

**КВАЛІФІКАЦІЙНА РОБОТА**

**КР.ІІЗ – 11.00.00.000 ІІЗ**

**Група ІІЗ-2017**

**Олійник Д.П.**

**2021**

**ЗАКЛАД ВИЩОЇ ОСВІТИ  
УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА**

**Факультет суспільних і прикладних наук  
Кафедра інформаційних технологій**

на правах рукопису

**Олійник Дмитро Петрович**

УДК 004.415

**Розробка вебзастосунку для обрахунку калорій за допомогою Node.js,  
Express та MongoDB**

Спеціальність 121 — «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра

Науковий керівник

к.т.н., доцент,

Мануляк Ірина Зиноївна

Івано-Франківськ — 2021

**ЗВО «Університет Короля Данила»  
Факультет суспільних і прикладних наук  
Кафедра інформаційних технологій**

Освітній ступінь: «бакалавр»

Спеціальність: 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри**

\_\_\_\_\_ 202\_\_ року  
«\_\_\_\_\_» \_\_\_\_\_

**З А В Д А Н Н Я  
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

\_\_\_\_\_  
(прізвище, ім'я, по-батькові)

1. Тема роботи

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



№ з/ п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примі тка
1	Основні поняття та принципи розробки вебзастосунків	29.03.2021	
2	Проектування та розробка застосунку	03.04.2021	
3	Програмна реалізація застосунку	25.04.2021	
4	Формування висновків	02.05.2021	
5	Охорона праці	10.05.2021	
6	Оформлення пояснювальної записки	27.05.2021	
7	Оформлення графічного матеріалу та підготовка до захисту роботи	01.06.2021	

Студент \_\_\_\_\_



	тренувань		сторінка
18	Програма харчування	40	Персональний офіс
20	Персональний тренер	40	Сторінка профілю
21	Бережи фігуру	41	Головна сторінка та сторінки верифікації
22	Власний розклад тренувань	42	Сторінки профілю, впливаючої, навігації та калькуляторів
23	Калькулятор тіла	43	Сторінки персонального офісу
24	Особистий щоденник	46	Команда перевірки Node.js
24	Калорійність	47	MVC
25	Форум	48	Project tree
29	SDLC	50	Метод Include
33	Діаграма баз даних	50	Перші кроки
33	Activity diagram	51	Nav-Bar
34	Class diagram	52	Форма калькулятора
35	State machine diagram	53	Форма калькулятора

36	Use case diagram	69	Схема профілю баз даних
----	------------------	----	----------------------------

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



# ЗМІСТ

ПЕРЕЛІК УМОВНИХ	
СКОРОЧЕНЬ.....	6
ВСТУП.....	8
РОЗДІЛ 1. ОГЛЯД АНАЛОГІВ ФІТНЕС	
ВЕБЗАСТОСУНКІВ.....	10
1.1 Огляд предметної області.....	10
1.2 Аналіз аналогів фітнес за стосунків.....	12
1.3 Постановка задачі.....	24
Висновки до розділу 1.....	25
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ФІТНЕС	
ЗАСТОСУНКУ.....	26
2.1 Огляд сфери розробки ПЗ.....	26
2.2 Конструювання власного ПЗ.....	27
2.3 Огляд технологій для конструювання ПЗ.....	41
Висновки до розділу 2.....	43
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ФІТНЕС ЗАСТОСУНКУ	
НА БАЗІ NODE.JS, EXPRESS ТА	
MONGODB.....	44
3.1 Налаштування робочого середовища.....	44
3.2 Стилзація та маршрутизація сторінок проєкту.....	48
3.3 Створення верифікації за допомогою сесій.....	52
3.4 Реалізація усіх додаткових запитів та процесів проєкту.....	53
Висновки до розділу 3.....	56
РОЗДІЛ 4. ОХОРОНА	

ПРАЦІ.....	57
4.1 Основні положення безпеки праці в ІТ-індустрії.....	57

4.2 Психологічні та фізіологічні особливості користувачів інформаційних технологій.....	63
4.3 Теоретичні основи ризику під час використання комп'ютеризованих систем.....	67
Висновки до розділу 4.....	73
ВИСНОВОК.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

1. IT - інформаційні технології
2. JS - JavaScript
3. Фреймворк - (від англ. framework) інфраструктура програмних рішень, що полегшує розробку складних систем
4. Internet Explorer - легендарний браузер від MicroSoft
5. PWA - progressive web application
6. Web application - клієнт-серверний застосунок для роботи в браузері
7. Desktop application - програма, яка самостійно запускається на комп'ютері
8. БЖВ - білки, жири, вуглеводи
9. Youtube - популярний відеохостинг
10. UI - user interface
11. Маркетплейс - (від англ. marketplace) майданчик для розміщення товарів іншими продавцями
12. Інтерфейс - сукупність засобів та методів для керування системою
13. Казуал - (від англ. casual) людина, яка любить простоту і не любить виділятися
14. Google - пошукова система
15. UX - user experience
16. CSS - Cascading Style Sheets
17. Flex - система для створення гнучких макетів за рахунок правильного розміщення елементів на сторінці
18. HTML - HyperText Markup Language
19. Бургер - впливаюче меню
20. POST - один із методів HTTP, переважно використовують для

передачі даних серверу

21. Трекінг - відстежування
22. MongoDB - документно-орієнтована система управління БД
23. HTTP - (HyperText Transfer Protocol) протокол передачі даних в мережі
24. GET - один із методів HTTP, запитує сервер вказаний ресурс
25. PUT - один із методів HTTP, завантажує вказаний ресурс на сервер
26. PATCH - HTTP-метод, завантажує тільки частину вказаного ресурсу
27. DELETE - один із методів HTTP, видаляє вказаний ресурс
28. Feedback - відгук, зворотний зв'язок
29. Supervisor - людина, яка слідкує за всім
30. Sprint - короткий часовий інтервал за який команда виконує поставлену задачу
31. ПЗ - програмне забезпечення
32. DB - data base
33. SQL - мова структурованих запитів
34. UML - Unified Modeling Language
35. RESTfull Api - програмний інтерфейс, який виконує більшість HTTP-запитів
36. Wireframe - образ дизайну низької точності.
37. Front - end - інтерфейс взаємодії з користувачем
38. Npm - node package manager
39. Ejs - embedded javascript
40. Рендерінг - процес отримання зображення за допомогою програми
41. Дебагінг - знаходження та виправлення помилок
42. Хмарка - хмарний сервіс для завантаження даних
43. Репозиторій - місце, куди заливаються і зберігаються дані
44. Пушити - заносити дані
45. Деплой - процес закидання проєкту на сервер

## 46. SDLC - Software development lifecycle

### ВСТУП

**Актуальність теми.**Програмні забезпечення зараз використовуються у всіх галузях діяльності людей, а будь-яка сучасна компанія не може обійтись без власного вебсайту чи сторінки в Facebook. На цю мить цифрові технології використовуються всюди, тому кожна людина вимушена так чи інакше бути ознайомленою з якимось ПЗ чи WEB - сайтом.

Однією з галузей, де вдало інтегрувались сучасні технології є фітнес та спорт. За допомогою вебдодатків є можливість ввести індивідуальний режим тренування та харчування. Зважаючи на ці фактори розробка вебзастосунку, який допоможе людям оптимізувати свій процес тренувань є актуальною темою для опрацювання.

**Об'єкт роботи.**Способи реалізації вебзастосунку для обрахунків та збереження даних у сфері фітнесу.

**Предмет роботи.**Вебзастосунок для збереження, опрацювання та збереження даних.

**Мета роботи.**Розробити вебзастосунок, для обрахунків персональних даних користувачів за допомогою спеціальних калькуляторів, збереження цих даних у базі даних, відображення їх на сторінці, а також можливість їх редагувати.

**Завдання роботи.**Відповідно до поставленої задачі потрібно виконати наступні завдання:

- пошук та збирання необхідної інформації для подальшої розробки проєкту
- проектування та створення дизайну майбутнього додатку
- вибір необхідних технологій для виконання роботи
- розробка застосунку

**Методи роботи.**Для виконання поставленої задачі будуть використані технології для створення користувацького інтерфейсу CSS та EJS. Перший використовується для стилізації зовнішнього вигляду, другий для розмітки структури документу, надання йому контенту та для інтеграції з серверною частиною. Для побудови RESTApi та іншої серверної логіки буде використовуватись Node.js, який у свою чергу використовує для синтаксису JavaScript. До нього під'єднаємо Express, щоб спростити написання API.

**Результати роботи.**Результатом дипломної роботи є інтернет-застосунок для обрахування та опрацювання індивідуальної інформації користувачів у сфері фітнесу.

**Структура роботи.**Розділи - 4. Загальний обсяг основної частини - 65. Список використаних джерел містить - 15 позицій.

# РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ АНАЛОГІВ ФІТНЕС ВЕБЗАСТОСУНКІВ

## 1.1 Огляд предметної області

На цей час важко уявити світ без цифрових технологій. Вони настільки інтегрувались у наше життя, що їх відсутність поверне нас далеко в минуле, ще навіть задовго до того як вони стали з'являтися. Вони полегшують наше життя у всьому: чи то у побуті, чи на роботі або навчанню. Завдяки мережі Інтернет ми з легкістю можемо здобути необхідну нам інформацію і уже не потрібно днями сидіти у бібліотеках і вивчати літературу. При цьому доступ до мережі може бути здійснений за допомогою простого кишенькового телефона. Різноманітні соціальні мережі дали можливість людям з різних точок світу взаємодіяти один з одними. Зараз ми з легкістю можемо ділитись знаннями та досвідом не виходячи з дому.

Розглянемо сферу фітнесу та спорту в ІТ і які вимоги стоять перед розробниками для створення вебзастосунків. Для початку потрібно зрозуміти, що ми живемо у світі, де люди не обмежуються лише одним комп'ютером. Зараз багато людей використовують крім стаціонарних комп'ютерів також ноутбуки і планшети, а присутність кишенькового смартфона уже стала звичайним явищем. З цього випливає, що розробник повинен враховувати всі розширення екранів для всіх пристроїв, а також потрібно враховувати, що кожен користувач користується різними телефонами в яких своя розмірність екранів і операційна система. Зрозуміло, що якщо програмний застосунок реалізований на Android, не зможе функціонувати IOS на і навпаки. Крім всіх цих нюансів також потрібно враховувати, що створюючи вебзастосунок на мові JS, неможливо повною мірою використовувати всі нововведення у цю мову, а також різні бібліотеки або фреймворки, оскільки кожен браузер диктує свої умови та



чи інша функція може не підтримуватись на приклад в легендарному Internet Explorer. Варто зазначити, що на сьогодні дедалі активніше застосовуються так звані PWA. Технологія, яка візуально та функціонально трансформує webapplication в desktopapplication. Так в не далекому майбутньому всі застосунки будуть побудовані за таким принципом, а ті хто швидше інтегрує цю технологію буде краще підготовлений до ринку IT.

Переважно перед людьми, які приходять в спортивний зал чи займаються самостійно стоять різні цілі, хтось хоче бути красивим, хтось сильним, хтось здоровим, але у всіх цих випадках потрібно слідкувати за своїм харчуванням. Під цим розуміється, що ми їмо, яку кількість калорій ми споживаємо, яке відношення білків жирів та вуглеводів і яка насиченість різними корисними елементами.

Для цього розробнику потрібно реалізувати інструменти для обрахунку калорійності продуктів і їхнє БЖВ, а також надати користувачеві можливість розрахувати свою денну норму калорій, щоб надати йому розуміння, як він має харчуватись. У кожної людини свої цілі, хтось хоче набрати масу, а хтось хоче її скинути, тому потрібно врахувати ці фактори і надати необхідний функціонал. Крім вище згаданих калькуляторів, було б не погано реалізувати наступний функціонал: калькулятори для обрахунку норми спожитої води, можливість розрахувати свій процент жиру і частоту пульсу для того, щоб не перевантажувати себе на тренуванні.

Можливість зробити необхідні розрахунки - це не щось нове на ринку, тому потрібно вдосконалювати свій застосунок, потрібно реалізувати можливість вести свій особистий щоденник в якому користувач міг би зберігати свої персональні дані, розрахунки, які проводились і можливість відстежувати статистику для кращого розуміння прогресу. Певні сайти також надають можливість розробити свій план тренування і наочно показують, як їх виконувати.

Розглянемо ринок вебзастосунків у сфері спорту та фітнесу.

## 1.2 Аналіз аналогів фітнес застосунків

Люди, які займаються спортом використовують різного типу додатки на своїх смартфонах чи вебзастосунки для того, щоб максимально ефективно покращити тренування. Все більш популярними стають фітнес браслети, які дають інформацію про те скільки кроків ти пройшов, скільки калорій спалено, відстежують серцебиття користувача, а також тривалість сну і все це підтримується додатками. Звичайно тренувань під поглядом кваліфікованих тренерів чи порад дієтологів це не замінить, проте це вже краще ніж нічого. Інколи не всі мають можливість звернутись до таких спеціалістів або придбати недешевий спортивний гаджет, тому часто люди шукають потрібну їм інформацію в мережі.

Розглянемо для прикладу досить прогресивний варіант, а саме: FITNESS BLENDER(<https://www.fitnessblender.com>) (рис. 1.1).

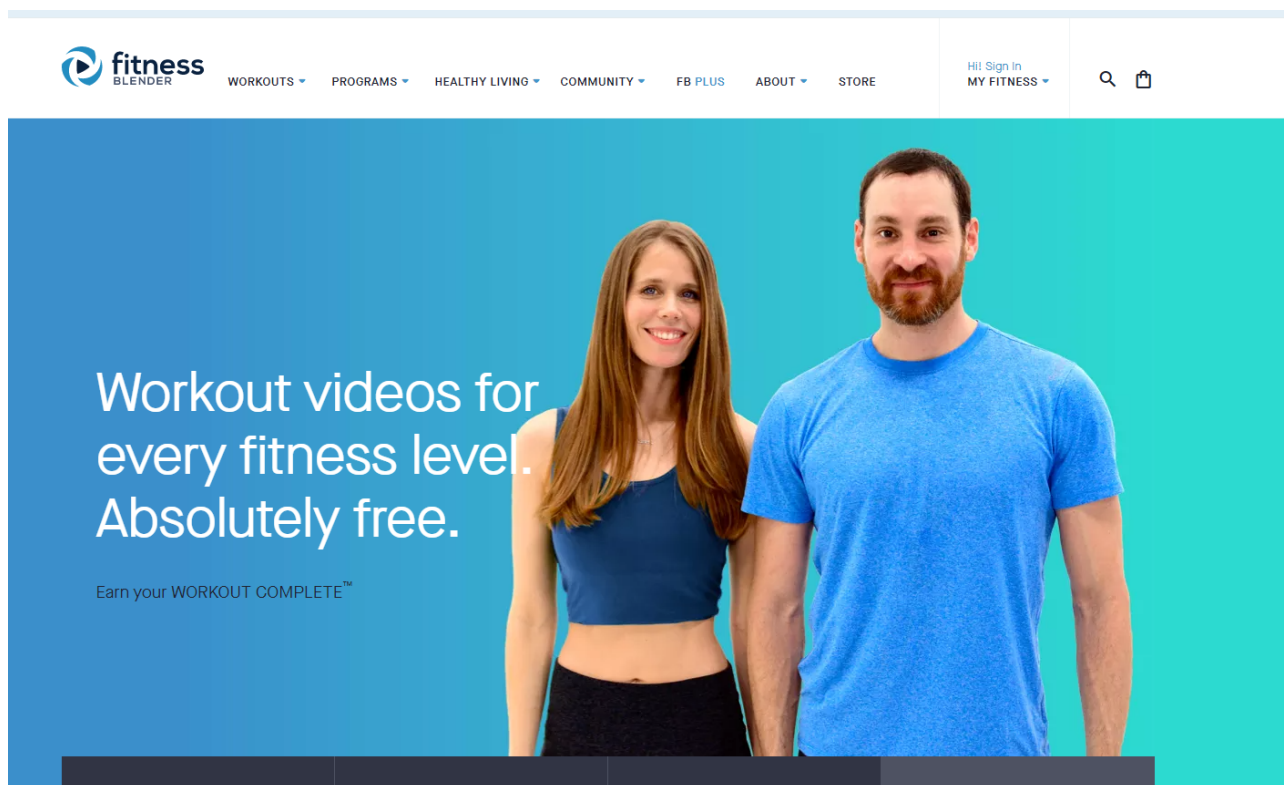


Рисунок 1.1 - Fitness Blender

Дизайн сайту досить приємний на око, а UI не дасть загубитися не досвідченому користувачу, що звичайно є плюсом. Сайт надає більше сотні

безкоштовних комплексів фізичних вправ. На самих відео наочно показано, як виконується та чи інша вправа, також збоку зручно розташований таймер та кількість спалених калорій. Для новачків набір цих вправ є достатнім для початку. На сайті також можна знайти більш різноманітний функціонал. Наприклад, кожен користувач може скласти свою програму тренувань в залежності від своїх даних, а також є уже заготовленні програми тренувань і користувачу залишається вибрати той, який йому підходить найближче. На сайті також можна знайти програми харчування і можливість скласти свою програму або вибрати наявну. Все це підкріплюється великою кількістю статей в яких користувач може знайти досить корисну для вас інформацію. Також існує додаток для смартфона та канал на Youtube. Отже, проєкт надає досить солідну базу інформації, але майже все з вище перелічених не є безкоштовним.

На сайті діє система підписки при, якій користувач отримує весь функціонал і доступ до любої інформації. Користувач може окремо придбати програму тренувань чи дієту за певну ціну. На рисунку 1.2 є можливість подивитись на тарифний план даного застосунку і зробити висновки. Ціна досить велика для більшої частини населення України, до того ж сайт повністю на англійській, що вимагає хорошого розуміння цієї мови.

З мінусів можна виділити відеоуроки, які не надають повного розуміння техніки виконання вправ, що може призвести до небажаних травм. Не озброєним оком помітно, що даний проєкт розробляла досвідчена команда розробників. Для одного ж програміста розробка даного проєкту займе не один місяць роботи, до того ж потрібна хороша база знань, щоб виконати його.

Тому як приклад для проєктної роботи даний сайт не підходить. Розглянемо більш примітивні варіанти.

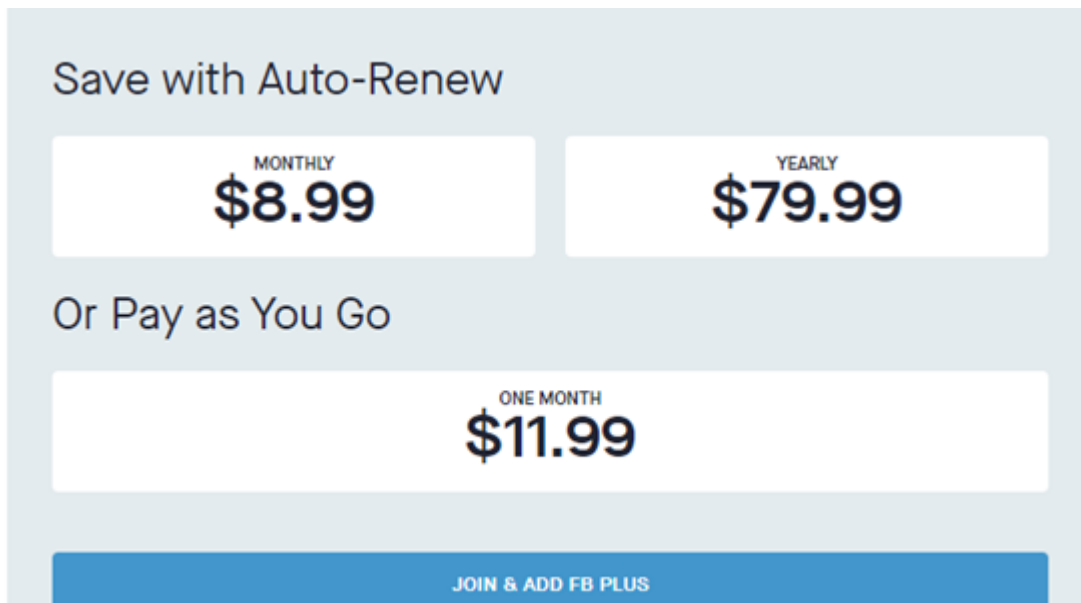


Рисунок 1.2 - Підписка на сайті

Розглянемо подібний варіант, проте на цей раз з трохи іншою системою забезпечення інформації та монетизації. Для прикладу обрано сайт: IQBODY(<https://iq-body.ru/>) (рис. 1.3). Сайт надає весь потрібний функціонал, який потрібний простому користувачу, а також виконаний у спрощеному стилі.

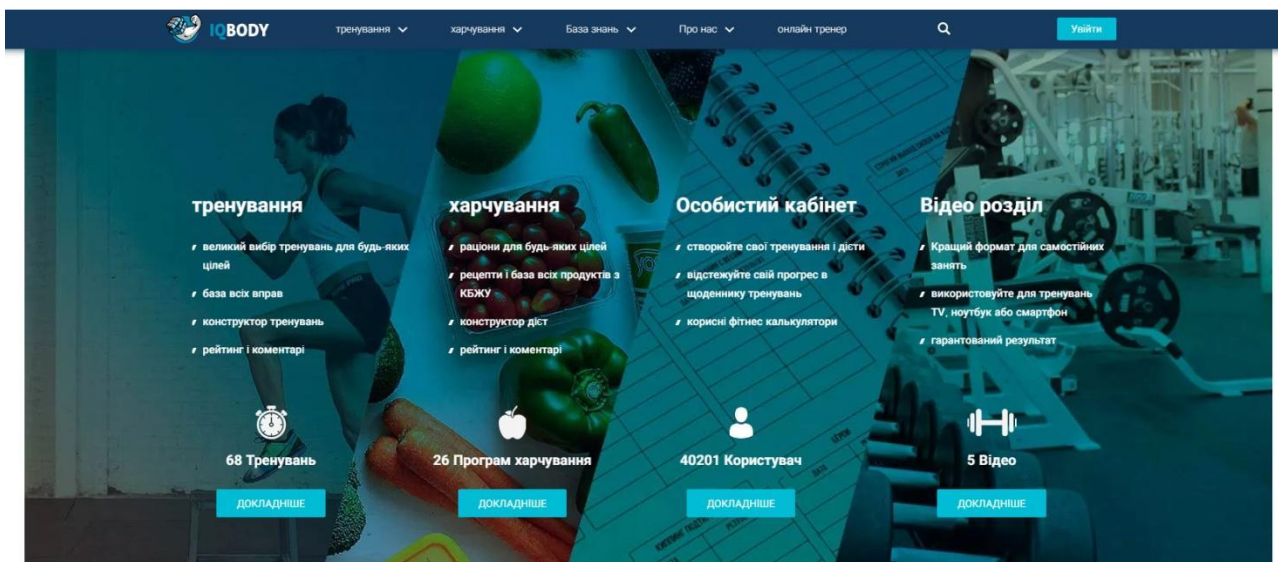


Рисунок 1.3 - IQBODY

В представленому вебзастосунку можна знайти все, що потрібно для якісних тренувань. У розділі тренування користувач має доступ до великої

кількості відеоуроків, де у доступному та зрозумілому форматі пояснюється як потрібно виконувати те чи інше навантаження. В цьому розділі приділяється увага правильності виконання вправи та надається все необхідне для того, щоб користувач мав чітке уявлення. Всі відеоуроки зручно поділити на категорії такі як:

- призначення;
- ціль тренувань;
- місце тренувань;
- рівень підготовки.

Їх можна сортувати за популярністю та новизною, тому користувач завжди зможе знайти те, що йому потрібно.

Кожна програма тренувань поділяється на певні часові рамки. Так для прикладу 1 програма може мати 15 тренувань, які проводяться протягом 5 тижнів та кожне тренування може тривати протягом 30-40 хвилин. На сторінці програми користувач може знайти всю інформацію по даному відеоуроку і вирішити чи вона йому точно підходить. Кожен користувач може знайти відгуки до даної програми та інформацію про тренера.

В підрозділі тренування на окрему групу м'язів доступна інформація про власне підхід та спосіб тренувань відповідних груп м'язів. Надана додаткова інформація по типу чим ця група м'язів відрізняється від інших і чому необхідно виконувати саме таку кількість підходів та повторів. Користувач може знайти для себе можливість розробити власну програму тренувань під свої вимоги за підтримки тренера (рис. 1.4).

У підрозділі харчування користувач може підібрати дієту, яка йому підходить найбільше. Так є можливість вибрати потрібний план харчування, передивившись наявні рецепти або скористатися послугами персональних дієтологів. В програмах харчування користувач як і в попередньому розділі може зручно відсортувати бажану дієту.

Головна / тренування / Програма тренувань з гантелями /


## Програма тренувань з гантелями

Для тих, у кого є можливість займатися тільки з гантелями

Рейтинг **9.2** серед програм **ТОП 10** переглядів **196K** Для чоловіків **♂**

[проголосувати](#)

[почати програму](#)

 197 осіб вже тренується

мета: **підготовча**

Місце тренувань: **будинки**  
[використовуваний інвентар](#)

Тривалість циклу: **8 тижнів**

Тренувань в тиждень: **3 дні**

Час тренування: **55-70 хвилин**

Рівень підготовки: **любитель**

Рисунок 1.4 - Програма тренувань

На сторінці кожної програми доступним чином пояснено, в чому плюси даної методики. Крім того, програма надає повну інформацію калорійності і поживності і розписаний повний план продуктів в конкретний проміжок часу. Користувач зручно може модифікувати вибрану програму під себе. Присутня можливість подивитись рецензії на дану програму й оцінити її актуальність. У цьому розділі також присутня можливість скласти власну програму харчування з допомогою кваліфікованих спеціалістів. Така методика забезпечить ретельний підхід до кожного користувача.

Головна / раціони / Бюджетний варіант набору маси /


## Бюджетний варіант набору маси

Як харчуватися на массанабор в умовах кризи хлопцям в 60-70 кг

Рейтинг **9.7** серед раціонів **ТОП 2** переглядів **100K** Для чоловіків **♂**

[проголосувати](#)

[Перейти до раціону](#)

 388 осіб вже використовують

мета: **набір маси**

вага: **60-70 кг**  
[Вибрати іншу вагу](#)

Прийомів їжі: **6**

інгредієнти: **3**  
[інгредієнти](#)

Рисунок 1.5 - Програма харчування

В ній будуть враховуватись всі фізичні дані особи, її вподобання, кількість калорій, яку вона витрачає протягом дня, вид діяльності та об'єм фізичного навантаження протягом дня (рис. 1.5).

В підрозділі база знань надана велика база інформації присвячена тематиці фітнесу та здоров'я. Він поділяється на 5 підрозділів, а саме: статті, новини, калькулятори, питання та визначення термінів.

В підрозділі статі подається велика кількість статей на різноманітну тематику. Користувач зручно може знайти потрібну статтю за допомогою категорій сортування. Самі ж статті надають солідну та кваліфіковану інформацію по кожній темі, які поданні у зручному форматі.

В підрозділі новини надається актуальна інформація про зміни у сучасному світі спорту, фітнесу та бодибілдінгу.

В підрозділі калькулятори надані для користування власне калькулятори з різними функціоналами.

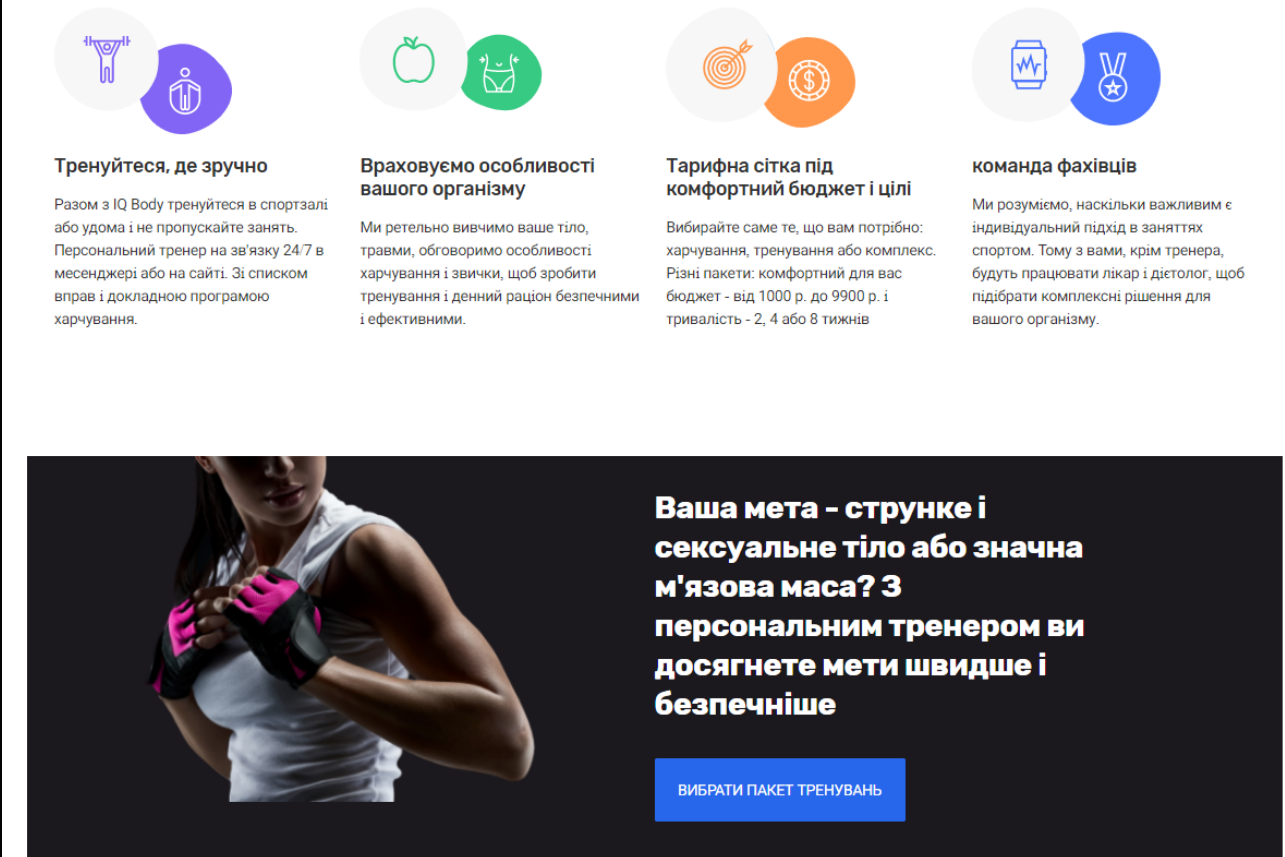
Калькулятор для підрахунку норми вживання калорій та БЖВ допоможе користувачу правильно порахувати кількість калорій, жирів, білків та вуглеводів для досягнення поставленої цілі (схуднення, набір маси чи підтримка форми).

«Робоча вага» дасть можливість підрахувати робочу вагу з урахуванням кількості підходів та максимальної ваги. За допомогою «Обчислення пульсу для кардіо» користувач отримає значення пульсу і пульсових зон. «Розрахунок вживання води» дасть поняття про те, скільки потрібно випивати воду протягом дня. Калькулятор «Індекс маси тіла» допоможе розрахувати індекс маси тіла (ІМТ) та ідеальної ваги для чоловік та жінок з урахування віку. «Дізнатись тип статури» дасть зрозуміти, яка у вас статура чи то ендоморф, чи мезоморф, чи екторморф.

В підрозділі питання та відповіді користувач може поставити своє питання адміністрації сайту і отримати відповідь, а в підрозділі терміни знайти визначення невідомих понять.

На сайті також присутня можливість персональної консультації з

тренером. З розвитком соціальних мереж виникає також потреба підтримувати себе в формі, з кваліфікованим тренером це зробити набагато легше. Проте чим більший попит на дану спеціалізацію, тим більша її вартість. Сайт надає ж цю можливість за доступними цінами (рис. 1.6).



**Тренуйтеся, де зручно**  
Разом з IQ Body тренуйтеся в спортзалі або удома і не пропускайте занять. Персональний тренер на зв'язку 24/7 в месенджері або на сайті. Зі списком вправ і докладною програмою харчування.

**Враховуємо особливості вашого організму**  
Ми ретельно вивчимо ваше тіло, травми, обговоримо особливості харчування і звички, щоб зробити тренування і денний раціон безпечними і ефективними.

**Тарифна сітка під комфортний бюджет і цілі**  
Вибирайте саме те, що вам потрібно: харчування, тренування або комплекс. Різні пакети: комфортний для вас бюджет - від 1000 р. до 9900 р. і тривалість - 2, 4 або 8 тижнів

**команда фахівців**  
Ми розуміємо, наскільки важливим є індивідуальний підхід в заняттях спортом. Тому з вами, крім тренера, будуть працювати лікар і дієтолог, щоб підібрати комплексні рішення для вашого організму.

**Ваша мета - струнке і сексуальне тіло або значна м'язова маса? З персональним тренером ви досягнете мети швидше і безпечніше**

[ВИБРАТИ ПАКЕТ ТРЕНУВАНЬ](#)

Рисунок 1.6 - Персональний тренер

Авторизувавшись на сайті, користувач отримує доступ до персонального кабінету, а також можливість вести особистий блог і спостерігати за змінами в персональному щоденнику.

Якщо оцінювати цей вебзастосунок зі сторони функціоналу, можна з впевненістю сказати, що такий приклад є одним з найкращих у своїй сфері. Він не тільки надає всі необхідні дані, зважаючи на різні критерії, але й дає можливість провести якісну роботу з людьми, які знають свою справу.

Розглянутий приклад використовує систему маркетплейсу, де багато кваліфікованих тренерів і дієтологів пропонують свої послуги. Інтерфейс та



дизайн виконаний в приємній для ока манері та за сучасними мірками. На сайті, як і в попередньому випадку, діє система підписки, при якій користувач має доступ до всіх програм тренувань та дієт. З мінусів можна виділити недолік в підрозділі калькуляторів, які не надають максимально точних обрахунків, як в прикладі, який буде розглянуто пізніше, а також в деяких моментах слабка робота адміністрації сайту. Сайт також має свій мобільний застосунок, з яким користувач може отримати доступ до функціонала сайту навіть під час тренувань в залі.

Останній аналог, який розглянуто є сайт: «Бережи фігуру» (<https://beregifiguru.ru/>) (рис. 1.7).

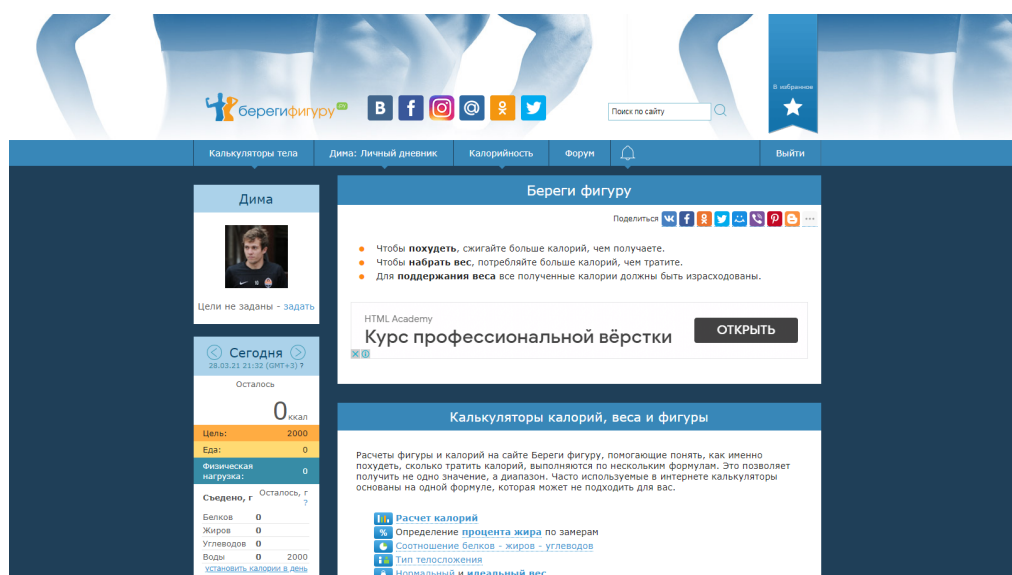


Рисунок 1.7 - Бережи фігуру

Оскільки згаданий приклад є найбільш приближений до тематики роботи, давайте детальніше зупинимось на функціоналі сайту. На сайті присутні 4 розділи, а саме: калькулятор тіла, особистий щоденник, калорійність та форум, розглянемо кожен із них.

В калькуляторі тіла представленні велика кількість різних калькуляторів, які допоможуть користувачеві зробити необхідні обрахунки. Цей розділ у свою чергу поділяється на: калькулятори вживання, калькулятори ваги, калькулятори пропорцій та фітнес калькулятори. Окрім того, що кожен

калькулятор надає можливість зробити відповідні розрахунки, користувач також може отримати доступ і до додаткових даних відносно того, що він вибрав.

Розглянемо калькулятор для розрахунку калорій, де користувач може побачити короткий опис даного функціоналу та свій розклад калорій на кожен день (рис.1.8).

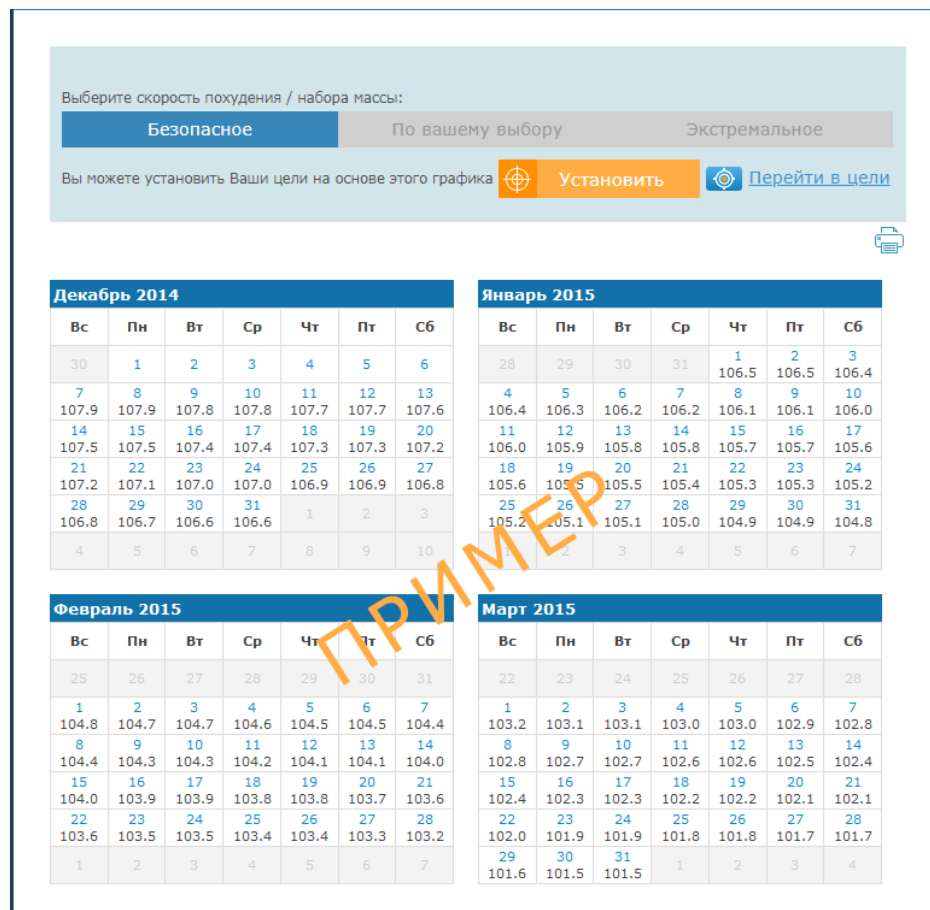


Рисунок1.8 - Власний розклад тренувань

Також представлено калькулятор, а далі уже все інше (рис.1.8). А саме вибрати, якому персональному сценарію вживання калорій користувач надає перевагу і надалі скласти свою програму.

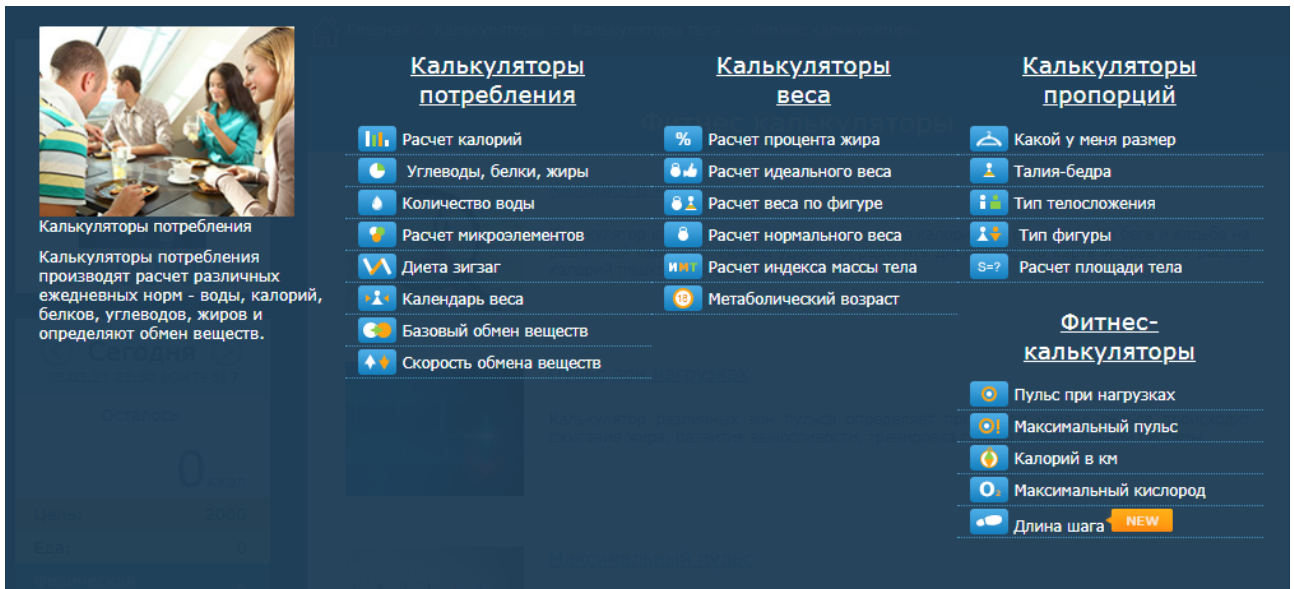


Рисунок 1.9 - Калькулятори тіла

Далі користувач може вибрати середній розкид і формули, за якими він буде робити обрахунок. Перерахунок всіх доступних калькуляторів можна побачити на (рис. 1.9).

Користувач може авторизуватись на сайті та отримати доступ до функцій сайту, який містить в собі розділ особистий щоденник. Авторизація на сайті досить швидка, користувач може зареєструватись, як з допомогою особистого e-mail, так з допомогою сторонніх сервісів, наприклад Google.

В особистому щоденнику користувач може вводити свої персональні дані, слідкувати за прогресом своїх тренувань і отримувати необхідну статистику для того, щоб вміло коригувати свою роботу і досягти своєї цілі максимально швидко. Розділ поділений на: власне особистий щоденник, щоденник харчування, щоденник вправ та результати. Перелік всього функціоналу можна побачити на (рис. 1.10).

В наступному розділі подана можливість обраховувати свою загальну калорійність. Він охоплює обрахунок калорійності продуктів, а також вирахування втрачених калорій при виконанні фізичних навантажень. Так в розділі поданні калорійність не тільки окремих продуктів, але цілих страв, що в рази полегшує обрахунки.

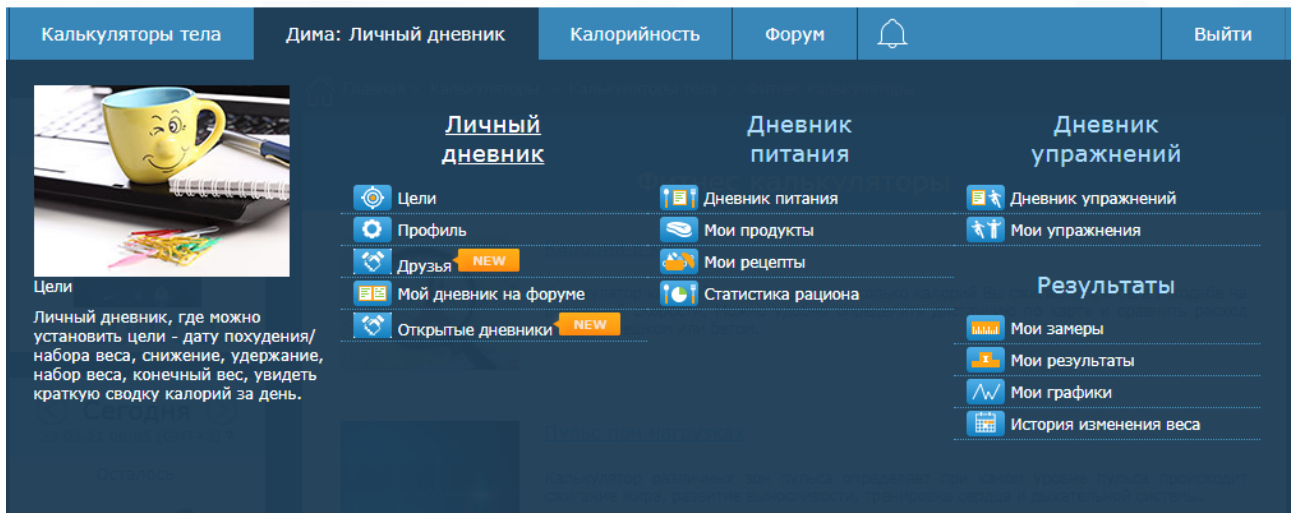


Рисунок1.10 - Особистий щоденник

На сайті представлена величезна база даних різноманітних продуктів в якій користувач може знайти не тільки калорійність конкретного продукту, але й вміст білків, жирів та вуглеводів, що допоможе правильно розраховувати денну пропорційність БЖВ. Вміст розділу можна побачити на (рис.1.11).

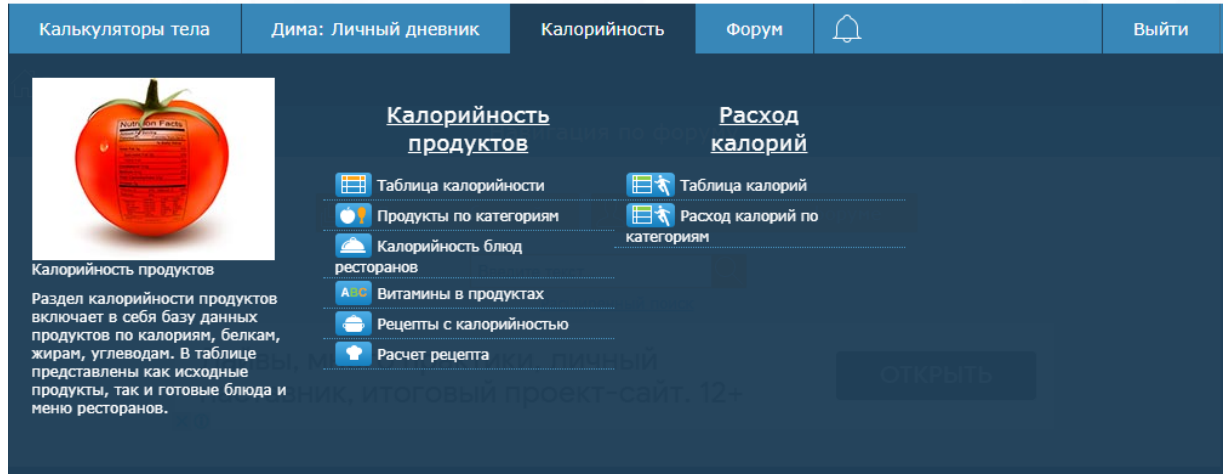


Рисунок1.11 - Калорійність

Кінцевим розділом є форум. Тут доволі все просто, переходячи у цей розділ, користувач потрапляє на сторінку, де є перелік певних тем. Тут він може комунікувати з іншими користувачами сайту і задавати необхідні для нього питання. Вигляд розділу «Форум» поданий на рисунку (рис.1.12).

**Форум**

---

**Дневники**

Личные дневники  
Личные дневники 2407 тем 39887 сообщ. [ANNA](#)  
[ANNA](#), 28 марта 2021 г. 20:19

---

**Консультации**

Консультации тренеров  
Вопросы по физической нагрузке, питанию, тренировкам 86 тем 522 сообщ. [Отправим Ваше коммерческое предложение владельцам/администраторам более 800 000 сайтов!](#)  
[Евгений](#), 26 марта 2021 г. 17:39

---

**Идеальная фигура**

Будем худеть  
Для тех, кто только собирается 254 тем 16779 сообщ. [Марафон стройности 2020](#)  
[Алена](#), 28 марта 2021 г. 4:39

Худеем  
Обсуждаем проблемы, делимся успехами 222 тем 2319 сообщ. [Как вы худеете?](#)  
[marmelad](#), 27 марта 2021 г. 16:06

Жизнь после похудения  
Вопросы поддержки веса, питания, фигуры 65 тем 960 сообщ. [О чем мечтаете?](#)  
[Cherry](#), 22 марта 2021 г. 10:31

Спорт  
О влиянии спорта на вес и фигуру 144 тем 1736 сообщ. [спорт для ленивых](#)  
[marmelad](#), 27 марта 2021 г. 16:07

Набор веса  
Об увеличении веса и наборе массы 33 тем 392 сообщ. [Вода](#)  
[Ольга](#), 18 марта 2021 г. 13:31

---

**Сайт**

Ваши вопросы и пожелания  
Обсуждение работы сайта, проблемы, пожелания 276 тем 2371 сообщ. [ВОДА](#)  
[incognito](#), 28 марта 2021 г. 14:11

Калькуляторы  
Вопросы по работе калькуляторов, разработке новых калькуляторов и их улучшения 45 тем 356 сообщ. [Интерес к работе](#)  
[marmelad](#), 27 марта 2021 г. 16:08

Рисунок 1.12 - Форум

Плюсами даного сайту однозначно є різноманітний функціонал, де користувач не тільки може робити необхідні розрахунки за допомогою наданих йому калькуляторів, а також може вибрати, яку формулу при цьому буде використовувати. Сайт надає план калорій в залежності, що потрібно: набір маси, збереження фігури чи схуднення. І такі плани надаються не тільки для калькуляторів по розрахунку калорій, а й для всіх інших. Інструментів достатньо, лишається тільки ними користуватися. Сайт при цьому є абсолютно безплатним, тому кожен охочий може ним скористатися, проте на сайті присутня реклама. На відмінно від попередніх аналогів на сайті, окрім інструкцій, немає різноманіття корисних статей. Щодо UI та UX немає, що й додати, сайт досить простий, хоча, можливо, це навіть плюс. Також на сайті

працює система сповіщень, яка тримає користувачів в курсі останніх новин.

### 1.3 Постановка задачі

Отже, із зазначеного вище можна зробити висновки, як має виглядати майбутній проєкт.

Вебзастосунок повинен підтримуватись на більшості сучасних пристроїв з різною розмірністю екрана:

- iPhone X 375 на 812 px;
- iPad 768 на 1024 px;
- iPhone 6/7/8 Plus 414 на 736 px;
- iPad Pro 1024 на 1366 px;
- Galaxy S5 360 на 640 px;
- Браузерна версія екрану з понад 1440 px в ширину.

У застосунку повинна бути реалізована система обрахунку необхідних калорій, БЖУ, води тощо.

Користувач повинен мати можливість зберігати дані для трекінгу своїх результатів і кращого розуміння в досягненні заданої цілі.

Для UI потрібно використати адаптивну верстку на базі CSS. Цього можливо буде досягти за допомогою media-запитів, а також використовуючи таку методологію блочного позиціонування, як Flex. Слід зазначити, що головна навігаційна панель при певному розширенні буде перетворюватись у так званий бургер.

Форма для введення даних буде реалізована за допомогою тегів HTML, далі JS буде приймати інформацію, обраховувати за певною формулою та виводити результат на сторінці. При умові, якщо користувач авторизований на сторінці, він отримає можливість зберігати ці дані, натиснувши відповідну кнопку і відіславши POST-запит на сервер.

Для реалізації такої системи буде використовуватись база даних MongoDB,

як місце, де будуть зберігатись усі персональні дані користувача. За допомогою різних HTTP-запитів користувач зможе взаємодіяти з сервером. GET-запит буде відображати сторінки сайту в браузері та відображати збережену в базі інформацію. За допомогою POST-запитів користувач зможе відсилати інформацію на сайт та зберігати її там. PUT та PATCH нададуть змогу модифікувати наявну інформацію, а DELETE - видаляти її.

## **Висновки до розділу 1**

В розділі було зроблено огляд предметної області, визначено який функціонал повинні носити в собі сучасні вебзастосунки в цілому та для даної тематики. Був зроблений аналіз наявних на ринку аналогів у сфері спорту та фітнесу, описано як сильні, так і слабкі сторони кожного з них. Розроблено план для виконання поставленої задачі та методи для його досягнення.

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ФІТНЕС ЗАСТОСУНКУ

### 2.1 Огляд сфери розробки ПЗ

Зараз практично всі системи менеджменту побудовані по принципу Agile. Також ще можна зустріти такі методології як: Kanban та Scrum.

Якщо розглядати Agile, то перше щоб впадає в око, те що ця методологія не опирається на завчасно підготовлену документацію. Принцип орієнтується на постійну зміну умов внутрішнього та зовнішнього середовища розробки та також враховується feedback від замовників і клієнтів.

В Scrum методології над проєктом працює універсальна команда спеціалістів, в яку входить Scrum-owner та Scrum-master. Якщо перший відповідає за зворотний зв'язок з клієнтом і слідкує чи все добре (такий собі supervisor), то другий керує процесом розробки, вирішує проблеми, проводить загальні зібрання (так званні Sprint-и), мотивує команду до роботи та слідкує, щоб всі підходи відбувалися правильно.

Суть Kanban полягає у збалансуванні роботи команди, яка працює над проєктом. Там немає такого, що розробники сидять без діла, а дизайнери працюють без перерв. Вся команда єдина і немає ніяких ролей на відміну від Scrum і тому якщо виникає якась проблема, то її вирішують всі.

Весь процес розробки програмного забезпечення можна розбити на декілька етапів (SDLC), їх можна побачити на рис. 2.1.

До кожного етапу властива своя документація. До стадії планування властиві High-level documentation та Low-level documentation, для стадії тестування - Test plans та Standarts і Test documentation. До User documentation відносяться System documentation, End-user-guides та Installation guides. Ну а вже на релізі додаються Final reports.



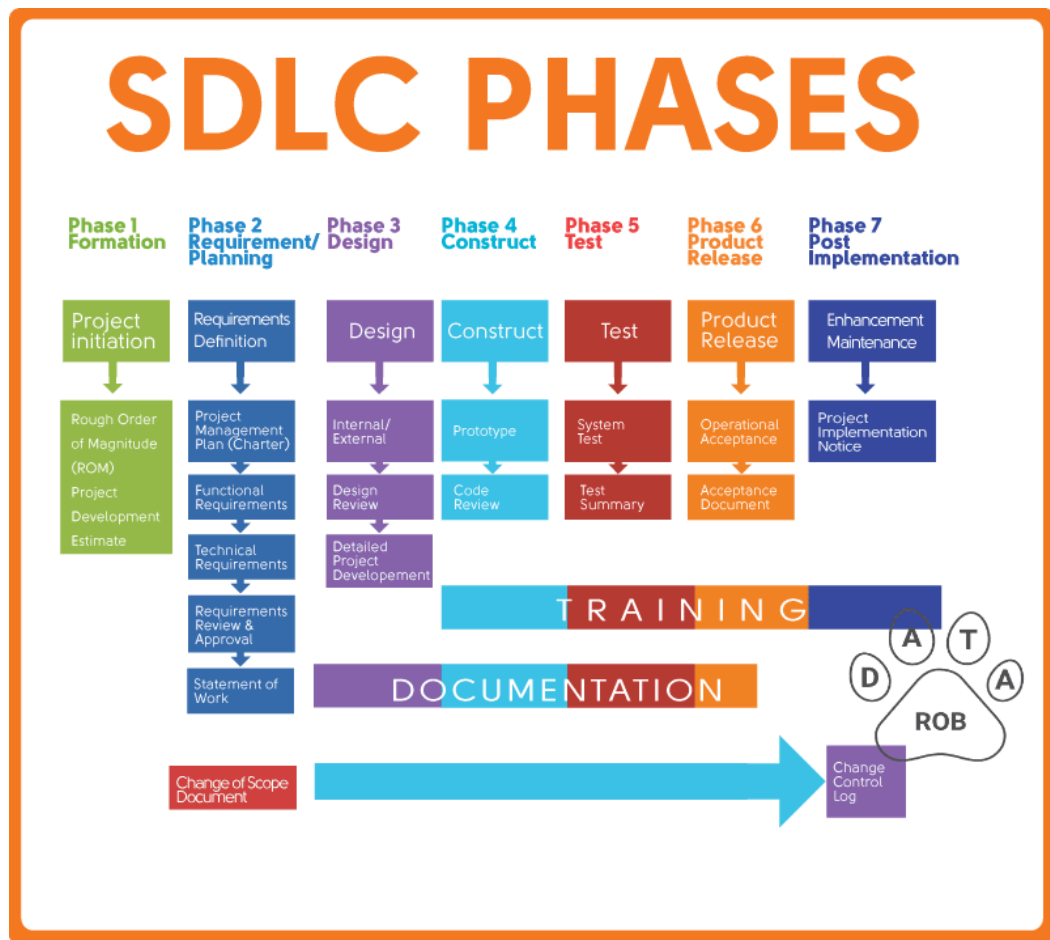


Рисунок 2.1 - SDLC

## 2.2 Конструювання власного ПЗ

Перейдемо безпосередньо до розробки нашого проєкту. В даному розділі не буде загострюватись увагу на документуванні усіх етапів розробки життєвого циклу ПЗ, а буде зроблено лише High-level documentation, яке містить в собі всі Еріс-и і дає приблизне уявлення, про те як буде виглядати проєкт. Дано визначення High-level documentation (табл. 2.1).

- Короткий опис та назва кожного модуля;
- Кожен модуль повинен мати опис функціоналу;
- Відношення та залежність між модулями;
- Визначення бази даних і її ключові елементи;
- Повноцінна архітектура діаграм з відповідними деталями.

Перейдемо до опису системи, для початку розділимо проєкт на 3 модулі та дамо їм опис. Після чого визначимо функціонал кожного модуля та дамо їм визначення. Останнім етапом буде опис залежностей між модулями (табл. 2.2).  
Короткий опис системи та модулів:

YourFit - це платформа для підрахунку калорій і ведення особистих даних.

Калькулятори тіла - можливість вибрати необхідний калькулятор для ініціалізації обрахунків. Калькулятори можуть бути поділені по категоріях.

Профіль - можливість кожного користувача при реєстрації отримати доступ до власного профілю, можливість вносити і зберігати дані (вік, стать, дата народження).

Особистий кабінет - дає можливість зберігати обрахунки в розділі калькуляторів, вводити кількість калорій, які були спожиті протягом дня і відстежування змін та підрахунок тижневої кількості спожитих калорій.

Таблиця 2.1

Короткий опис функціоналу модулів

<b>Назва модуля</b>	<b>Назва Функціоналу</b>	<b>Опис Функціоналу</b>
Калькулятори тіла	Денна норма калорій	Можливість розрахувати денну норму калорій
	Норма БЖВ	Можливість розрахувати денну норму БЖВ
	Кількість води	Можливість розрахувати денну норму води

	Максимальний пульс	Можливість розрахувати свій максимальний пульс при різних навантаженнях
	Процент жиру	Можливість вирахувати свій процент жиру
Профіль	Реєстрація	Можливість зареєструватись
	Авторизація	Можливість авторизуватись
	Особистий кабінет	Доступ до особистого кабінету
	Профіль	Доступ до профілю в якому можна ввести свої особисті дані(вік, стать, ім'я, дату народження, фото)
Особистий кабінет	Мої розрахунки	Можливість зберігати свої розрахунки в розділі калькулятори тіла на сайті
	Щоденник калорій	Можливість вводити кількість своїх спожитих калорій протягом дня, власноруч відстежувати зміни та отримувати тижневу кількість калорій
	Мої дані	Можливість вводити і зберігати

		свої виміри , ріст , процент жиру , бажану вагу, поточну вагу і можливість оновлювати її протягом часу.
--	--	---

Таблиця 2.2

Опис залежностей між модулями

Назва модуля	Назва залежного модуля	Залежність
Калькулятори тіла	Профіль	Переведення обрахунків до особистого кабінету
	Особистий кабінет	Збереження даних в особистому кабінеті
Профіль	Калькулятори тіла	Відкриває можливість для збереження даних
	Особистий кабінет	Відкриває доступ до особистого кабінету
Особистий кабінет	Профіль	З'являється доступ до кабінету після реєстрації
	Калькулятори тіла	Отримує дані з обрахунків, зберігає їх та модифікує по можливості

Перейдемо до DB diagram. В цьому випадку не буде використовуватись система реляційних таблиць в SQL, натомість буде застосовуватись MongoDB, яка використовує колекції для зберігання даних. Зв'язок між таблицю буде один

до багатьох. Головною таблицею буде верифікації, яка з'єднає в собі всі інші таблиці та по якій будемо знаходити всю іншу інформацію кінцевого користувача (рис. 2.2).

Відразу від бази даних перейдемо до Activity diagram, але для початку дамо визначення цієї діаграми. Activity diagram є ще одною важливою поведінковою UML діаграмою, яка описує динамічні аспекти системи. В представленому випадку діаграма поділена на 3 модулі, так само як це було показано в таблиці вище (рис. 2.3).

Принципи Activity diagram:

- Ідентифікує акторів проєкту та їхню взаємодію з системою;
- Моделює взаємодію між Use case;
- Моделює комплексну схему операцій над об'єктами;
- Моделює в деталях поведінку.

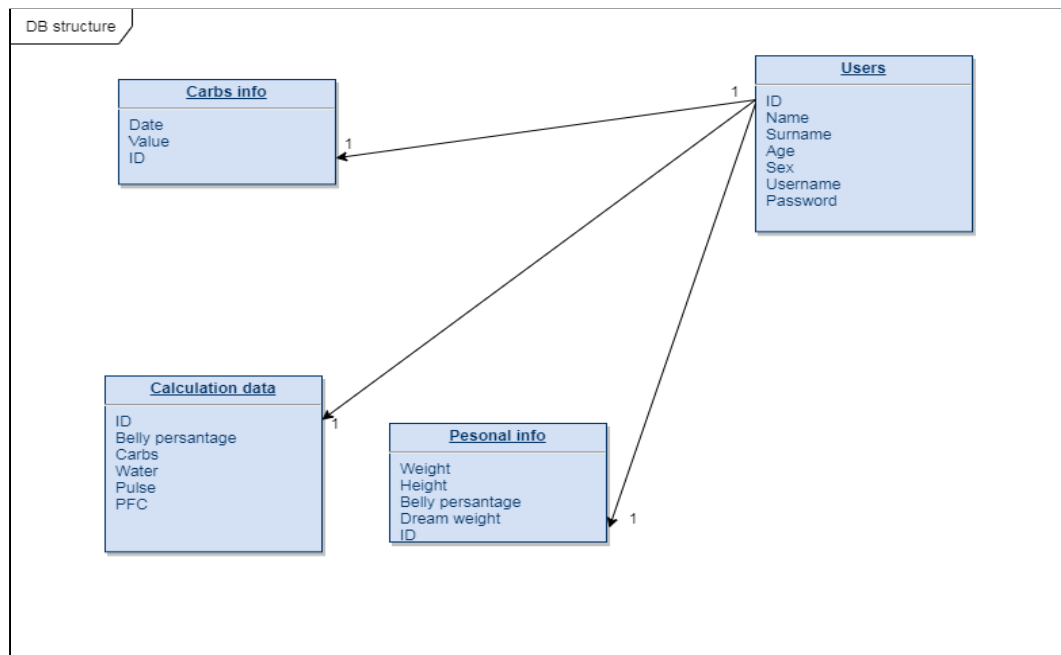


Рисунок 2.2 - Діаграма баз даних

На рисунку 2.3 представлено Activity diagram.

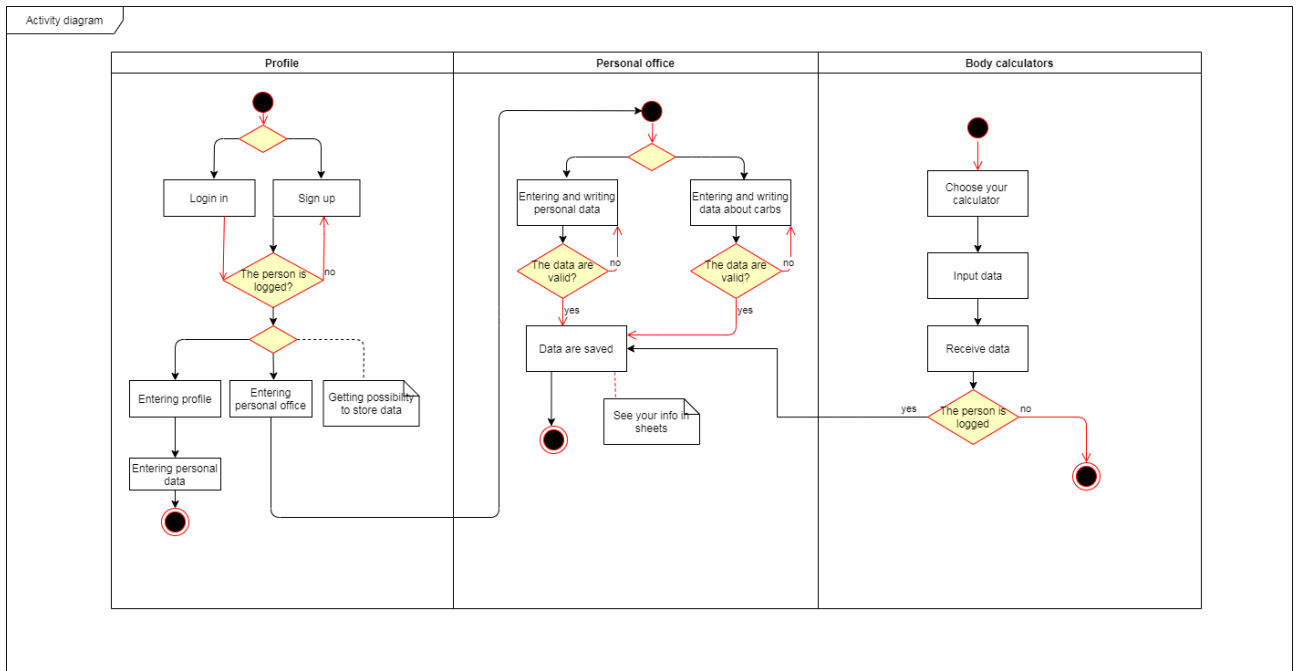
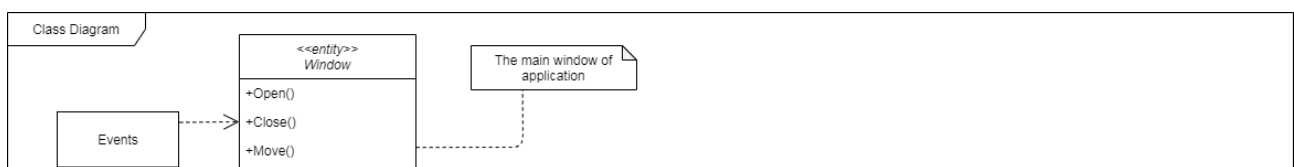


Рисунок 2.3 - Activity diagram

Далі по списку іде діаграма класів, в якій було зображено візуальну частину проєкту (UI), в ролі класів виступають головні об'єкти інтерфейсу (рис. 2.4). UML Class diagram буде використовуватись, як графічна схема, яка побудована за принципом об'єктно-орієнтованої системи. Класова діаграма відноситься за типом до статично-структурних діаграм, які описують систему показуючи:

- Класи;
- Їхні атрибути;
- Операції або методи;
- Відношення між об'єктами;

Класи можна сприймати як макет для об'єктів. Класи та об'єкти співпрацюють між собою. Не можливо розділяти ці два поняття один від одного. Вся суть об'єктно-орієнтованого дизайну не в об'єктах, а в класах, тому що ми використовуємо класи для створення об'єктів. Тож класи - це сутність, яка використовується до опису об'єкту, проте не являє собою як об'єкт. Сам об'єкт уже містить всю властиву йому інформацію.



## Рисунок 2.4 - Class diagram

Далі по списку іде State machine diagram (рис 2.5). UML state machine diagram показує різноманітні стани сутності ПЗ. Також діаграма показує, як певна сутність відповідає на певні події, міняючись від одного стану до іншого. Ця діаграма дасть можливість зрозуміти, як система взаємодіє з користувачем і це дасть краще уявлення при побудові RESTfull API.

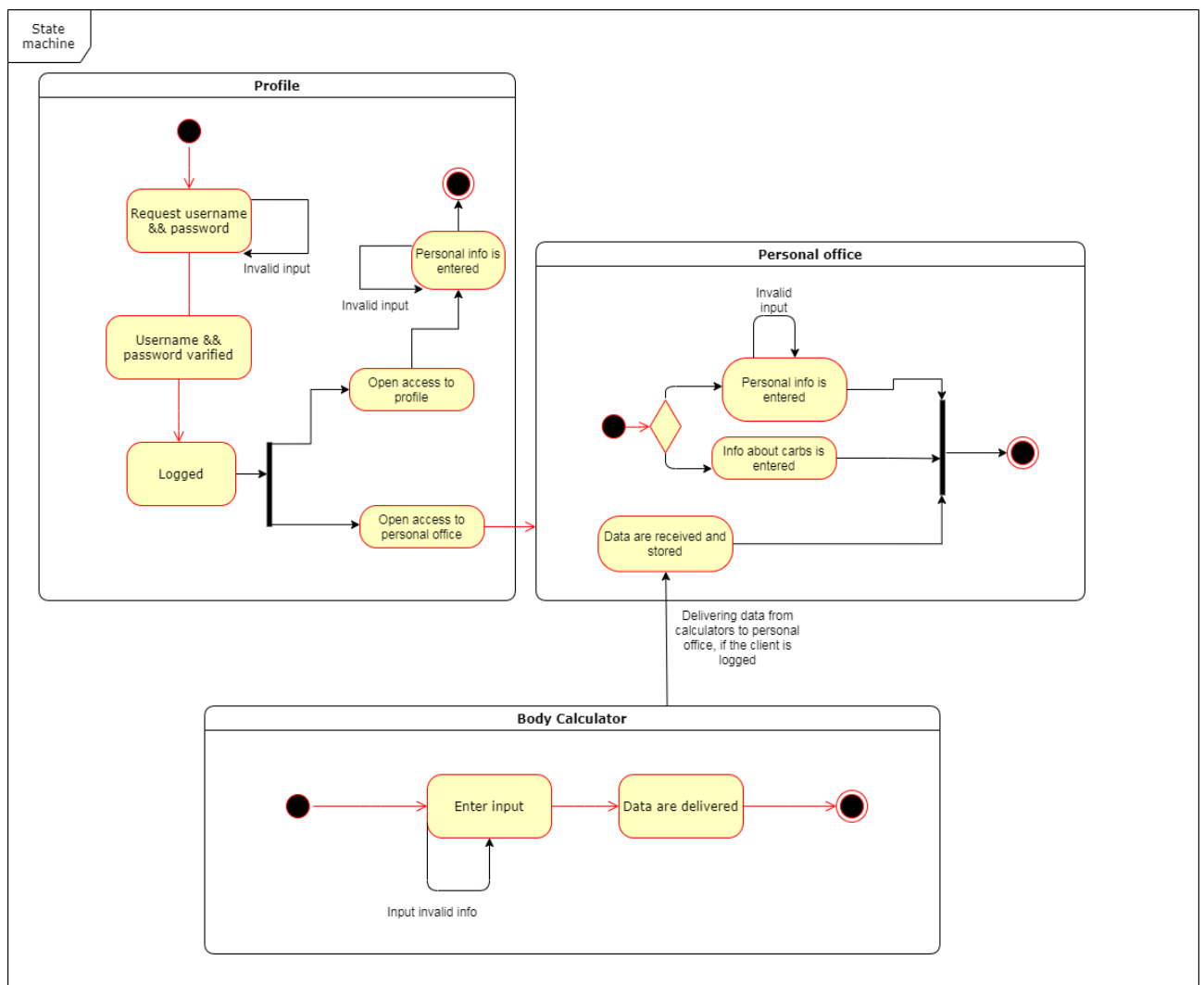


Рисунок 2.5 - State machine diagram

Наступною діаграмою для розгляду буде Use case diagram. UML use case diagram є початковою формою вимог системи/програми для створення нового програмного забезпечення. Use case показує очікувану поведінку (what) і метод який виконує цю поведінку (how). Ключовим концептом use case моделювання є

те, що воно допомагає спроектувати систему з ретроспективи кінцевого користувача. Зазвичай Use case diagram досить проста, вона не показує всіх деталей use case (рис 2.6):

Вона тільки сумує певні взаємодії між cases, actors і системою

Вона не показує спосіб досягнення певних цілей кожного випадку

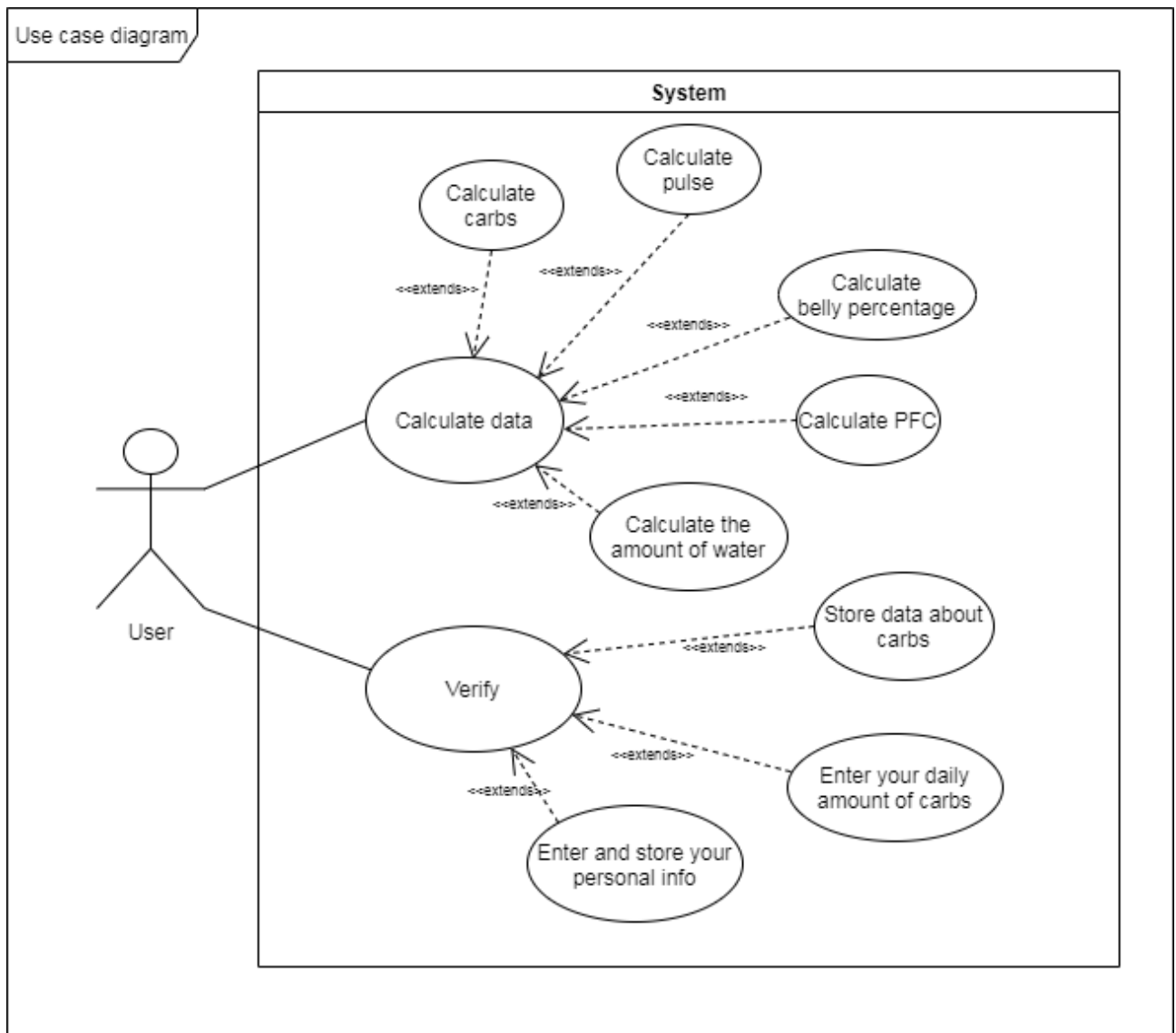


Рисунок2.6 - Use case diagram

Маючи приблизне уявлення про те, як буде виглядати проект можна зробити макет сайту, а саме wireframe. Чому саме wireframe? Даний сайт не вирізняється якимось особливим зовнішнім виглядом, головне це взаємодія користувача з сервером, тому в цьому випадку wireframe підходить краще ніж



mock-up або prototype. Wireframe-и будуть побудовані за допомогою софту Balsamiq. Для проєкту було спроектовано усі сторінки включаючи, як авторизовані версії та і неавторизовані. Для початку розглянемо головну сторінку проєкту (рис. 2.7).

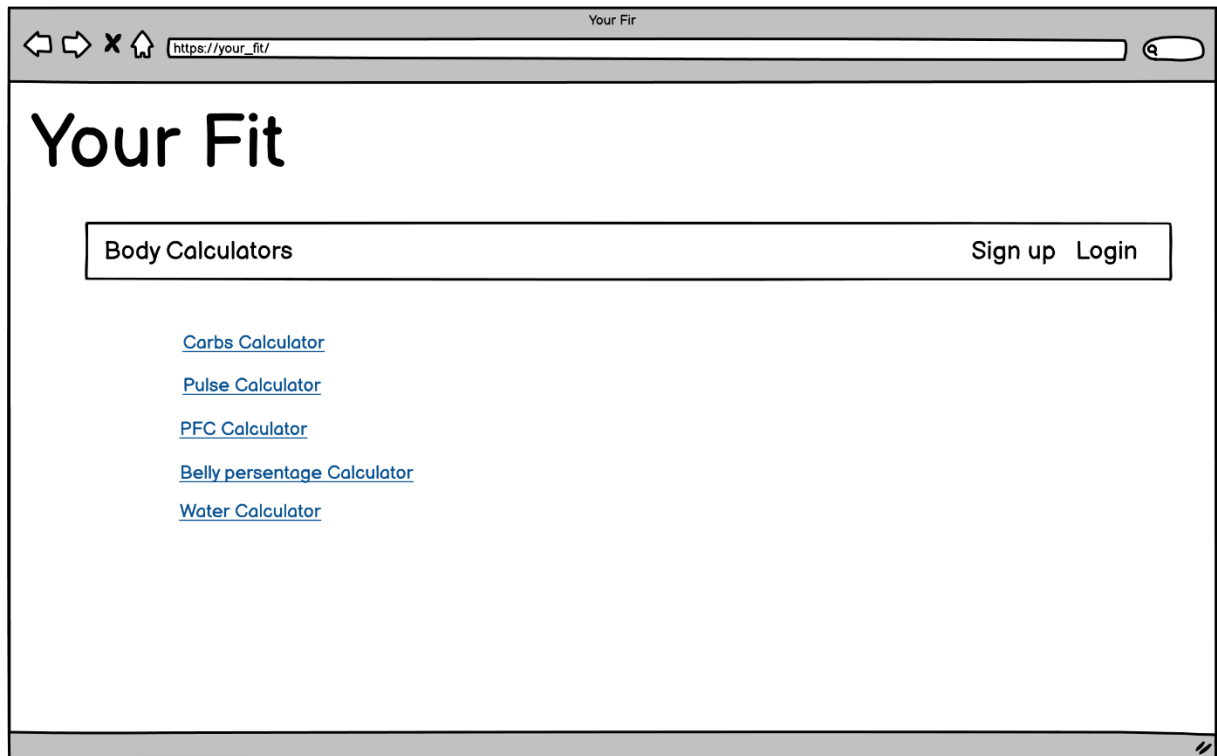


Рисунок 2.7 - Головна сторінка сайту

Все що, можливо на цю мить, поки користувач не авторизований це перейти в один з калькуляторів, та власне зареєструватись на сайті. Перейдемо в один з калькуляторів і оцінимо функціонал (рис. 2.8).

Кожен калькулятор буде в схожому стилі, тобто користувач зможе ввести та обрахувати потрібні дані. Також в даній вкладці буде короткий опис функціоналу для того, щоб користувач орієнтувався, що йому вводити. В залежності від калькулятора на сторінці можуть бути інші елементи. Зареєструємось на сайті або авторизуємось, якщо аккаунт уже присутній та поглянемо що зміниться.

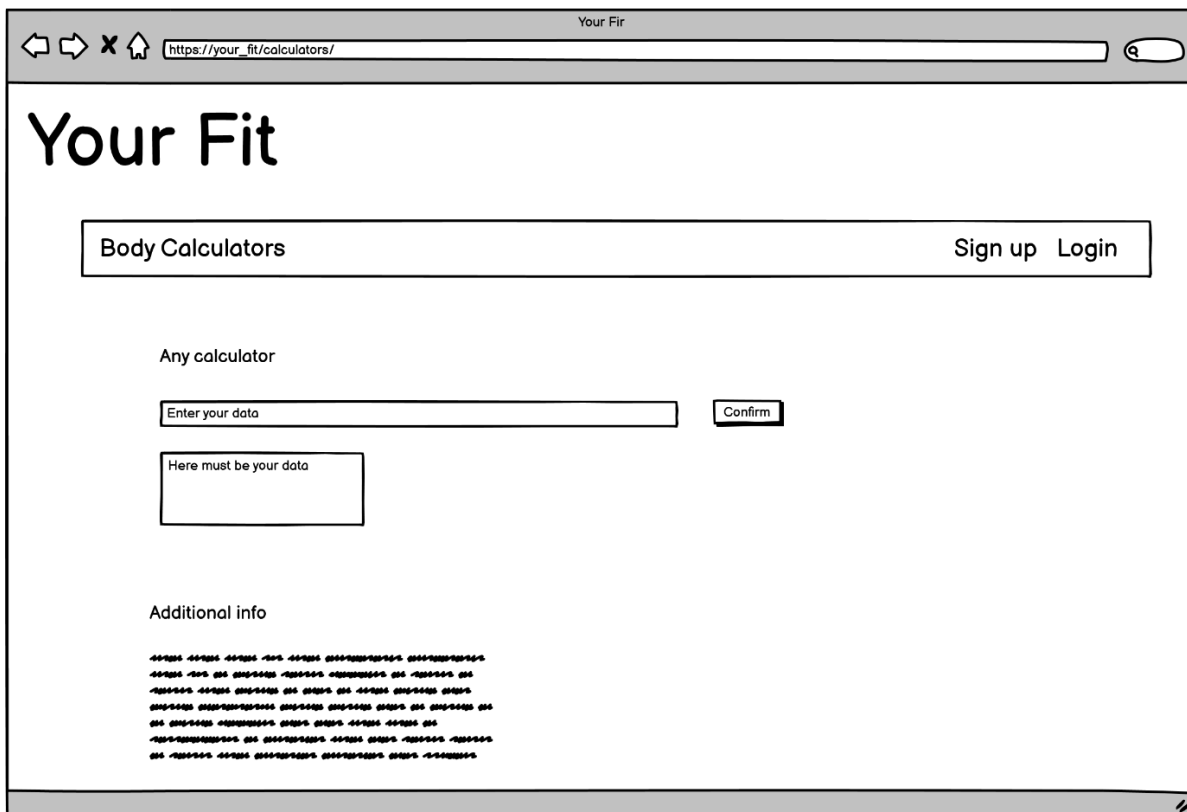
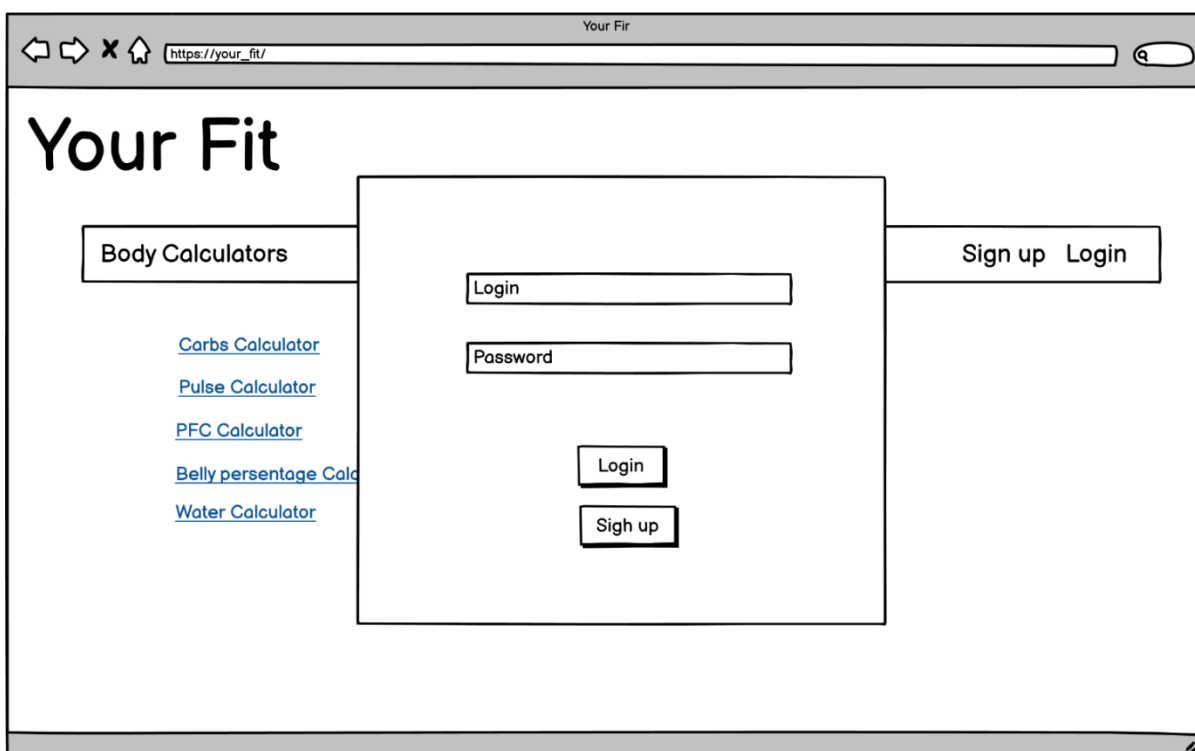


Рисунок 2.8 - Сторінка калькуляторів

Для користувача буде відкрита можливість зареєструватись на сайті вводячи пароль та логін (рис. 2.9).



## Рисунок 2.9 - Сторінка авторизації

Якщо користувач уже до цього вводив свої дані, він зможе швидко і без проблем авторизуватись і також вийти з нього не боячись, що його персональні дані пропадуть.

Як можна побачити, в навігації з'явилась вкладка персонального офісу, яка дає доступ до особистого кабінету користувача з подальшими діями, також в навігації кнопки логування та реєстрації змінилися на вкладку профілю, яка у свою чергу дає доступ до особистого профілю користувача (рис. 2.10).

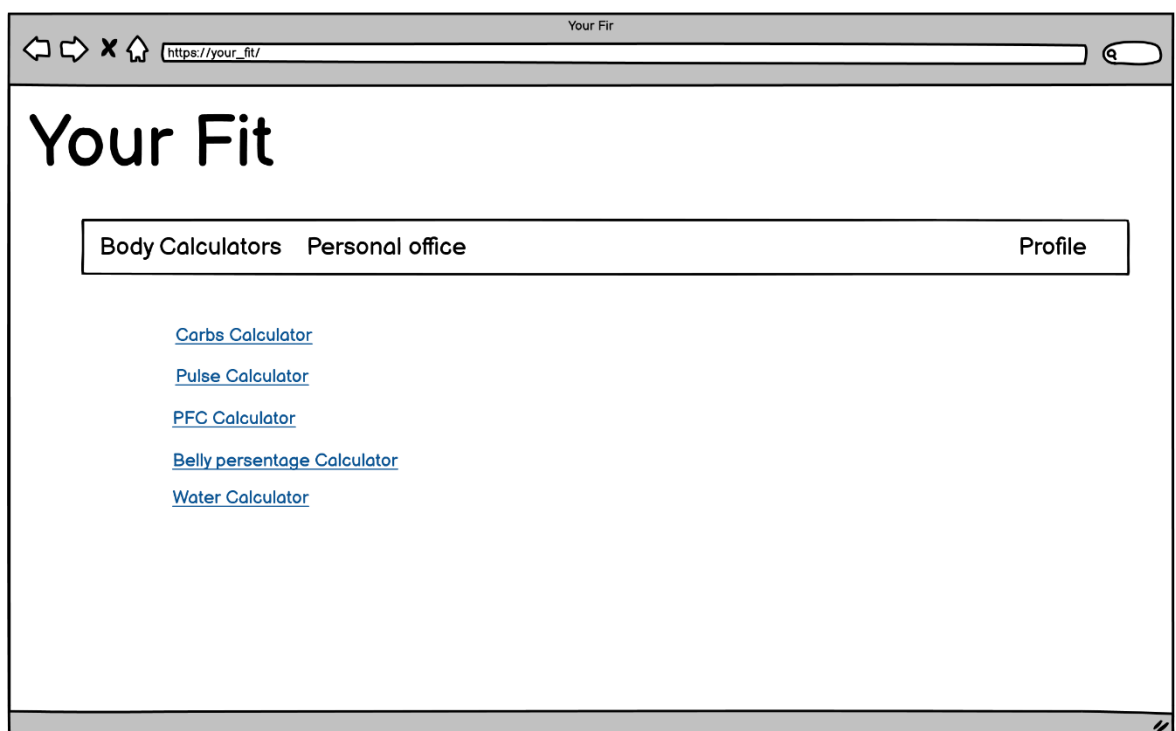


Рисунок 2.10 - Авторизована головна сторінка

Перейдемо в вкладку персонального офісу (рис 2.11). В персональному офісі в блоці «Виміри» користувач зможе ввести свої дані такі як: ріст, вага, бажана вага та процент жиру.

Також в цій вкладці дані, які вводились в розділі калькуляторів зберігаються, тобто є можливість переглянути попередні значення спожитих калорій. Можна вводити кількість калорій, які користувач споживав протягом дня та отримати загальну кількість калорій.

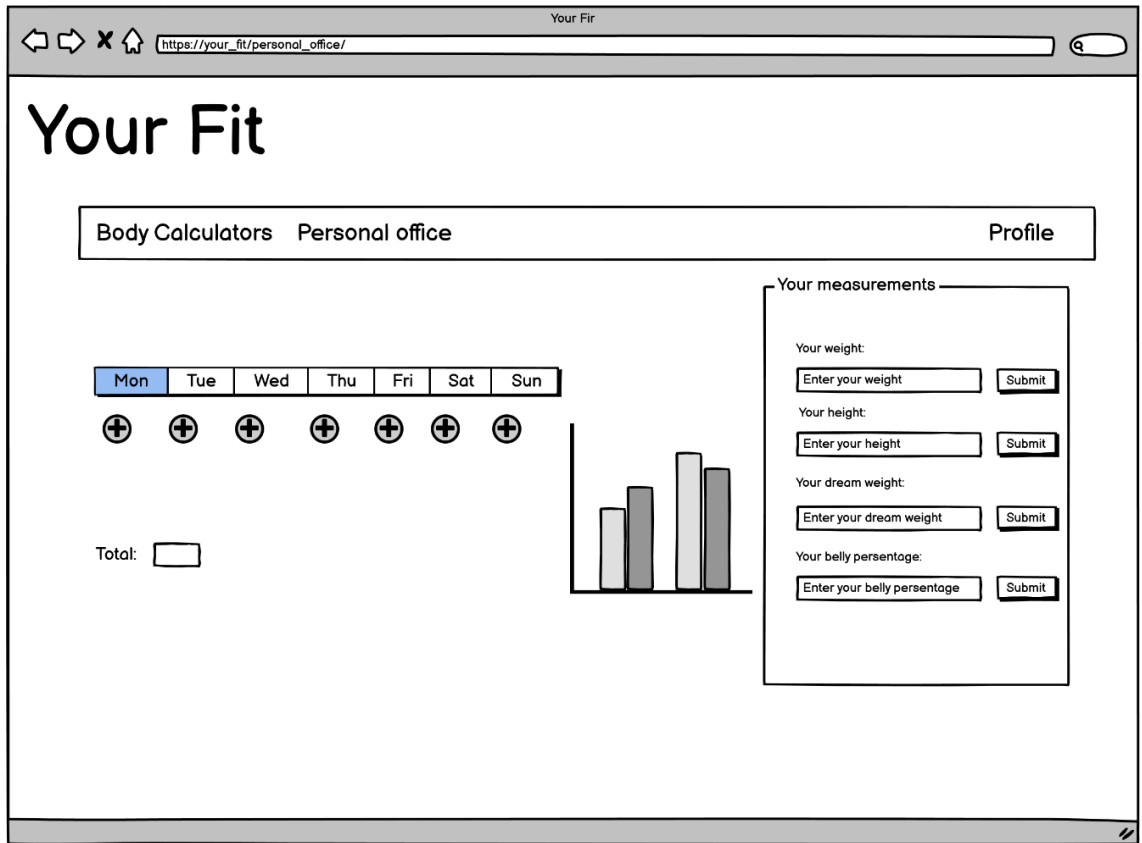


Рисунок 2.11 - Персональний офіс

Перейдемо у вкладку профілю (рис 2.12).

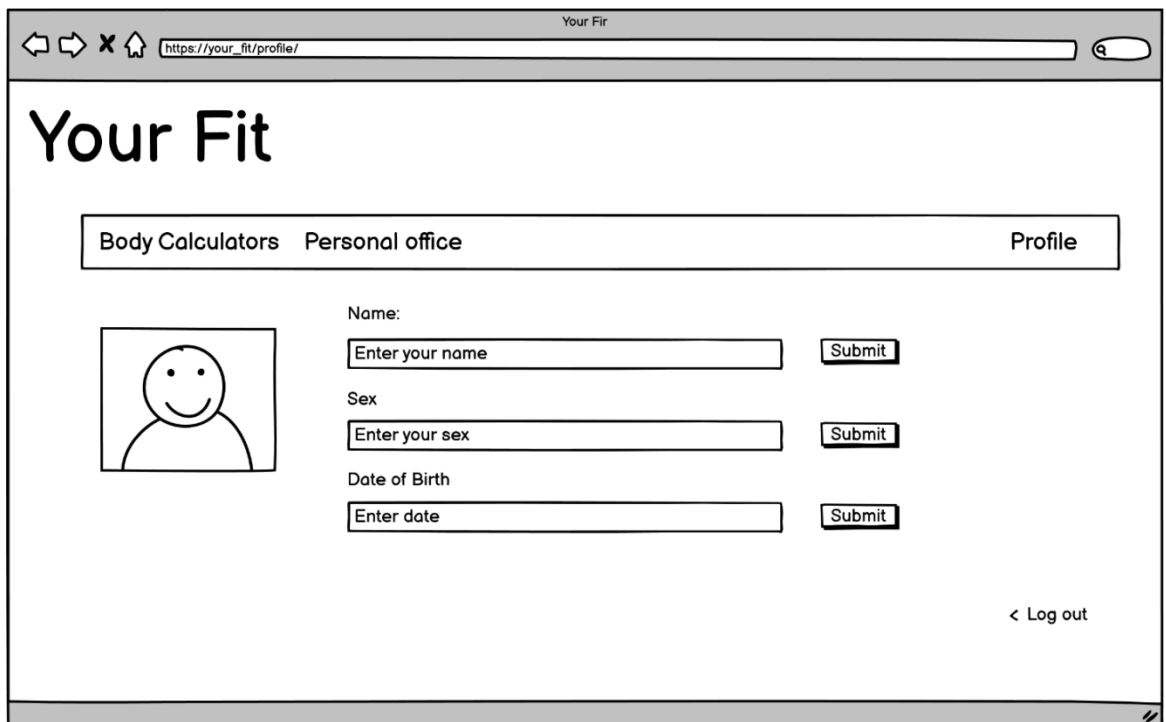


Рисунок 2.12 - Сторінка профілю

В профілі користувач зможе вводити такі особисті дані як: ім'я, стать та дату народження. Також буде можливість завантажити аватар профілю. При потребі тут можна вийти з облікового запису.

З оглядом браузерної версії закінчено, проте для того, щоб сайт був актуальним потрібно надати користувачу версію для смартфона, тож доступимо до огляду.

В деяких випадках будуть вказані стандартні вкладки та вкладки для авторизованих користувачів.

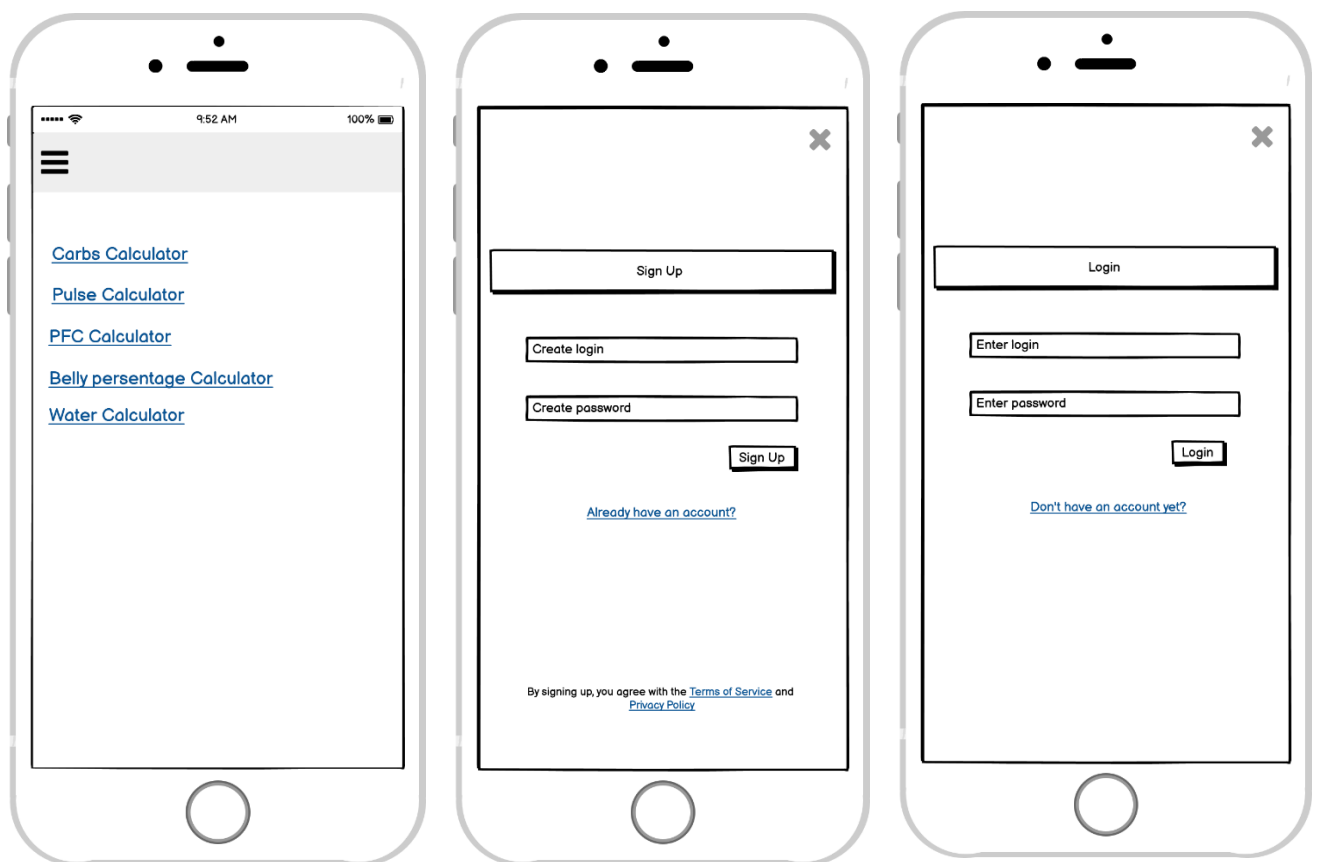


Рисунок 2.13 - Головна сторінка та сторінки верифікації

На перших трьох картинках буде показаний варіант мобільної версії для головної сторінки та сторінок верифікації (рис. 2.13).

Далі представлено варіанти впливаючої навігації (рис. 2.14).



Рисунок 2.14 - Сторінки профілю

Перейдемо до сторінок профілю та калькуляторів (рис. 2.15).

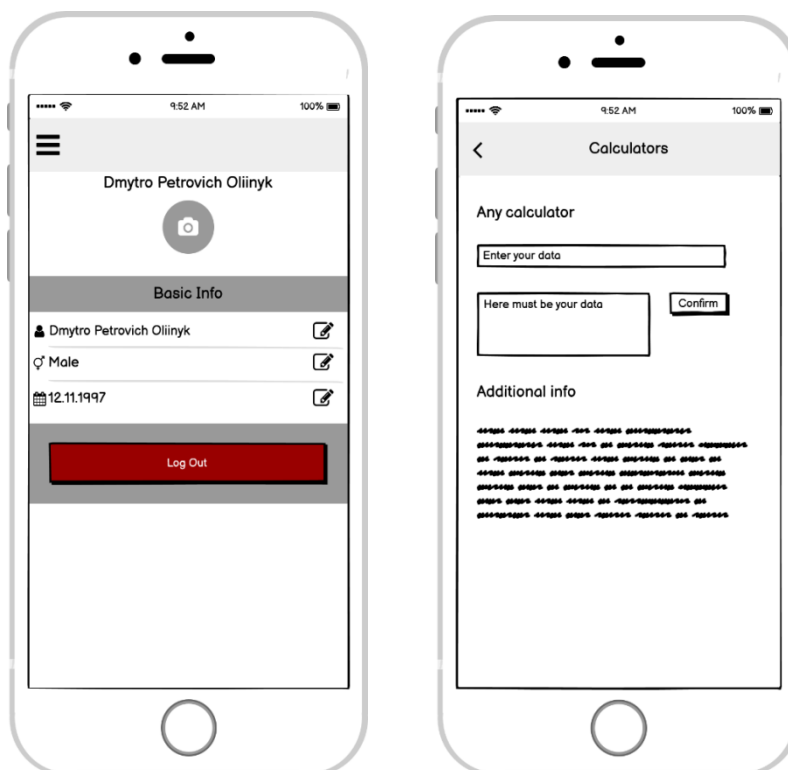


Рисунок 2.15 - Сторінки профілю та калькуляторів

Сторінка персонального офісу представлено на рис. 2.16. Всі макети побудовані для розмірності екрану під iPhone.

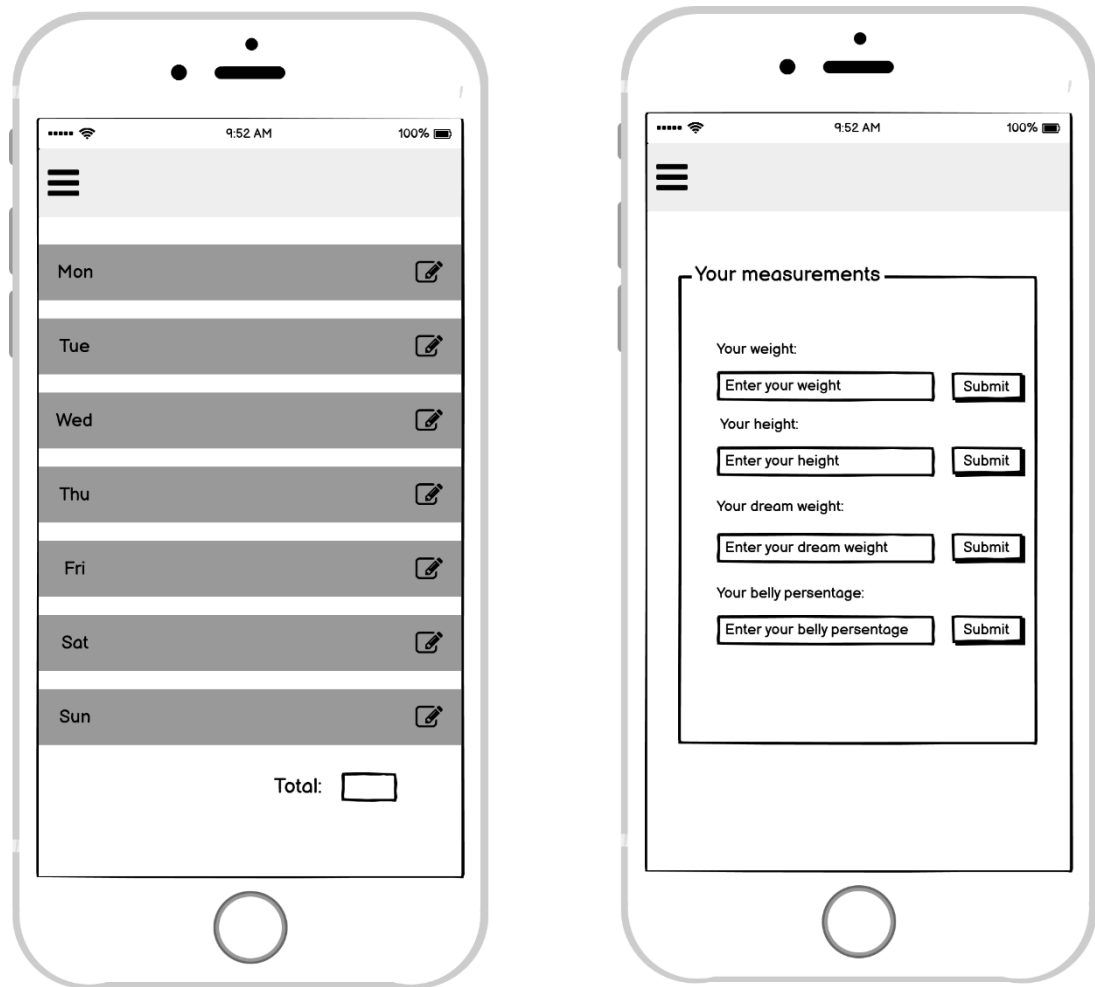


Рисунок 2.16 - Сторінки персонального офісу

На цьому з оглядом макетів закінчено, передивимось і розберемо технології, які будуть використані у проєкті.

### 2.3 Огляд технологій для конструювання ПЗ

Якщо з вибором мови для веброзробки все досить просто, оскільки на зараз у JS просто немає конкурентів. По суті JS є монополістом в цій сфері. Проте на чистому JS уже ніхто не пише, до нього додають різні фреймворки, бібліотеки або конфігурації. В JS є повна інтеграція з html/css, мова досить легка для вивчення і також вона підтримується більшістю браузерів. JS не раз оновлювався на своєму шляху, проте найбільш значущі зміни були внесені з

версією ES6. ES6 був випущений у 2015 році та він додав багато необхідного функціоналу: Let та Const, Arrow Functions, Promises, Modules, Spread Operator, Destructuring, Template Literals та інше. Також слід відзначити ES8, який привніс у світ програмування Async/Await.

Найбільш поширеним доповненням до JS є React, який призначений побудови користувацьких UI. React в разі полегшує створення інтерактивного інтерфейсу та він відрізняється від ванільного JS кращою швидкістю.

Технологія CSS буде використана для стилізації вебсторінки. На відміну від вище озвучених технологій, CSS не є мовою програмування (так само як і HTML), а являє собою мовою стилізації таблиць. Якщо говорити про HTML, то ця мова, буде використана для структуризації та відображення вебсторінки та її контенту. Контент буде відображатись в параграфах, списках, блоках, зображеннях і таблицях даних. У проєкті буде використовуватись синтаксис HTML, проте для іншої технології EJS, про яку буде згадано пізніше.

До цього було розглянуто Front-end архітектуру, перейдемо до Back-end, саме на ньому і буде побудовано проєкт, почнемо з Node.js. Раніше виконувати JS міг тільки браузер, це стало можливо і з Node.js, який в основному використовується для написання серверної частини, проте може бути використаний для написання Desktop-застосунків. За основу Node.js був взятий движок для Google Chrome, який має назву V8, який своєю чергою був відполірований до ідеалу розробниками з Google. Проте це не все, для того, щоб Node.js був повноцінною серверною мовою до нього додали трохи коду, а також можливість роботи з npm. Npm - це свого роду лавка, в яку ми будемо звертатися при потребі та брати необхідний товар (модуль), кожен модуль має свій функціонал. Якщо говорити про переваги Node.js над інших back-end мовами, то node максимально зручний і швидкий за рахунок однопоточності та асинхронності, ну і також він використовує одну і ту ж мову, як для клієнтської, так і для серверної сторони. До Node.js буде додано фреймворк Express для того, щоб спростити створення сервера і серверної логіки та не перевантажувати



проект великим об'ємом коду. А для клієнтської частини буде використовуватись движок представлення EJS для того, щоб UI був динамічнішим. EJS використовує схожий синтаксис, як і інші шаблонізатори такі як: HandleBars або Рас. Краще ознайомлення з сутністю цього движка буде уже на створенні проекту.

## **Висновки до розділу 2**

У розділі проведено аналітичну роботу над проектом. Охарактеризувавши сферу розробки програмного забезпечення, було створено документацію, в якій було розглянуто усі наявні модулі та всі залежності між цими модулями. Далі було описано та створено 5 UML-діаграм, а саме:

DB diagram

Activity diagram

Class diagram

State machine diagram


Use case diagram

Після чого були показані макети для мобільної версій та браузерної версії вебдодатка. І нарешті було розглянутий перелік технологій, які будуть використовуватись для розробки проекту.

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ФІТНЕС ЗАСТОСУНКУ БАЗИ NODE.JS, EXPRESSTAMONGODB

### 3.1 Налаштування робочого середовища

Для того, щоб розпочати проєкт потрібні дві речі: середовище розробки та node.js. В першому випадку було вибрано Visual studio code, який являє собою зручним інструментом для розробки вебзастосунків. Для того, щоб використовувати npm потрібно зайти на офіційний сайт node.js, вибрати відповідну версію і встановити його на комп'ютер. Далі командою в консолі перевіряємо наявність node так, як це показано на (рис. 3.1).



```
C:\Users\classuser52>node -v
v13.14.0
```

Рисунок 3.1 - Команда перевірки версії Node.js

Далі заходимо в VS code, створюємо директорію та ініціалізуємо проєкт командою `npm init`, що додає до проєкту файл `package.json`, в якому знаходиться вся інформація про проєкт. Перший таким модулем буде Express, пропишемо в консолі `npm i express`. Також знадобиться Ejs, для цього вводимо команду в консолі `npm i ejs express-ejs-layouts`, після чого в `package.json` додаються відповідні залежності та створюється файл `package-lock.json`. Переходимо в `package.json` та змінюємо рядок `main` на `server.js`, в якому буде міститися головна логіка сервера, також задаємо необхідні скрипти, щоб сервер запускався за допомогою `nodemon`. Переходимо в `server.js` і підключаємо всі модулі, які були завантажені до цього. Далі встановлюємо всі необхідні маршрути для функціонування ejs. Після цього задаємо проміжний обробник (`middleware`), для статичних файлів, які будуть зберігати в папці `public` і підключаємо слухач (`listen`) на порт 3000 і відтепер можливо запуснути сервер, проте на даний момент він нічого не відобразить. Щоб рухатись далі необхідно побудувати дерево наших директорій, для цього можемо використати патерн MVC (рис.

3.2).

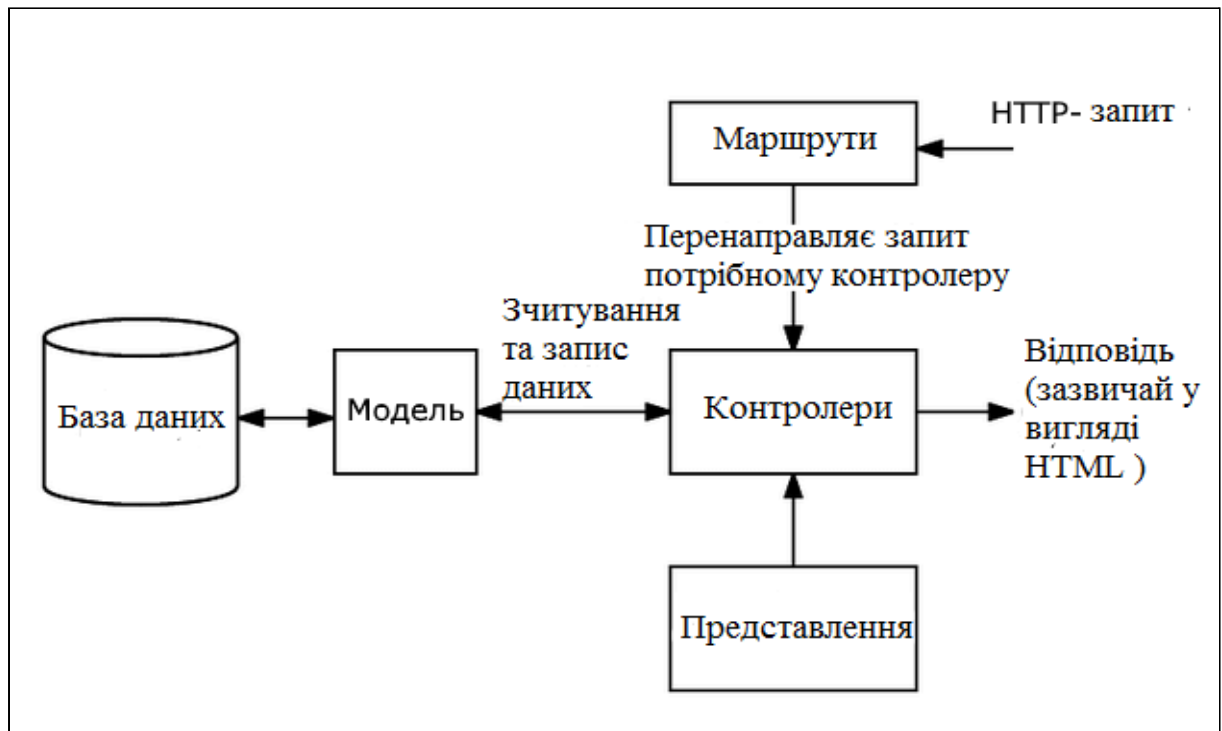


Рисунок 3.2 - MVC

Суть полягає в розділенні коду на model, view та controller. У model зберігається уся логіка, наприклад у моєму випадку я буду зберігати схеми для бази даних. У view - пропишемо усі сторінки, які будемо рендерити в браузері. А controller у свою чергу буде з'єднувати ці дві сутності між собою, такий собі міст між двома берегами, при цьому view та model не повинні взагалі знати один про одного. Така методика дозволяє покращити читабельність коду, що дуже важливо при великих проєктах та полегшить дебагінг коду при виникненні помилок. Controller було перейменовано на routes і в ньому буде задаватись маршрутизація для сторінок. Для кожної сторінки було задано свою папку у view, в яких було створено відповідні сторінки з розширенням ejs та їхні розгалуження. Повернемось до директорії public, яку було створено до цього та додамо директорію stylesheets, в якій будуть всі css-файли. На цьому проєктування основної гілки директорій завершено і на рисунку можна побачити, як виглядає дерево директорій (рис 3.3).

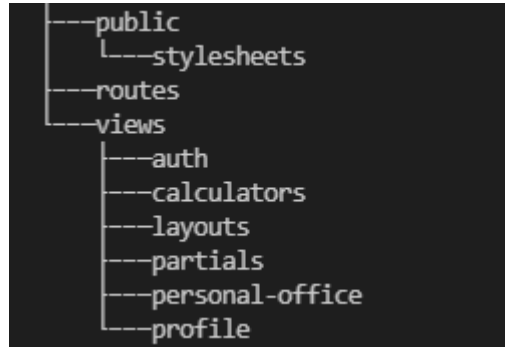


Рисунок 3.3 - Project tree

На даний момент в проєкті є робочий сервер, проте він нічого не відображає, тож для початку створимо головну сторінку проєкту і назвемо його `index.ejs` він буде задаватися як корінний маршрут нашого домену. Щоб сервер зміг його відобразити, в `routes` пропишемо початковий маршрут, для цього потрібно використати `expressrouter`. Далі переходимо в `server.js` і виконуємо всю необхідну логіку. Проте це ще не все оскільки даний файл потрібно помістити в `layouts`. Тож переходимо в даний файл і прописуємо стандартну `html`-розмітку в яку поміщаємо `<%-body%>`. Layout буде своєрідним шаблоном для кожної сторінки, в ньому також можна підключати файли, які повинні відображатись по дефолту на кожній сторінці такі як: `header`, `title` та `footer`. В `body` сервер буде поміщати сторінку, зважаючи на поточний маршрут.

Поки закінчимо з маршрутами, але, звичайно, повернемось до них пізніше. Наступним кроком буде підключення баз даних. Як уже говорилося раніше в проєкті буде застосовано `MongoDb`, яке використовує колекції для зберігання даних. Для початку потрібно перейти на `Atlas` та пройти всі необхідні процедури. `Atlas` - це своєрідний хостинг для баз даних в якому і буде зберігатись база і відповідно всі дані користувачів. Щоб база була інтегрована в код, копіюємо необхідну посилку, міняємо поля `login` та `password` і зберігаємо її в попередньо створений файл `.env`, який захистить наші персональні дані від небажаних очей. Для взаємодії з `MongoDb` необхідно встановити модуль

Mongoose ввівши в термінал команду `npmimongoose`. Переходимо в `server.js` і прописуємо всю логіку.

```
mongoose.connect(process.env.URL,{useNewUrlParser: true,useUnifiedTopology: true }
)
const db = mongoose.connection;
db.on("error", (error) => console.error(error));
db.once("open", () => console.log("Connected to Mongoose"))
app.listen(process.env.PORT || 3000, () => {
  console.log("server started on port:" + 3000);})
```

З базою поки що все, переходимо до Git. Git буде використовуватись як система контролю версії, яка встановлюється локально на комп'ютер та допомагає зберігати код і всі зміни, при цьому не втратити нічого важливого. Для збереження коду на хмарці, перейдемо на хостинг GitHub зареєструємось та створимо репозиторій для збереження проєкту.

Ввівши необхідні команди, які показані на рисунку ініціалізуємо на репозиторій локально підключимо його до віддаленого репозиторію на GitHub та вкажемо головну гілку(branch). Далі в проєкті створюємо файл ігнору для GitHub і вносимо туди файли, які важливо не пушити, а саме `node_modules` та `.env`. Відтепер все, що буде потрібно це запусити зміни за допомогою команд `git add`(зберігає всі наші зміни), `git commit -m "commit"`(створює commit) та `git push`(завантажує всі зміни на GitHub).

Наступним кроком потрібно задеплоїти сайт на хост. Для хостингу було використано платформу Heroku, перейдемо туди та створимо новий проєкт. Пройшовши всі необхідні процедури, вибираємо спосіб деплою за допомогою GitHub. Також Heroku потребує визначити зміни, які було вказано в `.env`, після чого сайт повноцінно працює на даній платформі і кожен користувач мережі інтернет має до нього доступ.

З первинними налаштуваннями розібралися, далі будемо створювати всі сторінки, вказувати їхні маршрути та стилізувати їх.

### 3.2 Стилiзацiя та маршрутизацiя сторiнок проєкту.

Для початку створимо шапку i будемо її рендерити на кожнiй сторiнцi. В цю шапку буде входити назва сайту, яка при цьому буде ссилкою на головну сторiнку. Пiд тайтлом буде знаходитись панель навігацiї, завдяки якiй будемо переходити на рiзні сторiнки. В цю навігацiю будуть входити перехiд на авторизацiю, реєстрацiя, калькулятори, профiль та персональний офiс, всi вони по собi є переходами на вiдповiднi сторiнки. Все вище перераховане пропишемо в файлi header.ejs, який створимо в папцi partials. Для того, він був пiдключений на кожнiй сторiнцi використаємо синтаксис ejs i пропишемо вiдповiдний рядок, який показаний на рисунку в layouts.ejs (рис. 3.4).

```
<%- include('../partials/header.ejs') %>
```

Рисунок 3.4 - Метод Include

На цю мить, все що є в проєктi це лише ссилки на вiдсутнi сторiнки (рис. 3.5).

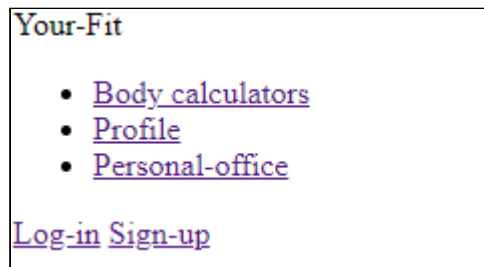


Рисунок 3.5 - Першi кроки

Створимо кожну сторiнку та пропишемо в нiй базовi форми. Для початку зробимо сторiнку для калькуляторiв. Для цього перейдемо views/calculators i створимо там файли: calculators.ejs, belly.ejs, carbs.ejs, water.ejs, pfc.ejs, pulse.ejs. В calculators.ejs пропишемо ссилки на всi види калькуляторiв. У всiх калькуляторах буде схожа форма - це title(в якому будуть вiдповiднi назви калькуляторiв), input(в який будуть вводитись данi), button(який буде вiдсилати

дані на опрацювання та окрема кнопка для зберігання даних в базі) і р(в якому при пропишемо необхідний текст). Для того, щоб підключити їх, переходимо в routes і прописуємо всю логіку, яка відображена на (рис. 3.6).

```
router.get('/', (req,res)=>{
  res.render('calculators/calculators')
})
router.get('/belly', (req,res)=>{
  res.render('calculators/belly')
})
router.get('/carbs', (req,res)=>{
  res.render('calculators/carbs')
})
router.get('/pfc', (req,res)=>{
  res.render('calculators/pfc')
})
router.get('/pulse', (req,res)=>{
  res.render('calculators/pulse')
})
router.get('/water', (req,res)=>{
  res.render('calculators/water')
})
```

Рисунок 3.6 - Маршрути калькуляторів

Далі переходимо в server.js і підключаємо всі написані маршрути, в схожому стилі, як це було зроблено з index.js, проте з відповідними роутерами.

Все те ж саме робимо і для сторінок верифікації, персонального офісу та профілю. На сторінках створюємо відповідні форми для введення даних і будемо їх стилізувати.

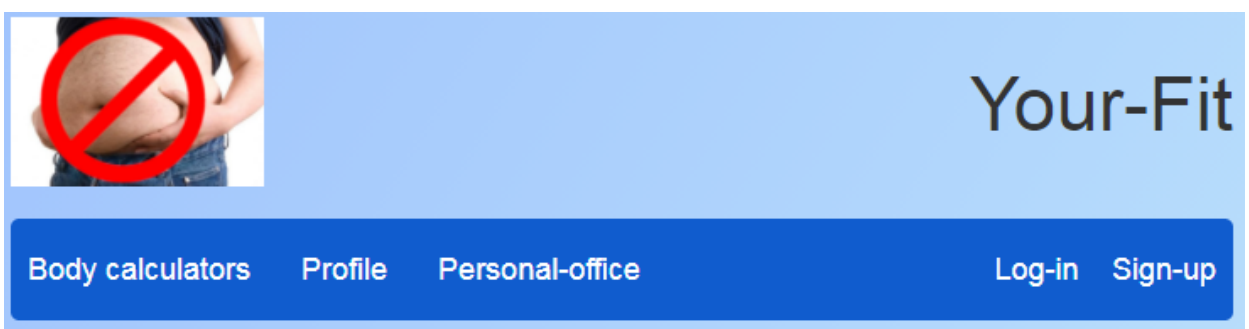
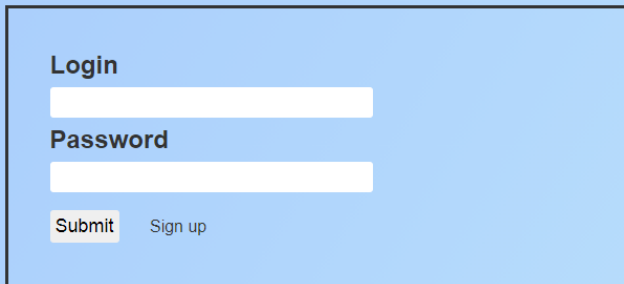


Рисунок 3.7 - Nav-Bar

Для початку повернемося до шапки проекту і дамо їй вигляд повноцінного title та nav-bar. До назви нашого сайту добавимо картинку, яка буде виконувати роль лого, збільшимо шрифт та розташуємо їх за допомогою Flex. Переходимо

до nav-bar, засунемо всі наші переходи у блок, задамо цьому блоку відповідної форми та кольору. Для усіх ссилек заберемо непотрібні стилі та розташуємо їх в блоці за допомогою того ж само flex, додамо псевдоклас, кольори та шрифт, те що вдалося зробити показано на (рис. 3.7).

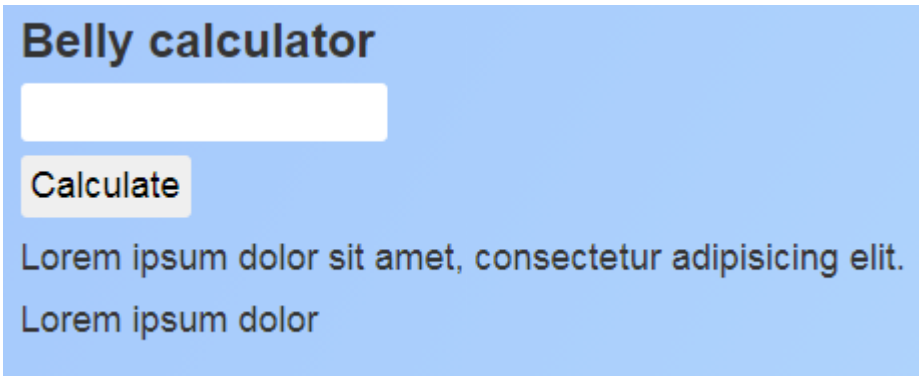
Перейдемо до сторінок верифікації та стилізуємо їх. Обидві сторінки будуть у схожому стилі, помістимо форму у блок, додамо контури, стилі шрифту та тексту, результат можна подивитись в прикладі нижче (рис. 3.8).



The image shows a login form on a light blue background. It has a title "Login" in bold. Below it is a white input field. Underneath is the label "Password" in bold, followed by another white input field. At the bottom, there are two buttons: "Submit" and "Sign up".

Рисунок. 3.8 - Авторизація

В розділі калькуляторів змінимо стилі ссилек та розташуємо їх ближче до центру, також вставимо картинку для заповнення порожнього простору. Перейшовши в конкретний калькулятор задамо стилі для форми, назви та тексту(рис. 3.9).



The image shows a calculator form on a light blue background. It has a title "Belly calculator" in bold. Below it is a white input field. Underneath is a button labeled "Calculate". Below the button, there are two lines of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit." and "Lorem ipsum dolor".

Рисунок 3.9 - Форма для калькулятора

Перейдемо до сторінок персонального офісу та профілю та задамо всім елементам у схожому з попередніми прикладами стилі.



Далі займемось адаптивністю, оскільки в проєкті використовувався flex та при можливості розмірність задавалась у відсотках елементи будуть самі підлаштовуватись під розміри екрану, проте це ще не все. При ширині екрану в 768px за допомогою media-запитів nav-bar буде трансформуватись в бургер. Таким чином з'являється нова кнопка при натисканні на яку буде впливати нове вікно, в яку будуть переміщені всі ссылки.

Для того, щоб форми змогли відсилати дані на сервер потрібно визначити моделі для них. Переходимо в папку model, пишемо схеми для кожної форми в проєкті та визначимо моделі для них всіх. Для початку визначимо схему для профілю. В цій моделі пропишемо рядки, які будуть зберігатися в базі даних, а саме name, sex, birth. при цьому потрібно враховувати, що назви цих рядків повинні сходитися з атрибутом name, який був заданий для кожної нашої форми. В кожен рядок буде записуватись значення, яке буде знаходитись res.body, який був відісланий на сервер після спрацьовування submit, після чого дані будуть зберігатись в базі. Усю цю логіку будемо прописувати уже в POST-запитах route. Створюємо подібним шляхом моделі для форми на сторінці персонального офісу, калькуляторів і пізніше створимо для верифікації, а поки що пропишемо все необхідне в route і відтепер введені дані будуть зберігатися в базі(рис. 3.10).

```
const mongoose = require('mongoose')

const profileSchema = mongoose.Schema({
  name: { type: String,
    required: true},
  sex: {type: String,
    required:true  },
  birth:{type: String,
    required:true}
})

const Profile = mongoose.model('Profile', profileSchema)

module.exports = Profile
```

Рисунок 3.10 - Схема профілю для баз даних

### 3.3 Створення верифікації за допомогою сесій

До форми реєстрації добавимо POST-запрос для того, щоб наші дані зберігалися в базі, проте попередньо потрібно зашифрувати пароль, щоб люди у яких є доступ до бази, не могли побачити персональні дані користувачів на сайті. Для цього потрібно завантажити модуль `bcrypt` командою `npm i bcrypt`, підключити і прописати його в POST-запросі `sign-up.js`. Наступним кроком буде приведення авторизації до нормально функціоналу. В даній ситуації не буде великою проблемою перевірити чи логін та пароль збігаються в базі даних, але для того, щоб система розуміла і розпізнавала кожного користувача потрібні сесії. Для керування сесіями будемо використовувати Passport, пропишемо в консолі `npm i passport passport-local express-session express-flash`. Щоб не нагромаджувати поточну сторінку, створимо окремий файл `passport-config.js` в якому будемо реалізовувати логіку Passport і імпортуємо його на необхідну сторінку. В файлі `passport-config.js` створюємо функцію ініціалізації попередньо, підключивши модуль `passport-local`. В самій функції створюємо новий екземпляр `LocalStrategy` і передаємо в параметр функцію, яка буде перевіряти наявність користувача в системі. Для цього будемо порівнювати пароль та логін, які знаходяться в системі, оскільки пароль зашифрований за допомогою `bcrypt`, необхідно підключити цей модуль і порівнювати введений пароль і пароль в базі за допомогою `bcrypt.compare`. Також всередині функції ініціалізації потрібно оголосити функції, які будуть відповідати за вхід та вихід з сесії.

```
function initialize(passport, getUserByEmail, getUserById) {
  const authenticateUser = async (email, password, done) => {
    const user = getUserByEmail(email)
    if (user == null) {
      return done(null, false, { message: 'No user with that email' })
    }
    try {
      if (await bcrypt.compare(password, user.password)) {
```

```

return done(null, user)
} else {
return done(null, false, {message: 'Password incorrect' })
}
} catch (e) {
return done(e)
}}

passport.use(new LocalStrategy({ usernameField: 'email' }, authenticateUser))

passport.serializeUser((user, done) => done(null, user.id))

passport.deserializeUser((id, done) => {
return done(null, getUserById(id))
})}

```

Переходимо в файл логування і підключаємо функцію ініціалізації пропустивши через неї параметри паролю та логіну. Далі потрібно підключити парочку middleware для flash та session попередньо підключивши ці модулі. Два наступних middleware підключають passport.initialize та passport.session. Зробивши ще парочку необхідних процедур, відтепер на сайті присутня проста, але робоча авторизація. Ще одна річ про яку потрібно подбати - це те, щоб не зареєстровані та зареєстровані користувачі не могли зайти на закриті для них сторінки просто змінивши домен. Для цього створимо спеціальну функцію checkAuthenticated, в якій створимо всю необхідну логіку і яку ще використаємо надалі.

### **3.4 Реалізація усіх додаткових запитів та процесів проєкту**

Перш ніж випустити проєкт на реліз потрібно закінчити RESTfull Арі. Для початку перейдемо на сторінку профілю та зробимо так, щоб після введення даних наші input зникали, а на їх місце з'являлись поле з введеним текстом. Також стане доступна кнопка , завдяки якій зможемо редагувати дану інформацію при введенні нового значення в форму за допомогою PUT-запроса.

```
router.put('/form1', async (req, res) => {
  let name
  try{
    name=awaitname.findById(req.params.id)
    name.name=req.body.name
    await name.save()
  }catch{
    if(name==null){
      console.log('nonamefind')
    }}})
```

Для профілю потрібно реалізувати можливість завантажувати фото. Для цього використаємо бібліотеку Filepond і виконаємо всю необхідну логіку для її реалізації.

Переходимо в персональний офіс і тут схожа ситуація після того, як користувач підтвердив введений текст замінюємо форми на поле з відображенням результату, при цьому буде доступна кнопка при натисканні на яку, відкривалось поле вводу для внесення відповідних змін. Також всі дані, які вводились в калькуляторах будуть відображатися на цій сторінці та відповідно, якщо користувач обрав нове значення воно оновиться. Також для поля вводу калорійності по днях тижня, будемо виводити загальну тижневу калорійність.

Повернемося до калькуляторів, для авторизованих користувачів буде доступна функція збереження результату. Для того, щоб така кнопка з'явилась потрібно застосовувати перевірку на авторизованість за допомогою функції `checkAuthenticated`. Також введені дані будуть прийматися та виводитися в залежності від формули. Розглянемо всі формули використанні у проєкті:

Формули для розрахунку калорій (4.1):

Для жінок:  $ООР = 655,1 + (9,6 * \text{маса тіла (кг)}) + (1,85 * \text{ріст(см)}) - (4,68 * \text{вік (роки)})$ .

Для чоловіків:  $ООР = 66,47 + (13,75 * \text{маса тіла (кг)}) + (5,0 * \text{ріст(см)}) - (6,74 * \text{вік (роки)})$ . (4.1)

де ООР - це денна норма калорій.

Для прикладу в формулі БЖВ візьмемо денну кількість калорій в 1700 ккал (4.2):

білки:  $(0,3 * 1700) / 4 = 127,5$  г;

жири:  $(0,3 * 1700) / 9 = 56$  г;

(4.2)

вуглеводи:  $(0,4 * 1700) / 4 = 170$  г.

Для розрахунку спожитої води вистачить знати лиш вагу тіла (4.3):

Чоловіки: Вага тіла x 35.

Жінки: Вага тіла x 31. (4.3)

За допомогою формули Астранда можна розрахувати максимальну частоту пульсу (4.4):

МЧСС для чоловіків = 220 - вік.

МЧСС для жінок = 226 - вік. (4.4)

Формула для розрахунку проценту жиру в організмі (4.5):

Для чоловіків:

$$\frac{495}{1.0324 - 0.19077 \times (w-n) + 0.15456 \times h} - 450$$

Для жінок: (4.5)

$$\frac{495}{1.29579 - 0.35004 \times (w+hip-n) + 0.22100 \times h} - 450$$

де  $h$  - ріст(см),  $n$  - об'єм шиї(см),  $w$  - об'єм талії(см),  $hip$ (тільки для жінок) - об'єм стегон(см).

За допомогою тієї ж функції `checkAuthenticated` будемо ховати кнопки `login` та `sign-up` в `nav-bar` на головній сторінці проєкту, якщо користувач авторизувався та виводи кнопки профілю та персонального офісу, якщо користувач зайшов в систему. Ця ж операція буде відбуватися і в бургер меню.

В деяких формах додамо кнопку видалення для того, щоб користувач міг видаляти небажані для нього дані.

```
router.delete('/', async (req, res) => {
  let name
  try {
    author = await name.findById(req.params.id)
    await name.remove()
  } }catch{
if(name == null) {
console.log('no name find')
}}
```

### Висновки до розділу 3

В розділі було проведено розробку програмного забезпечення з детальним, покроковим описом процесу розробки. Для початку побудували архітектуру нашого проєкту, потім запустили його на репозиторій і задеплоїли його на хост. Після чого побудували маршрути для всіх сторінок, надали

структуру і стилізували, щоб користувач міг взаємодіяти з сервером. Наступним кроком була реалізація системи авторизацій за допомогою сесій. І останнім кроком було підключено всі необхідні HTTP-запити та покращено функціонал UI.

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

### 4.1 Основні положення безпеки праці в ІТ-індустрії

Перед суспільством на усіх етапах його розвитку стояло завдання підвищення ефективності трудового процесу, удосконалення методів виробництва продуктів та засобів, які необхідні для його існування. У своїй практичній діяльності фахівець будь-якої галузі має вміти аналізувати потенційні небезпеки на різних етапах проєктування, експлуатації устаткування, визначати заходи усунення небезпек упроваджувати вимоги нормативних документів щодо нових розробок.

Забезпечення безпеки праці - необхідна умова будь-якого технологічного процесу, зокрема з застосування комп'ютерної техніки. Із зростанням кількості автоматизованих виробництв і комп'ютерних систем зазнають зміни чинники, які впливають на формування умов праці, які у свою чергу призводять до негативної дії на фізіологічну систему людини, знижуючи її працездатність і ініціюючи неправильні дії. Відомо, що кількість нещасних випадків, викликаних небезпечними діями, значно більше, ніж кількість нещасних випадків, викликаних небезпечними умовами праці.

На сьогодні, враховуючи інтенсивний розвиток комп'ютерної техніки, різко зросло і далі зростатиме кількість галузей і сфер діяльності людини, у яких використовуються інформаційні технології. Сучасний світ комп'ютерної техніки настільки широкий і різноманітний і так швидко розвивається, що кожна молода людина, яка сьогодні вчиться в середньому або вищому навчальному закладі, поза усяких сумнівів знайде в ньому своє місце. Тому надзвичайно важливо, щоб на початку своєї діяльності студент набув відповідних навичок зі збереження власного і суспільного здоров'я.

«Комп'ютерні захворювання», як і галузь, яка породила їх появу, ще



достатньо молода, але на відміну від самої техніки ще мало вивчені. Значною мірою поява «комп'ютерних захворювань» зумовлена тим, що шкідлива дія такого популярного і, на перший погляд, безпечного сучасного способу праці, яким є комп'ютер, ще далеко не всім відома. Крім того, на користувачів комп'ютерами впливає цілий комплекс чинників малої інтенсивності, негативна дія яких розвивається поступово і приховано. Тому захворювання виявляються тільки після багатьох місяців або навіть років праці, коли боротися з ними вже украй важко. У багатьох випадках важко навіть установити головну причину захворювань.

У таких умовах все більше значущим і необхідним стає потреба формування у майбутніх фахівців відповідних знань щодо особливостей впливу несприятливих виробничих чинників на комп'ютеризованих робочих місцях, способів і засобів, направлених на мінімізацію такого впливу, збереження здоров'я і професійного довголіття користувачів.

Отже, дисципліна «Безпека праці в індустрії інформаційних технологій» сприяє систематизації знань з безпечних методів праці щодо напряму інформаційних технологій та створенню комплексного підходу до організації робочих місць користувачів інформаційних технологій, що відповідають сучасним вимогам безпеки.

Вивчаючи цю тему, необхідно пам'ятати, що державна політика України в галузі безпеки праці спрямована на створення безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Вона базується на ряді принципів, основними з яких є пріоритет життя і здоров'я працівників, повна відповідальність роботодавця за створення безпечних та належних умов праці, підвищення рівня промислової безпеки, комплексне розв'язання завдань з охорони праці, соціальний захист працівників, повне відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань.

Україна є членом Міжнародної організації праці (МОП). Вона

ратифікувала 63 конвенції МОП, із них 14 - за роки незалежності. Положення цих конвенцій лягли в основу чинного в Україні законодавства, що регулює соціально-трудові відносини.

Законодавство Євросоюзу у сфері охорони праці можна умовно розділити на дві групи:

- директиви ЄС щодо захисту працівників;
- директиви ЄС щодо випуску товарів на ринок

(включаючи

обладнання, устаткування, машини, засоби колективного та індивідуального захисту, які використовують працівники на робочому місці).

Необхідно вивчити міжнародну систему охорони безпеки та здоров'я, основою якої є міжнародний стандарт OHSAS 18001 «Система менеджменту 12 охорони здоров'я та безпеки персоналу» будується на принципі добровільного дотримання вимог охорони праці, виходячи із встановленого законодавством допустимого рівня ризику для життя і здоров'я працівників. Міжнародний стандарт OHSAS так само, як і система управління якістю ISO 9000 і система управління охороною навколишнього середовища ISO 14000, побудований на основі циклу Демінга. Вивчаючи цю тему, треба знати вимоги, що встановлює підхід OHSAS.

Вимоги про те, якою має бути ергономіка в країнах ЄС, закріплено в законодавстві Союзу й, насамперед, у директиві № 89/391 «Про заходи щодо поліпшення безпеки й здоров'я трудящих». Відповідно до неї роботодавці зобов'язані оцінювати виробничі ризики й забезпечувати адекватні захисні і профілактичні заходи, гарантувати відповідне навчання й інструктаж працівників з дотримання заходів безпеки, а також надавати працівникам інформацію та консультації і дозволяти їм брати участь в обговоренні всіх питань із забезпечення безпеки й гігієни праці.

Розробкою загальних єдиних нормативних документів для користувачів

інформаційних технологій займаються декілька міжнародних організацій:

International Organization for Standardization (ISO) -  
міжнародна організація зі стандартизації;

Ergonomics committee ISO (TC 159) - комітет з ергономіки  
міжнародної організації зі стандартизації;

European Standardization Organization (CEN) - європейська  
організація зі стандартизації.

Серед низки розроблених нормативних документів з використання  
ВДТ найбільш часто застосовуються такі стандарти:

- ISO 9241-3, який регламентує ергономічні вимоги за  
умовами праці і охорони здоров'я користувачів;
- ISO 9001, який визначає якість і рівень виробництва  
апаратури; - ISO DIN 9995, який встановлює принципи розміщення  
елементів клавіатури для роботи з текстом в офісних системах;
- IEC 950, який визначає норми безпеки  
електротехнічного устаткування. Треба мати на увазі, що основним  
нормативним документом України із забезпечення охорони праці  
користувачів комп'ютерно-інформаційних технологій є НПАОП  
0.00-1.28-10 «Правила охорони праці при експлуатації  
електронно-обчислювальних машин».

У процесі трудової діяльності на людину впливає комплекс  
несприятливих чинників виробничого середовища і трудового процесу, у  
результаті чого можливі виробничі травми і професійні захворювання.  
Комплексний характер впливу цих чинників визначає необхідність системного  
підходу під час вирішення питань безпеки праці. Усі етапи технологічного  
процесу характеризуються інформаційним навантаженням, їх виконання  
вимагає уваги, зосередженості, що необхідно враховувати під час створення  
безпечних і нешкідливих умов праці користувачів ІТ.

Необхідно розуміти, що комп'ютер як технічний засіб може бути

джерелом небезпечних і шкідливих виробничих чинників. З принципу роботи відеодисплейного терміналу виходить, що він є джерелом:

- для моніторів з електронно-променевою трубкою (ЕПТ) - іонізуючих випромінювань, а саме рентгенівського випромінювання, яке виникає усередині колби під час різкої зупинки електронів, що швидко рухаються;
- оптичного випромінювання у видимому діапазоні, а також в УФ діапазоні, яке виникає завдяки взаємодії електронів з шаром люмінофора;
- електромагнітного випромінювання в радіочастотному діапазоні. При цьому високочастотні поля виникають під впливом електронного променя і пов'язані з частотою формування елемента зображення й інтенсивністю променя, низькочастотні поля виникають у системі горизонтальної розгортки, а дуже низькочастотні поля пов'язані з генерацією (вертикальною розгорткою);
- магнітного поля, що виникає через наявність відхиляючого пристрою;
- електростатичного поля, що виникає у зв'язку з високим потенціалом анодів ЕПТ.

Опрацьовуючи тему, слід згадати ДСТУ 12.0.003-74 «Небезпечні і шкідливі виробничі фактори», згідно з яким під час роботи на ПК мають місце фізичні та психофізіологічні небезпечні та шкідливі виробничі чинники. Хімічні та біологічні чинники - відсутні. Необхідно визначити та згадати засоби захисту від відповідних фізичних та психофізіологічних виробничих чинників.

Для об'єктивної оцінки категорії тяжкості праці на робочому місці на основі вивчення сукупного впливу всіх матеріально-виробничих елементів умов праці на організм людини (хімічних, фізичних і психологічних) розроблені спеціальні критерії. Кожний виробничий елемент умов праці  $X_i$  на робочому

місці одержує бальну оцінку від 1 до 6, якщо він впливає на працівника протягом усього робочого часу. В тих випадках, якщо він впливає на працівника не повний робочий день, а лише частково, елемент оцінюється його тривалістю і визначається за спеціальними діаграмами з урахуванням часу їх впливу.

Аналіз причин виробничого травматизму показує, що від 30 до 47 % нещасних випадків трапляється внаслідок наявності небезпечних та шкідливих факторів. У результаті впливу небезпечних і шкідливих виробничих факторів і шкідливих речовин мають місце нещасні випадки і професійні захворювання.

Професійними захворюваннями вважаються ті, які виникли в результаті професійної діяльності застрахованої особи і викликані впливом шкідливих виробничих чинників.

Нещасний випадок на виробництві, раптове погіршення стану здоров'я чи настання смерті працівника під час виконання ним трудових обов'язків внаслідок короткочасного (тривалістю не довше однієї робочої зміни) впливу небезпечного або шкідливого чинника.

Праця користувача комп'ютера відбувається за умови надмірного нервово-емоційного й зорового напруження, довготривалих статичних навантажень, обмеженої рухової активності, електростатичних полів, радіочастотного та рентгенівського випромінювання, які призводять до неврозів, відхилень у психіці, погіршення пам'яті, захворювань опорно-рухового апарату, серцево-судинної, імунної та статевої систем, органів зору, шкіри та ін.

Вивчаючи цю тему, необхідно знати усі потенційні травми та професійні захворювання, такі, як: електротравми, професійна офтальмопатія, карпальний тунельний синдром, фізіологічні та гінекологічні порушення, інформаційні неврози і нервові перенапруження, синдром комп'ютерного стресу, синдром тривалого статичного навантаження.

Розслідування та облік нещасних випадків, професійних захворювань і аварій на виробництві проводяться згідно з НПАОП 0.00-6.02-04. Вивчаючи цю

тему, згадати методи аналізу травматизму. Методи технічного аналізу дозволяють виявити технічні й організаційні причини травматизму. Існує кілька методів: монографічний метод, груповий метод та економічний метод.

Як організаційно-інформативні заходи щодо попередження травматизму застосовують знаки безпеки.

Існують чотири групи знаків безпеки праці: вказівні, заборонні, попереджувальні й приписувальні.

Вивчаючи цю тему, студент має знати сутність та особливості кожного методу виявлення причин травматизму, а також відрізнити знаки безпеки за групою

#### **4.2 Психологічні та фізіологічні особливості користувачів інформаційних технологій**

Відомо, що праця - це сукупність фізіологічних та психічних процесів, які спонукають, програмують і регулюють діяльність людини. Важлива роль у забезпеченні безпеки праці користувачів комп'ютеризованих систем відводиться їх психологічним властивостям і можливостям. Психологічні явища, що відбуваються в організмі оператора, є регуляторами його діяльності. Вони можуть посилювати або послаблювати дію того чи іншого сигналу на людський організм.

Вивчаючи тему, треба розуміти, що психологічні можливості користувача комп'ютера не є постійними. Вони залежать від інформаційного перевантаження, високого темпу роботи, перенапруження зорового та слухового аналізаторів, емоційного стану людини. Так, після конфліктних ситуацій, виробничих невдач, незаслужених образ з боку керівництва чи колег обсяг уваги різко знижується, порушується пам'ять. Оператор забуває послідовність дій, неправильно оцінює ситуацію, припускається грубих помилок. Тому люди, у яких психологічні можливості обмежені, а емоційний стан нестійкий, не

можуть бути призначені на відповідальні роботи, до яких належить й робота за комп'ютером.

Важливою умовою, що забезпечує безпеку праці, є збереження працівниками високого рівня працездатності. Під працездатністю розуміють потенційні можливості людини для виконання трудової діяльності протягом заданого часу з певною ефективністю. Великою, зворотною працездатністю, є стомлення.

Вивчаючи тему, необхідно знати чинники, що збільшують або зменшують ризик виникнення втоми та перевтоми операторів.

Однією з найважливіших задач ефективної організації трудового процесу є запобігання професійному стресу. Серед виробничих стрес-факторів можна виділити: фізичні (вібрація, шум, забруднена атмосфера); фізіологічні (змінний графік, відсутність режиму харчування); соціально-психологічні (конфлікт ролей, перевантаження або недовантаження працівників, міжособистісні конфлікти, висока відповідальність, дефіцит часу); структурно-організаційні («організаційний стрес»).

Слід зауважити, що найгіршим проявом професійного стресу для працівника є професійне вигорання. За визначенням Всесвітньої організації охорони здоров'я (2001), синдром професійного вигорання (burnout syndrome) -це стан фізичного, емоційного або мотиваційного виснаження, що характеризується порушенням продуктивності роботи та втомою, підвищенням схильності до соматичних захворювань, а також вживанням алкоголю чи інших психоактивних речовин з метою отримання тимчасового полегшення, що має тенденцію до розвитку фізіологічної залежності та (у багатьох випадках) суїцидальної поведінки. Згідно з клінічним підходом, синдром вигорання розцінюється як психічний розлад, який зумовлює особистісні властивості суб'єкта, зокрема, особливості його реагування на стрес. Відповідно до цього підходу, професійне вигорання класифікується як патологічний синдром у Міжнародному класифікаторі хвороб (МКХ-10).

Основними механізмами, що протидіють виникненню синдрому професійного вигорання є саморегуляція і само менеджмент. Вивчаючи тему, необхідно приділити увагу психічній саморегуляції та знати її методи. Самоменеджмент розуміють як послідовне і цілеспрямоване використання ефективних методів, прийомів, технік і технологій, спрямованих на саморозвиток свого творчого потенціалу з метою свідомого управління своїм життям і максимальної самореалізації у всіх його сферах.

Однією з важливих задач з управління діяльністю підприємства, пов'язаних з охороною праці, є забезпечення таких умов на підприємстві, що роблять можливим виконання роботи в безпечній і комфортній обстановці.

Одним з таких умов є мотивація. Мотивація включає внутрішній стан людини, званий потребою, і зовнішні чинники, що впливають на мотивацію, так звані стимули. Поведінка людини визначається потребами і стимулами, які домінують у даний момент часу.

Неефективна система мотивації може викликати у працівників незадоволеність, що завжди викликає зниження продуктивності праці. З іншого боку, ефективна система стимулює продуктивність персоналу, підвищує ефективність людських ресурсів.

Необхідно розуміти, що стимулювання до належного рівня мотивації може існувати у двох взаємозв'язаних формах:

- Матеріальне стимулювання виявляється в тому, що працівник при обумовленому обсязі і якості виконуваної роботи може розраховувати на отримання тих або інших матеріальних благ у натуральній або грошовій формі;
- Нематеріальне стимулювання виявляється в тому, що працівник при обумовлених вигляді, обсязі і якості виконуваної роботи може розраховувати на отримання тих або інших благ у нематеріальній формі, яка не має прямої грошової оцінки;

Студент має знати види матеріального та нематеріального стимулювання



до мотивації безпеки праці.

Вивчаючи тему, необхідно розглянути методи управління мотивацією праці:

- адміністративні методи засновані на ухваленні рішень керівниками у сфері нематеріальних заохочень і покарань, розподілу повноважень і відповідальності, визначенням графіка роботи тощо;
- економічні методи засновані на застосуванні матеріальної винагороди як компенсації за трудові зусилля працівника і досягнуті їм результати. Це ціла система мотивів і стимулів, що спонукають усіх працівників плідно працювати на загальне благо;
- соціально-психологічні методи засновані на застосуванні до працівника спеціальних стимулів, таких, як ставлення колективу, статус тощо. Соціальні методи пов'язані з соціальними відносинами, з моральним і психологічним впливом.

Вивчаючи ергономічні вимоги до робочих місць користувачів комп'ютерів, слід мати на увазі, що площа, виділена на одне робоче місце з відеотерміналом або ПК, має бути не менше 6 кв.м, а об'єм - не менше 20 куб.м. Площа, виділена для одного робочого місця без відеотерміналу або ПК, має бути не менше 4,5 кв.м, а об'єм - не менше 15 куб.м. Приміщення для відпочинку, прийому їжі, психологічного розвантаження й інші побутові приміщення повинні мати площу й об'єм з розрахунку на одну людину: площа - 4,5 кв.м і об'єм - 15 куб.м на одну людину з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Розташовуючи робочі місця з відеотерміналами і ПК, необхідно дотримуватися таких вимог:

- робочі місця з відеотерміналами і ПЕОМ розташовуються на відстані не менше 1 м від стін зі світловими отворами;

- відстань між бічними поверхнями відеотерміналів має бути не менше 1,2 м;
- відстань між тильною поверхнею одного відеотерміналу й екраном іншого має бути не менше 2,5 м;
- прохід між рядами робочих місць має бути не менше 1 м.

Необхідно вивчити вимоги до конструкції робочих місць. Організація робочого місця має забезпечувати відповідність усіх елементів робочого місця та їхнього розташування ергономічним вимогам ДСТУ 12.2.032-78, робоче місце при виконанні робіт сидячи, загальні ергономічні вимоги.

Необхідно розуміти, що для підвищення працездатності операторів вводять додаткові технологічні перерви в роботі, які залежать від виду діяльності.

За характером і складністю розв'язання задач працю користувачів ПК розділяють на три групи. Вивчаючи тему, студент має знати та розрізняти ці групи.

Під час проектування обладнання для оператора комп'ютеризованих систем викликає інтерес визначення міри складності інформаційної панелі, яка має відповідати рівню реакції людини

Умовну одиницю огляду панелі  $\beta$  розраховують залежно від лінії погляду по вертикалі та горизонталі робочого місця та від центрального поля зору.

### **4.3 Теоретичні основи ризику під час використання комп'ютеризованих систем**

Аналіз нещасних випадків дозволяє виділити три основні ланки, які зображено на рис. 4.1.

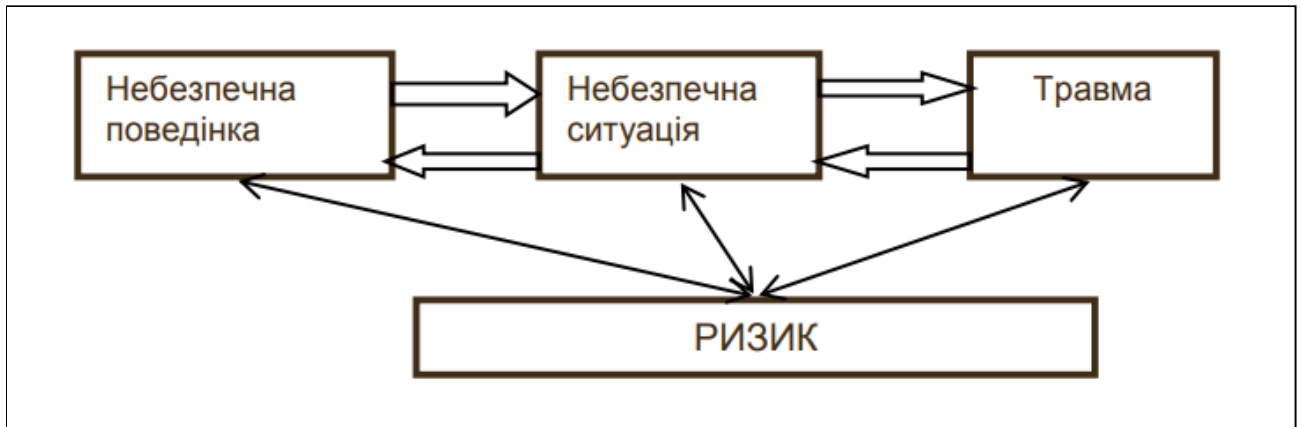


Рисунок 4.1 - Першопричини появи ризику

Перша ланка є однією з основних причин, які викликають травму або створюють небезпечну ситуацію, яка далі може призвести до нещасного випадку. У свою чергу, небезпечна поведінка - це наслідок психологічного характеру людини.

Студенту з курсу БЖД треба згадати, що ризик - це міра кількісного вимірювання небезпеки. Ризик є багатокомпонентною величиною і містить показники:

- величину збитку від дії чинника небезпеки;
- вірогідність виникнення чинника небезпеки;
- невизначеність величини збитку і вірогідності.

Система безпеки на будь-якому виробництві містить функції управління ризиком, яка спрямоване на забезпечення рівня безпеки максимально досяжного з погляду економічних і соціальних чинників.

Ризик-менеджмент - процес прийняття і виконання управлінських рішень, спрямованих на зниження ймовірності виникнення несприятливого результату і мінімізацію можливих втрат, викликаних його реалізацією.

Процес менеджменту ризику відповідно до ISO 31010:2009 складається з таких елементів:

- обмін інформацією і консультації;
- визначення галузі застосування ризик-менеджменту;

- оцінку ризику (включаючи ідентифікацію ризику, аналіз і порівняльну оцінку ризику);
- обробку ризику;
- моніторинг і аналіз ризику.

Оцінка ризику є частиною процесу менеджменту ризику, це структурований процес, у якому ідентифікуються засоби досягнення мети, проводять аналіз наслідків і вірогідності виникнення небезпечних подій для прийняття рішення про необхідність обробки ризику.

Оцінка ризику дозволяє відповісти на такі основні запитання:

- які події можуть відбутися і їх причина (ідентифікація небезпечних подій);
- які наслідки цих подій;
- яка вірогідність їх виникнення;
- які чинники можуть зменшити несприятливі наслідки або зменшити вірогідність виникнення небезпечних ситуацій.

Студент має розглянути основні методи оцінки ризиків згідно з ISO/IEC 31010:2009 Risk management - Risk assessment techniques:

- мозковий штурм,
- метод Дельфі,
- попередній аналіз небезпек (PHA - Preliminary Hazard Analysis),
- дослідження безпеки та працездатності (HAZOP - Hazard and Operability Study),
- аналіз першопричин (RCA - Root Cause Analysis),
- аналіз видів і наслідків відмов та критичності відмов (FMEA - Failure Mode Effect Analysis),
- Байєсівський аналіз і мережа Байєса,
- криві FN,
- аналіз ефективності витрат,

метод Файн-Кінні,  
аналіз дерева несправностей (FTA - Fault Tree Analysis),  
аналіз дерева подій (ETA- Event Tree Analysis).

Таблиця 4.1

Серйозність наслідків	Признаки
1. Незначні	Подія викликає короточасне захворювання або порушення здоров'я, які не потребують звертання по медичну допомогу. Можлива відсутність на роботі не більше 3-х днів. Наприклад, головний біль або синяк.
2. Помірно значні	Подія викликає значні і тривалі наслідки. Потребує звертання по медичну допомогу. Викликає від 3 до 30 днів відсутності на роботі. Наприклад, різана рана або слабкі опіки.
3. Серйозні	Подія викликає постійні і незворотні пошкодження. Потребує стаціонарного лікування і викликає відсутність на роботі більше 30 днів. Наприклад, серйозні професійні захворювання, стійка непрацездатність або смерть.

Критерії визначення серйозності наслідків

Також існує методика визначення величини ризику згідно зі стандартом BS 8800. 1996 Guide to Occupational Health and Safety Management Systems. British Standard Institution (Руководство по системам управління охороною труда. Британский институт стандартизации). У даному випадку величина ризику утворюється з вірогідності небезпечної події і значущості (серйозності) її наслідків. Значущість наслідків означає серйозність заподіюваної здоров'ю людини шкоди, що викликається небезпечною подією (табл. 4.1).

На вірогідність події впливають багато явних і прихованих чинників, при цьому найзагальнішим з них є частота прояву шкідливої дії; тривалість шкідливої дії; можливість передбачати наперед появу шкідливої дії; можливість запобігти шкідливій дії.

Існує декілька інструкцій загального характеру для визначення вірогідності за допомогою методу, наведеного в даній методичці (табл. 4.2).

Таблиця 4.2

Критерії визначення вірогідності події

Вірогідність події	Признаки
1. Мала вірогідність	Подія, яка виникає рідко і нерегулярно. Наприклад, поверхня тротуарів взимку стає слизькою від льоду.
2. Середня вірогідність	Подія, яка виникає час від часу, але нерегулярно. Наприклад, під час техобслуговування підйомника вантаж потрібно піднімати власноруч.
3. Висока вірогідність	Подія, яка виникає часто і регулярно. Наприклад, регулярний рух навантажувача викликає небезпеку зіткнення.

Величину ризику можна визначити за допомогою таблиці (табл. 4.3).

Таблиця 4.3 Величина ризиків (BS 8800)

Вірогідність події	Серйозність наслідків		
	Незначні	Помірно значні	Серйозні

Мала	1 Малозначущий ризик	2 Малий ризик	3 Помірний ризик
Середня	2 Малий ризик	3 Помірний ризик	4 Значний ризик
Висока	3 Помірний ризик	4 Значний ризик	5 Неприпустимий ризик

Також можна скористатися зведеною табл. 4.4 для визначення величини ризику, що викликається фізичними чинниками.

Таблиця 4.4

Визначення величини ризику, що викликається фізичними чинниками

	Наслідки		
	Вірогідність події	Легкі Незручний стан, роздратування, легка хвороба, що проходить	Середньої тяжкості. Опіки, тривалі серйозні впливи, тривала легка шкода
Мала. Вплив 10-50 % від нормативної величини	1 - Малозначущий ризик	2 - Малий ризик	3 - Помірний ризик

Середня. Вплив 50-100 % від нормативної величини	2 - Малий ризик	3 - Помірний ризик	4 - Значний ризик
Висока. Вище нормативних значень	3 - Помірний ризик	4 - Значний ризик	5 - Неприпустимий ризик

Відносною межею проведення заходів може слугувати різниця у величині ризиків за таблицею ризиків. Якщо величина ризику 1 або 2, то вона не потребує проведення спеціальних заходів. Якщо величина ризику 3, 4 і 5, то ризик потрібно мінімізувати. Черговість заходів можна визначити, користуючись таблицею (табл. 4.5).

Таблиця 4.5

Інструкції про значущість ризику й ухвалення рішення про необхідність і черговість заходів

Величина ризик	Необхідні заходи для зменшення ризику
1 - Малозначущий ризик	Ризик такий малий, що заходи непотрібні
2 - Малий ризик	Заходи не обов'язкові, але за ситуацією потрібно слідкувати, щоб ризик був керованим
3 - Помірний ризик	Заходи для зменшення ризику необхідні, але їх проведення можна спланувати і провести точно за графіком.
4 - Значний ризик	Заходи щодо зниження величини ризику обов'язкові і їх проведення слід почати терміново.



	Робота в умовах ризику має бути негайно припинена, і її не можна відновлювати перш, ніж ризик буде зменшений.
5 - Неприпустимий ризик	Заходи щодо ліквідації ризику обов'язкові і їх проведення необхідно почати негайно. Робота в умовах ризику має бути негайно припинена, і її не можна відновлювати перш, ніж ризик буде ліквідований.

#### **Висновки до розділу 4**

В розділі було наведено та детально розглянуто основні положення безпеки в ІТ-індустрії, після чого описано основні фактори негативного впливу на людину при роботі у даній спеціальності, були розглянуті психологічні особливості користувачів інформаційних технологій задля забезпечення оптимального трудового процесу. Були наведені основні ризики при роботі з комп'ютеризованими системами, в таблиці були описані всі признаки, наслідки, та необхідні заходи при роботі в цій області.

## ВИСНОВОК

В роботі було розглянуто основні вимоги до сучасних вебзастосунків, поставлено вимоги, а також переглянуто декілька аналогів, які відповідають цим вимогам. Завдяки зібраній інформації була описана документація, після чого побудовано декілька діаграм та побудовані макети, як для браузера, так і для мобільної версії. І нарешті була реалізована система для обробки, обрахунку та збереження даних у вебзастосунку.

Робота була виконана на стеку технологій Node.js, Express, Ejs, Css та при підключені бази даних MongoDB. Проєкт підтримується в більшості браузерів та адаптується під мобільне розширення. Також було реалізована система авторизації з подальшою прив'язкою даних до певного користувача. Користувацький інтерфейс був виконаний у простій манері, тому користування даним застосунком не складе великої проблеми.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Посібник поExpress: Вебресурс.* URL: <https://glosbe.com/uk/>
2. *Посібник по HTML5 і CSS3: Вебресурс.* URL:  
<https://metanit.com/web/html5/>
3. *Express - фреймворк вебзастосунків Node.js: Вебресурс.* URL:  
<https://expressjs.com/ua/>
4. *EJS-EmbeddedJavaScripttemplates: Вебресурс.* URL:  
<https://ejs.co/>
5. *Node.js | EJS: Вебресурс.* URL:  
<https://metanit.com/web/nodejs/4.10.php>
6. *NpmDocs: Вебресурс.* URL: <https://www.npmjs.com/>
7. *Документація | Node.js: Вебресурс.* URL:  
<https://nodejs.org/uk/docs/>
8. *MongoDB | Вступ: Вебресурс.* URL:  
<https://metanit.com/nosql/mongodb/1.1.php>
9. *Node.js | Mongoose: Вебресурс.* URL:  
<https://metanit.com/web/nodejs/6.6.php>
10. *П'ять простих кроків для розуміння JSONWebTokens (JWT): Вебресурс.* URL: <https://habr.com/ru/post/340146/>
11. *Уроки Express.js. Шаблонізації з EJS: Layout, Block, Partials: Вебресурс.* URL:<https://soshace.com/ru/express-js-ejs-templating/>
12. *HelloWorld. GitHub: Вебресурс.* URL:  
<https://guides.github.com/activities/hello-world/>
13. *Сучасний посібник JavaScript: Вебресурс.* URL:  
<https://learn.javascript.ua/>
14. *Керівництво по JavaScript: Вебресурс.* URL:  
<https://metanit.com/web/javascript/>

15. *JavaScript* | *MDN:Вебресурс*. URL:

<https://developer.mozilla.org/ru/docs/Web/JavaScript>