МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет економіки, підприємництва та інформаційних технологій

Кафедра інформаційних технологій

Допущено до захисту

"_____" _____ 2026 р.

Зав. кафедри, к.т.н., доцент

_____ Ліщук Р.І.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНДИВІДУАЛЬНОГО ПІДБОРУ РЕЖИМУ СПОРТИВНИХ ТРЕНУВАНЬ

Виконав – студент 4 курсу, групи 41 кн
освітньої програми "Комп’ютерні науки"
спеціальності 122 Комп’ютерні науки
Корсанюк Максим Олександрович

Науковий керівник – к.е.н., доцент
Концеба Сергій Михайлович

Умань – 2026

Реферат
на кваліфікаційну роботу "Розробка програмного забезпечення для індивідуального підбору режиму спортивних тренувань "
студента факультету економіки, підприємництва та інформаційних технологій
Уманського національного університету
Корсанюка Максима Олександровича

Структура та обсяг роботи. Пояснювальна записка кваліфікаційної роботи складається з трьох розділів, містить 48 сторінок, 1 таблиця, 12 рисунків, 1 додаток, 37 джерел наукової літератури.

Метою даної роботи є розробка програмного забезпечення для індивідуального підбору режиму спортивних тренувань, яке забезпечує автоматизований аналіз параметрів користувача та формування оптимального комплексу вправ.

У першому розділі проаналізовано потреби користувачів та способи виконання задач. Здійснено постановку задач з розробки програмного забезпечення для індивідуального підбору режиму спортивних тренувань.

У другому розділі здійснено проектування автоматизації індивідуального підбору тренувань для спортсменів з врахування антропометрії.

У третьому розділі зроблено програмне забезпечення для автоматизації індивідуального підбору тренувань.

Анотація

В ході виконання роботи зроблено програмне забезпечення для автоматизації індивідуального підбору тренувань. Написання коду виконано у Microsoft Visual Code, мови програмування HTML, CSS, Javascript. Перевагою такого методу розробки веб-сайту є те, що дану версію можна вбудувати в будь-який інший сайт. Дана мультиплатформність облегшить роботу і представлення такого роду технологію тренерам, котрі не мають ПК.3

Ключові слова: тренування, автоматизація, веб-сайт, комплекси тренувань, вебпрограмування, верстка сайту.

Annotation

During the work, a website was created to automate individual training selection. The code was written in Microsoft Visual Code, programming languages HTML, CSS, Javascript. The advantage of this method of website development is that this version can be embedded in any other website. This multi-platform feature will facilitate the work and presentation of this type of technology to trainers who do not have a PC.3

Keywords: training, automation, website, training complexes, web programming, website layout

ЗМІСТ

ВСТУП 4
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМ ДЛЯ СПОРТСМЕНІВ 6
1.1. Аналіз предметної області 6
1.2. Аналіз потреб користувачів та способи виконання задач 7
1.3. Аналіз аналогів розробки 10
1.4. Постановка задачі 19
РОЗДІЛ 2 ПРОЄКТУВАННЯ АВТОМАТИЗАЦІЇ ІНДИВІДУАЛЬНОГО ПІДБОРУ ТРЕНУВАНЬ ДЛЯ СПОРТСМЕНІВ З УРАХУВАННЯМ АНТРОПОМЕТРІЇ 21
2.1. Обґрунтування технологій розробки 21
2.2. UML-Діаграми 34
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ 38
3.1. Процес користування сайтом 38
3.2. Аспекти розробки 41
ВИСНОВКИ 44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46
ДОДАТКИ 49

ВСТУП

У сучасних умовах розвитку інформаційних технологій та зростання інтересу до здорового способу життя актуальним є створення програмних засобів, що дозволяють автоматизувати процес підбору індивідуальних програм тренувань. Особливої важливості це набуває у випадках, коли необхідно враховувати індивідуальні фізичні характеристики користувача, рівень підготовки, вік та специфіку обраного виду спорту.

Метою даної роботи є розробка програмного забезпечення для індивідуального підбору режиму спортивних тренувань, яке забезпечує автоматизований аналіз параметрів користувача та формування оптимального комплексу вправ.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- проаналізувати існуючі програмні рішення у сфері автоматизації тренувального процесу та визначити їхні переваги і недоліки;
- визначити основні параметри користувача, які впливають на формування індивідуальної програми тренувань (вік, вага, зріст, рівень підготовки, соматотип тощо); розробити модель підбору вправ на основі вхідних даних користувача;
- спроєктувати архітектуру програмного забезпечення з урахуванням можливості масштабування та розширення функціоналу;
- реалізувати алгоритми обробки даних та формування індивідуальних тренувальних програм;
- розробити зручний інтерфейс користувача для введення даних і отримання результатів;
- забезпечити доступ до системи через веб-інтерфейс для можливості використання на різних пристроях;
- провести тестування програмного забезпечення та оцінити його ефективність.

Об’єктом дослідження є процес організації та оптимізації тренувальної діяльності спортсменів.

Предметом дослідження є методи та засоби автоматизованого підбору індивідуальних програм тренувань із використанням інформаційних технологій.

Очікуваним результатом є створення програмного продукту, який дозволяє швидко та ефективно формувати індивідуальні комплекси вправ, мінімізуючи вплив людського фактора та підвищуючи якість тренувального процесу.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМ ДЛЯ СПОРТСМЕНІВ

1.1 Аналіз предметної області

У XXI столітті автоматизація людської діяльності стала невід'ємною складовою розвитку сучасних технологій та програмного забезпечення. Її впровадження охоплює дедалі ширший спектр сфер — від промисловості до освіти і повсякденного життя. Навіть ті галузі, які раніше не асоціювалися з інформаційними технологіями, поступово інтегрують цифрові рішення, що є закономірним результатом безперервної еволюції технічного прогресу.

Особливого значення набуває впровадження інформаційних технологій у сферу освіти. Воно не лише сприяє інноваційному використанню технічних засобів у навчанні, але й відкриває нові можливості для оптимізації організації освітнього процесу. Своєчасне застосування ІТ-рішень дозволяє значно підвищити ефективність управління навчальною діяльністю, автоматизувати облік та аналіз даних, а також покращити взаємодію між учасниками освітнього процесу.

Автоматизація дає змогу позбутися значної частини рутинної та монотонної роботи, що, у свою чергу, забезпечує економію часу та ресурсів. До ключових напрямів автоматизації навчання належать: управління навчальними процесами, облік відвідуваності, оцінювання знань, планування занять, моніторинг результатів та їх діагностика. Важливу роль при цьому відіграють навчальні моделі, які становлять основу для побудови ефективних автоматизованих систем.

Окрім традиційного навчання, автоматизація активно використовується і в тренувальних процесах. Сучасний ринок пропонує широкий вибір рішень - від мобільних додатків для домашніх занять до комплексних програм із доступом за підпискою. Проте більшість таких продуктів не враховують індивідуальні

особливості користувачів і пропонують універсальні набори вправ без належної адаптації.

Саме тому доцільним є створення системи, що базується на індивідуальному підході до кожного користувача. У даному випадку акцент зроблено на формуванні персоналізованого комплексу вправ для розвитку м’язових груп, задіяних у дзюдо. Важливою особливістю є орієнтація на дітей як основну цільову аудиторію. Дитячий організм має вищу адаптивність до фізичних навантажень, що дозволяє досягати результатів із використанням меншої кількості параметрів, а також забезпечує значний потенціал розвитку за умови правильно організованого тренувального процесу.

У процесі дослідження було проведено аналіз існуючих програмних рішень і тренувальних комплексів. Вони умовно поділяються на кілька категорій. До першої належать мобільні додатки, які пропонують різноманітні програми тренувань - як для занять вдома, так і у спортивних залах, із теоретичними матеріалами або візуальними прикладами виконання вправ [3]. Друга категорія - це кросфіт-програми за підпискою, орієнтовані переважно на дорослу аудиторію різного віку. Третю групу становлять інформаційні статті, що базуються на загальнодоступних даних і не містять достатньої конкретики. Четверта категорія - відеоуроки, розміщені на популярних відеохостингах, які, хоча й є доступними, але не передбачають персоналізації та адаптації під індивідуальні потреби користувача.

Таким чином, аналіз існуючих рішень підтверджує необхідність розробки інтелектуальної системи, здатної забезпечити персоналізований підхід до формування тренувальних програм, що є особливо актуальним у контексті розвитку дитячого спорту.

1.2 Аналіз потреб користувачів та способи виконання задач

У сучасних умовах стрімкого розвитку технологій стає дедалі очевиднішою необхідність використання автоматизованих засобів для обробки та аналізу

даних, зокрема і в спортивній сфері. Ефективне опрацювання отриманої інформації дозволяє не лише підвищити якість тренувального процесу, але й досягти прогнозованих результатів у вирішенні конкретних завдань, пов’язаних із фізичним розвитком спортсменів.

Формування тренувального комплексу для дітей є складним і відповідальним процесом, що передбачає підбір оптимальних вправ відповідно до індивідуальних особливостей організму. Якість такого підбору значною мірою залежить від кваліфікації тренера, його досвіду та здатності враховувати велику кількість параметрів. Проте навіть висококваліфікований спеціаліст не застрахований від помилок під час оцінювання фізичного стану дитини або розрахунку навантажень. Саме тому доцільним є впровадження автоматизованих систем, які здатні мінімізувати людський фактор.

Одним із перспективних підходів є створення веб-орієнтованої системи, яка збирає необхідні дані про спортсмена (вік, фізичні показники, рівень підготовки тощо), аналізує їх і визначає соматотип користувача. На основі отриманих результатів система формує індивідуальний комплекс вправ, найбільш придатний для конкретного користувача. Такий підхід забезпечує більш точний і обґрунтований підбір тренувальних програм.

Інформаційні технології відіграють важливу роль не лише в освіті, але й у сфері фізичного виховання та спорту. Незважаючи на значну кількість програмних продуктів, орієнтованих на підтримку фізичної активності, більшість із них не враховують потреби тренерів або не мають вузької спеціалізації за окремими видами спорту. Це створює передумови для розробки нових рішень, які б поєднували аналітичні можливості та практичну користь для професійної діяльності тренерів.

У сучасному суспільстві проблема недостатньої фізичної активності набуває все більшої актуальності. Обмежений час, малорухливий спосіб життя та недостатня обізнаність у питаннях здоров’я призводять до поширення таких явищ, як гіподинамія та гіпокінезія. Гіподинамія - це порушення функціонування організму, що виникає внаслідок зниження рухової активності і супроводжується

погіршенням роботи опорно-рухового апарату, серцево-судинної та інших систем [13].

Гіпокінезія, у свою чергу, характеризується обмеженням рухової активності, зниженням обсягу та інтенсивності рухів і часто асоціюється із сидячим способом життя [14].

Хоча відвідування спортивних залів частково вирішує проблему фізичної неактивності, воно не гарантує досягнення бажаних результатів без належних знань і контролю. Відсутність кваліфікованого супроводу може призвести до неправильного виконання вправ і навіть погіршення стану здоров’я.

Найбільш поширеними засобами підтримки фізичної активності є мобільні додатки. До їх основних переваг належать доступність, зручність використання, відсутність фінансових витрат на тренера чи абонемент у спортивний зал, гнучкість графіку тренувань та широкий вибір програм. Користувач може самостійно обирати вправи, адаптувати тренування під власні потреби та займатися у комфортних умовах.

Водночас такі додатки мають і суттєві недоліки. Зокрема, вони не забезпечують належного рівня контролю за правильністю виконання вправ, не враховують індивідуальні фізіологічні особливості користувача та не передбачають безпосередньої взаємодії з фахівцем. Це може призвести до неефективності тренувань або навіть травм.

Окрему категорію становлять кросфіт-програми, що надаються за підпискою. Вони, як правило, передбачають певний рівень персоналізації та інколи включають консультації з тренерами. Проте такі рішення мають фінансові обмеження та не завжди гарантують відповідність потребам конкретного користувача.

З огляду на зазначені особливості було обрано напрям розробки системи з вузькою спеціалізацією - допомога тренеру з дзюдо у формуванні індивідуального комплексу вправ для дітей. Ключовим елементом є визначення соматотипу дитини, що потребує глибоких знань і професійного підходу.

Запропонована система покликана спростити цей процес і підвищити точність прийняття рішень.

Важливими критеріями оцінювання такої системи є її функціональні можливості та якість контентного наповнення. Саме поєднання цих аспектів забезпечує коректність результатів і ефективність використання програмного продукту.

Особливу увагу приділено забезпеченню доступності системи. Вона повинна працювати в онлайн-режимі та бути доступною з будь-якого пристрою - смартфона, планшета чи персонального комп’ютера. Для отримання результату користувачеві достатньо виконати простий алгоритм дій: відкрити веб-сайт, ввести необхідні параметри спортсмена та отримати індивідуально сформований комплекс вправ.

Таким чином, запропонований підхід поєднує сучасні технологічні можливості з практичними потребами спортивної підготовки, що стало основою для формування концепції та реалізації даного проєкту.

1.3 Аналіз аналогів розробки

Розберемо спочатку мобільні додатки з певною спорт спеціалізацією:

Перелік вузькоспеціалізованих спортивних програм є значно менш обширним порівняно із загальною кількістю фітнес-додатків, проте для проведення детального аналізу було виявлено програмний продукт для персонального комп’ютера, функціонально подібний до веб-рішень. Йдеться про програму Judoist, яка за своїми можливостями частково відповідає концепції розроблюваного проєкту. Однак вона має суттєві обмеження у використанні, зокрема через те, що є запатентованою розробкою та не отримала подальшого розвитку. Її функціональність залишилася на завершеному рівні без оновлень, що значно знижує її актуальність.

Крім того, доступ до цієї програми є ускладненим: без наявності конкретного джерела знайти її у відкритому доступі практично неможливо. Це

робить її малопридатною для широкого використання, зокрема серед тренерської спільноти України.

На відміну від цього, розроблюваний проєкт передбачає відкриту доступність для широкого кола користувачів. Хоча ефективне використання системи передбачає наявність певного рівня кваліфікації, зокрема у тренерів, для користувачів без спеціальної підготовки також будуть передбачені короткі пояснення та рекомендації. Це дозволить підвищити зручність використання та отримання коректних результатів навіть для менш досвідчених користувачів.

Такий підхід обрано з метою популяризації розроблюваного ресурсу, а також підвищення інтересу до дзюдо як виду спорту. Демонстрація ефективних методів тренування та доступність інформації можуть сприяти залученню нових користувачів і розширенню аудиторії.

Розберемо спочатку мобільні додатки з певною спорт спеціалізацією.

Мобільний додаток Heria Pro є сучасною платформою для організації тренувального процесу, орієнтованою передусім на розвиток сили, витривалості та функціональної підготовки. Додаток створений відомим спортсменом і популяризатором калістеніки Chris Heria, що значною мірою визначає його спрямованість на тренування з власною вагою, а також поєднання елементів функціонального тренінгу та класичних вправ (рис. 1.1).

У додатку представлено велику кількість тренувальних програм різного рівня складності - від початкового до просунутого. Користувач отримує доступ до структурованих планів занять, які включають відеоінструкції, пояснення техніки виконання вправ та рекомендації щодо інтенсивності. Особливістю є акцент на візуалізації: більшість вправ супроводжується демонстрацією правильного виконання, що полегшує процес навчання навіть для новачків.

До переваг додатку можна віднести його високу доступність та зручність використання. Інтерфейс інтуїтивно зрозумілий, що дозволяє швидко розпочати тренування без додаткового навчання. Значною перевагою є можливість займатися у будь-якому місці - як вдома, так і на відкритих спортивних майданчиках, без необхідності використання спеціалізованого обладнання.

Також варто відзначити наявність готових програм, що економить час користувача на самостійне планування тренувань. Регулярне оновлення контенту забезпечує різноманітність вправ і підтримує мотивацію до занять.

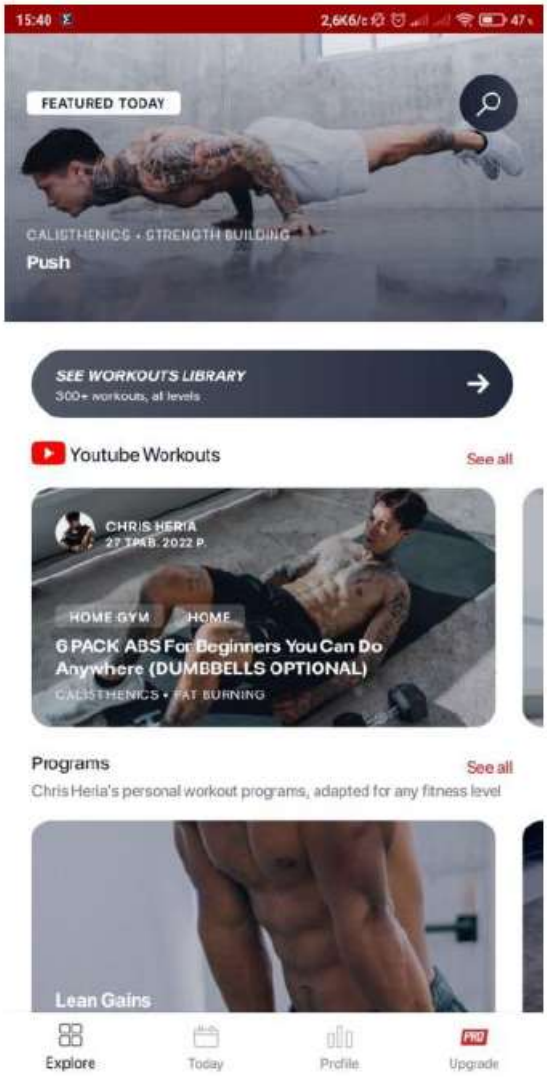


Рис. 1.1. Приклад мобільного додатку Heria pro [15]

Однак, попри значну кількість переваг, додаток має і певні недоліки. Насамперед, відсутня глибока персоналізація тренувальних програм - більшість із них є універсальними і не враховують індивідуальні фізіологічні особливості користувача. Це може знижувати ефективність тренувань для окремих категорій користувачів. Крім того, додаток не забезпечує повноцінного контролю за правильністю виконання вправ, що може призвести до помилок і навіть травм у разі неправильного підходу. Частина функціоналу доступна лише за платною

підпискою, що обмежує можливості безкоштовного використання. Також відсутня безпосередня взаємодія з тренером, що є важливим фактором для початківців або дітей.

Таким чином, Heria Pro є потужним інструментом для самостійних тренувань, особливо у сфері калістеніки, проте потребує доповнення у вигляді персоналізації та професійного супроводу для досягнення максимально ефективних результатів.

Мобільний додаток Fitness Trainer FitProSport FULL є універсальним інструментом для організації тренувального процесу, орієнтованим на користувачів різного рівня підготовки. Додаток містить значну кількість вправ, розподілених за групами м’язів, а також готові програми тренувань, які можуть використовуватися як у домашніх умовах, так і в спортивному залі. Основна увага приділяється силовим вправам, бодібілдингу та загальному фізичному розвитку (рис. 1.2).

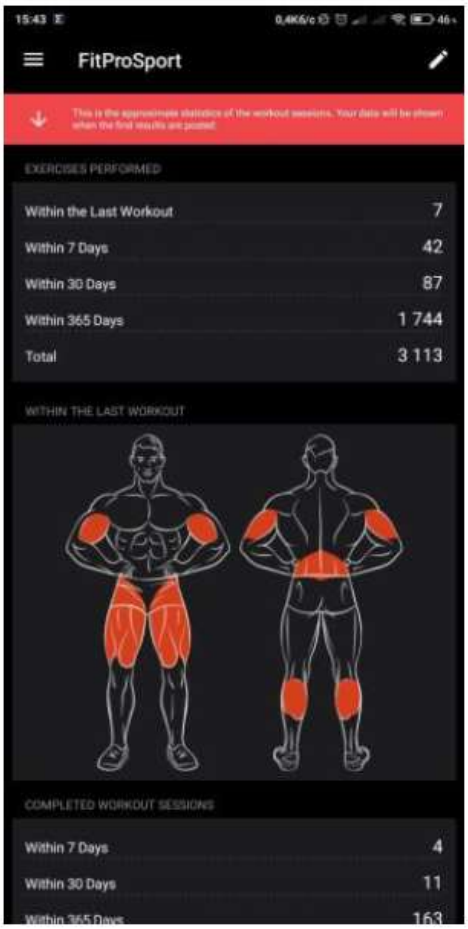


Рис. 1.2. Приклад мобільного додатку Fitness Trainer FitProSport FULL [16]

Функціонал додатку передбачає наявність детальних описів вправ, інколи із візуальним супроводом, що допомагає користувачу зрозуміти правильну техніку виконання. Також присутня можливість самостійного формування тренувальної програми, що дає змогу адаптувати заняття під власні потреби. Додаток може використовуватися як довідник з вправ або як базовий інструмент для планування тренувань.

До переваг даного додатку можна віднести його простоту та доступність. Він не потребує складної реєстрації або підключення до інтернету для базового функціоналу, що робить його зручним у використанні в будь-яких умовах. Велика база вправ дозволяє користувачу ознайомитися з різними техніками тренування та урізноманітнити свої заняття. Крім того, можливість самостійного складання програм є корисною для більш досвідчених користувачів, які розуміють принципи побудови тренувального процесу.

Водночас додаток має низку недоліків. Насамперед, він не забезпечує індивідуального підходу до користувача, оскільки не аналізує фізичні параметри або рівень підготовки автоматично. Уся відповідальність за підбір навантаження покладається на самого користувача. Також відсутній контроль за правильністю виконання вправ, що може призвести до помилок або травм. Інтерфейс у деяких версіях виглядає застарілим і не завжди є інтуїтивно зрозумілим. Додатково варто зазначити обмежену кількість інтерактивних функцій і відсутність зв’язку з тренером або іншими користувачами.

Таким чином, Fitness Trainer FitProSport FULL є корисним базовим інструментом для ознайомлення з вправами та організації тренувань, однак його функціональність обмежена відсутністю персоналізації, сучасних інтерактивних можливостей та професійного супроводу.

Мобільний додаток Freeletics є сучасною платформою для організації індивідуального тренувального процесу, орієнтованою на розвиток сили, витривалості та загальної фізичної підготовки. Основною особливістю додатку є використання алгоритмів адаптації, які формують персоналізовані тренувальні

програми на основі введених користувачем даних, таких як рівень підготовки, цілі та прогрес у виконанні вправ (рис. 1.3).



Рис. 1.3. Приклад мобільного додатку Freeletics

Freeletics спеціалізується переважно на тренуваннях із власною вагою тіла, що дозволяє займатися без необхідності використання додаткового обладнання. Додаток пропонує структуровані плани занять, які змінюються в процесі тренувань залежно від результатів користувача. Кожне тренування супроводжується відеоінструкціями та поясненнями техніки виконання вправ, що сприяє правильному засвоєнню матеріалу.

До переваг додатку можна віднести високий рівень адаптивності. Програма аналізує результати тренувань і автоматично коригує навантаження, що дозволяє досягати поступового прогресу. Також важливою перевагою є можливість тренувань у будь-якому місці, оскільки більшість вправ не потребують спеціального інвентарю. Інтерфейс додатку є сучасним і зручним, а мотиваційні елементи, такі як досягнення та відстеження прогресу, стимулюють користувача до регулярних занять.

Серед недоліків варто відзначити обмежену глибину персоналізації. Хоча додаток адаптує тренування, він не враховує специфічні фізіологічні характеристики користувача, такі як соматотип або медичні показники. Частина функціоналу доступна лише за платною підпискою, що обмежує можливості безкоштовного використання. Також відсутній повноцінний контроль за правильністю виконання вправ у реальному часі, що може призводити до помилок у техніці. Крім того, додаток не орієнтований на вузькоспеціалізовані види спорту та не передбачає безпосередньої взаємодії з тренером.

Таким чином, Freeletics є ефективним інструментом для самостійних тренувань із базовою персоналізацією та високим рівнем автоматизації, однак має обмеження щодо глибини індивідуального підходу та професійного супроводу.

Мобільний додаток Fitbod є сучасним інструментом для планування та оптимізації тренувального процесу, орієнтованим насамперед на силові тренування у спортивному залі. Основною особливістю додатку є використання алгоритмів, які формують індивідуальні програми тренувань на основі фізичних показників користувача, його цілей та історії попередніх занять (рис. 1.4).

Fitbod аналізує навантаження на різні групи м’язів і враховує рівень їх відновлення, що дозволяє уникати перенавантаження та більш ефективно розподіляти тренувальний процес. Користувач може вказати доступне обладнання, після чого система підбирає відповідні вправи, що робить додаток зручним як для занять у повністю оснащеному залі, так і в умовах обмеженого інвентарю. Кожне тренування супроводжується описом вправ, а також рекомендаціями щодо кількості підходів і повторень.

До переваг додатку можна віднести досить високий рівень персоналізації порівняно з іншими фітнес-додатками. Програма автоматично адаптується до прогресу користувача, змінюючи інтенсивність і структуру тренувань. Важливою перевагою є врахування відновлення м’язів, що позитивно впливає на безпеку та ефективність тренувань. Також додаток має зручний інтерфейс і дозволяє вести облік результатів, що сприяє систематичності занять.

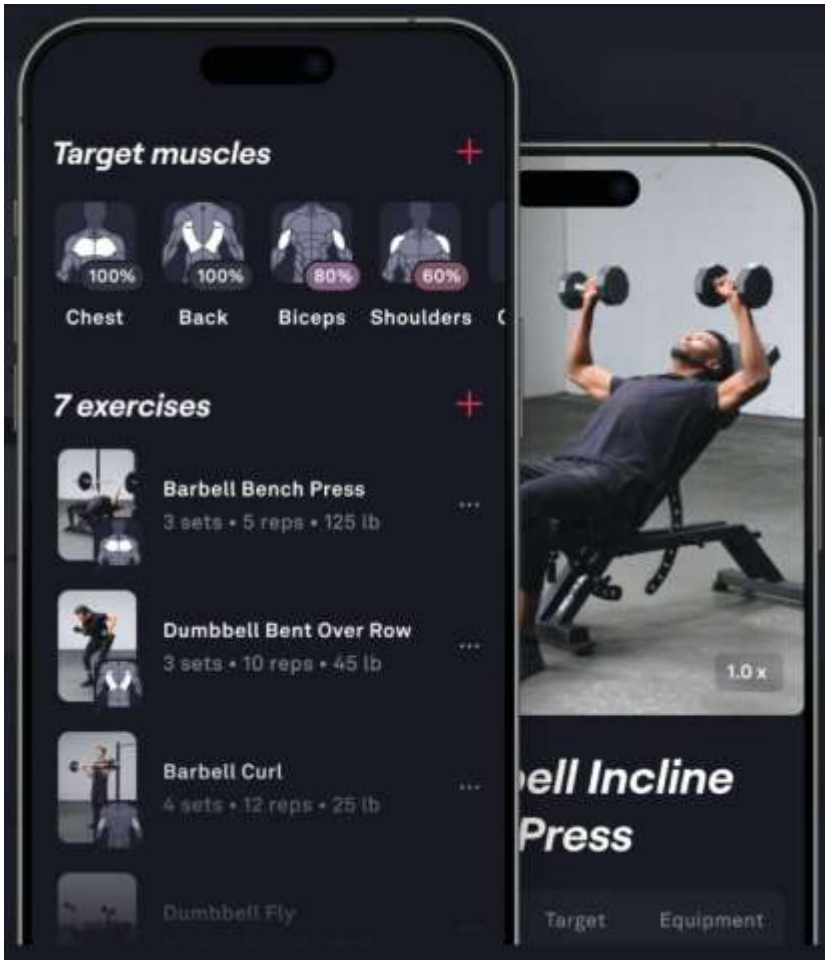


Рис. 1.3. Приклад мобільного додатку Fitbod

Водночас Fitbod має і певні недоліки. Значна частина функціоналу доступна лише за платною підпискою, що обмежує можливості безкоштовного використання. Для коректної роботи додатку користувач повинен регулярно вводити дані про виконані вправи, що може бути незручним і потребує додаткового часу. Крім того, додаток орієнтований переважно на силові тренування і не враховує специфіку окремих видів спорту. Відсутній також контроль правильності виконання вправ у реальному часі, що може призвести до технічних помилок. Для новачків певною складністю може бути необхідність розуміння базових принципів тренувального процесу.

Таким чином, Fitbod є ефективним інструментом для автоматизованого планування силових тренувань із достатнім рівнем персоналізації, проте має обмеження щодо універсальності, доступності функцій та контролю техніки виконання вправ.

Порівняльний аналіз додатків наведено в таблиці 1.1

1.1. Порівняльний аналіз мобільних додатків для тренувань

Додаток	Переваги	Недоліки
Heria Pro	Орієнтація на калістеніку та функціональні тренування; якісні відеоінструкції; структуровані програми різного рівня складності; зручний інтерфейс; можливість тренувань без обладнання	Обмежена персоналізація; відсутність глибокого аналізу фізичних параметрів; немає контролю техніки в реальному часі; частина функцій доступна лише за підпискою
Fitness Trainer FitProSport FULL	Велика база вправ; різноманітні тренувальні програми; можливість створення власних тренувань; використання як довідника вправ	Відсутність автоматичного підбору тренувань; немає адаптації під користувача; базовий рівень функціональності; відсутність аналітики
Freeletics	Адаптивні тренування; автоматична корекція навантаження; тренування без обладнання; мотиваційна система та відстеження прогресу	Більшість персоналізації платна; відсутній глибокий біометричний аналіз; обмежена спеціалізація під конкретні види спорту
Fitbod	Розумне формування тренувань; врахування відновлення м’язів; підтримка різного обладнання; автоматична адаптація навантаження; зручне відстеження прогресу	Потребує постійного введення даних; орієнтація переважно на силові тренування; більшість функцій платні; складність для новачків

Усі розглянуті додатки мають спільну рису - вони забезпечують певний рівень автоматизації або структуризації тренувального процесу, однак відрізняються глибиною персоналізації.

- Heria Pro та FitProSport FULL більше орієнтовані на готові програми та базовий контент.
- Freeletics забезпечує адаптацію навантаження, але без глибокого індивідуального аналізу.
- Fitbod є найбільш алгоритмічно розвиненим, але обмежений силовими тренуваннями.

Жоден із цих додатків не забезпечує повної інтеграції вузькоспеціалізованого підбору тренувань з урахуванням індивідуальних фізіологічних параметрів та специфіки конкретного виду спорту (наприклад, дитячого дзюдо), що підкреслює актуальність розробки нової системи.

1.4 Постановка задачі

Актуальність даної розробки зумовлена стрімким розвитком інформаційних технологій та зростаючою потребою у цифрових засобах автоматизації процесів у сфері спорту і фізичної підготовки. У сучасних умовах значна кількість людей прагне підтримувати фізичну форму, однак не завжди має доступ до кваліфікованих тренерів або достатній рівень знань для самостійного складання ефективних тренувальних програм. Це створює потребу у створенні програмних рішень, які можуть автоматизувати процес підбору вправ та забезпечити індивідуальний підхід до кожного користувача.

Оскільки основним завданням розробки є створення веб-сайту для автоматизації тренувального процесу, визначення актуальності проєкту базується на необхідності вирішення ряду ключових задач, серед яких:

– аналіз потреб потенційних користувачів системи з метою визначення функціональних вимог до програмного забезпечення;

– дослідження існуючих програмних продуктів і мобільних додатків із подібною або вузькоспеціалізованою функціональністю;

– порівняльний аналіз існуючих аналогів із запропонованим рішенням для визначення його переваг та інноваційності;

– оцінка можливостей різних технологічних платформ щодо реалізації веб-системи з необхідним рівнем продуктивності та доступності;

– проєктування та розробка веб-сайту, який забезпечує автоматизований підбір тренувальних програм на основі введених користувачем даних.

Додатково важливим аспектом є забезпечення універсальності та доступності системи, що дозволить використовувати її на різних пристроях без необхідності встановлення спеціального програмного забезпечення. Це підвищує практичну цінність розробки та робить її зручною для широкого кола користувачів.

Таким чином, реалізація даного проєкту є актуальною як з точки зору розвитку інформаційних технологій, так і з позиції підвищення ефективності організації спортивних тренувань шляхом їх автоматизації та індивідуалізації.

РОЗДІЛ 2 ПРОЄКТУВАННЯ АВТОМАТИЗАЦІЇ ІНДИВІДУАЛЬНОГО ПІДБОРУ ТРЕНУВАНЬ ДЛЯ СПОРТСМЕНІВ З УРАХУВАННЯМ АНТРОПОМЕТРІЇ

2.1 Обґрунтування технологій розробки

Даний проєкт було реалізовано таким чином, щоб інтерфейс веб-сайту мав вигляд єдиної форми для введення даних із максимально компактним розміщенням елементів. Незважаючи на наявність великих інформаційних блоків, їх перегляд здійснюється за допомогою прокручування сторінки, що забезпечує зручну навігацію та логічну структуру подачі інформації.

Користувач, який вперше відвідує веб-ресурс, отримує коротке пояснення щодо його призначення, а також детальні підказки до кожного параметра, що вводиться у систему. Це дозволяє уникнути помилок під час заповнення форми та робить процес використання більш інтуїтивно зрозумілим.

Зазначені рішення спрямовані на підвищення зручності та доступності системи для широкого кола користувачів, зокрема для батьків, які планують залучити своїх дітей до занять спортивними єдиноборствами. Завдяки цьому кожен користувач отримує можливість об’єктивно оцінити фізичні можливості дитини та сформувати базу для подальшого розвитку та підготовки до спортивних занять (рис. 2.1).

Для розробки було обрано такі мови програмування: HTML, CSS, Javascript.

В мові Javascript, було обрано скористатися бібліотекою JQuery[18] для полегшення і покращення роботи кінцевого продукту.

Розберемо коротко історію, переваги і недоліки всіх використаних мов та фреймворків.

Перша версія HTML (HyperText Markup Language) була розроблена Тімом Бернерсом-Лі у 1993 році як базова мова для створення та структуризації веб-сторінок у межах концепції Всесвітньої павутини. Відтоді HTML пройшов

значний шлях розвитку та зазнав численних оновлень, кожне з яких розширювало його можливості та покращувало стандартизацію веб-технологій.

Однією з найпоширеніших версій у 2000-х роках стала HTML 4.01, яка була офіційно затверджена консорціумом W3C у 1999 році. Вона довгий час залишалася основним стандартом для створення веб-сторінок і широко використовувалася розробниками у всьому світі.

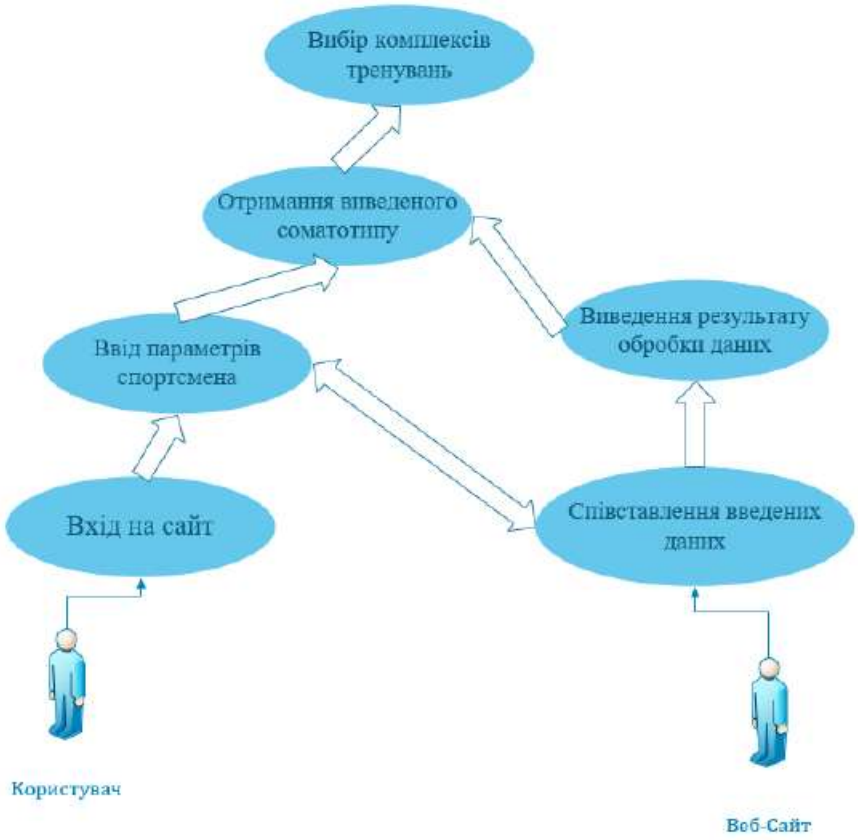


Рис. 2.1. UML-діаграма варіантів використання

Паралельно з розвитком HTML було створено XHTML (Extensible HyperText Markup Language), який є реформатованою версією HTML на основі XML (Extensible Markup Language). XML використовується як універсальна мова для створення інших мов розмітки, таких як MathML, SVG, RSS та інші. Завдяки строгій структурі XML забезпечується краща сумісність між різними системами та платформами, що робить обмін даними більш стандартизованим і надійним.

XHTML був офіційно рекомендований W3C у 2000 році та оновлений у 2002 році. Він мав суворіші правила синтаксису порівняно з HTML, що забезпечувало

більш коректну обробку документів різними програмними засобами. Однак через надмірну строгость XHTML не набув такого широкого поширення, як очікувалося, оскільки розробники віддавали перевагу більш гнучкому HTML.

Починаючи з 2004 року, консорціум W3C спільно з організацією WHATWG (Web Hypertext Application Technology Working Group) розпочав розробку HTML5 — нового покоління мови розмітки, орієнтованого на сучасні веб-додатки. HTML5 включає підтримку мультимедіа без використання сторонніх плагінів, покращену семантичну структуру, інтеграцію з JavaScript API, а також можливості для створення веб-додатків, які працюють офлайн.

Станом на сьогодні HTML5 є основним стандартом веб-розробки. Він офіційно рекомендований W3C з 2014 року і постійно оновлюється через WHATWG як “живий стандарт” (living standard), що означає безперервний розвиток без фіксованих версій. Сучасні браузери повністю підтримують HTML5 та його розширення, що робить його фундаментальною технологією для створення веб-сайтів і веб-додатків у 2020-х роках.

Таким чином, HTML пройшов еволюцію від простої мови розмітки до потужної основи сучасного вебу, яка постійно вдосконалюється відповідно до потреб новітніх технологій та користувачів.

HTML має низку суттєвих переваг, які забезпечили йому статус базової технології веб-розробки. Насамперед, HTML є дуже простою для вивчення мовою розмітки. Вона характеризується зрозумілою структурою тегів і не потребує глибоких знань програмування для початкового використання. Код HTML легко читається, що дозволяє швидко розуміти чужі проєкти та вносити необхідні зміни. Крім того, мова є досить «поблажливою» до помилок — незначні неточності, наприклад незакриті теги, зазвичай не призводять до критичних збоїв у роботі сторінки.

Важливою перевагою HTML є його безкоштовність та відкритість. Для роботи з ним не потрібне спеціальне програмне забезпечення або ліцензії, достатньо звичайного текстового редактора. Це робить технологію економічно вигідною для розробників та компаній будь-якого рівня.

HTML підтримується всіма сучасними браузерами, що забезпечує його кросплатформеність і стабільне відображення веб-сторінок на різних пристроях. Завдяки цьому розробникам не потрібно створювати окремі версії сайтів для різних браузерів, хоча оптимізація все ж залишається важливим етапом.

Ще однією перевагою HTML є його висока сумісність із пошуковими системами. Завдяки правильно структурованій розмітці веб-сторінки легше індексуються пошуковими роботами, що позитивно впливає на SEO-оптимізацію та швидкість завантаження сторінок. Крім того, HTML легко інтегрується з іншими технологіями, такими як CSS, JavaScript, PHP, Node.js та іншими, що дозволяє створювати повноцінні веб-додатки.

HTML також є легкою мовою з точки зору обсягу коду, що сприяє швидкому завантаженню сторінок і ефективній роботі веб-сайтів. Результати змін у коді можна побачити миттєво після оновлення сторінки, що значно спрощує процес розробки та тестування. Простота редагування дозволяє працювати з HTML-файлами навіть у базових текстових редакторах без використання складних середовищ розробки.

Окремо слід відзначити, що HTML є фундаментальною технологією веб-розробки. Знання HTML є обов’язковим для фронтенд-розробників і корисним навіть для бекенд-спеціалістів, оскільки він використовується як основа для інтеграції з іншими мовами та технологіями.

Водночас HTML має і певні недоліки. Насамперед, це статична мова розмітки, яка не забезпечує динамічної поведінки веб-сторінок без використання додаткових технологій, таких як JavaScript. Сам по собі HTML не призначений для обробки даних або реалізації логіки, що обмежує його функціональність.

Ще одним недоліком є обмежений рівень безпеки, оскільки HTML не містить вбудованих механізмів захисту даних. Також при створенні складних веб-сторінок може виникати значний обсяг коду, що ускладнює його підтримку та масштабування.

Крім того, розробники змушені дотримуватися стандартів мови, що обмежує свободу реалізації нестандартних рішень без використання додаткових

технологій. Таким чином, HTML потребує обов’язкового поєднання з іншими мовами програмування та інструментами для створення повноцінних сучасних веб-додатків.

Історія появи CSS (Cascading Style Sheets) значною мірою пов’язана з норвезьким розробником Хоконом Віумом Лі, який у 1994 році запропонував концепцію універсальної мови для опису стилів веб-сторінок. Його ідея полягала у відокремленні структури документа (HTML) від його візуального оформлення, що дозволяло значно підвищити гнучкість і керованість веб-дизайну.

Перші практичні експерименти з CSS були реалізовані у браузері Arena, де Лі вперше протестував свою концепцію стилів. Згодом він разом із Тімом Бернерсом-Лі та Робертом Кайо брав участь у розробці та стандартизації перших специфікацій CSS, включаючи CSS1 (1996) та CSS2 (1998), які були затверджені консорціумом W3C. У цей період закладалися основи сучасної веб-візуалізації, що суттєво вплинуло на розвиток інтернет-дизайну та доступності веб-ресурсів.

Подальший розвиток технології привів до появи CSS3, робота над яким розпочалася наприкінці 1990-х років і який поступово впроваджувався як набір модульних специфікацій. На відміну від попередніх версій, CSS3 не є монолітним стандартом, а складається з окремих модулів (анімації, градієнти, flexbox, grid тощо), які розвивалися та впроваджувалися поступово. Саме ці можливості зробили можливим створення сучасних адаптивних і інтерактивних інтерфейсів.

Станом на сьогодні CSS є одним із ключових стандартів веб-розробки разом із HTML та JavaScript. Сучасні браузери повністю підтримують CSS3 та його модулі, а сама специфікація розвивається як «живий стандарт» у рамках роботи організації WHATWG та W3C. Це означає, що CSS постійно оновлюється без фіксованої фінальної версії, адаптуючись до нових вимог веб-технологій.

Важливим етапом розвитку CSS стало його широке впровадження у сфері адаптивного дизайну, мобільної оптимізації та веб-доступності. Підтримка різних пристроїв, екранів і форматів відображення зробила CSS незамінним інструментом для створення сучасних інтерфейсів.

CSS пройшов шлях від експериментальної ідеї до фундаментальної технології вебу, яка визначає зовнішній вигляд більшості сучасних веб-сайтів і веб-додатків.

CSS (Cascading Style Sheets) має низку суттєвих переваг, які зробили його невід'ємною частиною сучасної веб-розробки. Насамперед, CSS значно покращує користувацький досвід (UX). При використанні лише HTML веб-сторінки виглядають примітивно і не забезпечують належного візуального комфорту. Натомість CSS дозволяє створювати привабливі, структуровані та інтуїтивно зрозумілі інтерфейси. Завдяки підтримці анімацій, переходів та адаптивного дизайну користувач отримує більш сучасний і зручний досвід взаємодії з веб-ресурсом, що безпосередньо впливає на утримання аудиторії.

Важливою перевагою CSS є принцип централізованого управління стилями. Один файл стилів може застосовуватися до великої кількості сторінок, що дозволяє змінювати зовнішній вигляд всього веб-сайту шляхом редагування лише одного місця в коді. Це значно спрощує підтримку проєктів і забезпечує єдність стилістичного оформлення на всіх сторінках.

Окремо слід відзначити високу ефективність CSS у контексті економії часу розробки. Оскільки немає потреби вручну змінювати кожну сторінку окремо, процес оновлення дизайну стає значно швидшим і зручнішим. Крім того, можливість повторного використання стилів на різних сторінках сприяє стандартизації оформлення веб-ресурсу.

CSS також позитивно впливає на продуктивність веб-сайтів. Скорочення обсягу HTML-коду та винесення стилів у зовнішні файли зменшує розмір сторінок, що пришвидшує їх завантаження. Це, у свою чергу, позитивно впливає на SEO-оптимізацію, оскільки пошукові системи враховують швидкість завантаження сторінок і якість їх структури при формуванні рейтингу.

Ще однією перевагою є спрощення процесу розробки та супроводу веб-додатків. Використання CSS дозволяє розділити логіку структури (HTML) та оформлення (CSS), що робить код більш організованим, читабельним і легким у

підтримці. Це особливо важливо при роботі над великими проєктами, де задіяно кілька розробників.

CSS є ключовою технологією для реалізації адаптивного дизайну, що забезпечує коректне відображення веб-сторінок на різних пристроях і екранах — від смартфонів до настільних комп'ютерів. Завдяки медіа-запитам (media queries) розробник може створювати інтерфейси, які автоматично підлаштовуються під роздільну здатність екрана користувача.

Також CSS дозволяє використовувати зовнішні стилі, що сприяє «чистоті» HTML-коду. Це позитивно впливає на індексацію сторінок пошуковими системами, оскільки структурований і менш перевантажений код легше аналізується пошуковими роботами. У результаті це може покращити позиції веб-сайту у видачі пошукових систем.

CSS є потужним інструментом, який не лише покращує зовнішній вигляд веб-сторінок, але й підвищує ефективність розробки, продуктивність сайтів та якість взаємодії з користувачем.

Попри значні переваги CSS, дана технологія має і певні недоліки, які необхідно враховувати при розробці веб-ресурсів. Однією з ключових проблем є питання безпеки. Оскільки CSS є відкритою технологією, будь-який користувач із доступом до коду може вносити зміни у стилі сторінки. Це потенційно може призвести до порушення візуальної цілісності веб-сайту або небажаних змін у його відображенні. Крім того, CSS не має вбудованих механізмів захисту, що робить його вразливим у контексті сучасних вимог кібербезпеки.

Ще одним недоліком є наявність різних версій і специфікацій CSS, таких як CSS1, CSS2 та CSS3, а також модульний розвиток сучасних стандартів. Різниця між цими версіями іноді викликає плутанину, особливо у початківців, оскільки різні функції підтримуються нерівномірно в різних середовищах. Це ускладнює вибір оптимального підходу для реалізації конкретного проєкту.

Важливою проблемою також є кросбраузерна сумісність. Незважаючи на те, що сучасні браузери (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) здебільшого підтримують сучасні стандарти CSS, все ще існують відмінності в

інтерпретації стилів. Це вимагає додаткового тестування та адаптації коду для забезпечення коректного відображення веб-сторінок у різних середовищах. У разі недостатнього тестування можуть виникати проблеми з відображенням інтерфейсу, що негативно впливає на якість користувацького досвіду.

Ще одним обмеженням є залежність від версій браузерів. Старі або застарілі браузери можуть не підтримувати сучасні можливості CSS, через що дизайн веб-сторінки відображається некоректно або спрощено. У таких випадках користувачі бачать спрощену версію сайту, що знижує його візуальну привабливість і може негативно впливати на загальне сприйняття ресурсу.

Хоча CSS є потужним інструментом для оформлення веб-сторінок, він має низку технічних і практичних обмежень, пов’язаних із безпекою, сумісністю та підтримкою різних версій браузерів.

JavaScript є однією з ключових мов програмування для веб-розробки та входить до трійки основних технологій сучасного вебу поряд із HTML і CSS. Якщо HTML відповідає за структуру веб-сторінки, а CSS — за її візуальне оформлення, то JavaScript забезпечує інтерактивність, динамічну поведінку та логіку роботи веб-додатків. Завдяки йому користувач може взаємодіяти з елементами сторінки, отримуючи зміну контенту без необхідності перезавантаження сторінки.

Спочатку JavaScript був створений як клієнтська мова, тобто виконувався безпосередньо у браузері користувача. Однак із появою середовища Node.js його можливості значно розширилися, і сьогодні JavaScript активно використовується також на серверній стороні, що дозволяє створювати повноцінні веб-застосунки за єдиним технологічним стеком.

З моменту появи JavaScript швидко набув популярності завдяки своїй доступності, відносній простоті вивчення та великій спільноті розробників з відкритим вихідним кодом. Він став однією з найпоширеніших мов програмування у світі, випередивши багато інших технологій, зокрема Java та Flash у сфері веб-розробки. Його популярність пояснюється також високою універсальністю та можливістю швидкої розробки масштабованих застосунків.

JavaScript широко використовується у веб-екосистемі: за різними оцінками, він застосовується на переважній більшості сучасних веб-сайтів, включаючи великі платформи, такі як Meta Platforms (Facebook), Google (зокрема Google Maps) та eBay. Його роль у розвитку сучасного інтернету є критичною, оскільки без JavaScript неможливо уявити інтерактивні веб-додатки, онлайн-сервіси та динамічні інтерфейси.

У сучасній розробці JavaScript займає провідні позиції серед мов програмування. За даними різних аналітичних досліджень та репозиторіїв коду, зокрема GitHub, він стабільно залишається однією з найпопулярніших мов у світі за кількістю проєктів і активних розробників. Це пояснюється постійним розвитком екосистеми JavaScript, появою нових фреймворків і бібліотек (React, Angular, Vue.js), а також розширенням сфер його застосування — від веб-інтерфейсів до мобільних і серверних додатків.

JavaScript є фундаментальною технологією сучасного вебу, яка забезпечує інтерактивність, динамічність і високий рівень функціональності веб-застосунків, продовжуючи активно розвиватися та розширювати свої можливості.

JavaScript має низку суттєвих переваг, які забезпечили йому провідну роль у сучасній веб-розробці. Насамперед, однією з головних його переваг є висока швидкодія. Оскільки код JavaScript зазвичай виконується безпосередньо у браузері користувача, він не потребує постійних звернень до сервера, що зменшує затримки та підвищує продуктивність. Сучасні браузери також використовують технології JIT-компіляції (Just-In-Time), які дозволяють виконувати код майже миттєво без попередньої компіляції.

Ще однією важливою перевагою є відносна простота синтаксису. JavaScript був частково натхненний мовою Java, проте є значно більш гнучким і легшим для засвоєння у порівнянні з такими мовами, як C++ або Java, що робить його доступним для початківців у програмуванні.

JavaScript має надзвичайно широку сферу застосування. Він використовується практично на всіх сучасних веб-сайтах, а з появою середовища

Node.js його можливості розширилися до серверної розробки. Завдяки великій кількості навчальних матеріалів, активній спільноті розробників та численним відкритим проєктам на платформах, таких як GitHub і Stack Overflow, JavaScript залишається однією з найпопулярніших мов програмування у світі.

Важливою перевагою є також універсальність використання. JavaScript можна інтегрувати у будь-яку веб-сторінку та поєднувати з іншими мовами програмування, такими як PHP, Python або Perl, що робить його гнучким інструментом для створення складних веб-систем.

Оскільки JavaScript виконується на стороні клієнта, він значно зменшує навантаження на сервер, що особливо важливо для масштабованих веб-додатків. У деяких випадках прості веб-застосунки можуть навіть не вимагати постійної серверної взаємодії.

JavaScript також дозволяє створювати інтерактивні елементи інтерфейсу користувача, такі як перетягування елементів, слайдери, модальні вікна та інші динамічні компоненти. Це суттєво покращує користувацький досвід і робить веб-сайти більш сучасними та зручними.

Окремо варто відзначити можливість розширення функціональності веб-сторінок за допомогою сторонніх скриптів і доповнень, що дозволяє розробникам створювати додаткові інструменти для браузерів і веб-додатків.

З розвитком платформи Node.js JavaScript став повноцінною мовою як для фронтенд-, так і для бекенд-розробки. Використовуючи сучасні технологічні стекі, такі як Node.js, Express та MongoDB, можна створювати повноцінні застосунки виключно на JavaScript, що спрощує розробку та підтримку проєктів.

Крім того, JavaScript постійно розвивається. Починаючи з ECMAScript 5, стандарт мови регулярно оновлюється, і сьогодні ECMA International щорічно впроваджує нові можливості, що підвищують продуктивність, безпеку та зручність використання мови.

Попри значні переваги JavaScript, дана мова програмування має також ряд недоліків, які необхідно враховувати під час розробки веб-додатків. Однією з головних проблем є питання безпеки. Оскільки JavaScript-код виконується на

стороні клієнта і є доступним для перегляду користувачем, зловмисники можуть аналізувати або використовувати його у шкідливих цілях. Це включає можливість несанкціонованого доступу до логіки роботи сайту, підміни скриптів або впровадження шкідливого коду, що може загрожувати безпеці даних.

Ще одним суттєвим недоліком є проблема кросбраузерної сумісності. Різні браузери можуть по-різному інтерпретувати JavaScript-код, що вимагає додаткового тестування та адаптації програмного забезпечення. Крім того, старі версії браузерів часто не підтримують сучасні можливості мови, що може призводити до некоректної роботи веб-додатків або втрати частини функціональності.

JavaScript також має обмеження у сфері налагодження та обробки помилок. Хоча сучасні середовища розробки та браузери надають інструменти для дебагінгу, вони не завжди є настільки зручними та потужними, як у традиційних мовах програмування, наприклад C++ або Java. Додатковою складністю є те, що не всі помилки одразу відображаються у вигляді явних повідомлень, що ускладнює їх виявлення.

До технічних обмежень JavaScript можна віднести відсутність повноцінної підтримки множинного наслідування в об’єктно-орієнтованому програмуванні. Хоча сучасні версії мови та механізми прототипного наслідування частково компенсують це обмеження, у деяких складних системах це може створювати певні труднощі при проєктуванні архітектури програмного забезпечення.

Також варто відзначити особливості роботи з числовими даними. JavaScript використовує 64-бітне представлення чисел з плаваючою комою, тоді як побітові операції виконуються над 32-бітними значеннями. Це призводить до додаткових перетворень типів даних під час виконання операцій, що може впливати на продуктивність у високонавантажених обчислювальних задачах.

Ще одним недоліком є те, що помилка в JavaScript-коді може призвести до зупинки виконання скрипта на сторінці. У деяких випадках це створює враження некоректної роботи або відсутності функціональності. Хоча сучасні браузери є досить «толерантними» до помилок і намагаються продовжувати виконання

інших частин коду, це не завжди дозволяє повністю уникнути негативного впливу на користувацький досвід.

Таким чином, JavaScript, попри свою потужність і універсальність, має певні обмеження, пов’язані з безпекою, сумісністю, продуктивністю та особливостями виконання коду, які необхідно враховувати при розробці сучасних веб-систем.

jQuery є однією з найвідоміших і найпоширеніших бібліотек JavaScript, яка тривалий час відігравала ключову роль у розвитку веб-розробки. Особливо активно вона використовувалася наприкінці 2000-х років, коли сформувалася велика екосистема веб-сайтів, плагінів і фреймворків, що базувалися на її можливостях. Протягом багатьох років jQuery фактично став стандартом де-факто для спрощення роботи з JavaScript у браузері.

Бібліотека була створена Джоном Резігом у 2005 році та вперше представлена публічно у 2006 році на конференції BarCamp NYC. Її поява була відповіддю на складність роботи з «чистим» JavaScript у той час, коли різні браузери мали значні відмінності у реалізації стандартів.

Основною перевагою jQuery став зручний та інтуїтивно зрозумілий API для роботи з DOM (Document Object Model). Бібліотека значно спростила маніпуляції з елементами веб-сторінок, зокрема їх пошук, зміну та обробку подій. На відміну від стандартного JavaScript того часу, jQuery дозволяв використовувати складні селектори для вибору елементів, включаючи логічні та структурні зв’язки між ними. Згодом механізм вибору елементів був виокремлений у незалежну бібліотеку Sizzle, що ще більше підвищило гнучкість системи.

Другою важливою перевагою jQuery була кросбраузерна сумісність. У період його активного розвитку веб-розробники стикалися з великою кількістю відмінностей у роботі браузерів, і jQuery абстрагував ці розбіжності, забезпечуючи однакову поведінку коду в різних середовищах. Це значно зменшувало складність розробки та тестування веб-додатків.

З часом jQuery став настільки популярним, що навколо нього сформувалася організаційна структура управління проєктом. У 2011 році була створена jQuery

Board, а у 2012 році — jQuery Foundation. Надалі відбулося об’єднання з Dojo Foundation у 2015 році, що призвело до створення JS Foundation. У 2019 році JS Foundation об’єдналася з Node.js Foundation, утворивши OpenJS Foundation, де jQuery і сьогодні залишається одним із ключових проєктів екосистеми JavaScript.

У сучасній веб-розробці роль jQuery дещо зменшилася через розвиток сучасних фреймворків і можливостей самого JavaScript (ES6+), однак бібліотека все ще використовується у великій кількості існуючих проєктів та залишається важливою частиною історії розвитку веб-технологій.

jQuery має низку суттєвих переваг, які забезпечили його широке використання у веб-розробці протягом тривалого часу. Однією з ключових переваг є кросбраузерна сумісність. Бібліотека забезпечує однакову поведінку коду в різних сучасних браузерах, що дозволяє розробникам бути впевненими у коректному відображенні інтерфейсу для всіх користувачів незалежно від платформи або браузера.

Ще однією важливою перевагою є компактність та читабельність коду. Багато операцій, які у «чистому» JavaScript потребують написання значної кількості рядків коду, у jQuery реалізуються значно простіше — іноді всього в декілька рядків. Це робить код більш структурованим, зрозумілим і легким для супроводу, що особливо важливо при розробці великих проєктів.

jQuery також забезпечує зручну роботу з подіями та візуальними ефектами. За його допомогою легко реалізувати інтерактивні елементи інтерфейсу, такі як підказки при наведенні курсора, анімації, динамічні переходи, ефекти завантаження сторінки або обробку дій користувача. Це значно спрощує створення інтерактивного користувацького досвіду без необхідності написання складної логіки.

Окремо слід відзначити наявність добре структурованої та детальної документації. Офіційні ресурси jQuery містять повний опис функцій бібліотеки, приклади використання та практичні рекомендації, що значно полегшує навчання та впровадження технології у реальні проєкти.

Важливою перевагою є також велика кількість готових плагінів і розширень. У відкритому доступі існує безліч рішень для реалізації типових задач, таких як галереї зображень, слайдери, модальні вікна та інші компоненти інтерфейсу. Це дозволяє значно скоротити час розробки, використовуючи вже готові рішення.

Попри свої переваги, jQuery має і певні недоліки. Одним із них є продуктивність. У порівнянні з сучасними JavaScript-фреймворками, такими як React або Vue.js, jQuery може поступатися у швидкодії при складних та масштабних застосунках. Хоча різниця у продуктивності у сучасних умовах часто мінімальна через потужність апаратного забезпечення та високу швидкість інтернет-з’єднання, у великих проєктах це може мати значення.

Ще одним недоліком є відносно невеликий розмір бібліотеки, який становить приблизно 20 КБ у стисненому вигляді. Хоча це значення є незначним для сучасних умов, у минулому воно могло впливати на швидкість завантаження сторінок, особливо при повільному інтернет-з’єднанні або використанні застарілих пристроїв.

jQuery є потужною, зручною та історично важливою бібліотекою JavaScript, яка значно спростила веб-розробку, однак у сучасних умовах частково поступається новішим технологіям, що пропонують більш гнучкі та продуктивні підходи до створення веб-додатків.

2.2 UML-Діаграми

Уніфікована мова моделювання [20] (UML) - це мова маркування або побудови графіків, призначена для визначення, візуалізації та документування програмно-орієнтованих об’єктів моделі. UML не є методом розробки, іншими словами, структура цієї мови не говорить вам, що робити спочатку, а що робити в останню чергу, і не дає інструкцій, як створити свою мову, але ця мова чітко враховує макет і робота з іншими. Розробник. UML був розроблений для розробки структури, орієнтованої на програмні об’єкти. Ця мова має дуже обмежені можливості для програмування на основі інших парадигм.

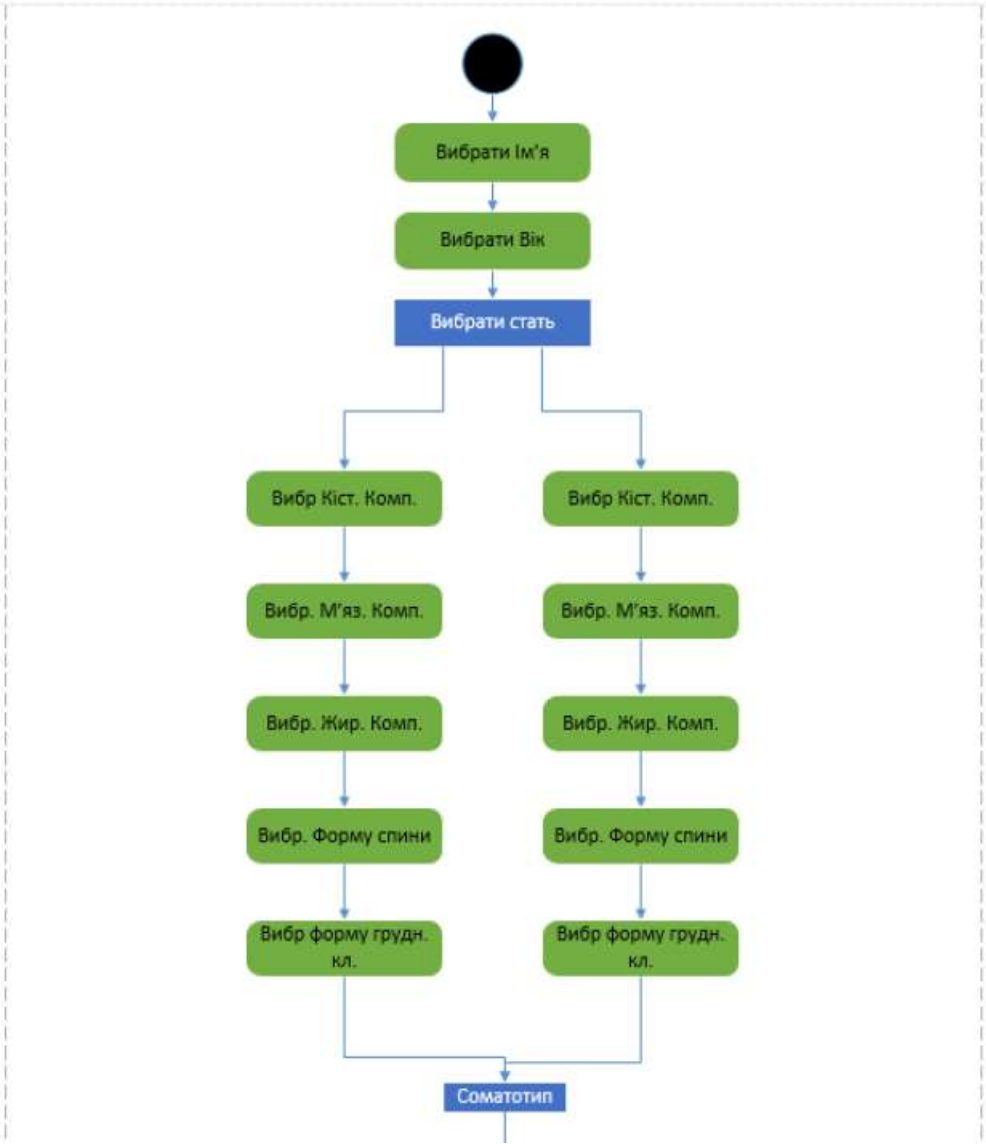


Рис. 2.2 Перша частина діаграми, що описує алгоритм роботи з сайтом/додатком

Діаграма відображає алгоритм визначення соматотипу користувача в рамках інформаційної системи підбору тренувальних комплексів. Процес починається з ініціалізації (стартова точка) та послідовного введення базових персональних даних: користувач обирає ім’я, вік та стать. Ці параметри виступають початковими вхідними даними для подальшого аналізу.

Після вибору статі відбувається розгалуження алгоритму на дві умовні гілки, що відповідають різним фізіологічним особливостям (умовно чоловічий та жіночий варіанти обробки). У кожній з гілок виконується послідовний набір кроків оцінювання антропометричних характеристик.

Спочатку здійснюється визначення кісткової комплекції, яка характеризує загальну будову скелета. Далі аналізується м’язова комплекція, що відображає рівень розвитку м’язової маси. Наступним етапом є оцінювання жирової комплекції, яка дозволяє врахувати особливості тілобудови та відсоток жирової тканини.

Після цього виконується визначення форми спини, що є важливим показником постави та особливостей опорно-рухового апарату. Завершальним етапом є вибір форми грудної клітки, яка також впливає на загальний тип тілобудови та функціональні можливості організму.

Після проходження всіх етапів аналізу результати з обох гілок зводяться до фінального блоку - визначення соматотипу. Отриманий соматотип використовується як основа для подальшого формування індивідуального комплексу фізичних вправ і тренувальних рекомендацій, що дозволяє забезпечити персоналізований підхід до тренувального процесу.

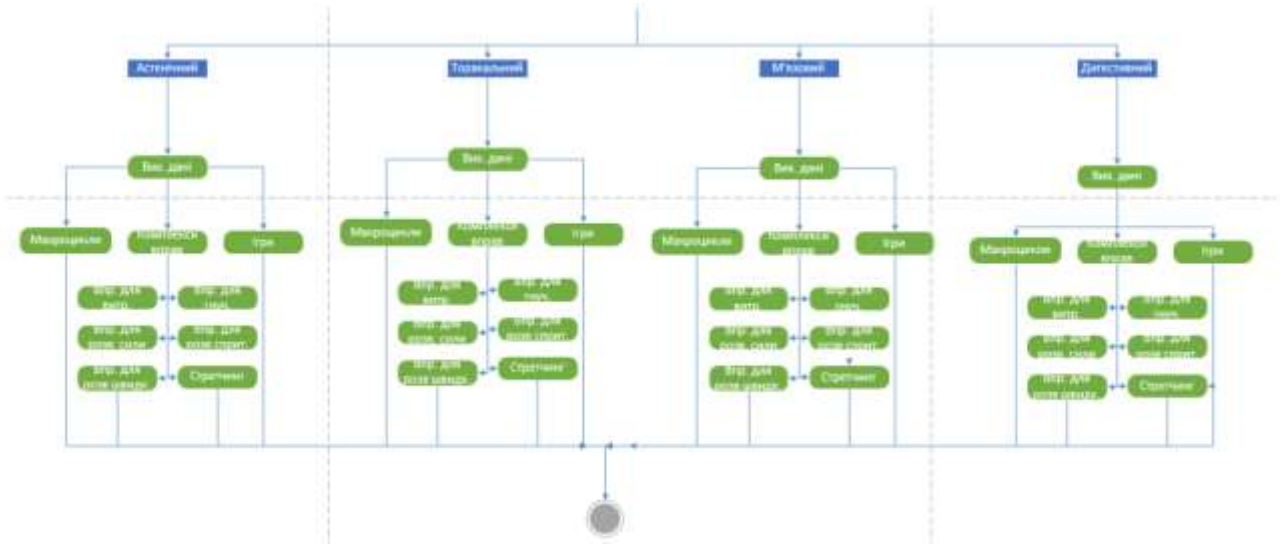


Рис. 2.3 Друга частина діаграми, що описує алгоритм роботи з сайтом

Діаграма відображає структуру формування індивідуальних тренувальних рекомендацій залежно від визначеного соматотипу користувача. У верхній частині представлено чотири основні типи тілобудови: астенічний, торакальний, м’язовий та дигестивний. Кожен із цих типів виступає окремою гілкою алгоритму, але загальна логіка їх обробки є подібною.

Після визначення соматотипу система переходить до етапу використання вихідних даних, на основі яких формується подальша структура тренувального процесу. Для кожного типу тілобудови передбачено три основні напрями формування навантаження: макроцикли, комплекси вправ та ігрові елементи.

У блоці макроциклів визначаються загальні тренувальні акценти, що включають вправи на розвиток витривалості, сили та швидкісних якостей. Це дозволяє сформувати довгострокову структуру фізичної підготовки відповідно до особливостей організму.

Блок комплексів вправ деталізує тренувальний процес і включає більш конкретні вправи, спрямовані на розвиток окремих фізичних якостей, зокрема витривалості, сили та швидкості. Окремо також враховується стретчинг як елемент розвитку гнучкості та профілактики травм.

Третій блок - ігри - містить вправи ігрового характеру, які спрямовані на розвиток загальної фізичної підготовки, координації та швидкості реакції. Зокрема, використовуються вправи для розвитку витривалості, сили та швидкісних якостей у формі рухливих ігор, включаючи елементи стретчингу.

Після проходження всіх трьох блоків інформація з кожної гілки зводиться до єдиного результату, який формує підсумковий комплекс тренувань. Таким чином, діаграма демонструє логіку персоналізованого підбору фізичних навантажень на основі соматотипу з урахуванням різних форматів тренувальної діяльності.

Оскільки сайт створений за допомогою мов програмування HTML, CSS, Javascript, то його функціонал та дизайн безмежно може покращуватися і розширятися, заміняючи попередні варіанти прямо на хостингу, у випадку інтеграції в інтернет. Тому не потрібно буде користувачам завантажувати оновлені версії додатку для тренувань, це значно полегшить експлуатацію продукту. В майбутньому буде додана функція запису даних про організм спортсмена взятих з “OMRON BF 511”, що допоможе точніше визначати нормування вправ.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1 Процес користування сайтом

У процесі розробки проєкту було використано програму-конструктор інтерфейсів та макетів сайтів Figma, за допомогою якої створено попередній (чорновий) варіант веб-проєкту (Рисунок 3.1).



Рис. 3.1 Варіант веб-сторінки

Figma є сучасним веб-орієнтованим інструментом для дизайну інтерфейсів та прототипування. Вона дозволяє виконувати широкий спектр завдань у сфері графічного дизайну, зокрема розробку макетів веб-сайтів, мобільних додатків, інтерактивних прототипів, а також створення візуального контенту для цифрових платформ і соціальних мереж.

Відмінною особливістю Figma є її робота у браузері, що забезпечує кросплатформенність та доступ до проєктів з будь-якого пристрою без необхідності встановлення додаткового програмного забезпечення. Це значно

спрощує процес розробки та дозволяє працювати над проєктом у режимі реального часу з різних пристроїв або в команді.

Додатковою перевагою є наявність безкоштовного тарифного плану, який дозволяє одночасно створювати та зберігати до трьох активних проєктів. Цього функціоналу достатньо для навчальних цілей, експериментів та реалізації невеликих і середніх за складністю проєктів.

У представленому макеті стрілками вниз позначено поля для вибору оціночних параметрів спортсмена відповідно до заданих критеріїв. Числові значення біля кожного параметра відображають кількість доступних варіантів вибору, з яких користувач повинен обрати один. Після заповнення всіх необхідних полів у блоці «Соматотип» автоматично відображається результат класифікації користувача у вигляді одного з чотирьох можливих типів тілобудови, сформованого на основі введених даних.

У правій частині інтерфейсу передбачено меню, яке дозволяє обрати відповідні комплекси вправ, макроцикли тренувань та ігрові елементи. Усі ці елементи підбираються індивідуально відповідно до визначеного соматотипу спортсмена, що забезпечує персоналізований підхід до формування тренувального процесу.

Процес роботи користувача з веб-системою передбачає послідовне виконання низки кроків, спрямованих на збір вихідних даних та подальше формування індивідуального результату:

- користувач вводить ім’я досліджуваної особи (це може бути як сам користувач, так і його учень або інша особа);
- здійснюється введення вікових даних, що дозволяє врахувати вікові особливості організму при подальшому аналізі;
- користувач обирає стать досліджуваного, що є важливим фактором для коректного визначення фізіологічних характеристик;
- виконується оцінювання кісткового компоненту тілобудови з подальшим вибором відповідного варіанту із запропонованих;

- проводиться аналіз м’язового компоненту, який відображає рівень розвитку м’язової маси, після чого користувач обирає найбільш відповідний варіант;
- виконується оцінювання жирового компоненту організму з вибором відповідного рівня;
- здійснюється аналіз форми спини, що дозволяє врахувати особливості постави та будови опорно-рухового апарату;
- проводиться оцінка форми грудної клітки як одного з ключових антропометричних параметрів;
- після введення та оцінювання всіх параметрів система автоматично обробляє дані та формує результат у вигляді визначеного соматотипу обстежуваного;
- користувач отримує доступ до другого блоку системи, який містить вихідні дані та рекомендації;
- на завершальному етапі користувач обирає один із представлених варіантів (зазвичай до восьми категорій), що відповідають персоналізованим тренувальним програмам та комплексам вправ.

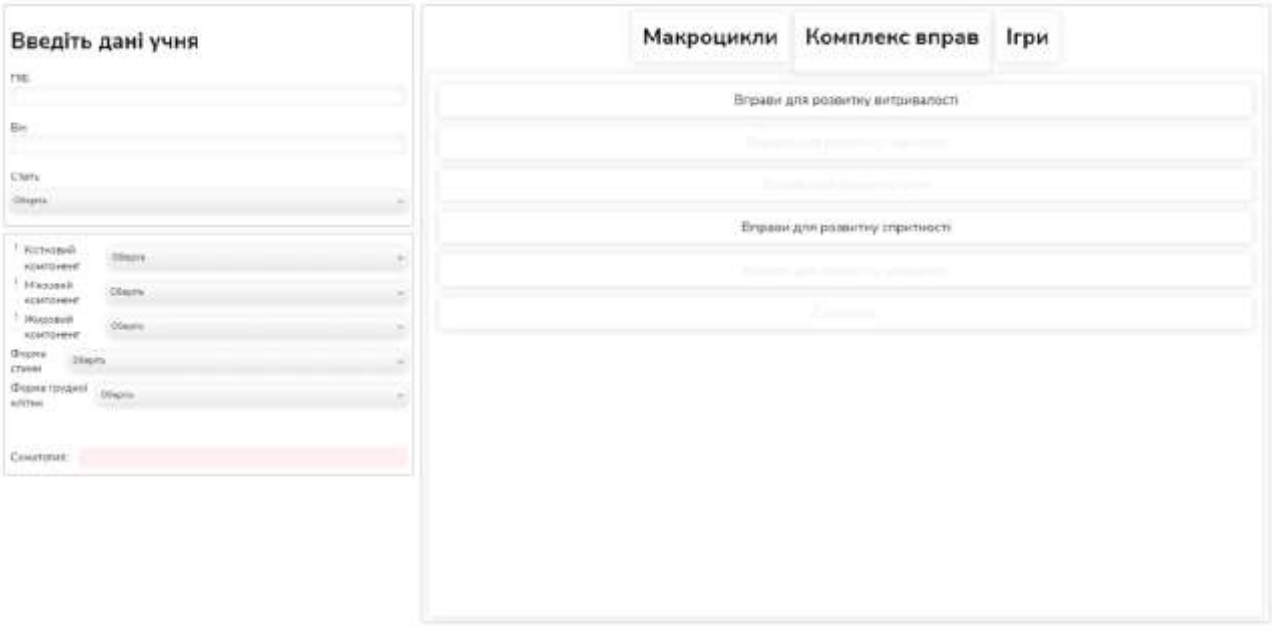


Рис. 3.2 Кінцевий вигляд сайту



Рис. 3.3 Приклад виведення довідки про параметр

3.2 Аспекти розробки

Загалом реалізація проєкту не потребує знань, що виходять за межі базової верстки веб-сторінок та основ веб-розробки. Окрему увагу слід звернути на процес підключення бібліотеки jQuery, оскільки вона не є вбудованим інструментом жодної з використаних мов програмування і потребує додаткового імпорту через зовнішні джерела. Оскільки проєкт не передбачає використання серверної частини (бекенду), увесь функціонал реалізовано виключно за допомогою JavaScript на стороні клієнта.

```
$("#macrocycle").click(
    (e) => {
        $(".title").removeClass("open-title");
        $(".title").addClass("close-title");
        $(e.currentTarget).addClass("open-title");
        $(e.currentTarget).removeClass("close-title");
        $(".menu-block").hide();
        $("#macro-prikol").show();
    }
)
```

Рис. 3.4 Приклад функції приховування і відкриття блоку по нажаттю.

Принцип роботи інтерактивного елементу полягає в тому, що при натисканні на певний блок він активується та частково перекриває сусідній елемент, забезпечуючи ефект перемикання видимості. При повторному натисканні або виборі іншого блоку попередній елемент автоматично приховується, що дозволяє реалізувати зручну навігацію та компактне відображення інформації.

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="save_styles.css">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"
        integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
    <script defer src="script.js"></script>
    <title>Judoist web program</title>
</head>
```

Рис. 3.5 Приклад підключення бібліотеки JQuery.

Водночас використання jQuery має певні недоліки. Для підключення бібліотеки необхідно додавати зовнішні посилання у HTML-документ, що збільшує обсяг коду в секції <head> та залежність від сторонніх ресурсів. Крім того, через постійний розвиток бібліотеки та регулярні оновлення, використовувані посилання можуть ставати застарілими, що вимагає їх періодичної актуалізації. Оновлення версій також можуть призводити до несумісності з уже написаним кодом, оскільки зміни у функціоналі бібліотеки інколи мають фундаментальний характер і можуть порушувати роботу раніше

реалізованих скриптів. Це, у свою чергу, потребує доопрацювання або часткового переписування коду для підтримки стабільної роботи проєкту.

Для реалізації логіки визначення соматотипу у проєкті було використано умовний оператор if у мові JavaScript (Рис. 3.6), що дозволило задати всі необхідні логічні умови для коректного формування результату на основі введених користувачем даних.

```
function somatotype() {
    bones = $('#bones').val();
    muscle = $('#muscle').val();
    fat = $('#fat').val();
    if (bones == 1 && muscle == 2 && fat == 1) {
        $('#somatype').text('Астенічний');
        $('#macro-win').attr('src', 'media/macro1.bmp')
    }
```

Рис. 3.6 Приклад функції if в Javascript.

ВИСНОВОК

У ході виконання даної кваліфікаційної роботи було спроєктовано та розроблено веб-сайт для автоматизованого індивідуального підбору тренувальних комплексів на основі визначення соматотипу користувача. Основною метою проєкту було створення програмного інструменту, який дозволяє спростити процес аналізу фізичних параметрів спортсмена та забезпечити формування персоналізованих рекомендацій для тренувального процесу, зокрема у сфері дитячої спортивної підготовки.

У процесі розробки було проведено аналіз сучасних інформаційних систем і мобільних додатків спортивного спрямування, таких як Heria Pro, Freeletics, Fitbod та інші, що дозволило визначити їхні переваги та недоліки. На основі цього аналізу було сформовано вимоги до власної системи, зокрема акцент на індивідуалізацію, простоту використання та доступність без необхідності складного технічного або професійного супроводу.

Розроблений веб-ресурс реалізовано із використанням базових веб-технологій: HTML, CSS та JavaScript, що забезпечило кросплатформеність і відсутність потреби у серверній частині (бекенді). Для підвищення інтерактивності інтерфейсу було застосовано бібліотеку jQuery, яка спростила реалізацію динамічних елементів взаємодії користувача з системою. Також було реалізовано алгоритм логічної обробки даних із використанням умовних конструкцій JavaScript, що дозволило автоматизувати процес визначення соматотипу.

Особливу увагу приділено структурі користувацького інтерфейсу, який побудований у формі послідовного заповнення даних. Це забезпечує зрозумілу логіку взаємодії користувача із системою та мінімізує ймовірність помилок під час введення інформації. Результатом роботи є система, що після введення антропометричних даних автоматично формує тип тілобудови користувача та пропонує відповідні тренувальні комплекси.

Отриманий програмний продукт може бути використаний як допоміжний інструмент для тренерів, викладачів фізичного виховання, а також батьків, які займаються фізичним розвитком дітей. Запропоноване рішення сприяє підвищенню ефективності тренувального процесу за рахунок індивідуалізації навантажень та зменшення впливу суб’єктивного фактора при підборі вправ.

Таким чином, поставлені у роботі завдання виконано повністю, а мета дослідження досягнута. Подальший розвиток проєкту може бути спрямований на впровадження серверної частини, розширення бази тренувальних програм, а також використання алгоритмів машинного навчання для більш точного та автоматизованого підбору індивідуальних тренувальних режимів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Всесвітнє павутиння URL: https://uk.wikipedia.org/wiki/Всесвітнє_павутиння
2. Дідик Д.М., Ковалюк О.О., Автоматизована система пошуку квитків на транспорт. LI Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації: зб. матеріалів конференції. Вінниця, 2022. URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15888
3. Мобільні додатки URL: https://play.google.com/store/apps?hl=ru&gl=US
4. Сергієнко Л.П. Спортивний відбір: теорія та практика. У 2 кн. Книга 2. Відбір у різні види спорту: Підручник. Тернопіль: Навчальна книга. Богдан, 2010. 784 с. ISBN 978-966-10-0565-7.
5. OMRON BF 511 URL: https://www.manualspdf.ru/omron/bf511/инструкция
6. Вілмор Дж. Фізіологія спорту / Дж. Вілмор, Д.Л. Костілл. Київ.: Олімпійська література, 2003. 655 с. ISBN 966-7133-60-5.__
7. Wilmore J.H. Physiology of sport and exercise/ Wilmore J.H., Costill D.L. Champaign, Illinois: Human Kinetics, 2004. 726 p.
8. HD McCarthy, TJ Cole, T Fry, SA Jebb and AM Prentice. International Journal of Obesity (2006) 30, P. 598–602.
9. Гіподинамія URL: https://uk.wikipedia.org/wiki/Гіподинамія.
10. Гіпокінезія URL: https://esu.com.ua/search_articles.php?id=30196
11. Heria Pro: URL: https://play.google.com/store/apps/details?id=com.iomango.chrisheria&hl=ru&gl=US
12. Fitness Trainer FitProSport FULL: URL: https://play.google.com/store/apps/details?id=fitness.fitprosportfull
13. JQuery URL:https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&u act=8&ved=2ahUKEwiK_4zUk7P4AhViBRAIHWrkCwoQFnoECAYQAQ&url

=htt
ps%3A%2F%2Fjquery.com%2F&usg=AOvVaw1yb1TgbSxtZNKnsTynd_HN
14. Сучасний посібник з Javascript URL: https://uk.javascript.info
15. UML URL: https://evergreens.com.ua/ru/articles/uml-diagrams.html
16. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 4th ed. Hoboken : Pearson, 2021. 1136 p.
17. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 775 p.
18. Bishop C. M. Pattern Recognition and Machine Learning. New York : Springer, 2006. 738 p.
19. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow. 3rd ed. Sebastopol : O'Reilly Media, 2022. 851 p.
20. Aggarwal C. C. Recommender Systems: The Textbook. Cham : Springer, 2016. 518 p.
21. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. Cham : Springer, 2022. 1548 p.
22. Jannach D., Zanker M., Felfernig A., Friedrich G. Recommender Systems: An Introduction. Cambridge : Cambridge University Press, 2017. 344 p.
23. Nielsen J. Usability Engineering. San Francisco : Morgan Kaufmann, 1994. 362 p.
24. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. 3rd ed. Berkeley : New Riders, 2014. 216 p.
25. Garrett J. J. The Elements of User Experience: User-Centered Design for the Web and Beyond. 2nd ed. Berkeley : New Riders, 2011. 192 p.
26. Chaffey D., Ellis-Chadwick F. Digital Marketing. 8th ed. Harlow : Pearson, 2022. 688 p.
27. Kaushik A. Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity. Indianapolis : Wiley Publishing, 2009. 504 p.
28. Sommerville I. Software Engineering. 10th ed. Boston : Pearson Education, 2016. 816 p.

29. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 976 p.
30. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Boston : Addison-Wesley, 2018. 448 p.
31. Mozilla Developer Network Web Docs. Web Performance. URL: https://developer.mozilla.org/en-US/docs/Web/Performance (дата звернення: 10.06.2026).
32. Google Developers. Lighthouse Documentation. URL: https://developer.chrome.com/docs/lighthouse/ (дата звернення: 10.06.2026).
33. Google Search Central. Search Engine Optimization (SEO) Starter Guide. URL: https://developers.google.com/search/docs/fundamentals/seo-starter-guide (дата звернення: 10.06.2026).
34. World Wide Web Consortium (W3C). Web Content Accessibility Guidelines (WCAG). URL: https://www.w3.org/WAI/standards-guidelines/wcag/ (дата звернення: 10.06.2026).
35. Nielsen Norman Group. UX Research and Consulting. URL: https://www.nngroup.com/ (дата звернення: 10.06.2026).
36. Google Analytics Help Center. URL: https://support.google.com/analytics/ (дата звернення: 10.06.2026).
37. OpenAI API Documentation. URL: https://platform.openai.com/docs/ (дата звернення: 10.06.2026).