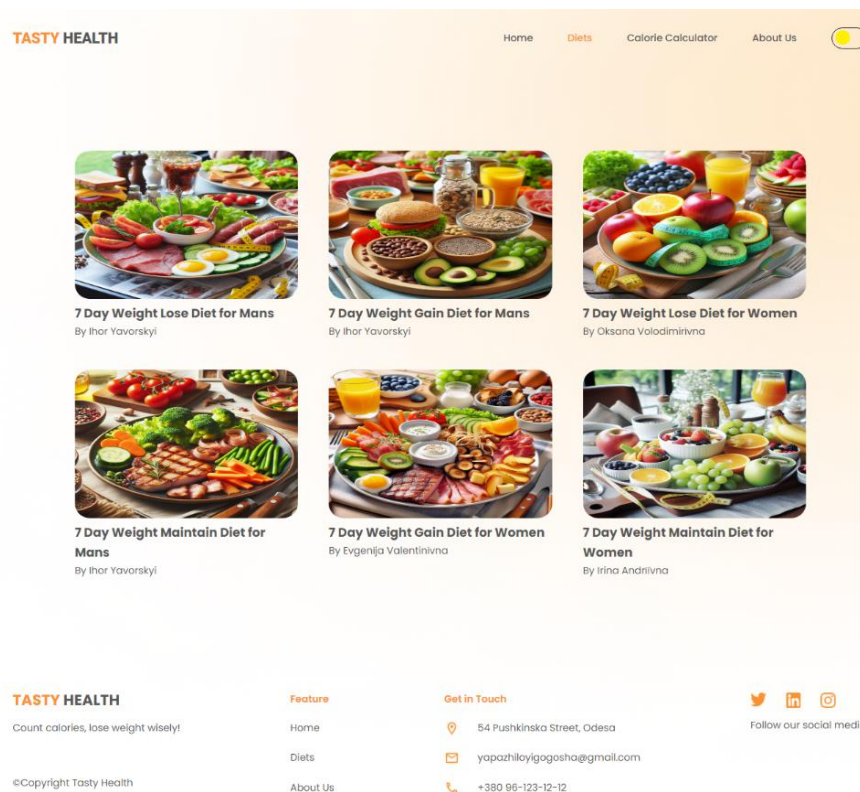


Рисунок 1.19. Сторінка “Diets”

Розглянемо веб-сторінку дієти.

Веб-сторінка з дієтою може дати користувачам кілька важливих функцій:

1. Користувачі можуть переглядати апетитні зображення страв, що допомагає їм візуально оцінити запропоновані варіанти харчування.
2. Сторінка містить розклад прийомів їжі, де зазначено час кожного прийому разом із калорійністю страв. Це дозволяє користувачам легко планувати

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 07 001. 00 ДП ПЗ

Арк.

44

свій день і забезпечувати збалансоване харчування, враховуючи калорійні потреби.

На рис. 1.20. зображено сторінку з дієтою.

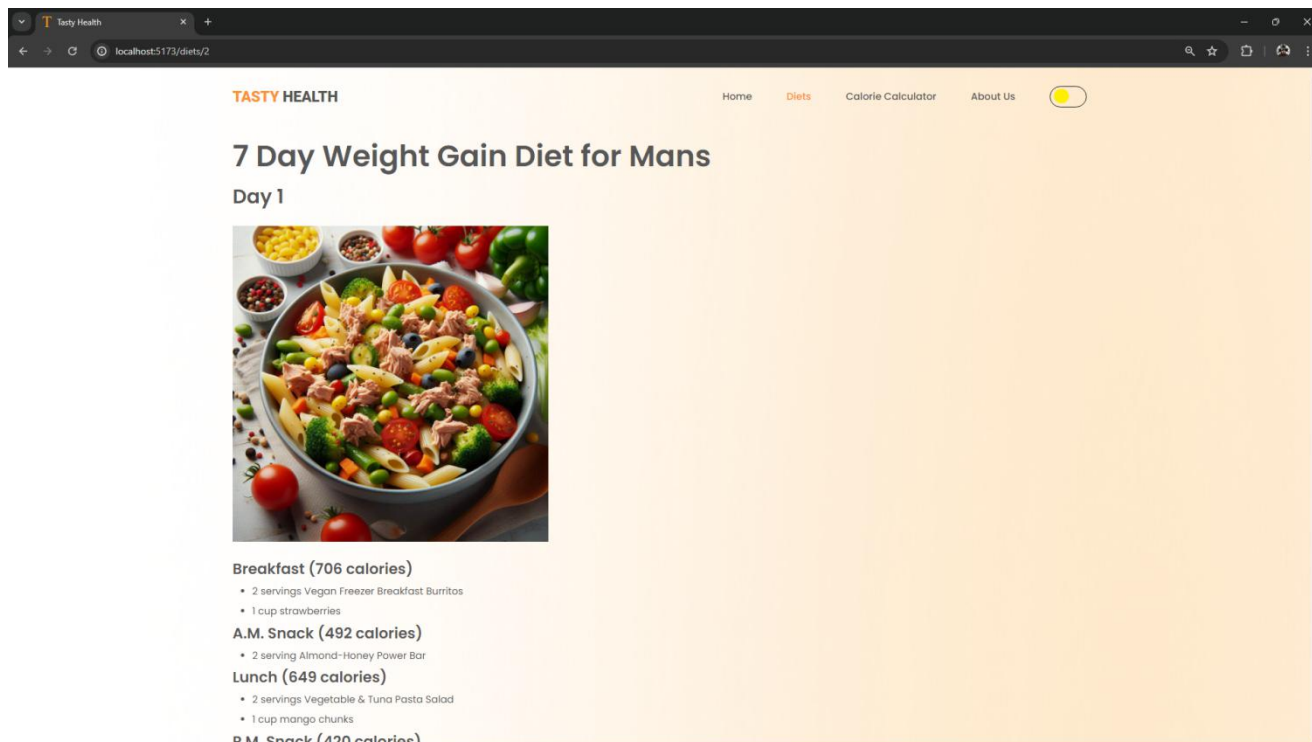


Рисунок 1.20. Сторінка з дієтою

1.4.3 Функціональний огляд сторінки “Calorie Calculator”

Розглянемо веб-сторінку – “Calorie Calculator”.

Веб-сторінка з калькулятором калорій надає користувачам корисний функціонал:

Користувачі можуть вводити свої індивідуальні параметри, такі як вік, вага, зріст і рівень фізичної активності, для отримання персоналізованих рекомендацій щодо щоденної кількості калорій, необхідних для досягнення їхніх цілей у харчуванні та здоров'ї.

На рис. 1.21. зображено сторінку “Diets”.

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
						45
Ізм.	Лист	№ докум.	Підпис	Дата		



Weight (kg):

Height (cm):

Age:

Gender:

Male Female

Activity Level:

Sedentary Lightly active Moderately active Very active Super active

Goal:

Lose Weight Maintain Weight Gain Weight

Calorie Calculator

Our calorie calculator helps you determine the daily calorie intake required to achieve your goals: lose weight, maintain weight, or gain weight. Simply input your parameters (weight, height, age, gender), choose your activity level and desired goal. The calculator will automatically calculate the optimal calorie intake needed to maintain your health and reach your desired results.

How to use:

- Enter your weight, height, and age.
- Select your gender.
- Indicate your activity level.
- Choose your goal.
- Click the "Calculate" button to find out your daily calorie intake.

TASTY HEALTH

Count calories, lose weight wisely!

©Copyright Tasty Health

Feature

Home

Diets

About Us

Get in Touch

54 Pushkinska Street, Odessa

yapazhiloyigogasha@gmail.com

+380 96-123-12-12



Follow our social media.

Рисунок 1.21. Сторінка “Calorie Calculator”

1.4.4 Функціональний огляд сторінки “About Us”

Розглянемо веб-сторінку – “About Us”.

Веб-сторінка "About Us" надає користувачам одну важливу функцію:

Користувачі можуть дізнатися про історію компанії, що допомагає зрозуміти її розвиток, цінності та основні етапи становлення. Це створює глибше розуміння місії та бачення компанії.

На рис. 1.22. зображено сторінку “About Us”.

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 07 001. 00 ДП ПЗ

Арк.

46



EST. 2024

Our Story

In those early days when we were just starting out, our team consisted of a few enthusiasts passionate about a healthy lifestyle. We often gathered together, discussing the benefits of proper nutrition and sharing healthy food recipes. Our dream was to create an online resource that would be a reliable guide in the world of healthy eating.

Over time, our project grew, and so did the number of people who shared our vision. We expanded our team, bringing in experienced dietitians and fitness experts. We developed unique tools, such as a calorie calculator, to help people control their diet and achieve their desired results.

Today, we are pleased to welcome you to our website, where you will find not only information about our team but also a wealth of useful materials to help you live well and make the most of your health. Welcome to our community, where caring for your health is our main goal!

TASTY HEALTH

Count calories, lose weight wisely!

©Copyright Tasty Health

Feature

Home

Diets

About Us

Get in Touch

54 Pushkinska Street, Odessa

yapazhilayigogosha@gmail.com

+380 96-123-12-12



Follow our social media.

Рисунок 1.22. Сторінка “About Us”

1.4.5 Функціональний огляд темного режиму у веб-застосунку

В хедері, перемикач темного режиму має важливий функціонал для користувачів. Він створює можливість миттєво змінювати вигляд сторінки залежно від особистих уподобань та потреб. При цьому, дизайн переключника відповідає загальному стилю веб-сайту, що додає єдність та гармонію в інтерфейсі.

Розглянемо головну сторінку в темному режимі, яка знаходиться на рис. 1.23.

					КГ 07. 07 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		47



Рисунок 1.23. Головна сторінка в темному режимі

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 07 001. 00 ДП ПЗ

Арк.

48

1.4.6 Функціональний огляд мобільної версії веб-застосунку

Мобільна версія веб-системи призначена для зручного користування на смартфонах і планшетах. Її інтерфейс оптимізований для малих екранів, що дозволяє користувачам легко та зручно взаємодіяти з контентом навіть на пристроях з обмеженим простором відображення.

Розглянемо мобільну версію головної сторінки, яка знаходиться на рис. 1.24.

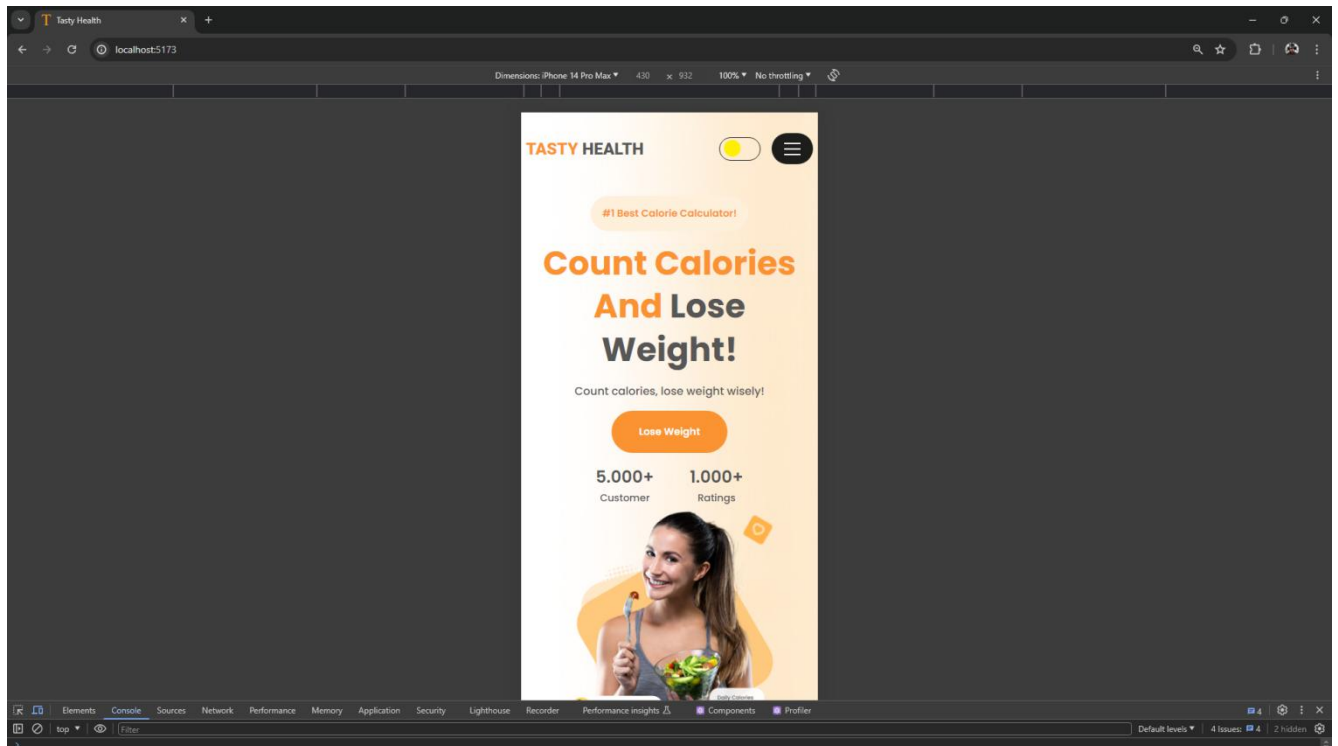


Рисунок 1.24. Мобільна версія головної сторінки

					<i>КГ 07. 07 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		49

2 ЕКОНОМІЧНИЙ РОЗДІЛ

У дипломному проєкті розроблена веб-система, яка надає користувачам доступ до потужної експертної системи для індивідуального підбору дієт харчування.

Цей застосунок створений з урахуванням найновіших технологічних рішень і має зручний інтерфейс користувача.

Основна функція експертної системи полягає в тому, щоб допомагати користувачам знайти оптимальні дієтичні плани, враховуючи їхні індивідуальні потреби.

При оцінці ефективності створюваного сайту слід виходити з того, що залежно від характеру ефекту, що досягається, можуть бути визначені наступні види ефективності сайту: економічна, функціональна та соціальна ефективність.

У дипломному проєкті веб-система, що розроблялася, не є комерційною та не приносить доходу, тому оцінюються лише витрати на її розробку, а також функціональну та соціальну ефективність веб-системи.

Загальні витрати (B_3) на створення сайту складаються з декількох параметрів:

$$B_3 = B_p + B_v + B_e ,$$

де B_p – витрати на розробку сайту;

B_v – витрати на впровадження сайту;

B_e – витрати на експлуатацію сайту;

Витрати на розробку сайту (B_p) є одноразовими та складаються з вартості наступних видів робіт зі створення сайту:

1. Дизайн сайту: розробка макетів для головної та внутрішніх сторінок сайту; створення фірмового стилю і логотипу.
2. Розробка сторінок-шаблонів
3. Наповнення сайту інформацією: наповнення та форматування web-сторінок; обробка малюнків для публікації на web-сторінках, верстка (переклад в HTML-формат) web-сторінок

4. Програмна розробка сайту: створення програмного коду сайту, програмування динамічних елементів (анімаційних елементів, флеш-заставок)

Для визначення витрат на розробку сайту (V_p) розраховуємо оплату праці виконавців, безпосередньо притягнених до її виконання. Над реалізацією веб-застосунку працював дипломник.

Для визначення трудомісткості розробки сайту (V_p) складено план-графік по розробці веб-системи і тривалості виконання робіт. Розподіл робіт по етапах і видах виконавців наведено в таблиці 2.1.

Таблиця 2.1. План-графік по розробці Web-сайту

№	Назва етапу	Час виконання (годин)	Посада виконавця
1	Визначення функціональних вимог, архітектури сайту та його компонентів.	26 годин	Дипломник
2	Розробка дизайну користувацького інтерфейсу (UI) та його прототипування для визначення вигляду та взаємодії.	24 години	Дипломник
3	Створення реактивних компонентів на React для кожного елемента інтерфейсу.	42 години	Дипломник
4	Вдосконалення швидкодії, оптимізація завантаження та покращення взаємодії з користувачем.	32 години	Дипломник
5	Створення анімацій для реактивних компонентів	20 годин	Дипломник
ВСЬОГО:		144 години	

Для оцінки складності розробки веб-сайту (V_p) був складений план-графік робіт з розробки веб-системи та визначення тривалості виконання завдань. Розподіл завдань за етапами та видами виконавців наведено у таблиці 2.2.

Таблиця 2.2. Витрати на заробітну плату

№	Персонал	Етапи розробки	Кількість робочих годин (або днів)	Погодинна ставка (або денна ставка), грн.	Заробітн а плата, грн.
1	Дипломник	Визначення основних функцій. Аналіз потреб користувачів.	26 годин	130грн./год.	3380 грн.
2	Дипломник	Проектування UI/UX. Створення стилістичного керівництва. Визначення користувацьких сценаріїв.	24 години	130грн./год.	3120 грн.
3	Дипломник	Розбиття інтерфейсу на компоненти. Визначення структури компонентів. Розробка компонентів. Комунікація між компонентам. Тестування компонентів	42 години	130грн./год.	5460 грн.

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 07. 17 002. 00 ДП ПЗ

Арк.

52

4	Дипломник	Аналіз продуктивності. Розбиття коду. Відкладене завантаження. Зменшення розміру бандла. Зменшення затримок. Відгук на дії користувача	32 години	130грн./год	4160грн.
5	Дипломник	Аналіз взаємодії. Вибір бібліотеки для анімування.. Розробка анімацій. Перевірка продуктивності. Оптимізація анімацій.	20 годин	130грн./год	2600 грн.
ВСЬОГО:					$V_{зп} = 18$ 720 грн.

До складу витрат на оплату праці також включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Розмір єдиного соціального внеску складає 22% від заробітної плати, розраховується за наступною формулою:

$$V_{есв} = V_{зп} \times 0,22 = 18\,720 \times 0,22 = 4118 \text{ грн.}$$

Загальні витрати (V_p) на розробку веб-сайту розраховуються як сума витрат на заробітну плату праці персоналу ($V_{зп}$) та єдиного соціального внеску ($V_{есв}$):

$$V_p = V_{зп} + V_{есв} = 18\,720 + 4118 = 22\,838 \text{ грн.}$$

Витрати на впровадження сайту (V_v) складаються з двох складових :

1. Витрати на реєстрацію доменного імені .com.ua становить 505 грн (V_{v1}).

					КГ 07. 17 002. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		53

2. Реєстрація в пошукових системах, таких як Google через інструмент Google Search Console, або Bing через Bing Webmaster Tools, не потребує жодних фінансових витрат.

$$B_B = B_{B1} + B_{B2} = 505 \text{ грн.}$$

Витрати на експлуатацію сайту (B_e) включають в себе кілька складових. По-перше, це підтримка сайту в робочому стані та продовження доменного імені на один рік. Також сюди входить вартість хостингу на рік.

Окрім технічного обслуговування, важливо враховувати витрати на регулярне наповнення сайту контентом. Це включає статті, новини та інші оновлення, а також розробку і впровадження індивідуальних дієт для користувачів. Для цього потрібні послуги контент-менеджерів, дієтологів та розробників, які забезпечують актуальність і якість інформації на сайті.

У таблиці 2.3 представлені постійні витрати, що складаються з витрат на впровадження та експлуатацію сайту протягом року. Дані взяті з відповідних прайс-листів та технічних специфікацій провайдерів послуг.

№	Стаття витрат	Вартість за рік, грн.
1	Хостинг на 1 рік	3 000 грн.
2	Продовження доменного імені	505 грн.
3	Наповнення контентом	8 000 грн.
4	Розробка індивідуальний дієт	5 000 грн.
Всього:		$B_e = 16 505$ грн.

Загальні витрати (B_3) на розробку, впровадження та експлуатацію веб-сайту розраховуються за наступною формулою:

$$B_3 = B_p + (B_B + B_e) = 18 720 + (505 + 16 505) = 35 730 \text{ грн.}$$

Функціональна ефективність веб-орієнтованої експертної системи для індивідуального підбору дієт харчування полягає в забезпеченні повноти, точності і доступності інформації про харчування та дієти у будь-який слушний для користувача час доби. Система надає користувачам можливість отримувати

персоналізовані плани харчування, відстежувати прогрес та адаптувати свої дієтичні потреби на основі науково-обґрунтованих даних.

Важливим аспектом є оптимізація технологічних процесів, зокрема швидкості роботи системи. Система використовує сучасні технології оптимізації коду, що забезпечує високу швидкість реакції на запити користувачів. Це дозволяє значно скоротити час на доступ до необхідної інформації та на виконання різноманітних операцій. Оптимізація швидкості роботи системи забезпечує безперебійну роботу навіть при великій кількості одночасних користувачів, що підвищує загальну продуктивність та зручність користування.

Зручність користування системою також підвищується завдяки наявності темного режиму, що дозволяє користувачам вибирати темний фон і білі шрифти. Це зменшує навантаження на очі під час тривалого користування сайтом, особливо у вечірній та нічний час. Додатково, система має мобільну версію, що забезпечує доступ до всіх функцій та інформації з будь-якого мобільного пристрою. Це дозволяє користувачам легко коригувати свої дієти, отримувати зворотний зв'язок та поради в будь-який зручний для них час та місце, що сприяє досягненню їхніх цілей у здоровому харчуванні.

Соціальна ефективність веб-орієнтованої експертної системи для індивідуального підбору дієт харчування залежить від типу об'єкту сайтобудування і може виражатися в наступному.

Розширення складу засобів просування товарів і послуг, що надаються. Система сприяє просуванню здорового способу життя та раціонального харчування серед широкої аудиторії. Це дозволяє залучати нових користувачів, які зацікавлені у покращенні свого здоров'я через індивідуально підібрані дієти. Завдяки інтерактивним функціям та зручному інтерфейсу, користувачі можуть легко ознайомитися з послугами системи та отримати доступ до корисної інформації про харчування. Важливо, що послуги системи надаються безкоштовно, що робить їх доступними для широкого кола користувачів.

Інформування найбільшого числа зацікавлених осіб про діяльність організації та умови користування її послугами. Веб-орієнтована система

					<i>КГ 07. 17 002. 00 ДП ПЗ</i>	Арк.
						55
Ізм.	Лист	№ докум.	Підпис	Дата		

дозволяє ефективно інформувати потенційних користувачів про можливості, які вона надає. Детальна інформація про функціональність, методи роботи та результати, які можна досягти завдяки системі, сприяє залученню нових користувачів та зміцненню довіри до організації.

Формування позитивного іміджу організації. Використання сучасних технологій та інноваційних підходів у розробці веб-орієнтованої експертної системи підвищує репутацію організації як надійного та професійного партнера. Позитивний імідж сприяє зміцненню довіри користувачів та покращенню взаємовідносин з клієнтами та партнерами.

					<i>КГ 07. 17 002. 00 ДП ПЗ</i>	Арк.
						56
Ізм.	Лист	№ докум.	Підпис	Дата		

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Вступ

У цьому розділі розглядаються питання охорони праці, виробничої санітарії, пожежної безпеки та охорони навколишнього середовища, що виникають при розробці веб-орієнтованої експертної системи для індивідуального підбору дієт харчування. Метою є забезпечення безпеки працівників, зменшення шкідливого впливу виробничих факторів та забезпечення безпечних умов праці.

3.2 Аналіз умов праці й забезпечення безпеки при виконанні основних видів робіт на об'єкті дипломного проєктування

Враховуючи тему дипломного проєкту "Розробка веб-орієнтованої експертної системи для індивідуального підбору дієт харчування", робочим місцем для аналізу умов праці є офісний простір, де розташоване робоче місце програміста. Це середовище, в якому здійснюється повний цикл розробки та тестування веб-сайту, включаючи як технічні, так і творчі аспекти.

3.2.1 Небезпечні та шкідливі виробничі фактори

Основні небезпечні та шкідливі виробничі фактори які виникають при роботі програміста включають:

1. Напруження очей: постійна робота за комп'ютером призводить до тривалого фокусування очей на моніторі, що може викликати зорову втоми, сухість очей, а іноді й погіршення зору. Психоемоційне напруження.
2. Статичні навантаження на м'язи: робота за комп'ютером передбачає тривале сидіння в одній позі, що створює статичне навантаження на м'язи спини, шиї та плечей, що може призводити до болю та дискомфорту.
3. Психоемоційне напруження: інтенсивна розумова діяльність, часті дедлайни та необхідність вирішення технічних проблем можуть викликати високе психоемоційне напруження у працівників. Це може призводити до

					КГ 07. 17 003. 00 ДП ПЗ	Арк.
						57
Ізм.	Лист	№ докум.	Підпис	Дата		

стресу, вигоряння, зниження продуктивності та погіршення загального самопочуття.

4. Недостатнє освітлення: відсутність достатнього освітлення може призводити до напруження очей, головного болю, швидкої втомлюваності та зниження продуктивності.
5. Неправильне розташування робочих місць: Неправильне розташування робочих місць може викликати дискомфорт, погану поставу та проблеми з опорно-руховим апаратом.
6. Електромагнітне випромінювання від комп'ютерної техніки: постійна робота за комп'ютером призводить до впливу електромагнітних полів, які можуть мати негативний вплив на здоров'я, викликаючи головний біль, втому та інші симптоми.

3.2.2 Заходи щодо забезпечення безпеки праці

Для забезпечення безпеки праці пропонується вжити наступні заходи:

1. Встановлення природного з поєднанням штучного освітлення для забезпечення комфортних робочих умов.
2. Уникнення блисків та тіней на екранах комп'ютерів для покращення зорового комфорту працівників.
3. Забезпечення робочих місць ергономічними меблями, що дозволяють правильно регулювати положення тіла.
4. Організація регулярних перерв для відпочинку та фізичних вправ для підтримки здоров'я м'язів і спини.
5. Проведення гімнастики для очей для запобігання втомі та напруженню зору.
6. Надання інструкцій з правильного використання комп'ютерної техніки та уникнення травматичних ситуацій.
7. Забезпечення працівників засобами індивідуального захисту, такими як спеціальні окуляри для комп'ютерної роботи або паузи на виконання спеціальних вправ для зменшення навантажень.

					КГ 07. 17 003. 00 ДП ПЗ	Арк.
						58
Ізм.	Лист	№ докум.	Підпис	Дата		

3.3 Гігієнічні вимоги до виробничого середовища:

3.3.1 Приміщення

Приміщення для розробки веб-сайту повинно відповідати таким вимогам:

1. Площа приміщення повинна бути достатньою для розміщення всіх працівників та обладнання. Відповідно до санітарних вимог на одне робоче місце повинно бути: площа – 6,0 м² , об'єм – 20 м³ . Це забезпечує зручні умови для роботи, запобігає перенаселеності та сприяє нормальному функціонуванню команди.
2. Приміщення повинно бути добре вентильованим для забезпечення постійного подачі свіжого повітря, що є важливим для здоров'я та концентрації працівників.
3. Наявність кондиціонування в приміщенні дозволяє підтримувати оптимальні температурні умови, що сприяє комфортній роботі в умовах підвищених температур або вологості.

3.3.2 Мікроклімат

Забезпечення комфортного мікроклімату включає:

1. Температура повітря повинна утримуватися у межах 18-22°C. Такий режим сприяє комфортній роботі працівників, запобігає перегріванню або охолодженню, що може негативно впливати на їх здоров'я та продуктивність.
2. Відносна вологість повітря має бути у діапазоні 40-60%. Таке значення вологості сприяє збереженню оптимального рівня вологи для здоров'я дихальних шляхів та шкіри, запобігає пересушуванню чи занадто високій вологості в приміщенні.

3.3.3 Освітлення

Освітлення робочих місць повинно відповідати нормам:

					КГ 07. 17 003. 00 ДП ПЗ	Арк.
						59
Ізм.	Лист	№ докум.	Підпис	Дата		

1. Використання комбінованого освітлення, яке включає природне світло та штучне освітлення. Такий підхід дозволяє забезпечити оптимальні умови освітлення навіть у різних часах доби та погодних умовах.
2. Інтенсивність освітлення на робочому місці повинна бути не менше 500 лк. Це значення забезпечує достатню яскравість освітлення для комфортної роботи, запобігає напруженню зору та сприяє збереженню здоров'я очей працівників.

3.3.4 Шум

Рівень шуму на робочому місці не повинен перевищувати 50 дБ (децибел). Це значення дозволяє забезпечити тишу та комфортне оточення для працівників, не допускаючи надмірного шуму, який може впливати на їх концентрацію та здоров'я.

3.3.5 Організація робочого місця

Організація робочого місця включає:

1. Використання ергономічних стільців та столів сприяє забезпеченню правильної позиції тіла працівників під час роботи. Це дозволяє уникнути перенапруження м'язів і хребта, що може виникати при тривалому сидінні за комп'ютером.
2. Розміщення монітора на рівні очей працівника допомагає зменшити навантаження на шийний відділ хребта. Це важливо для підтримки правильної позиції тіла та запобігання виникненню проблем зі здоров'ям шийного відділу хребта.

3.3.6 Електробезпека

Забезпечення електробезпеки включає:

1. Необхідно використовувати справне електрообладнання, яке відповідає вимогам безпеки та проходить регулярні перевірки на наявність пошкоджень чи несправностей. Це дозволяє уникнути аварійних ситуацій та травм працівників через електробезпеку.

					КГ 07. 17 003. 00 ДП ПЗ	Арк.
						60
Ізм.	Лист	№ докум.	Підпис	Дата		

2. Необхідно проводити регулярну перевірку електромережі та заземлення, щоб забезпечити їхню надійність та відповідність вимогам безпеки. Це важливо для запобігання виникненню коротких замикань, перенапруг та інших небезпечних ситуацій.

3.4 Пожежна безпека

Забезпечення пожежної безпеки на робочому місці - це надзвичайно важлива задача, яка вимагає комплексного підходу та вжиття конкретних заходів. Першим кроком є розміщення портативних вогнегасників відповідно до стандартів безпеки. Важливо, щоб вогнегасники були легкодоступними та відповідали потребам конкретного приміщення, ураховуючи його розміри та можливі ризики пожежі. Для цього слід використовувати наступні типи вогнегасників:

1. Вуглекислотні вогнегасники застосовують для гасіння пожеж класу В та С, тобто газових горючих матеріалів, а також електричних приладів.
2. Порошкові вогнегасники є універсальними, для гасіння пожеж класу А, В, С, а також електричних приладів.
3. Хладонові вогнегасники використовуються для гасіння електричних приладів і комп'ютерного обладнання.

Крім того, персонал повинен бути навчений правильно використовувати вогнегасники, включаючи практичні тренування для швидкого та ефективного реагування у разі пожежі.

Другим необхідним заходом є встановлення автоматичних систем пожежного оповіщення та сигналізації. Це може включати димові детектори, системи розпізнавання вогню та температурні датчики, які автоматично спрацьовують у разі виявлення загрози пожежі. Такі системи дозволяють оперативно реагувати на небезпеку та приймати необхідні заходи безпеки, зменшуючи ризик для працівників та відвідувачів приміщення.

					КГ 07. 17 003. 00 ДП ПЗ	Арк.
						61
Ізм.	Лист	№ докум.	Підпис	Дата		

3.5 Оптимальні рішення щодо забезпечення безпечного виконання робіт:

На основі аналізу умов праці та виробничих факторів, було розроблено наступні оптимальні рішення для забезпечення безпечного виконання робіт:

1. Для зниження напруження очей та статичних навантажень, рекомендується організувати регулярні перерви кожні 1-2 години роботи за комп'ютером.
2. Для фільтрації синього світла на моніторах і зменшення впливу на зір використовується програмне забезпечення.
3. Важливо забезпечити належний рівень освітлення та вентиляції у приміщенні для комфортної та безпечної роботи.

Для зменшення рівня шуму та покращення умов праці, рекомендується встановити шумопоглинаючі матеріали.

					<i>КГ 07. 17 003. 00 ДП ПЗ</i>	Арк.
						62
Ізм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломному проєкті створено програмний продукт – веб-орієнтовану експертну систему для індивідуального підбору дієт харчування. Цей продукт покликаний допомогти користувачам визначити оптимальні дієтичні рекомендації на основі їх індивідуальних потреб та характеристик, забезпечуючи персоналізований підхід до харчування. Система враховує різні параметри, такі як вік, вага, рівень фізичної активності, що дозволяє пропонувати найбільш ефективні та безпечні дієти для кожного окремого випадку.

Створений програмний продукт включає в себе:

1. Розділ відгуків, де користувачі можуть ділитися своїм досвідом та результатами.
2. Розділ дієт, що містить детальні описи та рекомендації.
3. Темну тему для комфортного використання у вечірній час.
4. Калькулятор калорій для точного розрахунку добової норми калорій.

При створенні програмного продукту було задіяно такий інструментарій: React, Figma, JavaScript, HTML та CSS. Використання React дозволило забезпечити високу швидкодію та інтерактивність додатку, роблячи його більш чутливим до дій користувача. Figma була використана для розробки макетів та прототипів, що забезпечило зручність і естетичну привабливість дизайну. Завдяки JavaScript вдалося реалізувати складну логіку додатку, а HTML та CSS допомогли створити структурований і стильний інтерфейс. Це все разом забезпечило високу якість дизайну та зручність використання програмного продукту.

Дана пояснювальна записка охоплює всі аспекти, визначені в технічному завданні до дипломного проєкту. У ній проведено глибокий аналіз предметної області, детально описано технології та інструменти, задіяні в розробці проєкту. Також здійснено розрахунок економічної ефективності впровадження проєкту, проаналізовані питання охорони праці та надано список використаних джерел.

					<i>КГ 07. 17 000. 00 ДП ПЗ</i>	Арк.
						63
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Calorie Calculator. [Веб-сайт]. URL: <https://www.calculator.net/calorie-calculator.html>.
2. Calorie calculator – Calculate your calorie needs. [Веб-сайт]. URL: <https://ch.esn.com/en/pages/kalorienrechner>.
3. Daily Calorie Intake Calculator. [Веб-сайт]. URL: <https://www.calories.info/calorie-intake-calculator>.
4. HTML: HyperText Markup Language - MDN Web Docs. [Веб-сайт]. URL: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML.
5. CSS: Cascading Style Sheets - MDN Web Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
6. JavaScript - MDN Web Docs. [Веб-сайт]. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript>.
7. React. [Веб-сайт]. URL: <https://react.dev/learn>.
8. LambdaTest. [Веб-сайт]. URL: <https://www.lambdatest.com/blog/best-javascript-frameworks/>.
9. What Is a Single Page Application? [Веб-сайт]. URL: <https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application>.
10. Figma [Веб-сайт]. URL: <https://uk.wikipedia.org/wiki/Figma>.
11. Що таке Routing? [Веб-сайт]. URL: <https://tseivo.com/b/memecode/t/xzylpoelz>.
12. React Router [Веб-сайт]. URL: <https://reactrouter.com/en/main/start/tutorial>.
13. Using React as a templating engine [Веб-сайт]. URL: <https://www.cleverage.com/en/using-react-as-a-templating-engine/>.
14. View React Component [Веб-сайт]. URL: <https://framework7.io/react/view>
15. Lazy [Веб-сайт]. URL: <https://react.dev/reference/react/lazy>
16. К. В. Двірничук, Д. О. Вацек. Веб-програмування та веб-дизайн: Навчальний посібник. – «Чернівецький національний університет імені Юрія Федьковича», 2022.

					КГ 07. 17 000. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		64

17.В. В. Босько, Л. В. Константинова, К. М. Марченко, О. С. Улічев. Web-програмування. Частина 1 (Frontend): Навчальний посібник. – «Кропивницький центральноукраїнський національний технічний університет», 2022.

					<i>КГ 07. 17 000. 00 ДП ПЗ</i>	Арк.
						65
Ізм.	Лист	№ докум.	Підпис	Дата		

ДОДАТОК А. Програмний код основної логіки веб-застосунку

```
// main.jsx

import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";
import "./index.scss";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
);

// App.jsx

import { createBrowserRouter, RouterProvider } from "react-router-dom";

import MainPage, {
  DietsPage,
  DietsSinglePage,
  KcalCalcPage,
  AboutUsPage,
  NotFoundPage,
} from "./pages";

import "./App.scss";

const router = createBrowserRouter([
  {
    path: "/",
    element: <MainPage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/kcal-calc",
    element: <KcalCalcPage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/diets",
    element: <DietsPage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/diets/:id",
    element: <DietsSinglePage />,
    errorElement: <NotFoundPage />,
  },
  {
    path: "/about-us",
    element: <AboutUsPage />,
    errorElement: <NotFoundPage />,
  },
]);
```

```

const App = () => {
  return <RouterProvider router={router} />;
};

export default App;

// Header.jsx

import { NavLink, Link } from "react-router-dom";
import { motion } from "framer-motion";

import DarkMode from "../darkMode/DarkMode";
import BurgerMenu from "../burgerMenu/BurgerMenu";

import "../Header.scss";

const Header = () => {
  return (
    <motion.header
      className="header"
      initial={{ opacity: 0 }}
      animate={{ opacity: 1 }}
      transition={{ duration: 0.3 }}
    >
      <div className="container">
        <div className="header__menu menu">
          <Link to="/" className="menu__logo">
            <span>Tasty</span> Health
          </Link>
          <nav className="menu__body">
            <ul className="menu__list">
              <li className="menu__item">
                <NavLink to="/" className="menu__link">
                  Home
                </NavLink>
              </li>
              <li className="menu__item">
                <NavLink to="/diets" className="menu__link">
                  Diets
                </NavLink>
              </li>
              <li className="menu__item">
                <NavLink to="/kcal-calc" className="menu__link">
                  Calorie Calculator
                </NavLink>
              </li>
              <li className="menu__item">
                <NavLink to="/about-us" className="menu__link">
                  About Us
                </NavLink>
              </li>
            </ul>
          </nav>
          <div className="menu__buttons">
            <DarkMode />
            <BurgerMenu />
          </div>
        </div>
      </div>
    </motion.header>
  );
};

```

```

export default Header;

// Footer.jsx

import { Link } from "react-router-dom";
import { motion } from "framer-motion";

import location from "/images/footer/location.png";
import mail from "/images/footer/mail.png";
import call from "/images/footer/call.png";

import twitter from "/images/footer/twitter.png";
import linkedin from "/images/footer/linkedin.png";
import instagram from "/images/footer/instagram.png";

import "./Footer.scss";

const Footer = () => {
  return (
    <motion.footer
      className="footer"
      initial={{ y: 150 }}
      whileInView={{ y: 0 }}
      transition={{ duration: 0.3 }}
      viewport={{ once: true }}
    >
      <div className="container">
        <div className="footer__content">
          <div className="footer__column">
            <div className="footer__logo">
              <p>
                <span>TASTY</span> HEALTH
              </p>
            </div>
            <p className="footer__text">
              Count calories, lose weight wisely!
            </p>
            <p className="footer__copyright">©Copyright Tasty Health</p>
          </div>
          <div className="footer__column">
            <h4 className="footer__title">Feature</h4>
            <ul className="footer__items">
              <li className="footer__item">
                <Link to="/" className="footer__link">
                  Home
                </Link>
              </li>
              <li className="footer__item">
                <Link to="/diets" className="footer__link">
                  Diets
                </Link>
              </li>
              <li className="footer__item">
                <Link to="/about-us" className="footer__link">
                  About Us
                </Link>
              </li>
            </ul>
          </div>
          <div className="footer__column">
            <h4 className="footer__title">Get in Touch</h4>

```

```

        <ul className="footer__items">
          <li className="footer__item">
            <img className="footer__icon" src={location} alt="" />
            <Link to="/" className="footer__link">
              54 Pushkinska Street, Odesa
            </Link>
          </li>
          <li className="footer__item">
            <img className="footer__icon" src={mail} alt="" />
            <Link
              to="mailto:yapazhiloyigogosha@gmail.com"
              className="footer__link"
            >
              yapazhiloyigogosha@gmail.com
            </Link>
          </li>
          <li className="footer__item">
            <img className="footer__icon" src={call} alt="" />
            <Link to="tel: +380961231212" className="footer__link">
              +380 96-123-12-12
            </Link>
          </li>
        </ul>
      </div>
      <div className="footer__column">
        <ul className="footer__socialLinks">
          <li className="footer__socialLink">
            <Link to="/" className="footer__image">
              <img src={twitter} alt="" />
            </Link>
          </li>
          <li className="footer__socialLink">
            <Link to="/" className="footer__image">
              <img src={linkedin} alt="" />
            </Link>
          </li>
          <li className="footer__socialLink">
            <Link to="/" className="footer__image">
              <img src={instagram} alt="" />
            </Link>
          </li>
        </ul>
        <p className="footer__follow">Follow our social media.</p>
      </div>
    </div>
  </div>
</motion.footer>
);
};

export default Footer;

// KcalCalc.jsx

import { ErrorBoundary } from "react-error-boundary";

import Header from "../header/Header";
import KcalHero from "../kcalHero/KcalHero";
import Footer from "../footer/Footer";

const KcalCalc = () => {
  return (

```

```

        <div className="wrapper">
          <Header />
          <ErrorBoundary
            fallback={
              
            }
          >
            <KcalHero />
          </ErrorBoundary>
          <Footer />
        </div>
      );
    };

```

```
export default KcalCalc;
```

```
// KcalHero.jsx
```

```
import KcalForm from "../kcalForm/KcalForm";
import KcalInfo from "../kcalInfo/KcalInfo";
```

```
import "./KcalHero.scss";
```

```
const KcalHero = () => {
  return (
    <main className="main">
      <section className="kcalCalc">
        <div className="container">
          <div className="kcalCalc__content">
            <KcalForm />
            <KcalInfo />
          </div>
        </div>
      </section>
    </main>
  );
};
```

```
export default KcalHero;
```

```
// KcalForm.jsx
```

```

import { useState } from "react";
import { Link } from "react-router-dom";
import { useFormik } from "formik";
import { motion } from "framer-motion";
import * as Yup from "yup";

import arrow from "/images/main/whyChooseUs/arrow.png";

import "./KcalForm.scss";

const validationSchema = Yup.object({
  weight: Yup.number()
    .required("Required")
    .min(10, "Too low")
    .max(500, "Too high")
});

```

```

        .positive("Must be a positive number")
        .typeError("Must be a number"),
    height: Yup.number()
        .min(50, "Too low")
        .max(300, "Too high")
        .required("Required")
        .typeError("Must be a number")
        .positive("Must be a positive number"),
    age: Yup.number()
        .min(4, "Too low")
        .max(100, "Too high")
        .required("Required")
        .typeError("Must be a number")
        .positive("Must be a positive number")
        .integer("Must be an integer"),
  });

const KcalForm = () => {
  const [gender, setGender] = useState("male");
  const [activity, setActivity] = useState("1.2");
  const [goal, setGoal] = useState("maintain");
  const [calories, setCalories] = useState(null);

  const formik = useFormik({
    initialValues: {
      weight: "",
      height: "",
      age: "",
    },
    validationSchema: validationSchema,
    onSubmit: values => {
      const { weight, height, age } = values;
      const weightNum = parseFloat(weight);
      const heightNum = parseFloat(height);
      const ageNum = parseInt(age, 10);
      const activityNum = parseFloat(activity);

      let bmr;
      if (gender === "male") {
        bmr = 10 * weightNum + 6.25 * heightNum - 5 * ageNum + 5;
      } else {
        bmr = 10 * weightNum + 6.25 * heightNum - 5 * ageNum - 161;
      }

      let totalCalories = bmr * activityNum;

      if (goal === "lose") {
        totalCalories -= 300;
      } else if (goal === "gain") {
        totalCalories += 300;
      }

      setCalories(totalCalories.toFixed(0));
    },
  });
});

const dietLinks = {
  male: {
    gain: "/diets/2",
    lose: "/diets/1",
    maintain: "/diets/4",
  },
};

```

```

    female: {
      gain: "/diets/5",
      lose: "/diets/3",
      maintain: "/diets/6",
    },
  };

const linkPath = dietLinks[gender]?.[goal] || dietLinks[gender]?.maintain;
return (
  <motion.form
    initial={{ y: 150 }}
    whileInView={{ y: 0 }}
    transition={{ duration: 0.3 }}
    viewport={{ once: true }}
    onSubmit={formik.handleSubmit}
    className="kcalCalc__form text"
  >
    <div className="kcalCalc__top">
      <div className="kcalCalc__item">
        <label>Weight (kg):</label>
        <input
          name="weight"
          onChange={formik.handleChange}
          onBlur={formik.handleBlur}
          value={formik.values.weight}
        />
        {formik.touched.weight && formik.errors.weight ? (
          <div className="error">{formik.errors.weight}</div>
        ) : null}
      </div>
      <div className="kcalCalc__item">
        <label>Height (cm):</label>
        <input
          name="height"
          onChange={formik.handleChange}
          onBlur={formik.handleBlur}
          value={formik.values.height}
        />
        {formik.touched.height && formik.errors.height ? (
          <div className="error">{formik.errors.height}</div>
        ) : null}
      </div>
      <div className="kcalCalc__item">
        <label>Age:</label>
        <input
          name="age"
          onChange={formik.handleChange}
          onBlur={formik.handleBlur}
          value={formik.values.age}
        />
        {formik.touched.age && formik.errors.age ? (
          <div className="error">{formik.errors.age}</div>
        ) : null}
      </div>
    </div>
    <div className="kcalCalc__gender">
      <div className="kcalCalc__item">
        <label>Gender:</label>
        <div className="kcalCalc__buttons">
          <button
            type="button"

```

```

        className={`kcalCalc__button ${gender} === "male" ?
"btn-active" : ""}`}
        onClick={() => setGender("male")}
      >
        Male
      </button>
      <button
        type="button"
        className={`kcalCalc__button ${gender} === "female" ?
"btn-active" : ""}`}
        onClick={() => setGender("female")}
      >
        Female
      </button>
    </div>
  </div>
</div>
<div className="kcalCalc__activity">
  <div className="kcalCalc__item">
    <label>Activity Level:</label>
    <div className="kcalCalc__buttons">
      <button
        type="button"
        className={`kcalCalc__button ${activity} === "1.2" ?
"btn-active" : ""}`}
        onClick={() => setActivity("1.2")}
      >
        Sedentary
      </button>
      <button
        type="button"
        className={`kcalCalc__button ${activity} === "1.375" ?
"btn-active" : ""}`}
        onClick={() => setActivity("1.375")}
      >
        Lightly active
      </button>
      <button
        type="button"
        className={`kcalCalc__button ${activity} === "1.55" ?
"btn-active" : ""}`}
        onClick={() => setActivity("1.55")}
      >
        Moderately active
      </button>
      <button
        type="button"
        className={`kcalCalc__button ${activity} === "1.725" ?
"btn-active" : ""}`}
        onClick={() => setActivity("1.725")}
      >
        Very active
      </button>
      <button
        type="button"
        className={`kcalCalc__button ${activity} === "1.9" ?
"btn-active" : ""}`}
        onClick={() => setActivity("1.9")}
      >
        Super active
      </button>
    </div>
  </div>
</div>

```

```

        </div>
      </div>
      <div className="kcalCalc__goal">
        <div className="kcalCalc__item">
          <label>Goal:</label>
          <div className="kcalCalc__buttons">
            <button
              type="button"
              className={`kcalCalc__button ${goal === "lose" ? "btn-
active" : ""}`}
              onClick={() => setGoal("lose")}
            >
              Lose Weight
            </button>
            <button
              type="button"
              className={`kcalCalc__button ${goal === "maintain" ?
"btn-active" : ""}`}
              onClick={() => setGoal("maintain")}
            >
              Maintain Weight
            </button>
            <button
              type="button"
              className={`kcalCalc__button ${goal === "gain" ? "btn-
active" : ""}`}
              onClick={() => setGoal("gain")}
            >
              Gain Weight
            </button>
          </div>
        </div>
      </div>
      <div className="kcalCalc__submit">
        <button type="submit">Calculate</button>
      </div>
      <div className="kcalCalc__myDiet">
        {calories && (
          <div className="kcalCalc__result">
            Your Daily Calories: {calories}
            <Link to={linkPath} className="kcalCalc__link">
              My Diet
              <img src={arrow} alt="arrow" />
            </Link>
          </div>
        )}
      </div>
    </motion.form>
  );
};

export default KcalForm;

// KcalInfo.jsx

import { motion } from "framer-motion";

import "./KcalInfo.scss";

const KcalInfo = () => {
  return (
    <>

```

```

    <div className="kcalCalc__info info">
      <motion.div
        className="info_calculator"
        initial={{ y: 150 }}
        whileInView={{ y: 0 }}
        transition={{ duration: 0.3 }}
        viewport={{ once: true }}
      >
        <h2 className="info_title title">Calorie Calculator</h2>
        <p className="info_desc text">
          Our calorie calculator helps you determine the daily
          calorie intake required to achieve your goals: lose weight,
          maintain weight, or gain weight. Simply input your parameters
          (weight, height, age, gender), choose your activity level and
          desired goal. The calculator will automatically calculate the
          optimal calorie intake needed to maintain your health and reach
          your desired results.
        </p>
      </motion.div>
      <motion.div
        className="info_use"
        initial={{ y: 150 }}
        whileInView={{ y: 0 }}
        transition={{ duration: 0.3 }}
        viewport={{ once: true }}
      >
        <h2 className="info_title title">How to use:</h2>
        <ul className="info_list text">
          <li className="info_item">
            Enter your weight, height, and age.
          </li>
          <li className="info_item">Select your gender.</li>
          <li className="info_item">Indicate your activity
            level.</li>
          <li className="info_item">Choose your goal.</li>
          <li className="info_item">
            Click the "Calculate" button to find out your daily
            calorie intake.
          </li>
        </ul>
      </motion.div>
    </div>
  </>
);
};

export default KcalInfo;

```

ДОДАТОК Б. Слайди мультимедійної презентації

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНАХТ»

Спеціальність: 123 – «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

ДИПЛОМНИЙ ПРОЕКТ

КГ 07.17.000.00 ДП

Яворський Ігор Олегович

НА ТЕМУ:

«Розробка веб орієнтованої експертної системи для індивідуального підбору дієт харчування»

Стек технологій для розробки веб-застосунку



HTML - мова розмітки документів для створення веб-сторінок.



CSS - мова стилів, яка використовується для опису зовнішнього вигляду і форматування веб-сторінок.



JavaScript - мова програмування, яка використовується для створення інтерактивних та динамічних веб-сторінок.

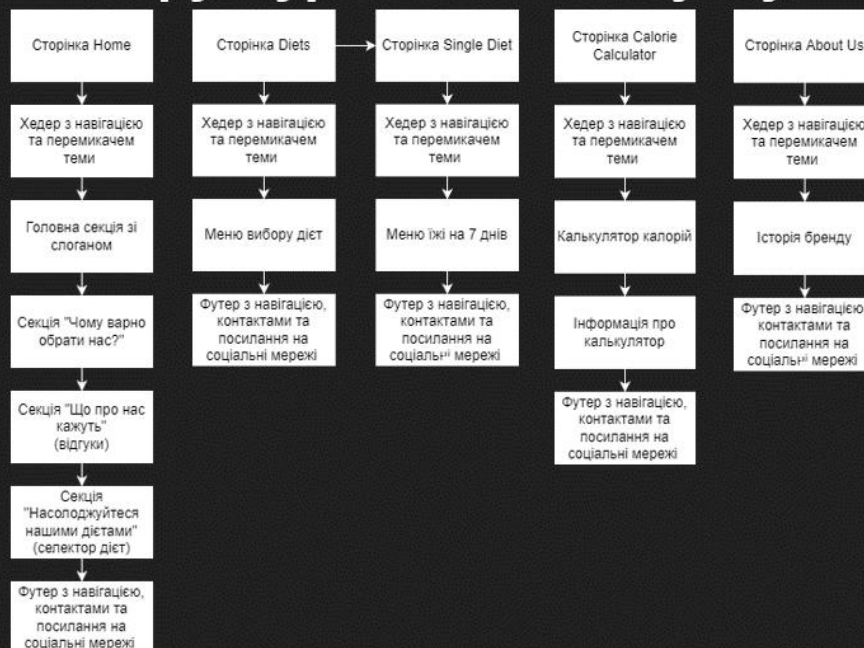


React Js - популярна бібліотека JavaScript для побудови користувацьких інтерфейсів.

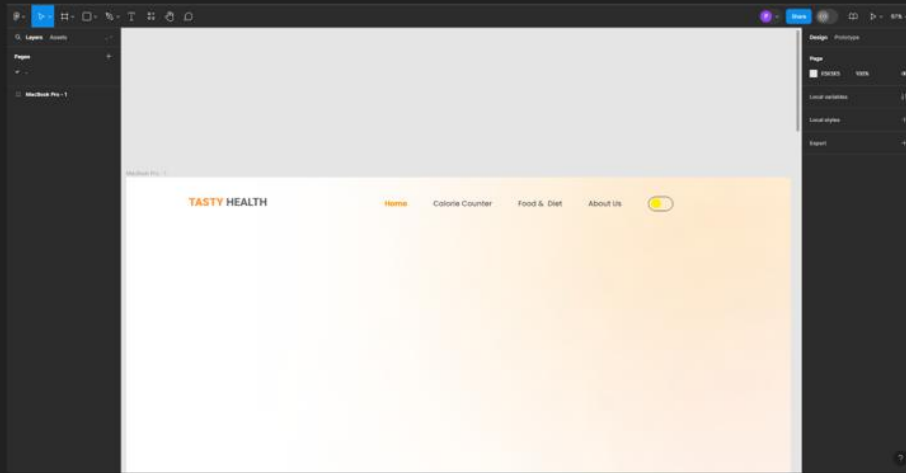
Етапи створення веб-застосунку

1. Розробка структури веб-застосунку
2. Розробка дизайну веб-застосунку
3. Розробка Front-end веб-застосунку
 - o Створення компонентів
 - o Створення сторінок
 - o Створення роутингу
 - o Створення анімацій
4. Розробка скриптів
5. Оптимізація веб-застосунку
6. Тестування роботи веб-застосунку

Структура веб-застосунку



Розробка дизайну хедеру веб-застосунку



Розробка хедеру веб-застосунку, який складається з логотипу, навігаційного меню та перемикача

Кольори веб-застосунку:

 #FB9333


 #FFFFFF

акцентні кольори


 #FB9333

 575757 100%

колір заголовків

 828282

колір тексту

 #FFFFFF

фоновий колір

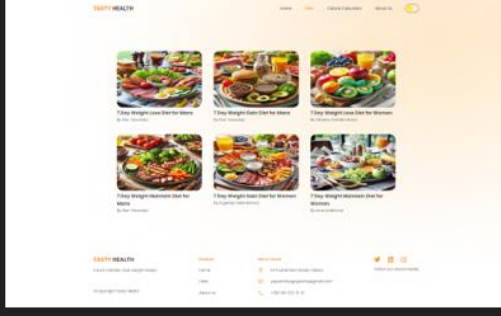

```
module.exports = ({ page }) => {
  return (
    <div style={styles.container} className="calculator-form">
      <div style={styles.form}>
        <div style={styles.inputs}>
          <input type="text" value={height} style={styles.input} />
          <input type="text" value={weight} style={styles.input} />
          <input type="text" value={age} style={styles.input} />
          <input type="text" value={gender} style={styles.input} />
          <input type="text" value={activity} style={styles.input} />
        </div>
        <div style={styles.buttons}>
          <button type="button" value="Calculate" style={styles.button} />
          <button type="button" value="Reset" style={styles.button} />
        </div>
        <div style={styles.result}>
          <div style={styles.resultText}>
            <span>Your BMI is: <span style={styles.bmi}>{bmi}</span></span>
          </div>
          <div style={styles.resultText}>
            <span>Your BMR is: <span style={styles.bmr}>{bmr}</span> kcal/day</span>
          </div>
          <div style={styles.resultText}>
            <span>Your TDEE is: <span style={styles.tdee}>{tdee}</span> kcal/day</span>
          </div>
          <div style={styles.resultText}>
            <span>Your maintenance calories are: <span style={styles.maintenance}>{maintenance}</span> kcal/day</span>
          </div>
          <div style={styles.resultText}>
            <span>Your deficit calories are: <span style={styles.deficit}>{deficit}</span> kcal/day</span>
          </div>
          <div style={styles.resultText}>
            <span>Your surplus calories are: <span style={styles.surplus}>{surplus}</span> kcal/day</span>
          </div>
        </div>
      </div>
    </div>
  );
};
};
```

Форма калькулятора калорий

Скріншоти сторінок веб-застосунку



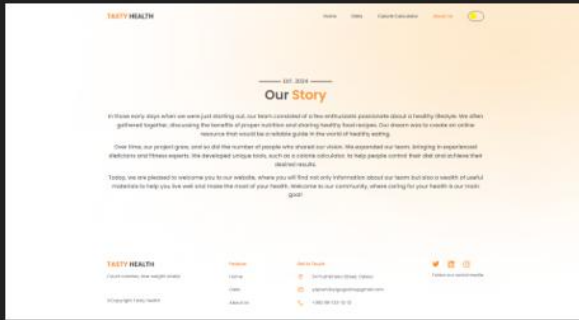
Головна сторінка



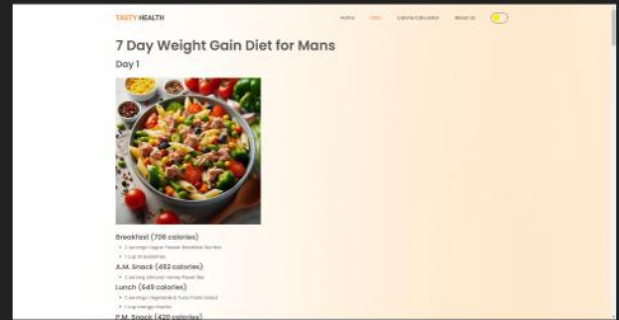
Сторінка з дієтами



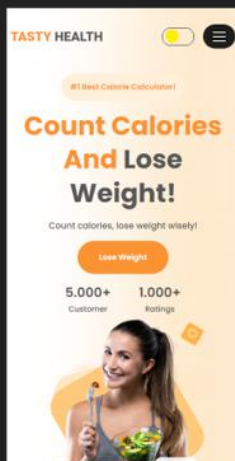
Сторінка з калькулятором калорій



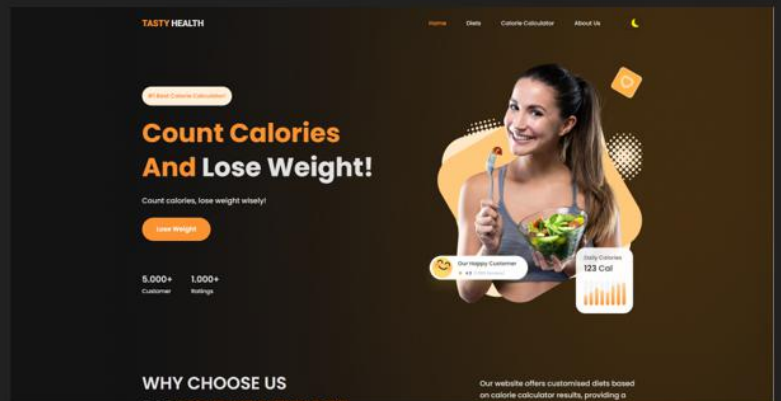
Сторінка "Про нас"



Сторінка з окремою дієтою



Головна сторінка (мобільна версія)



Головна сторінка (темна версія)

Дякую за увагу!

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Яворського Ігоря Олеговича

(прізвище, ім'я та по батькові)

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Тема дипломного проекту: Розробка веб-орієнтованої експертної системи для індивідуального підбору дієт

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 82 сторінки. У пояснювальній записці описано етапи розробки веб-орієнтованої експертної системи для індивідуального підбору дієт засобами JavaScript фреймворку React. Графічна частина складається з окремих слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувач освіти Яворський Ігор поступово та послідовно виконував всі етапи, проявив ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувач освіти Яворський Ігор під час роботи над дипломним проектом вивчив достатньо багато літературних та інтернет-джерел за даною тематикою.

Вважаю, що теоретична підготовка дипломника достатня і він готовий до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти Яворський Ігор показав вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як Microsoft Visual Code, React, Microsoft PowerPoint, Microsoft Visio та ін.

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Добре

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Жадан Артур Сергійович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис _____

«10» серпня 2024 р.

РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Яворського Ігоря Олеговича

(прізвище, ім'я та по батькові)

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна графіка і Web-дизайн»

Керівник дипломного проекту (роботи) Жадан Артур Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) *Розробка веб-орієнтованої експертної системи для індивідуального підбору дієт*

Обсяг розрахунково-пояснювальної записки 82 сторінок

Обсяг графічної (презентаційної) частини 13 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі індивідуальному підбору дієт та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування веб-системи, реалізації веб-системи, тестування веб-системи), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 13 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Реалізовано веб-орієнтовану систему, що підбирає індивідуальні дієти на основі вхідних параметрів.
Розроблене програмне базується на сучасній технології – React.js.
Стилізація графічного інтерфейсу відповідає тематиці обраної предметної області.

д) основні недоліки дипломного проекту _____
Веб-орієнтована система не містить позначки, що зображення страв в дістах були згенеровані штучним інтелектом.
В основному розділі пояснювальної поверхнево описано головний алгоритм підбору дієт, відсутнє графічне представлення алгоритму. В деяких частинах пояснювальної записки присутні помилки оформлення.

Оцінка розрахункової частини _____	<u>Добре</u>
Оцінка графічної частини _____	<u>Добре</u>
Загальна оцінка _____	<u>Добре</u>

Прізвище, ім'я, по батькові рецензента Васіліу Євген Вікторович

Місце роботи і посада рецензента Державний університет інтелектуальних технологій і зв'язку, д.т.н., проф. кафедри КБ та ТЗІ



_____ червня 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016336203

Дата перевірки:
08.06.2024 20:25:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.06.2024 21:12:29 EEST

ID користувача:
100011688

Назва документа: 4КГ-07_Яворський_I

Кількість сторінок: 44 Кількість слів: 6159 Кількість символів: 48074 Розмір файлу: 5.22 MB ID файлу: 1016136989

1.14% Схожість

Найбільша схожість: 0.47% з Інтернет-джерелом (<https://blog.logrocket.com/using-bootstrap-react-tutorial-examples>)

1.14% Джерела з Інтернету

137

Сторінка 46

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

2

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Яворський Ігор Олегович,
здобувач освіти гр. 4КГ-07, та

Жадан Артур Сергійович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка веб-орієнтованої експертної системи для індивідуального підбору дієт» (автор роботи – Яворський І.О., керівник роботи – Жадан А.С.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Яворський І.О. /

Керівник



/ Жадан А.С. /

«10» червня 2024 р.