**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ   
Державний торговельно-економічний університет   
Факультет інформаційних технологій   
Кафедра інженерії програмного забезпечення та кібербезпеки**

**ЗВІТ   
З КОЛЕКТИВНОЇ-ІНДИВІДУАЛЬНОЇ РОБОТУ   
на тему: «Розроблення вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner"»**

Виконали студенти факультету ФІТ 3-4:   
Наталія АВЄРІНА \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
 (оцінка)   
Андрій ГРАБОВСЬКИЙ \_\_\_\_\_\_\_\_\_\_\_\_\_   
 (оцінка)   
Олена ЛИМАР \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
 (оцінка)   
Данило ХИМИЧ \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
 (оцінка)   
Олександр МАРКІН \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
 (оцінка)   
Анастасія ГУМЕНЮК \_\_\_\_\_\_\_\_\_\_\_\_\_\_   
 (оцінка)   
Перевірив доктор техн. наук, професор, професор кафедри інженерії програмного забезпечення та кібербезпеки Цюцюра С.В.

Київ 2024

**ЗМІСТ**

[ВСТУП 4](#_Toc164039723)

[Розділ 1 Технічне задання вебсайту для персоналізованих рекомендацій щодо харчування «NutriPlanner» 5](#_Toc164039724)

[1.1. Вступ 5](#_Toc164039725)

[1.2. Характеристика об’єкта управління 6](#_Toc164039726)

[1.3. Призначення ІТ системи «NutriPlanner» 7](#_Toc164039727)

[1.4. Основні вимоги до ІТ системи «NutriPlanner» 9](#_Toc164039728)

[1.5. Техніко-економічні показники ІТ системи «NutriPlanner» 13](#_Toc164039729)

[1.6. Склад, зміст і організація робіт по створенню ІТ системи «NutriPlanner» 14](#_Toc164039730)

[Розділ 2 Розроблення вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner" 17](#_Toc164039731)

[2.1 Вступ 17](#_Toc164039732)

[2.2. Процес розробки вебсайту та опис використаних технологій 17](#_Toc164039733)

[2.3. Висновки 21](#_Toc164039734)

[Розділ 3 Розроблення системи менеджменту системи якості вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner" та його тестування 22](#_Toc164039735)

[3.1 Вступ 22](#_Toc164039736)

[3.2 Вимоги якості виконання функції ІТ системи “NutriPlanner” 22](#_Toc164039737)

[3.3. Розробка петлі якості за ISO 9001 23](#_Toc164039738)

[3.4 Тестування та перевірка 26](#_Toc164039739)

[3.5 Критерії оцінки ефективності діяльності із забезпечення якості 27](#_Toc164039740)

[3.6. Висновок 29](#_Toc164039741)

[Розділ 4 Проведення внутрішнього аудиту розробленого вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner" 31](#_Toc164039742)

[4.1. Вступ до ІТ-аудиту програмного продукту «NutriPlanner» 31](#_Toc164039743)

[4.2. Методологія та критерії оцінки аудиту для ІТ-продукту «NutriPlanner» 32](#_Toc164039744)

[4.3. План аудиту 33](#_Toc164039745)

[4.4. Результати оцінювання критеріїв аудиту для додатку «Nutriplanner» 34](#_Toc164039746)

[4.5. Рекомендації та вирішення ризиків, що були виявлені під час IT-аудиту «NutriPlanner» 38](#_Toc164039747)

[4.6. Висновки аудиту ІТ-проекту «NutriPlanner» 40](#_Toc164039748)

[ВИСНОВКИ 42](#_Toc164039749)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 43](#_Toc164039750)

[ДОДАТКИ 44](#_Toc164039751)

# **ВСТУП**

Розробка вебсайту для персоналізованих рекомендацій щодо харчування «NutriPlanner» є актуальним завданням в сучасному світі, де здоровий спосіб життя та правильне харчування набувають все більшого значення. Відстеження різноманітності та балансу харчових речовин допомагає забезпечити належне харчування та запобігти дефіцитам або переїданню певних макро- та мікронутрієнтів. Раціональне харчування сприяє підтримці оптимальної ваги тіла, зниженню ризику розвитку серцево-судинних захворювань, цукрового діабету, ожиріння та інших хронічних захворювань. Споживання правильних продуктів та забезпечення організму достатньою кількістю енергії допомагає підтримувати високий рівень ефективності та продуктивності протягом усього дня. Також правильне харчування може покращити настрій, знизити рівень стресу та підвищити загальний рівень задоволення від життя.

Саме тому вебсайт рекомендацій щодо харчування зможе допомогти зберегти здоров'я, покращити якість життя та досягти балансу між фізичним та психічним благополуччям.

Головною метою програми є полегшення процесу вибору здорового раціону на день.

Завданням розробки вебсайту для персоналізованих рекомендацій щодо харчування «NutriPlanner» є:

1. Написання технічного завдання
2. Програмування вебсайту за вимогами технічного завдання
3. Тестування та виправлення помилок
4. Внутрішній аудит

# **Розділ 1 Технічне задання вебсайту для персоналізованих рекомендацій щодо харчування «NutriPlanner»**

* 1. **Вступ**

1. Найменування IT:

Повне найменування: Вебсайт для персоналізованих рекомендацій щодо харчування “NutriPlanner”

Коротке найменування: “NutriPlanner”

1. Підстава для створення ІТ системи “NutriPlanner”

Робота ведеться на підставі навчального плану ДТЕУ за напрямом “Інженерія програмного забезпечення” дисципліни “Стандартизація та метрологія програмного забезпечення”

1. Терміни початку та закінчення роботи зі створення ІТ системи

Початок:12.02.24

Ранній фініш: 19.02.24

Пізній фініш: 24.04.24

1. Найменування розробника та замовника

Розробниками даного продукту є студенти групи ФІТ 3-4:

* Авєріна Наталія Ігорівна – керівник команди,
* Грабовський Андрій Олегович - архітектор технічного завдання,
* Лимар Олена Геннадіївна – аудитор,
* Гуменюк Анастасія Русланівна – тестувальник.

Команда програмістів:

* Химич Данило Олександрович,
* Маркін Олександр Костянтинович.

Замовник програмного продукту – кафедра інженерії програмного забезпечення та кібербезпеки.

1. Відомості про джерела і порядок фінансування

Дана робота виконується в рамках навчального плану та програми дисципліни “Стандартизація та метрологія програмного забезпечення” та немає ніякого фінансування на загальних підставах. Всі розробки належать ДТЕУ.

1. Загальні відомості

Проект розробки вебсайту "NutriPlanner" призначений для надання користувачам персоналізованих рекомендацій з харчування на основі їхніх індивідуальних потреб та цілей. Повсякденний вибір правильних продуктів є важливим елементом здорового способу життя, і "NutriPlanner" спростить цей процес, допомагаючи користувачам збалансувати своє харчування.

Мета проекту полягає у створенні інтуїтивно зрозумілої, ефективної та функціональної програми, яка забезпечуватиме користувачам не лише рекомендації з харчування, а й інформацію щодо харчових властивостей продуктів, порцій та калорійності.

* 1. **Характеристика об’єкта управління**

1. Опис складу об’єкта управління:

Тип об’єкта: вебсайт для персоналізованих рекомендацій щодо харчування "NutriPlanner".

Структурна схема об’єкта: веб-додаток з базою даних, алгоритмами рекомендацій та інтерфейсом користувача.

Технологічні процеси: збір та аналіз даних про користувачів (зріст, вага тощо), обробка інформації щодо індивідуальних харчових потреб користувачів, створення персоналізованих рекомендацій, відображення рекомендацій на інтерфейсі вебсайту.

1. Характеристики вхідних і вихідних матеріальних (інформаційних) потоків:

Вхідні матеріальні потоки: дані користувачів (особисті дані, вік, вага, зріст, стать, алергії, дієти, харчові цілі тощо), інформація про харчування (продукти, рецепти, харчові властивості).

Вихідні матеріальні потоки: Персоналізовані рекомендації щодо харчування (список страв, порції, розподіл на прийоми їжі).

1. Опис особливостей об’єкта управління:

Режим роботи: онлайн доступ до сервісу 24/7 для користувачів з усього світу.

Обмеження доступу: інформаційна конфіденційність (доступ до особистих даних тільки для авторизованих користувачів), обмеження прав доступу до функціональності за двома групами - адміністратори, звичайні користувачі.

Мапа вебсайту зображує послідовний список його сторінок. Вона наведена на рис. 1.1.

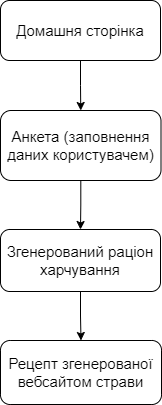


Рисунок 1.1 – Схема мапи вебсайту "NutriPlanner"

* 1. **Призначення ІТ системи «NutriPlanner»**

ПризначенняIT системи “NutriPlanner” полягає в розробці вебсайту для персоналізованих рекомендацій щодо харчування.

Аналіз цілей вебсайту "NutriPlanner" можна представити у вигляді дерева цілей, яке зображене на рис. 1.2.



Рисунок 1.2 - Дерево цілей

Щоб досягти зазначених цілей вебсайт повинен виконувати певні функції.

Функції вебсайту "NutriPlanner" можна представити у вигляді дерева функцій, яке зображене на рис. 1.3.

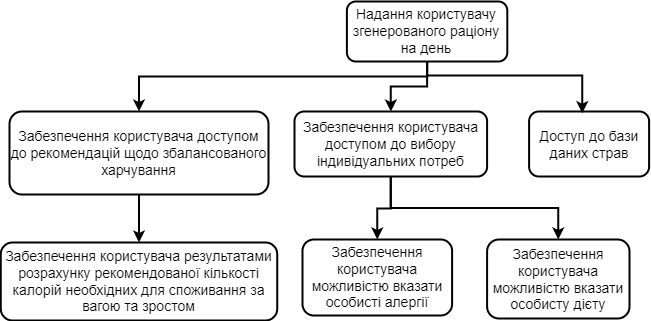


Рисунок 1.3 - Дерево функцій

**Опис загальної структури системи управління об’єктом**

Система складається з клієнтської та серверної частини.

Клієнтська частина - вебсайт, який відображає інтерфейс для користувачів.

Серверна частина - обробляє дані користувачів, робить розрахунки та формує рекомендації.

Можливий взаємозв’язок створюваної ІТ системи “NutriPlanner” з системами управління інших рівнів:

* Перший рівень: вебсайт "NutriPlanner" забезпечує зв'язок з користувачем.
* Другий рівень: можливе інтегрування з системами зберігання даних користувачів та статистичного аналізу.
* Третій (*перспективний*) рівень: можлива взаємодія з системами доставки їжі або онлайн-магазинами продуктів.

Перспективами розвитку IT системи “NutriPlanner” є власне впровадження можливості покупки готових страв чи відсутніх інгредієнтів в онлайн магазинах партнерів.

* 1. **Основні вимоги до ІТ системи «NutriPlanner»**

1. Вимоги до системи і до її частин
   1. Основні показники:

Автоматизоване управління об'єктом: Система повинна забезпечувати користувачам персоналізовані рекомендації щодо харчування відповідно до їхніх індивідуальних потреб та обраних цілей (схуднення, підтримка ваги, набирання ваги).

* 1. Структура ІТ системи:

Клієнтська частина (веб-інтерфейс)

Серверна частина (обчислення калорій, рекомендації страв)

База даних рецептів

* 1. Вимоги до функціонування ІТ системи:
     + Режим роботи: 24/7 доступність для користувачів
     + Способи обміну інформацією: Веб-запити з клієнтської сторони до серверної, база даних для зберігання та обміну даними.
  2. Ергономічні вимоги: Зручний та інтуїтивно зрозумілий інтерфейс для користувачів різних вікових категорій.
  3. Вимоги до збереження інформації при аваріях: Регулярне резервне копіювання бази даних, забезпечення її цілісності та доступності.

Наприклад автоматичне резервне копіювання щогодини на зовнішній сервер.

в) Вимоги до надійності:

Гарантування безперебійної роботи системи і захист від витоку або втрати даних користувачів.

Час відновлення після відмови, спричиненої некоректними діями користувача чи адміністратора не має перевищувати 15 хвилин.

Час безвідмовної роботи системи має складати >95% на рік при умові безвідмовної роботи мережі.

1. Вимоги якості виконання функції ІТ системи “NutriPlanner”
   1. Перелік функцій управління:
      * Введення даних про користувача (стать, вага, зріст, вік)
      * Введення індивідуальних потреб користувача (алергії, дієти, цілі харчування)
      * Розрахунок калорій за наданими користувачем параметрами
      * Розділення калорій на три прийомі їжі
      * Знаходження страв для кожного прийому їжі з заданими параметрами в базі даних рецептів
      * Відображення згенерованого денного раціону на інтерфейсі
      * Можливість повторного генерування денного раціону, конкретного прийому їжі чи страви
      * Можливість перегляду рецептів та харчової цінності згенерованих страв
   2. Вимоги до функціоналу:

Реалізація алгоритму розрахунку калорій за введеними параметрами користувача та забезпечення правильної роботи алгоритму, задля забезпечення здорового та збалансованого харчування

Підбір страв з врахуванням індивідуальних потреб та обмежень користувача (алергії, дієти, цілі харчування)

* 1. Вимоги до організації вихідних та вихідних даних:

Простий та доступний інтерфейс вводу даних користувача

Відображення згенерованих рекомендацій на інтерфейсі у зручному та зрозумілому для користувачів різних вікових категорій форматі

1. Вимоги до видів забезпечення ІТ системи "NutriPlanner"
   1. Вимоги до програмного, інформаційного, організаційного, технічного та інших видів забезпечення відповідно до вимог, встановлених на ІТ системи конкретних видів.
      * Вимоги до мов програмування:

Для реалізації статичних сторінок та шаблонів повинні використовуватися мови HTML 4.0 та CSS. Вихідний код повинен розроблятися відповідно до стандартів W3C (HTML 4.0).

Для реалізації інтерактивних елементів клієнтської частини повинна використовуватися мова JavaScript.

Система керування базами даних: MySQL

* + - Система керування версіями:

Використання Git для контролю версій програмного коду

* + - Вимоги до організації гіперпосилань:

Усі посилання на сайті мають бути відносними (за винятком зовнішніх).

* + - Вимоги до ілюстрацій

Усі малюнки та фото об'ємом більше 1 кБ (крім елементів дизайну сторінки) повинні бути виконані з текстом, що їх заміщає. Усі малюнки мають бути у форматі gif, jpg чи png.

* + - Вимоги до обсягу однієї сторінки

Обсяг однієї стандартної сторінки сайту, що завантажується, в середньому не повинен перевищувати 170 кБ.

* 1. Вимоги до інформаційного забезпечення:
     + Зберігання даних:

Усі дані страв та рецепти повинні зберігатися в базі даних.

Файли, такі як зображення страв або інші медіафайли, повинні зберігатися в файловій системі з посиланнями на них в базі даних.

* + - Стандарти:

Всі дані, передані між клієнтом та сервером, мають бути зашифровані за допомогою протоколу HTTPS.

* 1. Вимоги до організаційного забезпечення

Методи розробки:

Використання методології Agile для керування проектом та розробки програмного забезпечення.

* 1. Серверне обладнання:

Операційна система: Windows.

Мінімальні вимоги до обладнання: Intel Core i5 або еквівалентний, 8 ГБ оперативної пам'яті, 100 ГБ вільного місця на диску.

СУБД – MySQL версії не нижче 7.0

Вебсайт має бути доступний для повнофункціонального перегляду за допомогою наступних браузерів:

- MS IE 5.0 та вище;

- Opera 6.0 та вище;

- Mozilla Firefox 1.0 та вище

- Mozilla 1.7 та вище

- Chrome version 100 та вище

Мережеве забезпечення:

Забезпечення постійного доступу до Інтернету для користувачів та сервера.

* 1. Мовні версії:

Сайт повинен бути доступним на українській мові.

Існують *перспективи* перекладу вебсайту також на англійську мову.

* 1. Ергономіка та технічна естетика

Оптимізація для різних розмірів екрану:

Вебсайт повинен бути адаптивним та оптимізованим для перегляду на пристроях з різними розмірами екрану, включаючи комп'ютери та мобільні телефони.

Дизайн інтерфейсу:

Інтерфейс користувача повинен бути зручним та інтуїтивно зрозумілим, з урахуванням принципів UX/UI дизайну.

Використання приємних кольорових схем та естетичних елементів дизайну для покращення візуального враження користувачів.

* 1. **Техніко-економічні показники ІТ системи «NutriPlanner»**

Проект розробляється в ході вивчення дисципліни університету “Стандартизація та метрологія програмного забезпечення” і не передбачає комерційної мети. Таким чином, бюджет не надається і не є предметом розгляду. Вся робота проводиться з використанням студентських ресурсів та в рамках навчального процесу.

Будь-які витрати, які можуть виникнути в ході розробки, покриваються самими учасниками проекту на добровільній основі або засобами, наданими університетом для навчальних цілей.

Передбачувана сума витрат становить 0 грн.

Максимально допустима сума витрат в ході проекту: 1000 грн.

Коефіцієнт економічної ефективності витрат не застосовується через відсутність бюджету. Однак, можна врахувати ефективність використання трудових ресурсів студентського колективу та набуття навичок у процесі розробки проекту.

Ефективність трудових ресурсів повинна забезпечити вчасне виконання поставленої навчальним планом задачі – розробки даного проекту.

* 1. **Склад, зміст і організація робіт по створенню ІТ системи «NutriPlanner»**

**Стадія 1**: Аналіз вимог та планування, побудування технічного завдання (Термін: 2 тижні)

* Провести детальний аналіз вимог до системи, зокрема щодо функціональності, інтерфейсу користувача, та вимог до безпеки та захисту даних.
* Визначити ключові функціональні та нефункціональні вимоги, що включають в себе збір інформації від користувача, обробку даних та генерацію персоналізованих рекомендацій.
* Розробити план робіт, включаючи розподіл завдань між командою, визначення термінів виконання та необхідні ресурси.

**Стадія 2**: Розробка вебсайту "NutriPlanner" (Термін: 4 тижні)

* Розробка веб-інтерфейсу для користувачів з можливістю введення особистих даних.
* Розробка алгоритмів обчислення кількості калорій, необхідних для досягнення цілей харчування (схуднення, підтримка ваги, набирання ваги).
* Створення бази даних з рецептами страв, їх складовими та калорійністю.
* Розробка алгоритму підбору страв із бази даних, враховуючи кількість калорій та інші вимоги користувача.

**Стадія 3:** Забезпечення якості розроблюваного ПЗ у відповідності із стандартами (Термін: 2 тижні)

* Планування якості
* Вимоги до якості
* Контроль якості
* Впровадження вимог до якості

**Стадія 4:** Опис процедур проведення внутрішнього аудиту розробленого ПЗ (Термін: 2 тижні)

1. Планування аудиту:

* Визначення цілей аудиту, зокрема перевірка відповідності розробленого програмного забезпечення вимогам специфікації та безпековим стандартам.
* Складання робочого графіка аудиту, включаючи призначення відповідальних осіб та визначення термінів виконання кожного етапу.
* Вибір методів та інструментів, що будуть використані під час аудиту.

1. Виконання аудиту:

* Перевірка відповідності розробленого програмного забезпечення функціональним та нефункціональним вимогам, встановленим під час аналізу вимог та проектування системи.
* Оцінка якості коду та архітектури програмного забезпечення, виявлення можливих проблем та вразливостей.
* Перевірка документації, включаючи технічну документацію та інструкції з експлуатації.

1. Аналіз результатів:

* Визначення виявлених недоліків та проблем із програмним забезпеченням.
* Оцінка серйозності та пріоритетності виявлених проблем.
* Розробка рекомендацій щодо виправлення виявлених недоліків та покращення якості програмного забезпечення.

1. Підготовка звіту:

* Складання документованого звіту про результати внутрішнього аудиту, включаючи виявлені недоліки, рекомендації та плани подальших дій.
* Подання звіту керівнику групи та іншим відповідальним особам для подальшого аналізу та вжиття необхідних заходів.

1. Виправлення виявлених недоліків:

* Розробка та впровадження плану заходів щодо виправлення виявлених проблем та недоліків.
* Контроль за виконанням рекомендацій аудиту та вжиттям відповідних заходів з покращення якості програмного забезпечення.

1. Оцінка ефективності:

* Перевірка впроваджених заходів щодо виправлення недоліків на ефективність.
* Аналіз та оцінка покращень в якості програмного забезпечення після впровадження рекомендацій аудиту.
* Підготовка висновків щодо результатів внутрішнього аудиту та його впливу на якість програмного забезпечення.

# **Розділ 2 Розроблення вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner"**

## **2.1 Вступ**

Розробка програмного додатку є складним та многогранним процесом, що вимагає детального планування, аналізу вимог, проєктування, реалізації, тестування та регулярного оновлення.

Цей документ містить відомості про середовище розробки, включаючи інструменти та технології, що використані під час розробки програмного забезпечення. Описані технічні аспекти допомагають забезпечити консистентність та стандартизацію у всьому процесі розробки.

Відомості, представлені в цьому документі, призначені для розробників, менеджерів проекту та будь-яких зацікавлених сторін, які бажають отримати уявлення про те, як створений та функціонує програмний продукт.

## **2.2. Процес розробки вебсайту та опис використаних технологій**

Файл *package.json* містить інформацію про проект, його залежності, скрипти та інші налаштування (див. рис. 2.1. або Додатки). Залежності розробки включають різні пакети ESLint для аналізу та виявлення помилок в коді JavaScript, що допомагає підтримувати якість програмного забезпечення на високому рівні.

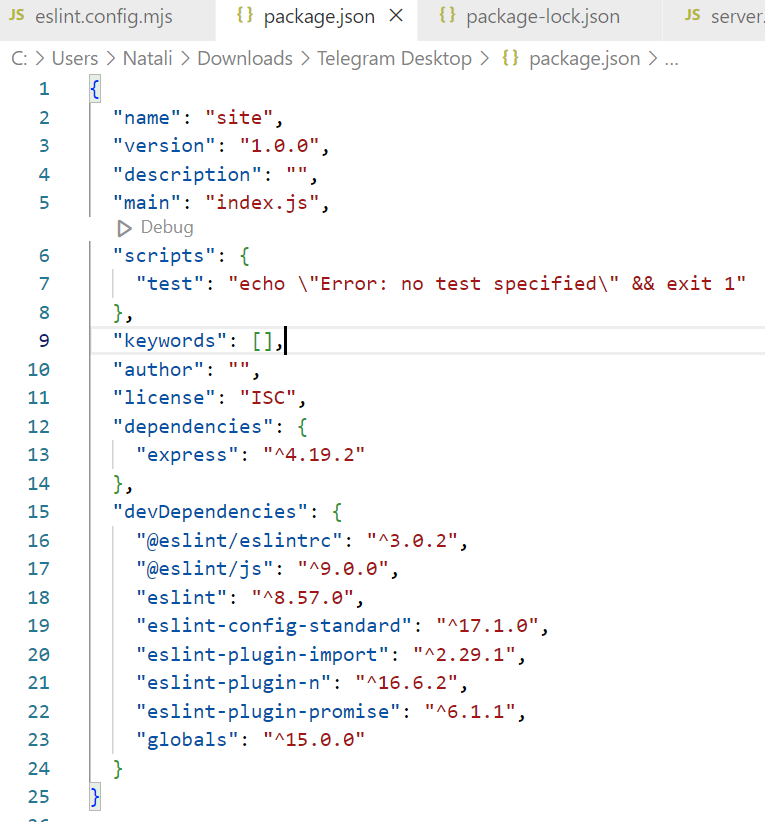


Рисунок 2.1 Знімок екрану коду файлу *package.json*

За допомогою фреймворку Express.js для Node.js. було створено простий веб-сервер, який обробляє статичні файли і відображає їх в браузері за адресою <http://localhost:3000> (див. рис. 2.2. або Додатки).



Рисунок 2.2 Знімок екрану коду файлу *server.js*

Файл конфігурації ESLint (*eslint.config.mjs*) встановлює правила лінтингу (аналізу та виявлення помилок) для JavaScript-коду в проєкті (див. рис. 2.3. або Додатки).



Рисунок 2.3 Знімок екрану коду файлу *eslint.config.mjs*

Також підключений файл *package-lock.json*, який містить точну інформацію про версії всіх пакетів, які встановлені у проєкті. Він гарантує, що всі члени команди або інші користувачі проєкту отримають ті ж самі версії пакетів, що дозволяє уникнути проблем, пов'язаних з різницею у версіях пакетів.

Код файлу *index.html* створює веб-сторінку з формою для введення даних користувача про їх ріст, вагу та ціль щодо досягнення певного результата (схуднення чи набору ваги) (код див. в Додатки). Після введення даних та їх підтвердження натисканням кнопки "Надіслати", вони відправляються на сервер для подальшої обробки, а відповідь виводиться на сторінці для користувача. Для зручності використання веб-сторінки використовується бібліотека *Axios* для здійснення HTTP-запитів на клієнтському боці.

Розроблений CSS-код встановлює стилі для всього документу, роблячи акцент на фоновому градієнті та фоновому зображенні (код див. в Додатки). За допомогою градієнту, від світло-синього до фіолетового, створюється привабливий та плавний перехід у фоні сторінки. Фонове зображення, що застосовується, доповнює цей ефект, розташовуючись справа сторінки та масштабуючись, щоб повністю заповнити фон. Встановлення стандартного шрифту *Arial* або *sans-serif* забезпечує читабельність тексту для користувачів на різних пристроях.

Додатково, визначаються стилі для заголовків (h1 та h2), форми зв'язку та кнопки відправки. Це включає встановлення відступів, рамок, та радіусу кутів для форми, а також налаштування вигляду для елементів вводу, текстових полів та випадаючих списків у формі. Кнопка відправки форми отримує вигляд з виділеним фоном при наведенні курсору для покращення інтерактивності та користувацького досвіду.

JS-код починає виконуватися після того, як весь HTML-документ завантажено в браузері (код див. в Додатки). Він отримує посилання на форму та кнопку відправки за їхніми ідентифікаторами. При поданні форми на сервер перешкоджається за допомогою функції, яка перехоплює подію відправки форми та збирає значення полів форми. Залежно від вибраної користувачем цілі (схуднення чи набір ваги), розраховується раціон харчування. Для цього використовуються випадкові страви з певною кількістю калорій для кожного прийому їжі (сніданок, обід, вечеря). Результат розрахунку представляється у вигляді рядка HTML-коду, який містить назву прийому їжі та перелік страв з кількістю калорій кожної страви. Після цього, цей результат виводиться на сторінці для користувача.

## **2.3. Висновки**

Вебсайт повинен мати чистий та організований код, що дотримується стандартів програмування та має мінімальну кількість помилок, тож використання інструментів, таких як ESLint, допомагає виявляти та виправляти помилки в коді. Управління залежностями за допомогою файлів package.json та package-lock.json допомагає забезпечити стабільність та сумісність версій пакетів під час розробки та розгортання веб-сайту. Файл server.js відповідає за створення веб-сервера та обробку запитів. Важливо забезпечити його надійність та оптимізацію для ефективної роботи веб-сайту. Також при розробці важливо пам'ятати про користувацький досвід, забезпечуючи зручний та привабливий інтерфейс, а також швидку та ефективну роботу веб-сайту.

Загалом, успішна розробка веб-сайту вимагає комплексного підходу, що включає якість коду, ефективне управління залежностями, стабільність серверної частини, увагу до користувацького досвіду та систематичне тестування.

# **Розділ 3 Розроблення системи менеджменту системи якості вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner" та його тестування**

## **3.1 Вступ**

Програмний продукт NutriPlanner розробляється з метою надання персоналізованих, збалансованих та здорових рекомендацій щодо харчування для користувачів. Однак, перед випуском продукту на ринок необхідно забезпечити його високу якість та ефективність. Для цього проводиться комплексне тестування, спрямоване на валідацію функціональності та надійності програми.

Метою тестування програми NutriPlanner є перевірка відповідності її функціоналу вимогам, забезпечення стабільної роботи та виявлення можливих недоліків перед випуском на ринок. Завдання включають перевірку встановлених вимог, валідацію функцій, перевірку взаємодії компонентів програми, виявлення та усунення помилок у коді, щоб забезпечити високу якість продукту.

## **3.2 Вимоги якості виконання функції ІТ системи “NutriPlanner”**

1. Перелік функцій управління:

* Введення даних про користувача (стать, вага, зріст, вік)
* Введення індивідуальних потреб користувача (алергії, дієти, цілі харчування)
* Розрахунок калорій за наданими користувачем параметрами
* Розділення калорій на три прийомі їжі
* Знаходження страв для кожного прийому їжі з заданими параметрами в базі даних рецептів
* Відображення згенерованого денного раціону на інтерфейсі
* Можливість повторного генерування денного раціону, конкретного прийому їжі чи страви
* Можливість перегляду рецептів та харчової цінності згенерованих страв

1. Вимоги до функціоналу:

Реалізація алгоритму розрахунку калорій за введеними параметрами користувача та забезпечення правильної роботи алгоритму, задля забезпечення здорового та збалансованого харчування

Підбір страв з врахуванням індивідуальних потреб та обмежень користувача (алергії, дієти, цілі харчування)

1. Вимоги до організації вихідних та вихідних даних:

Простий та доступний інтерфейс вводу даних користувача

Відображення згенерованих рекомендацій на інтерфейсі у зручному та зрозумілому для користувачів різних вікових категорій форматі

## **3.3. Розробка петлі якості за ISO 9001**

План розробки петлі якості (див. рис. 3.1.) за ISO 9001 для програми NutriPlanner може виглядати наступним чином:

1) Маркетинг (пошук і вивчення ринку)

2) Проєктування та розробка технічних вимог до ПЗ

3) Розробка коду

4) Тестування

5) Оцінка і аналіз помилок

6) Внесення змін

7) Повторна перевірка

8) Випуск

9) Підтримка та постійне вдосконалення

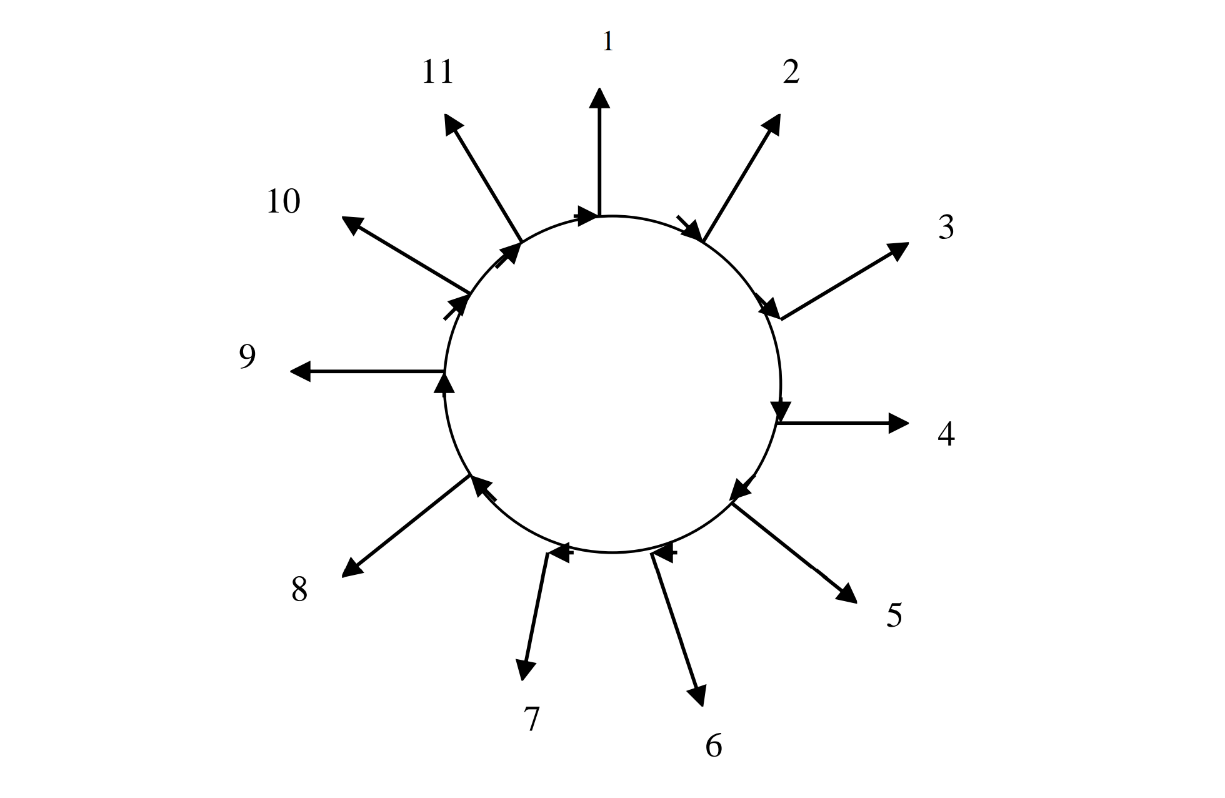


Рисунок 3.1 Петля якості

Перший етап розробки програми NutriPlanner - маркетингові дослідження. Він полягає у встановленні вимог до якості продукту, визначенні потреб користувачів та аналізі конкурентного середовища. Зокрема, важливо визначити очікування цільової аудиторії щодо персоналізованих харчових рекомендацій. Замовником програми NutriPlanner є компанія, яка спеціалізується на розробці програмного забезпечення для здорового способу життя.

Другий етап - розробка технічних вимог до програмного забезпечення. На цьому етапі формулюються функціональні та нефункціональні вимоги, що дозволять досягти поставлених цілей. Замовник визначає обсяг та основні функції програми, наприклад, можливість персоналізації рекомендацій з урахуванням індивідуальних характеристик користувача.

Після визначення технічних вимог розпочинається етап розробки коду програми NutriPlanner. Розробники переводять концепції та вимоги в програмний код, використовуючи відповідні технології та підходи. Важливо враховувати принципи чистого коду, які забезпечать його ефективність, зрозумілість та можливість подальшого розвитку. Розробка коду включає створення різних модулів та компонентів програми, написання функцій, класів, методів та їх взаємодію між собою. Код повинен бути добре структурованим і документованим, щоб інші члени команди могли легко розуміти його логіку та при необхідності вносити зміни.

На етапі тестування програми NutriPlanner проводяться приймальні випробування замовником для підтвердження відповідності програми вимогам і очікуванням. Під час цього тестування замовник перевіряє різноманітні аспекти програми, такі як правильність розрахунків, коректність виведення результатів, відповідність функціональності описаним вимогам та інші параметри, що є важливими для задоволення потреб користувачів. Це допомагає переконатися, що програма відповідає вимогам і очікуванням користувачів перед її впровадженням.

Після тестування ми оцінюємо всі виявлені помилки та аналізуємо, наскільки вони важливі та як вони можуть вплинути на роботу програми. Потім ми шукаємо причини цих помилок і розробляємо план дій, як виправити їх. Це допомагає зробити програму надійною і покращити її якість.

Далі після аналізу помилок та їх наслідків ми переходимо до етапу внесення змін. Це означає внесення виправлень у програмний код або внесення змін до дизайну програми з метою усунення виявлених проблем. Після внесення змін ми перевіряємо їх на відповідність вимогам та тестуємо програму, щоб переконатися, що вони були успішно впроваджені та не викликають нових проблем.

Після внесення змін у програмний код або дизайн програми проводиться повторна перевірка, щоб переконатися, що всі виправлення та зміни були виконані належним чином. Це важливий крок, оскільки дозволяє виявити та виправити будь-які помилки або проблеми, які можуть виникнути внаслідок внесених змін.

На етапі випуску програмний продукт готується для розгортання та використання користувачами. Він готовий до запуску в робочому середовищі та може бути поширений за різними каналами, такими як веб-сервери чи магазини додатків.

Після випуску програмного продукту на етапі підтримки важливо забезпечувати користувачів технічною підтримкою, вирішувати їхні проблеми та вдосконалювати продукт шляхом внесення змін і виправлення помилок. Крім того, на цьому етапі можна здійснювати постійний моніторинг роботи програми, збирати відгуки користувачів і вносити відповідні зміни для покращення якості та функціональності.

## **3.4 Тестування та перевірка**

3.4.1 Огляд стратегії тестування та методологій, які використовуються для валідації продукту.

Огляд стратегії тестування програми NutriPlanner та методологій, які використовуються для валідації продукту, включає в себе ряд етапів та методів, спрямованих на перевірку функціональності, надійності та ефективності програмного забезпечення.

Стратегія тестування включає в себе різні типи тестів, такі як функціональні тести, тести відповідності вимогам, тести на відмови, тести безпеки та інші, які спрямовані на перевірку різних аспектів роботи програми. Для валідації продукту можуть використовуватися різні методології, такі як модель V-моделі, Agile, Scrum тощо, які визначають порядок та послідовність тестування, а також взаємодію з розробкою продукту.

3.4.2 Перевірка функціональності генерації рекомендацій з харчування

Мета тесту: перевірка правильності та ефективності генерації персоналізованих рекомендацій з харчування для користувачів.

Кроки тесту:

1. Відкриття програми NutriPlanner.
2. Перехід до сторінки введення даних користувача.
3. Введення особистих даних, таких як ріст та вага.
4. Введення даних щодо поточного стану здоров'я та мети (наприклад, збільшення ваги або зниження ваги).
5. Збереження введених даних та перевірка їх коректного збереження у системі.

Очікуваний результат:

1. Усі введені дані коректно збережені у системі.
2. Система відображає коректні рекомендації щодо харчування на основі введених даних.
3. Всі функції введення даних працюють без помилок.

Критерії успішності:

1. Усі поля введення даних доступні та працюють коректно.
2. Система здійснює правильні розрахунки на основі введених даних.
3. Введені дані можна легко редагувати або видаляти за необхідності.

Цей сценарій тестування допоміг переконатися, що функціонал введення даних користувача працює правильно та надійно, що є важливим етапом у забезпеченні якості програми NutriPlanner.

## **3.5 Критерії оцінки ефективності діяльності із забезпечення якості**

За моделлю ISO/IEC 9126 можна створити основу для оцінки веб-сайту NutriPlanner (див. рис. 3.2.). Ці критерії допомагають забезпечити, що веб-сайт не лише відповідає функціональним вимогам, а й є надійним, ефективним та придатним для використання користувачами в різних умовах.

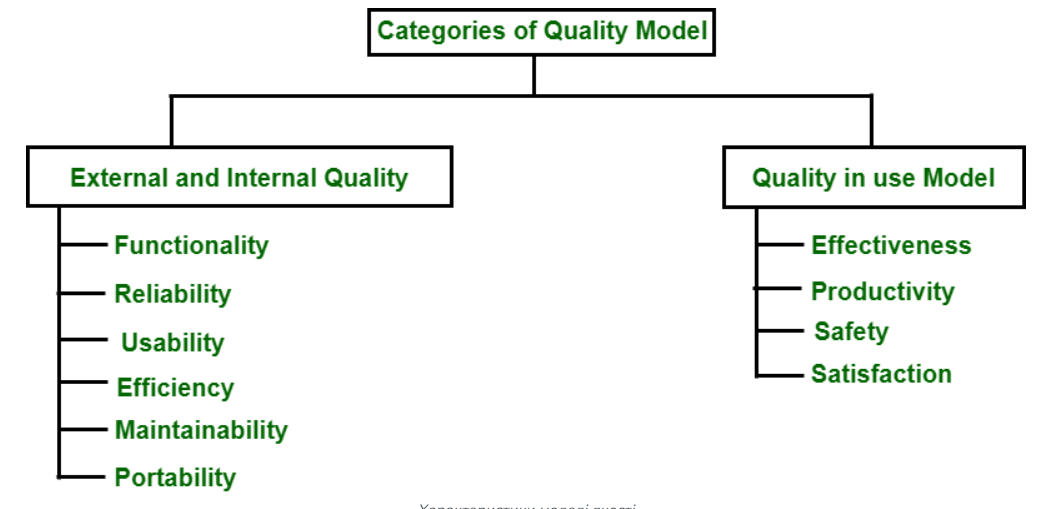


Рисунок 3.2 Категорії моделі якості

Ось критерії оцінки ефективності діяльності забезпечення якості для веб-сайту NutriPlanner, засновані на моделі ISO/IEC 9126:

Функціональність:

* Відповідність (Compliance): забезпечення відповідності веб-сайту функціональним вимогам, які включають можливість введення даних про харчування та отримання персоналізованих рекомендацій.
* Точність (Accuracy): відповідність рекомендацій харчування індивідуальним потребам користувачів, що базується на точних даних про їхнє харчування та харчових потребах.
* Інтероперабельність (Interoperability): можливість веб-сайту взаємодіяти з іншими додатками або сервісами для отримання додаткової інформації про харчування (наприклад, з фітнес-трекерами).

Надійність:

* Стабільність (Stability): забезпечення стабільної роботи веб-сайту без непередбачених вибоїв або падінь системи.
* Відновлення (Recoverability): здатність веб-сайту відновлюватися після помилок або відмови, щоб користувачі могли продовжувати отримувати рекомендації щодо харчування.

Відмовостійкість:

* Відновлення (Recoverability): здатність веб-сайту автоматично відновлюватися після відмови або збою.
* Тестова можливість (Testability): забезпечення можливості проведення тестів для виявлення та виправлення помилок у функціональності веб-сайту.

Ефективність:

* Часова ефективність (Time behavior): забезпечення швидкого та ефективного доступу до персоналізованих рекомендацій щодо харчування для користувачів.
* Використання ресурсів (Resource utilization): ефективне використання ресурсів сервера та мережі для забезпечення швидкої роботи веб-сайту.

Можливість підтримки:

* Модифікабельність (Modifiability): легкість внесення змін у веб-сайт для вдосконалення функціональності або виправлення помилок.
* Стабільність (Stability): забезпечення стабільної роботи веб-сайту після внесення змін або оновлень.

Адаптивність:

* Сприятливість (Adaptability): здатність веб-сайту адаптуватися до різних умов використання та потреб користувачів у персоналізованих рекомендаціях щодо харчування.
* Сприйняття (Acceptability): веб-сайт повинен бути прийнятим користувачами та відповідати їхнім очікуванням щодо якості рекомендацій та зручності використання.

## **3.6. Висновок**

Під час тестування було виявлено, що NutriPlanner має значні переваги щодо функціональності та користувацького досвіду, проте також були виявлені недоліки в забезпеченні безпеки даних. Слабкий рівень захисту може становити потенційну загрозу для конфіденційності та безпеки особистої інформації користувачів. Щоб забезпечити повну довіру користувачів до платформи, важливо вдосконалити систему захисту даних та розробити ефективні механізми запобігання можливим кіберзагрозам. Продовжуючи вдосконалювати якість та безпеку платформи, NutriPlanner може стати надійним інструментом для поліпшення харчових звичок та забезпечення здорового способу життя для своїх користувачів.

# **Розділ 4 Проведення внутрішнього аудиту розробленого вебсайту для персоналізованих рекомендацій щодо харчування "NutriPlanner"**

* 1. **Вступ до ІТ-аудиту програмного продукту «NutriPlanner»**

**Проєкт «NutriPlanner»** — це вебсайт, розроблений для надання персоналізованих дієтичних рекомендацій. Цей проект виконаний з використанням основних технологій веб-розробки: **HTML, CSS і JavaScript**. Він працює локально на комп'ютері без серверної частини, що виключає використання серверних технологій та баз даних. Проект розроблений студентами групи ФІТ 3-4 Державного торговельно-економічного університету в рамках дисципліни "Стандартизація та метрологія програмного забезпечення". Проєкт виконується на підставі навчального плану ДТЕУ і є частиною академічної програми, спрямованої на підвищення кваліфікації студентів у галузі розробки програмного забезпечення.

**Мета аудиту** полягає в тому, щоб переконатися, що розроблений сайт відповідає всім технічним та функціональним вимогам, викладеним у технічному завданні, а також стандартам безпеки та якості програмного забезпечення. Особлива увага приділяється захисту персональних даних користувачів, ефективності алгоритмів персоналізованих рекомендацій і взаємодії користувача з інтерфейсом сайту.

**Замовник проекту** — кафедра інженерії програмного забезпечення та кібербезпеки, яка вимагає відповідності проєкту актуальним академічним стандартам і забезпечення навчальних цілей. Розробка проекту не передбачає комерційного використання і здійснюється без зовнішнього фінансування.

**Цілі аудиту включають оцінку:**

1. Відповідності реалізації проекту поставленим вимогам та цілям
2. Якості коду та архітектури системи
3. Безпеки збереження та обробки персональних даних
4. Зручності та інтуїтивності користувацького інтерфейсу

**Методологія аудиту** охоплює ревізію документації, інспекцію коду, тестування функціональності та безпеки, а також анкетування розробників і користувачів системи для збору зворотного зв'язку.

Структура документа подальшого документа передбачає подальший розгляд кожного з аспектів аудиту в детальних розділах, що дозволяє охопити всі необхідні елементи перевірки проекту "NutriPlanner".

* 1. **Методологія та критерії оцінки аудиту для ІТ-продукту «NutriPlanner»**

Для ІТ-аудиту проекту "NutriPlanner", який є вебсайтом із переважно фронтенд-компонентами, слід обрати стандарти, критерії та методи збору результатів, які зосереджуються на клієнтській частині:

* + - 1. **Оцінка ефективності**

*Критерії:*

* 1. Час відгуку додатку.
  2. Зручність користувацького інтерфейсу.

*Методи та стандарти:*

* 1. Використання Google Lighthouse для аналізу швидкості завантаження сторінок. Це досить простий інструмент, який забезпечить базове розуміння продуктивності сайту.
  2. Проведення 2-3 сесії з залученням реальних користувачів(у цьому проекті одногрупників) для визначення інтуїтивності та зручності використання.
     + 1. **Оцінка безпеки**

*Критерії:*

* 1. Відсутність вразливостей XSS

*Методи та стандарти:*

* 1. Ручне перевірення JavaScript коду на предмет санітізації вводу. Для цього не потрібно складних інструментів, достатньо забезпечити, що дані, введені користувачами, не використовуються напряму в DOM без належної обробки.
  2. ISO/IEC 27001 і 27002: Використання для забезпечення адекватної безпеки інформації, перевірка санітізації вводу та безпеки зовнішніх бібліотек.
     + 1. **Оцінка доступності**

*Критерії:*

* 1. Відповідність WCAG рівня AА.
  2. Загальна доступність веб-інтерфейсу для людей з обмеженими можливостями.

*Методи та стандарти:*

1. Використання AChecker для перевірки сторінок на відповідність стандартам доступності. Це онлайн інструмент, який дозволяє швидко отримати звіт про доступність сайту.
   * + 1. **Оцінка підтримки**

*Критерії:*

* 1. Чистота та структурованість коду.

*Методи та стандарти:*

* 1. Перегляд коду: Ручний перегляд та використання інструментів як ESLint для оцінки якості JavaScript коду.
  2. W3C Validator: Для перевірки валідності HTML/CSS коду.

Ці методи та стандарти допоможуть забезпечити, що проект не тільки відповідає основним вимогам функціональності та доступності, але й має достатній рівень безпеки для користувачів, навіть не маючи складної серверної логіки. Такий підхід дозволить ефективно аудитувати проект, оптимізувавши процес під його специфіку.

* 1. **План аудиту**

Аудит проекту "NutriPlanner" проводиться аудитором **Лимар Оленою Геннадіївною**, студенткою кафедри інженерії програмного забезпечення та кібербезпеки. Вона відповідальна за збір інформації, перевірку відповідності розробленого сайту технічним вимогам і стандартам безпеки та аналіз зібраних даних для ідентифікації потенційних проблем або вразливостей.

Аудит розпочинається негайно після завершення розробки сайту, орієнтовно після **19 лютого 2024 року**, і має бути завершений до кінцевої дати проекту **24 квітня 2024 року**.

**Етапи аудиту:**

1. **Підготовчий етап (1 тиждень):** збір всіх необхідних матеріалів і документів, встановлення і налаштування інструментів для тестування.
2. **Виконання тестування і збір даних (1 тиждень):** проведення тестів на відповідність стандартам, доступності, безпеці, та функціональності.
3. **Аналіз результатів і підготовка звіту (1 тиждень):** аналіз зібраних даних, ідентифікація проблем, і складання звіту з рекомендаціями.
4. **Подача звіту і рекомендацій (1 тиждень):** презентація звіту керівнику групи та розробницькій команді, обговорення можливих покращень.
5. **План на випадок затримок:** включення додаткового тижня як буферу для неочікуваних затримок або проблем.
   1. **Результати оцінювання критеріїв аудиту для додатку «Nutriplanner»**

Сайт "NutriPlanner" був розміщений на локальному сервері, де він стабільно працював і займав приблизно **38 МБ** оперативної пам'яті. Вебсайт успішно відображається у різних браузерах та на різних пристроях. Інструмент Google Lighthouse був використаний для оцінки продуктивності, показавши відмінні результати з часом завантаження сторінки та високими балами по всіх категоріях. Дві сесії з реальними користувачами підтвердили інтуїтивну зручність та легкість використання додатку. Але слід зазначити, що не всі технічні цілі були виконані. У Технічному Завданні проекту стоять такі пункти, як авторизація користувачів та врахування персональних харчових звичок(алергія, непереносимість окремих видів продуктів) , але у кінцевому варіанті веб-сторінки ці умови не були виконані.

* + 1. **Оцінка ефективності**
* *Час відгуку додатку:* Сервер успішно обробляє запити, з часом відгуку в 0.2 секунди(див. Рис 4.1).

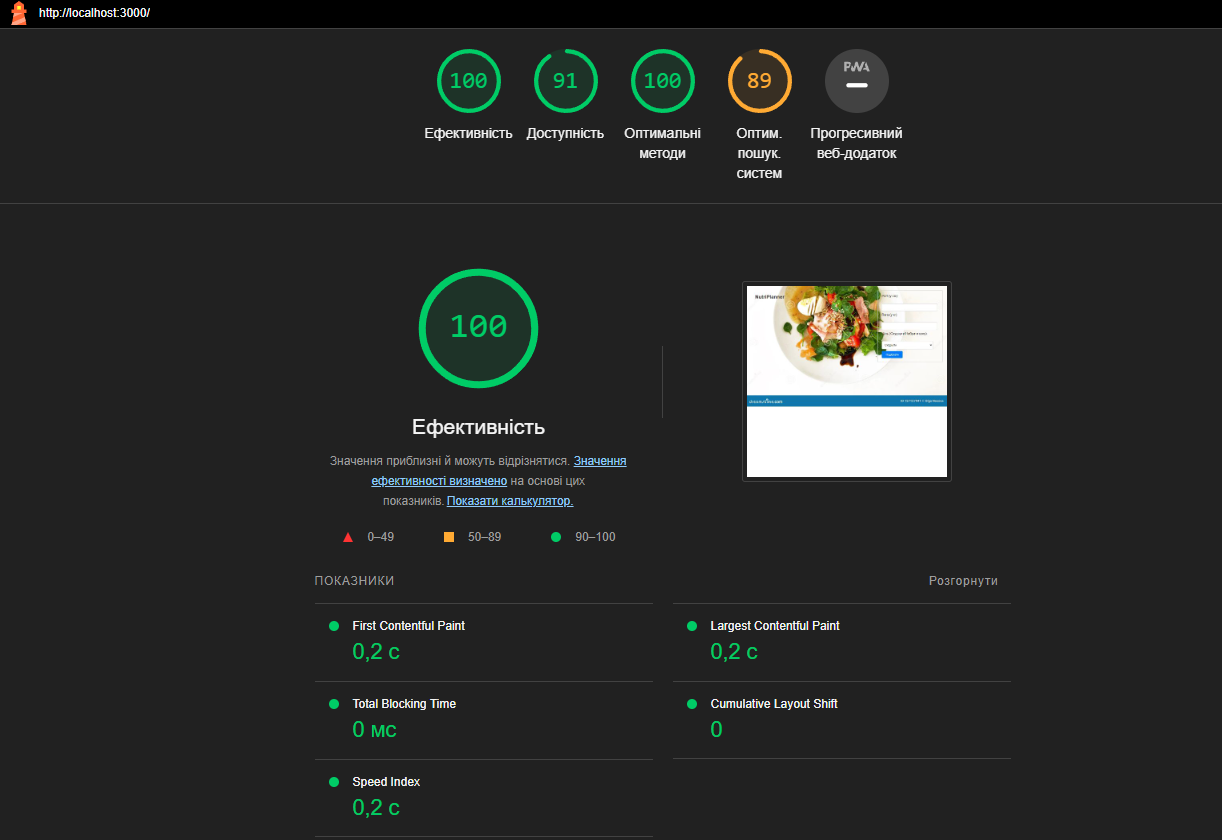


Рисунок 4.1 Результат аналізу часу відгуку сайту

* + - * *Оптимізація DOM:* Рекомендації по оптимізації DOM можуть сприяти кращій швидкодії сайту, зменшуючи використання ресурсів(див Рис.4.2)

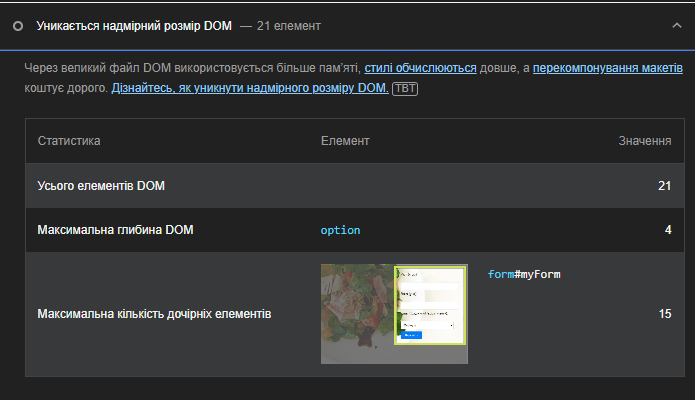


Рисунок 4.2 Виявлена проблема надмірного розміру DOM

* + 1. **Оцінка безпеки**
* *Відсутність вразливостей XSS:* Виявлено безпечне використання бібліотеки *`axios*` (див. Рис.4.3), ручний огляд коду не показав наявності XSS або CSRF вразливостей.



Рисунок 4.3 Використання бібліотеки *`axios`* у проекту «NutriPlanner»

* + - * *Відсутність шифрування:* Так як сайт запускався на локальному сервері, то протокол автоматично йде НТТР, тобто без SSL-шифрування. Це дуже погіршує безпеку сайту і конфіденційні дані, введені користувачами, не захищені.
    1. **Оцінка доступності**
* *WCAG 2.0 (Рівень AA):* Не виявлено жодних критичних проблем, відповідність стандартам доступності підтверджено.(див. Рис.4.4) Є деякі потенціальні проблеми, але ,переглянувши їх усі, було зроблено висновок, що вони майже не грають ролі при оцінюванні доступності.

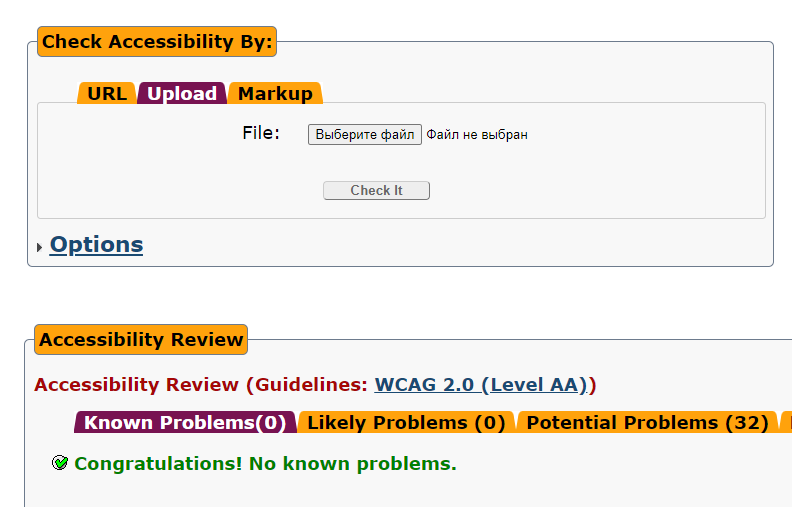


Рисунок 4.4 Успішна перевірка файлу index.html на рівень доступності *WCAG 2.0 AA* у додатку *AChecker*

* *Контрастність інтерфейсу:* Виявлено проблеми з низькою контрастністю, які потребують уваги для підвищення доступності сайту(див. Рис.4.5).

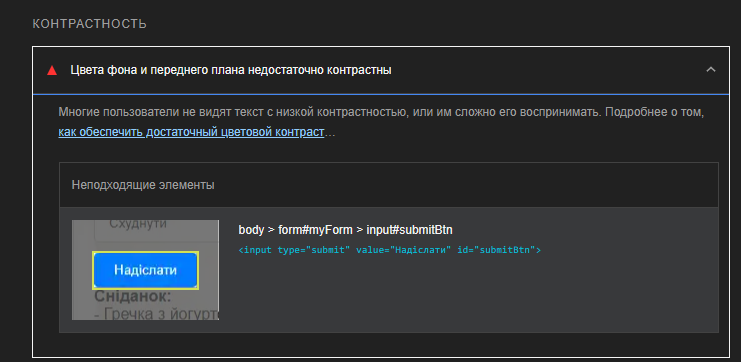


Рисунок 4.5 Виявлена проблема контрастності у програмному продукті «NutriPlanner»

* + 1. **Оцінка підтримки**
* Якість коду*: ESLint* виявив ряд питань, переважно стосовно стилю кодування у файлі *index.js,* які потребують уваги для покращення читабельності і підтримки коду(див. Рис.4.6).

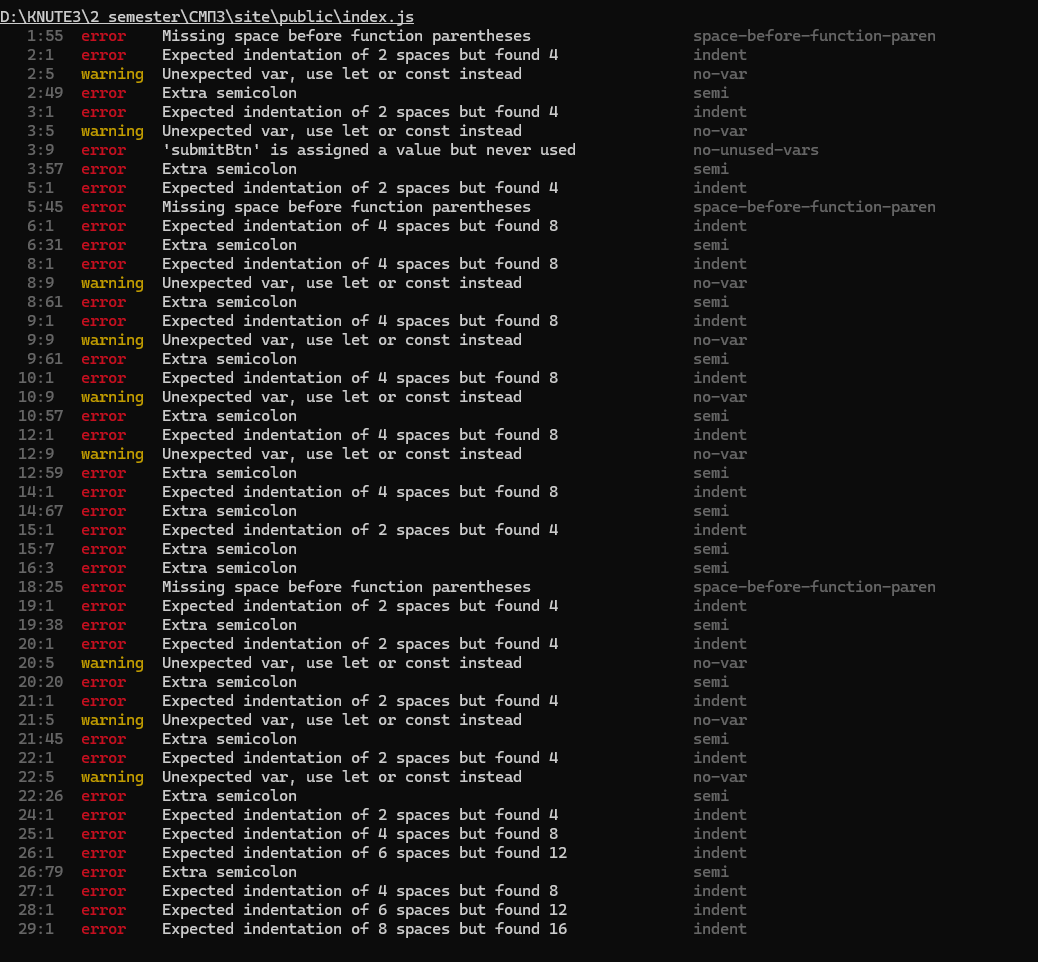


Рисунок 4.6 Виявлена проблеми у файлі *`index.js`* за допомогою інструменту *ESLint*

* Відповідність HTML стандартам: Одна помилка в HTML виявлена *W3C Validator*, що вказує на необхідність корекції для забезпечення валідності коду (див. Рис.4.7).

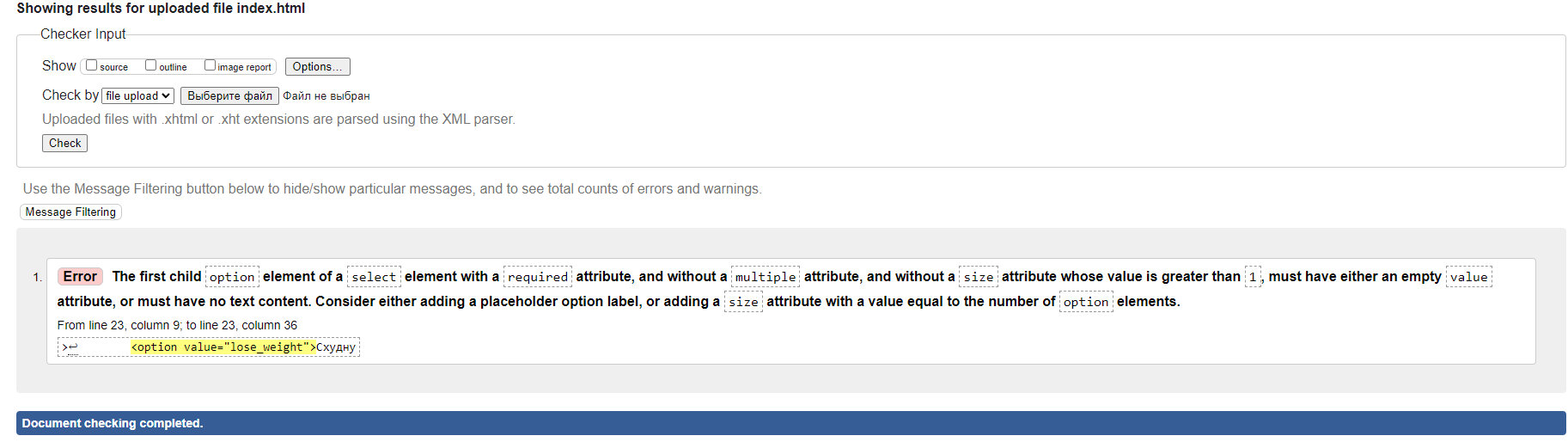


Рисунок 4.7 Помилка виявлена у файлі *index.html* за допомогою інструменту *W3C Validator*

**Висновки**

Аудит проекту "NutriPlanner" показав, що вебсайт успішно виконує свої основні функції та високо оцінюється за ефективність і доступність. Завдяки ретельному тестуванню, вебсайт продемонстрував швидку відповідь сервера та відмінну користувацьку інтуїтивність.

Однак, деякі технічні та візуальні аспекти вимагають уваги для подальшого покращення. Зокрема, потребують оптимізації структура DOM та контрастність інтерфейсу. Проблеми, ідентифіковані за допомогою інструментів ESLint та W3C Validator, потребують корекції для забезпечення кращої читабельності коду та валідності розмітки. Також рекомендовано виділити бюджет на придбання SSL-сертифікату та перейти на HTTPS-протокол.

* 1. **Рекомендації та вирішення ризиків, що були виявлені під час IT-аудиту «NutriPlanner»**

**Рекомендації:**

* + 1. *Оптимізація DOM:*

Переглянути та спростити структуру DOM, зменшити кількість елементів для поліпшення продуктивності.

* + 1. *Контрастність інтерфейсу:*

Підвищити контрастність тексту для кращої читабельності, особливо для користувачів зі зниженим зором.

* + 1. *Якість коду JavaScript:*

Виправити виявлені *ESLint* проблеми, зокрема питання форматування і використання змінних.

* + 1. *Валідність HTML:*

Усунути помилки в HTML-коді, виявлені *W3C Validator*, для відповідності стандартам веб-розробки.

**Вирішення ризиків:**

Після аналізу ІТ-продукту «NutriPlanner» усі рекомандії були надіслані команді програмістів, яка наразі вже виправила такі недоліки:

1. Контрастність**:** контраст деякої частини тексту був збільшений(див. Рис. 4.8 і Рис. 4.9), тому людям з вадами зору тепер набагато легше користуватись додатком та орієнтуватись у ньому.

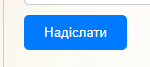
****

Рисунок 4.8 Контраст тексту на кнопці до виправлення

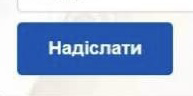


Рисунок 4.9 Контраст тексту на кнопці після виправлення

1. Виправлені помилки коду у файлі index.html, які були виявлені інструментом *W3C Validator:*Наразі вже забезпечена валідність коду у цьому документі(див. Рис.4.10. та Рис.4.11.)

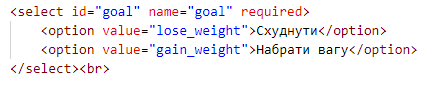


Рисунок 4.10 Частина коду у файлі index.html, де була виявлена помилка

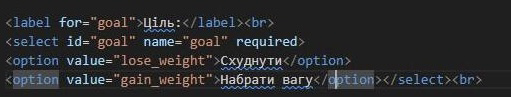


Рисунок 4.11 Виправлений фрагмент коду у файлі index.html

* 1. **Висновки аудиту ІТ-проекту** **«NutriPlanner»**

Проект "NutriPlanner" продемонстрував високу продуктивність з швидким часом відгуку та гарною роботою на локальному сервері. Це свідчить про те, що використані технології та розробка відповідають сучасним вимогам до швидкодії веб-додатків.

Використання бібліотеки axios та загальний аналіз безпеки не виявили критичних вразливостей, що підтверджує надійність вибраних рішень у контексті безпеки фронтенду.

Аудит підтвердив, що сайт відповідає стандартам WCAG 2.0 на рівні AA, що робить його доступним для широкої аудиторії користувачів, включаючи осіб з особливими потребами.

Аналіз коду виявив деякі проблеми, що вимагають уваги, зокрема, питання стилю кодування та валідності HTML, які вже виправлені, а також перехід на більш надійний протокол HTTPS і допрацювання недовиконаних цілей, що зазначені у Технічному Завданні проекту. Покращення в цих областях допомогло забезпечити більш стабільну підтримку і можливість легкого розширення проекту в майбутньому.

**Рекомендації:**

1. Провести оптимізацію DOM для поліпшення швидкодії.
2. Виправити виявлені ESLint проблеми коду JavaScript.

**Подальший план дій:**

Середньострокові дії (1 місяць):

1. Усунення помилок в коді та оптимізація DOM для поліпшення продуктивності.
2. Перегляд та рефакторинг JavaScript коду для виправлення виявлених *ESLint* проблем.
3. Повторна валідація HTML та CSS, з усуненням будь-яких виявлених помилок.
4. Придбання SSL-сертифікату та перехід на HTTPS-протокол.
5. Завершити етапи Технічного Завдання, які досі не були реалізовані( авторизація користувача та урахування харчових звичок користувача).

Довгострокові дії (2-3 місяці):

1. Проведення додаткових сесій користувацького тестування для збору зворотного зв'язку після внесених змін.
2. Планування та реалізація можливих функціональних розширень на основі зібраних даних та зворотного зв'язку.

Результати цього аудиту надають чітке визначення напрямків для покращення та розвитку проекту "NutriPlanner". Рекомендовані дії мають бути інтегровані в план розвитку проекту, щоб забезпечити його високу якість та відповідність очікуванням користувачів.

# **ВИСНОВКИ**

В ході розробки вебсайту для персоналізованих рекомендацій щодо харчування «NutriPlanner» було проведено важливі кроки, що забезпечили його успішне впровадження та функціонування. Від постановки завдання до завершення проєкту було віддано особливу увагу кожному етапу процесу.

Технічне завдання визначило основні вимоги до функціоналу та дизайну вебсайту, створивши чітку основу для подальшої розробки. Реалізація вебсайту відбувалася з урахуванням цих вимог, що забезпечило створення продукту, який відповідає потребам користувачів.

Тестування відіграло важливу роль у виявленні та виправленні потенційних проблем та помилок, забезпечуючи якість та стабільність вебсайту перед впровадженням.

Внутрішній аудит дозволив систематично переглянути та оцінити всі аспекти вебсайту, виявити можливі покращення та забезпечити його оптимальну ефективність та відповідність цілям проєкту.

Загалом вебсайт забезпечений зручним та привабливим інтерфейом, а також швидкою та ефективною роботою.

Завдяки злагодженій роботі команди та пильному контролю якості, «NutriPlanner» готовий до використання та може надати користувачам надійні та корисні інструменти для покращення їхнього харчування та здоров'я.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

* + - 1. Офіційна документація з JavaScript. [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
      2. Документація фреймворку Express.js для Node.js. [Електронний ресурс]. – Режим доступу: <https://devdocs.io/express/>
      3. Офіційна документація ESLint. [Електронний ресурс]. – Режим доступу: <https://eslint.org/docs/latest/>
      4. Наукові статті журналу "Journal of Nutrition"
      5. Офіційна документація з JavaScript. [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
      6. Документація DOM. [Електронний ресурс]. – Режим доступу: <https://developer.chrome.com/docs/lighthouse/performance/dom-size?utm_source=lighthouse&utm_medium=devtools>
      7. Ensure CSP is effective against XSS attacks . [Електронний ресурс]. – Режим доступу: [Ensure CSP is effective against XSS attacks](https://developer.chrome.com/docs/lighthouse/best-practices/csp-xss?utm_source=lighthouse&utm_medium=devtools)
      8. Google Lighthouse Overview . [Електронний ресурс]. – Режим доступу: [Overview  |  Lighthouse](https://developer.chrome.com/docs/lighthouse/overview/)
      9. AChecker Tool . [Електронний ресурс]. – Режим доступу: [Web Accessibility Checker for WCAG 2.0 ADA, AODA, EQA Compliance Monitoring (achecks.org)](https://achecks.org/checker/index.php)
      10. ESLint Tool. [Електронний ресурс]. – Режим доступу: [Find and fix problems in your JavaScript code - ESLint - Pluggable JavaScript Linter](https://eslint.org/)
      11. ISO/IEC 27001:2022 . [Електронний ресурс]. – Режим доступу: [ISO/IEC 27001:2022 - Information security, cybersecurity and privacy protection](https://www.iso.org/standard/27001)
      12. ISO/IEC 27002:2022 . [Електронний ресурс]. – Режим доступу:  [[ISO/IEC 27002:2022 - Information security, cybersecurity and privacy protection](https://www.iso.org/standard/75652.html)](https://www.iso.org/standard/27001)

# **ДОДАТКИ**

Програмний код файлу package.json

{

  "name": "site",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "express": "^4.19.2"

  },

  "devDependencies": {

    "@eslint/eslintrc": "^3.0.2",

    "@eslint/js": "^9.0.0",

    "eslint": "^8.57.0",

    "eslint-config-standard": "^17.1.0",

    "eslint-plugin-import": "^2.29.1",

    "eslint-plugin-n": "^16.6.2",

    "eslint-plugin-promise": "^6.1.1",

    "globals": "^15.0.0"

  }

}

Програмний код файлу eslint.config.mjs

import globals from 'globals'

import path from 'path'

import { fileURLToPath } from 'url'

import { FlatCompat } from '@eslint/eslintrc'

import pluginJs from '@eslint/js'

const \_\_filename = fileURLToPath(import.meta.url)

const \_\_dirname = path.dirname(\_\_filename)

const compat = new FlatCompat({ baseDirectory: \_\_dirname, recommendedConfig: pluginJs.configs.recommended })

export default [

  { languageOptions: { globals: globals.browser } },

  ...compat.extends('standard')

]

Програмний код файлу server.js

const express = require('express');

const app = express();

const port = 3000;

app.use(express.static('public'));

app.listen(port, () => {

    console.log(`Server is running at http://localhost:${port}`);

});

Програмний код файлу index.html

<!DOCTYPE html>

<html lang="uk"> <head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Форма зв'язку</title>

<link rel="stylesheet" href="index.css">

</head>

<body>

<h2>NutriPlanner</h2>

<form id="myForm">

<h3>Введіть дані</h3>

<label for="height">Ріст (у см):</label><br>

<input type="number" id="height" name="height" required><br>

<label for="weight">Вага (у кг):</label><br>

<input type="number" id="weight" name="weight" required><br>

<label for="goal">Ціль:</label><br>

<select id="goal" name="goal" required>

<option value="lose\_weight">Схуднути</option>

<option value="gain\_weight">Набрати вагу</option></select><br>

<input type="submit" value="Надіслати" id="submitBtn">

<br>

<div id="rationResult"></div>

</form>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

<script src="index.js"></script>

</body>

</html>

Програмний код файлу index.css

body {

    background: linear-gradient(to bottom, #add8e6, #9370db);

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: space-between;

}

body {

    background-image: url(https://cdn.discordapp.com/attachments/1224409838105268396/1229065946258346056/photo\_5251255109374042216\_y.jpg?ex=662e53dc&is=661bdedc&hm=756534b5a004e298d7685d5f4ca188e9be559f957ba3054fa3f112c2686e2e71&);

    background-size: cover;

    background-position: right;

    background-repeat: no-repeat;

}

h1, h2 {

    color: #333;

}

form {

    width: 320px;

    margin: 20px;

    padding: 20px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

label {

    display: block;

    margin-bottom: 10px;

}

input[type="number"],

textarea,

select {

    width: 90%;

    padding: 12px;

    margin-bottom: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

input[type="submit"] {

    background-color: #0056b3;

    color: #fff;

    padding: 15px 30px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    font-size: 18px;

    font-weight: bold;

}

input[type="submit"]:hover {

    background-color: #003c80;

}

h2 {

    margin: 20px;

    padding: 20px;

}

Програмний код файлу index.js

document.addEventListener('DOMContentLoaded', function() {

    var form = document.getElementById('myForm');

    var submitBtn = document.getElementById('submitBtn');

    form.addEventListener('submit', function(event) {

        event.preventDefault();

        var height = document.getElementById('height').value;

        var weight = document.getElementById('weight').value;

        var goal = document.getElementById('goal').value;

        var ration = calculateRation(height, weight, goal);

        document.getElementById('rationResult').innerHTML = ration;

    });

});

function calculateRation(height, weight, goal) {

    console.log(height, weight, goal);

    var ration = [];

    var bmi = weight / ((height / 100) \*\* 2);

    var totalCalories = 0;

    if (goal === 'lose\_weight') {

        if (bmi < 18.5) {

            return 'Ви занадто худі, щоб схуднути. Вам потрібно набрати вагу.';

        } else {

            while (totalCalories < 2500) {

                ration = generateRation();

                totalCalories = calculateTotalCalories(ration);

            }

        }

    } else if (goal === 'gain\_weight') {

        if (bmi > 25) {

            return 'Ви занадто повні, щоб набирати вагу. Вам потрібно скинути зайву вагу.';

        } else {

            while (totalCalories < 3000) {

                ration = generateRation();

                totalCalories = calculateTotalCalories(ration);

            }

        }

    }

    var rationStr = '';

    ration.forEach(function(meal) {

        rationStr += '<strong>' + meal.meal + ':</strong><br>';

        meal.foods.forEach(function(food) {

            rationStr += '- ' + food.name + ' (' + food.calories + ' ккал)<br>';

        });

        rationStr += '<br>';

    });

    return rationStr;

}

function generateRation() {

    var breakfastOptions = [

        { name: 'Вівсянка', calories: 300 },

        { name: 'Яйця з овочами', calories: 350 },

        { name: 'Гречка з йогуртом', calories: 400 }

    ];

    var lunchOptions = [

        { name: 'Куряча грудка з овочами', calories: 450 },

        { name: 'Рис з тушеними овочами', calories: 400 },

        { name: 'Салат з тунцем та оливковою олією', calories: 350 }

    ];

    var dinnerOptions = [

        { name: 'Творог з фруктами', calories: 250 },

        { name: 'Риба з овочами', calories: 300 },

        { name: 'Квасоля з овочами та горіхами', calories: 350 }

    ];

    var ration = [

        { meal: 'Сніданок', foods: getRandomSubset(breakfastOptions) },

        { meal: 'Обід', foods: getRandomSubset(lunchOptions) },

        { meal: 'Вечеря', foods: getRandomSubset(dinnerOptions) }

    ];

    return ration;

}

function calculateTotalCalories(ration) {

    var totalCalories = 0;

    ration.forEach(function(meal) {

        meal.foods.forEach(function(food) {

            totalCalories += food.calories;

        });

    });

    return totalCalories;

}

function getRandomSubset(array) {

    var shuffledArray = array.slice().sort(() => Math.random() - 0.5);

    return shuffledArray.slice(0, Math.floor(Math.random() \* shuffledArray.length) + 1);

}