

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
«\_\_\_\_» \_\_\_\_\_ 2019 р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**  
**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

**Розробка веб-додатку для автоматизації роботи фітнес-клубу**

---

---

---

Освітньо-кваліфікаційний рівень “бакалавр”

Керівник проекту:

\_\_\_\_\_

(підпис)

**М.Є. Щербакова**

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

**Я.О. Критська**

\_\_\_\_\_

(ініціали, прізвище)

Студент:

\_\_\_\_\_

(підпис)

**І.В. Хамула**

\_\_\_\_\_

(ініціали, прізвище)

Група:

\_\_\_\_\_

КН-15бд

Сєвєродоноєцьк 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітньо-кваліфікаційний  
рівень

бакалавр

Напрямок підготовки 6.050101 «Комп'ютерні науки»

(шифр і назва)

Спеціальність \_\_\_\_\_

(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_

І.С. Скарга-Бандурова

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**З А В Д А Н Н Я**  
**НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Хамули Ірини Вікторівни

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-додатку для автоматизації роботи фітнес-клубу

керівник проекту (роботи) Щербакова М.Є., к.т.н., доц

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " \_\_\_\_ " \_\_\_\_\_ 2019 р. № \_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи Мова програмування PHP

Мова структурування та подання вмісту HTML 5

Система управління базами даних MySQL

Віртуальний сервер

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області та постановка задачі;

вибір засобів для розробки; розробка веб-додатку для автоматизації роботи фітнес-клубу; охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Збір та вивчення інформації для написання дипломного проекту(роботи)	До	
2	Написання вступу	До 10.05.2019	
3	Оформлення першого розділу	До 18.05.2019	
4	Вимоги до розроблюваного додатку та написання другого розділу	До 30.05.2019	
5.	Розробка веб-сайту для автоматизації роботи фітнес-клубу та опис його у третьому розділі	До 06.06.2019	
6	Написання висновків	До 10.06.2019	
7	Оформлення розділу з охорони праці	До 15.06.2019	
8	Виправлення зауважень	До 15.06.2019	
9	Захист дипломного проекту(роботи)	20.06.2019	
		(Згідно графіку)	

**Здобувач вищої освіти**

\_\_\_\_\_ ( підпис )

**Науковий керівник**

\_\_\_\_\_ ( підпис )

**Хамула І.В.**

\_\_\_\_\_ (прізвище та ініціали)

**Щербакова М.Є.**

\_\_\_\_\_ (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра.  
70с., 27рис., 7 табл., 15 бібліографічних джерел посилань, 2 додатка.

Об'єкт розробки: веб-додаток для автоматизації роботи фітнес-клубу.

Метою дипломного проекту є розробка додатку, що автоматизує діяльність фітнес-клубу, а саме зберігає і оброблює поточні дані.

Основними модулями розробки є меню адміністратора, тренера та клієнта.

В проекті виконано:

1. Ознайомлення з літературою.
2. Визначені вимоги до розроблюваного веб-додатку.
3. Розроблена база даних веб-додатку.
4. Розроблений веб-додаток для роботи фітнес-клубу.

Дипломний проект може бути використаний фітнес-клубами і тренажерними залами у невеликих містах або для дистанційного ведення клієнтів з різних міст.

**Ключові слова:** ФІТНЕС-КЛУБ, WEB-ДОДАТКИ, СТРУКТУРА САЙТУ, PHP, CSS, HTML, MYSQL, DBFORGE, WEBMATRIX, БАЗА ДАНИХ.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	3
ВСТУП	4
1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Вибір засобів реалізації веб-додатку	6
1.1.1 Мова розмітки гіпертексту	6
1.1.2 Поняття про стилі CSS	7
1.1.3 Мова сценаріїв JavaScript	10
1.1.4 Серверна мова PHP	12
1.1.5 Обґрунтування вибору СКБД	15
1.2 Середовище розробки веб-додатку	17
1.3 Технічне завдання на розробку	18
1.3.1 Визначення завдання на розробку додатку	18
1.3.2 Вимоги до розроблюваного сайту	19
Висновки до першого розділу	20
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	21
2.1 Визначення етапів роботи над проектом	21
2.2 Розробка структури бази даних	22
Висновки до другого розділу	31
3 РОЗРОБКА СТРУКТУРИ ТА ОПИС ДОДАТКУ	30
3.1 Розробка структури додатку	30
3.2 Середовище роботи клієнта	32
3.3 Середовище роботи персоналу	41
3.3.1 Робота тренера	41
3.3.2 Робота адміністратора	49
Висновки до третього розділу	54
4 ОХОРОНА ПРАЦІ	55
4.1 Аналіз стану умов праці	55
4.1.1 Вимоги до приміщення	55

4.1.2	Вимоги до організації місця праці	56
4.2	Виробнича санітарія	57
4.2.1	Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	57
4.2.2	Пожежна безпека	58
4.2.3	Електробезпека	60
4.3	Гігієнічні вимоги до параметрів виробничого середовища	61
4.3.1	Мікроклімат	65
4.3.2	Освітлення	61
4.4	Вентилювання	63
4.5	Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	63
4.5.1	Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).	65
	Висновки до четвертого розділу	67
	<b>ВИСНОВКИ</b>	69
	<b>ПЕРЕЛІК ДЖЕРЕЛ І ПОСИЛАНЬ</b>	70
	ДОДАТОК А. Лістинг програми	72
	ДОДАТОК В. Комп'ютерна презентація	125

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

**HTML** – HyperText Markup Language

**PHP** – Hypertext Preprocessor

**CSS** – Cascading Style Sheets

**СКБД** – Система керування базами даних

**БД**– База даних

**HTTP** – Hyper Text Transfer

## ВСТУП

Для кожної людини поняття «фітнес» сприймається унікально. Для одних — це спорт, за допомогою якого можна створити ідеальну фігуру, для інших — модний тренд або спосіб провести вільний час. Та є така категорія людей для яких «фітнес» — це спосіб життя, без якого вони не можуть уявити себе в сучасному світі.

Збільшення і зміна характеру навантаження на людину у зв'язку з ускладненням життя суспільства призводить до тяжких наслідків для організму. Посилення екологічного, технологічного, психологічного і політичного впливу негативно впливає на здоров'я людини.

Спосіб життя людини безпосередньо впливає на стан і здоров'я її організму. У зв'язку з цим у суспільстві виникає необхідність дотримуватися правильного щоденного режиму, здорового харчування та спорту.

Здоров'я і здоровий спосіб життя зараз актуальні. Власне фітнес — є складовою, що формує таку цінність, як здоров'я людини.

Через велику завантаженість на роботі, людині важко виділити вільний час для особистих справ. Складнішим є завдання пошуку часу для організації тренувань. Проте візит до фітнес-клубу дозволяє ефективно використовувати вільний час.

З початком відвідування фітнес-клубу людина змінює старі та формує нові цілі.

Фітнес-клуб надає можливість отримати професійну допомогу в заняттях фітнесом, оскільки вправи виконуються за програмою складеною кваліфікованим тренером.

Фітнес-тренер обирає клієнту вправи і інтенсивність навантаження залежно від його здоров'я і цілей, які людина ставить перед собою (стати стрункішим, розвинути м'язи тощо). Тренер може працювати з групою або індивідуально. Він може спеціалізуватися на заняттях у тренажерному залі



(інструктор з гімнастики, інструктор з бодібілдингу) або в аеробіці та інших напрямках.

Без сумніву зрозуміло, що фітнес-клуб, котрий хоче ефективно працювати та мати попит на ринку, повинен організувати зручну систему для автоматизації роботи та мати гарний сайт для залучення клієнтів. Проте не кожен фітнес-клуб може це собі дозволити, адже таке задоволення коштує великих грошей.

Сьогодні проблема автоматизації робочого місця адміністратора і тренера особливо актуальна. Наприклад, тренери зазвичай використовують звичайні аркуші паперу в своїй повсякденній роботі з клієнтами, що не дуже зручно, оскільки існує великий ризик втрати інформації. Крім того, зберігання інформації у вигляді комп'ютерних даних, безумовно, більш доцільно, ніж в класичній формі, тобто у вигляді паперів. Таким чином пошук і заміна будь-яких необхідних даних значно спрощується.

Клієнтам в свою чергу важливо підтримувати постійний зв'язок з тренером та знати про останні події.

Мета дипломної роботи – розробка web-додатку, що автоматизує діяльність фітнес клубу, а саме організує збір, обробку та зберігання даних, а також оптимізує роботу з клієнтами.

Для досягнення мети необхідно:

- проаналізувати область дослідження;
- визначити засоби розробки;
- розробити базу даних для збереження поточних даних;
- розробити інтерфейс програмного продукту.

В результаті було створено недорогий web-продукт, що сприяє залученню клієнтів і подальшій роботі з ними.

# 1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Вибір засобів реалізації веб-додатку

### 1.1.1 Мова розмітки гіпертексту

Єдине, що необхідно для того, щоб успішно працювати в Інтернеті – навчитися говорити її специфічною мовою: HyperText Markup Language (мова гіпертекстової розмітки), або скорочено HTML.

HTML – це проста система, яка використовується для структурування і відображення веб-сторінки та її контенту.

HTML не є мовою програмування – це мова розмітки. Отже це означає, що це система ідентифікації та опису різних компонентів документа, наприклад, заголовків, абзаців і списків. Розмітка вказує на базову структуру документа.

У HTML немає головного атрибута, властивого будь-якій мові програмування – команди. На HTML не можливо вказати послідовність дій, а можна описати лише те, як браузер повинен відобразити документ на екрані.

HTML може використовуватися для виконання наступних можливостей:

- представлення інформації у мережі у вигляді електронних документів, з інформаційним наповненням у вигляді відформатованого тексту, списків, фотографій;
- розробки форми для взаємодії с віддаленими службами (пошуковими роботами або online-магазинами);
- меню опцій;
- виконання завантаження результатів запитів до баз даних;
- відображення простих документів з вбудованими графічними образами;

- гіпертекстовий перегляд існуючих масивів інформації.

HTML зберігається у текстових файлах із суфіксом html.

### 1.1.2 Поняття про стилі CSS

Каскадні таблиці стилів - скорочено CSS - надають творчий контроль над макетом і дизайном веб-сторінок. [1].

Таблиця стилів містить набір правил (стилів), які описують дизайн самої веб-сторінки та її окремих фрагментів. Ці правила визначають колір тексту і вирівнювання абзацу, інтервал між графічним зображенням і навколишнім текстом, наявність і параметри кадру в таблиці, колір фону веб-сторінки і багато іншого.

Кожен стиль повинен бути прив'язаний до відповідного елемента веб-сторінки (або самої веб-сторінки). Після прив'язки, описаного вибраними параметрами стилю, починають застосовуватися до цього елемента. Прив'язка може бути очевидною, у тому випадку, коли ми самостійно вказуємо, який стиль елемента веб-сторінки прив'язаний, або неявний, коли стиль автоматично пов'язаний з усіма елементами веб-сторінки, створеної за допомогою певного тегу.

Таблицю стилів можна зберігати безпосередньо в HTML-коді веб-сторінки або в окремому файлі. Крім того, окремі стилі можуть бути розміщені безпосередньо в тезі HTML, який створює елемент веб-сторінки. Цей підхід в даний час використовується досить рідко, і в основному в експериментах зі стилями.

Таблиці стилів записуються на спеціальній мові, яка називається - CSS. Специфікація для визначення першої версії цієї мови (CSS 1) з'явилась в 1996 році. Стандарт CSS2 широко підтримується і застосовується на практиці, а також розробляється стандарт CSS3, обмежена підмножина якого вже підтримується багатьма веб-браузерами.

Таблиці стилів, залежно від того, де вони зберігаються, поділяються на два типи. Зовнішні таблиці стилів зберігаються окремо від веб-сторінок, у файлах з розширенням CSS. Вони містять код CSS для визначення стилів.

Перевага зовнішніх таблиць стилів полягає в тому, що вони можуть бути прив'язані до декількох веб-сторінок одночасно. Існує тільки один недолік, і що один не є істотним - зовнішня таблиця стилів зберігається в окремому файлі, тому є шанс його втратити.

Внутрішня таблиця стилів записується безпосередньо до HTML-коду веб-сторінки. Вона укладається в парний тег `<style>` і розміщується в розділі заголовка. В іншому випадку вона не відрізняється від її зовнішньої. Перевага внутрішньої таблиці стилів полягає в тому, що вона є невід'ємною частиною веб-сторінки і, отже, ніколи не зникне. Недоліки лише два. По-перше, стилі, визначені у внутрішній таблиці стилів, застосовуються тільки до веб-сторінки, на якій знаходиться таблиця стилів. По-друге, внутрішня таблиця стилів не відповідає концепції Web 2.0, що вимагає відокремлення змісту веб-сторінки від її представлення. На тій же веб-сторінці може бути кілька таблиць стилів одночасно: кілька зовнішніх і внутрішніх.

Однією з основних особливостей властивостей CSS є їх каскадний характер, завдяки якому їх називають каскадними таблицями стилів. Каскадний метод - це метод, який використовується для вирішення потенційних конфліктів між різними типами стилів, що підтримуються браузером, і застосовується в порядку пріоритету, залежно від виробника стилів, методу, який використовується для створення стилю, і типів вибраних властивостей.

Всі сучасні браузери підтримують три основні типи таблиць стилів. У порядку пріоритету зверху вниз вони розташовані наступним чином.

1. Створені автором документа.
2. Створені користувачем.
3. Створені браузером.

Ці три набору таблиць стилів обробляються в зворотному порядку. Спочатку до документа застосовуються початкові налаштування браузера. Без них веб-сторінки, які не використовують таблиці стилів, виглядають жахливо. Вони включають в себе зовнішній вигляд, розмір і колір шрифту, відстань між елементами, обрамлення і відступи в таблицях, а також всі інші прийнятні стандарти, очікувані користувачем. Потім, якщо користувач створив кілька стилів, які переважно використовують як стандартні, вони заміщуються вихідними стилями браузера, які використовують суперечливі дані. В результаті, застосовуються будь-які стилі, створені автором поточного документа, замінюючи будь-які стилі, створені як оригінальні стилі браузера, або створені користувачем.

Існує три різні способи вибору стильових елементів. У порядку зменшення пріоритету їхній список виглядає наступним чином:

1. Звернення за індивідуальним ідентифікатором або селектором атрибутів.
2. Звернення в групах за класом.
3. Посилання на теги елементів.

Селектори обробляються відповідно до кількості та типів елементів, які підходять під правило, що дещо відрізняється від попередніх двох правил вирішення конфліктів. Причина полягає в тому, що правила не повинні негайно застосовуватися до одного типу селектора і можуть бути пов'язані з різними селекторами. Таким чином, спосіб, необхідний для визначення рівня пріоритету правил, може містити будь-яку комбінацію селекторів. Це робиться шляхом обчислення специфіки кожного правила шляхом побудови їх у порядку зменшення обсягу.

За допомогою CSS можна налаштувати чотири основних властивості шрифту: сімейство - `family`, стиль - `style`, розмір - `size` і насиченість - `weight`. Використовуючи ці властивості, можна правильно налаштувати спосіб відображення тексту на веб-сторінках та направляти його на друк.

Каскадні таблиці стилів пропонують велику кількість потужних команд форматування, які дозволяють задавати шрифти, колір, розмір, міжрядковий інтервал та інші властивості та атрибути, які впливають на візуальне сприйняття як окремих елементів веб-сторінки так і всієї веб-сторінку та сайту в цілому.

### 1.1.3 Мова сценаріїв JavaScript

JavaScript - це мова сценаріїв, яка запускається виключно на стороні клієнта всередині браузера.

JavaScript - це мова програмування, що дозволяє досягти найвищої продуктивності HTML-сторінок, заповнивши їх анімацією, інтерактивними елементами та динамічними візуальними ефектами.

Мова JavaScript здатна зробити веб-сторінки більш корисними, надаючи миттєвий відгук.

JavaScript генерує повідомлення про помилку відразу після того, як хтось намагається надіслати форму, яка не містить необхідної інформації.

Мова JavaScript також дозволяє створювати захоплюючі динамічні та інтерактивні інтерфейси. За допомогою JavaScript стає можливим перетворення статичної сторінки мініатюр у анімоване слайд-шоу. Або зробити щось ненав'язливе - додати додаткову інформацію на сторінку, не перевантажуючи її, розміщуючи вміст невеликими панелями, до яких відвідувач може отримати доступ за допомогою клацання миші.

Головною перевагою JavaScript є негайна відповідь. Мова JavaScript дозволяє веб-сторінкам миттєво реагувати на дії, коли відвідувач переходить через посилання, заповнює форму або просто керує вказівником миші по екрану.

У JavaScript відсутня повільність, яка пов'язана з скриптовими мовами сервера, такими як PHP, котрі взаємодіють на основі зв'язку між браузером і

веб-сервером. Оскільки ця мова не залежить від постійного завантаження та перезавантаження веб-сторінок, вона дозволяє створювати веб-сайти, які сприймаються та працюють швидше.

За допомогою JavaScript можна створювати як прості програми так і повноцінні веб-додатки.

Так як JavaScript є мовою програмування, для багатьох людей, включаючи веб-дизайнерів, процес роботи з нею здається досить важким. Ще більше ускладнює роботу те, що різні веб-браузери по-різному розуміють JavaScript. Тому для перевірки правильності роботи програми для всієї аудиторії може знадобитись досить велика кількість часу. У цьому випадку досить зручно використовувати jQuery.

jQuery - це бібліотека JavaScript, яка значно спрощує програмування на цій мові. Бібліотека JavaScript - це всеосяжний JavaScript-додаток, що одночасно спрощує складні завдання і вирішує проблеми сумісності з браузером. jQuery дозволяє подолати дві найбільші перешкоди в роботі з JavaScript: складність програмування і простота використання різних веб-браузерів.

JavaScript надає веб-сторінкам можливість свідомо реагувати на дії користувачів. За її допомогою можна:

- створювати розумні веб-форми, які будуть інформувати відвідувачів, якщо вони забудуть ввести важливу інформацію або допустять помилку;
- змушувати елементи з'являтися, зникати або переміщатися по екрану, і оновлювати вміст сторінки на основі інформації, отриманої від сервера, без необхідності перезавантаження.

Отже, мова JavaScript робить веб-сайти більш привабливими, корисними та ефективними, тому вона була використана мною в дипломній роботі.

### 1.1.4 Серверна мова PHP

Для реалізації WEB-додатку для автоматизації роботи фітнес-клубу було обрано мову програмування PHP.

PHP є двигуном мільйонів динамічних веб-додатків. Його широкий набір функцій, доступний синтаксис і підтримка різних операційних систем і веб-серверів зробили його ідеальною мовою як для швидкої веб-розробки, так і для методичної побудови складних систем. Однією з основних причин успіху PHP як мови веб-скриптів є її походження як інструменту для обробки HTML-форм і створення веб-сторінок.

За замовчуванням в кінці імен PHP-документів знаходиться розширення `.php`. Коли веб-сервер зустріне файл з цим розширенням у запитуваному файлі, він автоматично передає файл до PHP-процесора.

Веб-сервери мають досить широкий діапазон налаштувань, і деякі веб-розробники вибирають такий режим роботи, в якому файли з розширенням HTML або HTML примусово передаються для обробки PHP-процесора. Це зазвичай пояснюється тим, що розробники хочуть приховати факт використання PHP.

Синтаксис PHP легко вивчити, і його корені лежать в Perl, Java і C. Однак, на відміну від цих мов, PHP спочатку був розроблений як мова програмування, спеціально призначена для написання веб-додатків (сценаріїв), виконаних на веб-сервері. Це його перевага, оскільки його інструменти дозволяють швидко і ефективно створювати повнофункціональні веб-додатки.

Ще одна перевага PHP полягає в тому, що їм надається можливість реалізувати свої скрипти в HTML веб-сторінок, що значно спрощує завдання створення динамічних сайтів. Завдяки PHP, розробники можуть динамічно змінювати HTML сторінки в залежності від дій користувача сайту.

Коли справа доходить до розробки динамічних веб-сайтів, PHP краще,



швидше і простіше вивчати, ніж альтернативи. Завдяки PHP, ми отримуємо відмінну продуктивність, щільну інтеграцію з практично всіма доступними базами даних, стабільність, мобільність і практично необмежений набір функцій завдяки його розширюваності. Все це безкоштовно (з відкритим кодом PHP) і дуже керованою кривою навчання.

Для реалізації WEB-додатку необхідно вибрати зручну базу даних. Щоб виконати запити до БД на веб-сервері на основі PHP, повинен бути встановлений сервер, який буде приймати і виконувати запити з бази даних сценарію. Мова PHP, в цьому випадку, є найбільш сприятливою, адже має підтримку широкого кола баз даних. Мова програмування PHP містить багато вбудованих функцій, таких як: функція роботи з файловою системою, функція роботи з протоколом HTTP, функція роботи з датою і часом, функція обробки рядків і масивів і т. д.

Кожен програмний продукт, написаний на мові PHP, складається з набору спеціальних конструкцій. Такі конструкції можуть обслуговувати будь-які елементи, які використовуються в коді PHP, такі як цикли, оператори, функції тощо.

Головною особливістю цієї мови є те, що вона добре працює з усіма сучасними веб-технологіями і підтримує велику кількість сучасних веб-протоколів.

Мова PHP була створена для протоколу HTTP, тобто її основним завданням є реагування на запит HTTP і звільнення пам'яті та ресурсів.

Протокол HTTP, який лежить в основі Інтернету, не зберігає інформацію про стан сеансу. Це означає, що будь-який клієнтський запит трактується сервером як звернення до нового клієнта, і навіть якщо клієнт генерує запит на завантаження зображень з поточної сторінки, сервер сприймає його як запит від нового клієнта, не пов'язаний з тим, хто тільки що завантажив сторінку. Нажаль, для динамічних сторінок цей спосіб є абсолютно неприпустимим. У зв'язку з цим у протоколи HTTP запровадили

механізми сеансу та cookie, які тепер підтримуються всіма користувачами Інтернету: клієнтами, проксі-серверами та кінцевими користувачами.

Обробка сесії PHP є ключовою концепцією, яка дозволяє зберігати інформацію про користувача на всіх сторінках веб-сайту або програми.

Сесія представляє собою текстовий файл, який зберігає пари ключ / значення, не на машині клієнта, а на сервері.

Оскільки сервер збирає велику кількість файлів, що належать до різних сеансів клієнтів, то для їх ідентифікації кожному новому клієнтові призначається унікальний номер - ідентифікатор сеансу (SID), який передається через рядок запиту або ж cookies, якщо вони є.

До недоліків сесій відноситься неможливість контролю часу життя PHP-скриптів, так як цей параметр задається в конфігураційному файлі `php.ini`. Сесія ініціюється за допомогою функції `session_start()`. Функція повертає `true` у разі успішної ініціалізації сесії і `false` - в протилежному випадку. Для роботи з сесією функція `session_start()` повинна викликатися на кожній сторінці, де відбувається звернення до змінних сесії.

Підтримка cookie HTTP: PHP підтримує файли cookie HTTP. Cookie - це механізм для зберігання даних (малих файлів до 4 кілобайт інформації) у віддаленому браузері і, таким чином, відстеження або ідентифікація користувача. Встановити файли cookie можливо використовуючи функцію `setcookie()`. Файли cookie є частиною заголовка HTTP, тому функція `SetCookie()` повинна бути викликана до того, як будь-яка інформація буде надіслана в браузер. Будь-який файл cookie, надісланий від клієнта, автоматично перетворюється у змінну `PHP[5]`.

Іншою перевагою цієї мови є те, що вихідний текст скрипта PHP не можна переглядати в браузері, оскільки він компілюється безпосередньо перед відправкою за запитом користувача. Як результат, таке виконання скрипту не дозволяє крадіжки тексту кодів оригінальних команд PHP користувачами.

Оскільки PHP не містить коду, який буде орієнтований на певний веб-сервер, він добре працює з сучасними серверами, а саме : сервер Microsoft IIS, сервер Netscape Enterprise Server, сервер Apache, сервер Stronghold та інші.

Для запуску веб-додатку, побудованого мною у Microsoft Webmatrix, використано веб-сервер IIS 8 із підтримкою PHP версії 5.5.38.

### **1.1.5 Обґрунтування вибору СКБД**

Система керування базами даних (СКБД) MySQL популярна з багатьох причин. Вона швидка, і проста в налаштуванні, використанні та адмініструванні. Вона працює під багатьма різновидами Unix і Windows. Програми на основі MySQL можуть бути написані на багатьох мовах.

Система баз даних MySQL використовує архітектуру клієнт-сервер. Сервер mysql - це програма, яка фактично маніпулює базами даних. Щоб повідомити сервер, що робити, використовується клієнтська програма, яка повідомляє намір за допомогою операторів, написаних на мові структурованих запитів (SQL). Клієнтські програми написані для різних цілей, але кожна з них взаємодіє з сервером, підключаючись до нього, відправляючи оператори SQL для виконання операцій з базою даних і отримуючи результати.

Клієнти встановлюються локально на комп'ютері, з якого користувач хоче отримати доступ до MySQL, але сервер може бути встановлений де завгодно, якщо до нього можуть підключатися клієнти. Оскільки MySQL є мережевою системою баз даних, клієнти можуть обмінюватися даними з сервером, що працює локально на комп'ютері або в іншому місці на планеті.

Програма mysql є одним з клієнтів, включених в дистрибутиви MySQL. При інтерактивному використанні mysql запитує оператор, відправляє його на сервер MySQL для виконання і відображає результати. MySQL також

може бути використаний в пакетному режимі неінтерактивному читати заяви, що зберігаються в файлах або створені програмами. Це дозволяє використовувати MySQL з скриптів або завдань, або в поєднанні з іншими додатками.

Системи керування базами даних – це програмні інструменти, які дозволяють створювати бази даних і опрацьовувати дані.

В якості СКБД для реалізації WEB-додатку автоматизації роботи фітнес клубу було обрано MySQL.

Ця СКБД була обрана тому, що вона є безкоштовною і широко поширена, проста в установці та використанні.

MySQL – це реляційна система керування базами даних з вільним доступом. MySQL є зручною для використання в середовищі веб СУБД. Реляційна база даних зберігає інформацію в окремих таблицях, що призводить до збільшення швидкості і гнучкості.

Таблиці взаємодіють одна з одною за допомогою відношень, що дозволяє об'єднати дані з декількох таблиць при виконанні запиту.

MySQL - це клієнтська серверна система, що включає в себе багатопотоковий SQL-сервер, який підтримує різні комп'ютерні бази даних, а також декілька різних клієнтських програм, бібліотек, інструментів управління та широкого діапазону програмних інтерфейсів (API). Існує також велика кількість програмного забезпечення для MySQL, яке здебільшого є безкоштовне.

Для створення бази даних, таблиць та їх первинного заповнення використовувалася програма DBForge Studio для MySQL. Вона є дуже зручною, адже за допомогою графічного інтерфейсу дозволяє виконувати всі необхідні операції з базою даних MySQL.

## 1.2 Середовище розробки web-додатку

Майкрософт WebMatrix – це веб-конструктор сайтів і редактор HTML для Windows, орієнтований на веб-розробку.

WebMatrix дозволяє розробникам створювати веб-сайти за допомогою вбудованих шаблонів або популярних відкритих програм, з повною підтримкою ASP.NET, PHP, і HTML5. Microsoft розробила WebMatrix з метою надання веб-розробникам можливості кодування, налаштування та публікації в одному місці.

WebMatrix є наступником ASP.NET Web Matrix який був випущений в 2003 році і згодом припинений.

У 2011 році платформа WebMatrix була випущена для підтримки великої кількості систем керування контентом з відкритим кодом і для забезпечення легкого веб-середовища розробки для PHP і нових, спрощених веб-сторінок ASP.NET. Вона зосереджена на чистому, простому користувальницькому інтерфейсі, що дозволяє веб-розробникам створювати веб-сайти з нуля або шляхом налаштування систем управління відкритим кодом, таких як Orchard, DotNetNuke, Umbraco , Joomla! , Drupal і WordPress.

З 2011 по 2012 рік WebMatrix 2 Beta і RC випускають додаткову підтримку для Node.js, мобільних тренажерів, додаткових шаблонів веб-сайтів і підтримки публікацій на веб-сайтах Microsoft Azure. 6 вересня 2012 року офіційний випуск WebMatrix 2 став публічним. Випуск WebMatrix 3 був доступний 1 травня 2013 року. На відміну від WebMatrix 2, WebMatrix 3 вимагає Windows 7 або пізнішої версії.

WebMatrix – це набір програм для швидкого розгортання веб-додатків, що включає полегшену версію веб-сервера IIS Express і оновлену версію SQL Server Compact Edition. Він також включає новий аналізатор Razor для ASP.NET, який дозволяє розробникам використовувати код на Visual Basic або C # при створенні сторінок HTML.

Особливостями платформи Microsoft Webmatrix є:

- спрощене створення, публікація та синхронізація веб-сайтів-супутників;
- інтеграція з системами керування джерелами, включаючи Git і Team Foundation Server;
- заповнення коду та підсвічування синтаксису для HTML5 , CSS3 , JavaScript і TypeScript;
- редагування для серверних мов ASP.NET , PHP і Node.js;
- підтримка jQuery , jQuery Mobile , Less і Sass;
- мобільні тренажери;
- менеджер баз даних для MySQL , Microsoft SQL Server і SQL CE;
- інструменти розгортання файлів і баз даних;
- розгортання на хостингу, виділених серверах або Microsoft Azure;
- публікація веб-сайтів за допомогою FTP , HTTPS і Web Deploy ;
- вбудовані звіти про пошукову оптимізацію та ефективність;
- віддалене та автономне редагування ;
- завантаження віддалених сайтів для місцевого редагування;
- розширюваність за допомогою плагінів.

### **1.3 Технічне завдання на розробку**

#### **1.3.1 Визначення завдання на розробку додатку**

Мета дипломної роботи – розробка веб-додатку для автоматизації роботи фітнес-клубу, а саме для взаємодії з клієнтами та зберігання і аналізу інформації про них.

Основними користувачами веб-додатку будуть – адміністратори, тренери та клієнти фітнес клубу.

Адміністратор – це персонал фітнес-клубу, котрий може створювати, оновлювати, редагувати і видаляти будь-які адміністративні деталі,

включаючи тренерів та адміністратора.

Тренер – це персонал фітнес-клубу, котрий надає ряд послуг:

- слідкує за станом здоров'я клієнта, а саме веде облік медичних показників клієнта;
- веде облік програм тренувань;
- інформує клієнтів стосовно питань тренувального процесу, які їм не зрозумілі.

Клієнту надається можливість перегляду ознайомчих сторінок сайту фітнес клубу, власного профілю, визначення мети тренувань, та формування звітності власних тренувань для тренера.

Web-додаток надає можливість:

- зберігати та переглядати особисті дані по клієнтам, інформацію про тренування, стан здоров'я клієнта;
- формувати розклад роботи тренера фітнес-клубу.

Впровадження розробленого web-додатку значно підвищить ефективність роботи тренерів, оскільки призведе до систематизації, збільшення продуктивності і точності обробки даних, а також дозволить стандартизувати і полегшити обмін даними між тренерами та клієнтами.

### **1.3.2 Вимоги до розроблюваного сайту**

Загальні вимоги до сайту:

- розміщення контактної інформації;
- розміщення загальних даних про тренерів;
- наявність форми реєстрації нового клієнта;
- наявність розмежування прав доступу (адміністратор, тренер, клієнт);
- наявність бази даних про клієнтів і тренерів;

- надання можливості додавання, пошуку, редагування та видалення даних.

Вимоги до інтерфейсу:

- графічні елементи сайту повинні бути розроблені з урахуванням особливостей представлення інформації на веб-сайтах;
- сторінки сайту не повинні містити зайвої інформації;
- оформлення сайту має бути сучасним, привабливим та лаконічним;
- увагу користувача слід зосереджувати на вибраних елементах;
- на сайті має бути відсутня складна структурована інформація що відвертає увагу користувачів від перегляду необхідних даних;
- інтерфейс мусить надавати комфорт виконуваних дій та прискорювати адаптацію користувача до сайту.

Вимоги до навігації

Графічний інтерфейс має забезпечувати візуальну, інтуїтивно-зрозумілу структуру розміщеної інформації, щоб забезпечити швидкий перехід до вибраного розділу сайту.

Навігація сайту повинна бути зрозумілою користувачеві, посилання на сторінки повинні мати відповідні назви, щоб користувач міг легко перейти на потрібну сторінку.

### **Висновки до першого розділу**

У першому розділі було проведено аналіз предметної області та визначені завдання в рамках досягнення мети.

Проведено аналіз необхідного програмного забезпечення та визначено засоби розробки сайту.

Інформація розглянута при аналізі буде врахована з розробкою програмного продукту.



## 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

### 2.1 Визначення етапів роботи над проектом

Найбільша проблема в розробці додатків - це відсутність продуманості. Оскільки це може бути застосовано до додатків, керованими базою даних, процес проектування повинен включати ретельну оцінку бази даних. Вона повинна показувати, як дані пов'язані один з одним, і, що найбільш важливо, чи є вона гнучкою і масштабованою.

Масштабування і гнучкість дуже важливі, тому що незалежно від того, наскільки ретельно база продумана, можуть виникнути нові ідеї пізніше і потрібно буде скористатися ними.

Основні етапи процесу проектування наступні:

- визначення мети;
- проектування структури даних (таблиць і полів);
- визначення відношень;
- визначення і упровадження правил нормалізації;
- створення додатку.

Гарна розробка бази даних має вирішальне значення для високопродуктивного застосування. Без оптимізованих відносин база даних не буде працювати настільки ефективно, наскільки це можливо.

Роздуми про відношення і ефективність бази даних, яка включає в себе простоту обслуговування, мінімізацію дублювання і уникнення невідповідностей, є частиною нормалізації бази даних.

Крім проблеми продуктивності стоїть питання обслуговування. База даних повинна легко обслуговуватися. Це включає в себе зберігання тільки обмеженої кількості повторюваних даних. Якщо є багато повторюваних

даних і один екземпляр цих даних зазнає змін, ця зміна має бути виконана для всіх даних, що повторюються.

Усунути дублювання і розширити можливості по введенню даних, можна створивши таблицю можливих значень і використовувати ключі, щоб звернутися до значення. Таким чином, якщо значення змінює імена, зміна відбувається тільки один раз - в головній таблиці. Посилання залишається незмінним і в інших таблицях.

Переваги добре спланованої і розробленої бази даних численні, і чим більше роботи виконується заздалегідь, тим менше доведеться займатися виправленням пізніше.

Тому, перш ніж кодувати додаток, необхідно якісно розробити базу даних.

## 2.2 Розробка структури бази даних

Розроблена база даних використовує реляційну модель і представляє собою набір взаємопов'язаних таблиць. База даних містить 7 таблиць:

- «admin» – містить інформацію про адміністраторів та тренерів;
- «user» – містить інформацію про клієнтів;
- «health» – містить інформацію про показники здоров'я клієнта;
- «a\_presence» – містить інформацію про вільні години роботи тренера;
- «s\_workout» – містить інформацію про виконане тренування клієнтом та коментарі тренера;
- «p\_workout» – містить інформацію з програмою тренування для клієнтів;
- «ask» – містить інформацію з питаннями від клієнтів та відповідями тренера.

Таблиця “admin” зберігає дані про адміністраторів та тренерів, що застосовуються для авторизації користувача та визначення прав доступу (таблиця 2.1).

Таблиця 2.1 – Опис стовпців таблиці «admin»

Назва стовпця	Тип даних	Визначення типу	Опис
id	int	Ціле число	Унікальний ідентифікатор користувача, первинний ключ
username	varchar	символьний рядок змінної довжини	Унікальний ідентифікатор. Логін та ім'я тренера або адміністратора. Використовується для авторизації на сайті.
password	varchar	символьний рядок змінної довжини	Хеш-пароль тренера або адміністратора. Використовується для авторизації на сайті.
type	varchar	символьний рядок змінної довжини	Тип користувача. Визначає права доступу у системі.

Таблиця “user” зберігає анкетні дані про клієнтів, а також дані, що застосовуються для авторизації користувача (таблиця 2.2).

Таблиця 2.2 – Опис стовпців таблиці про клієнтів «user»

Назва стовпця	Тип даних	Визначення типу	Опис
id	int	Ціле число	Унікальний ідентифікатор користувача, первинний ключ
username	varchar	символьний рядок змінної довжини	Унікальний ідентифікатор. Логін та ім'я клієнта. Вказується щоб увійти до особистого кабінету.
password	varchar	символьний рядок змінної довжини	Хеш-пароль для авторизації у системі.

## Продовження таблиці 2.2 «user»

Назва стовпця	Тип даних	Визначення типу	Опис
f_name	varchar	символьний рядок змінної довжини	Ім'я клієнта
l_name	varchar	символьний рядок змінної довжини	Прізвище клієнта
gender	varchar	символьний рядок змінної довжини	Стать клієнта
birthday	varchar	символьний рядок змінної довжини	Дата народження клієнта.
address	varchar	символьний рядок змінної довжини	Адреса проживання клієнта
city	varchar	символьний рядок змінної довжини	Місто проживання клієнта
email	varchar	символьний рядок змінної довжини	Електронна адреса клієнта
phone	varchar	символьний рядок змінної довжини	Контактний номер
goal	text	текст	Містить інформацію про мету відвідування клубу клієнтом
image	longblob	Великий двійковий об'єкт. Може зберігати до 4 гігабайт інформації.	Зберігає фотокартку клієнта
admin	varchar	символьний рядок змінної довжини	Унікальний ідентифікатор, username тренера, зовнішній ключ від таблиці admin.
freeze	tinyint	Дуже мале ціле число.	Містить інформацію про активність акаунту клієнта.

Таблиця “ a\_presence ” зберігає дані тренера про вільні години тренувань (таблиця 2.3).

Таблиця 2.3 – Опис стовпців таблиці «a\_presence»

Назва стовпця	Тип даних	Визначення типу	Опис
id	int	Ціле число	Унікальний ідентифікатор наявності тренера, первинний ключ.
admin_id	varchar	символьний рядок змінної довжини	Унікальний ідентифікатор тренера, зовнішній ключ від таблиці admin.
monday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у понеділок.
tuesday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у вівторок.
wednesday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у середу.
thursday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у четвер.
friday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у п'ятницю.
saturday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у суботу.
sunday	varchar	символьний рядок змінної довжини	Містить дані про години роботи тренера у неділю.

Таблиця “health” зберігає медичні показники про стан здоров'я клієнта (таблиця 2.4).

Таблиця 2.4 – Опис стовпців таблиці «health»

Назва стовпця	Тип даних	Визначення типу	Опис
id	int	Ціле число	Унікальний ідентифікатор стану здоров'я, первинний ключ.

## Продовження таблиці 2.4 «health»

user_id	int	Ціле число	Унікальний ідентифікатор клієнта, зовнішній ключ від таблиці user.
height	int	Ціле число	Містить інформацію про зріст клієнта.
weight	int	Ціле число	Містить інформацію про вагу клієнта.
bmi	double	Число з плаваючою комою.	Містить показники про індекс маси тіла клієнта
ch	double	Число з плаваючою комою.	Містить показники холестерину клієнта
heart	double	Число з плаваючою комою.	Містить показники про серцевий ритм клієнта
blood	double	Число з плаваючою комою.	Містить показники про кров'яний тиск.
date	datetime	Зберігання дати та часу.	Містить інформацію про дату внесення показників

Таблиця “ s\_workout ” зберігає дані про тренування (таблиця 2.5).

Таблиця 2.5 – Опис стовпців таблиці «s\_workout»

Назва стовпця	Тип даних	Визначення типу	Опис
id	int	Ціле число	Унікальний ідентифікатор тренування, первинний ключ.
user_id	int	Ціле число	Унікальний ідентифікатор клієнта, зовнішній ключ від таблиці user.
w_session	text	текст (до 64 Кбайт)	Містить інформацію про виконане тренування клієнта
comment	text	текст	Містить коментар тренера
w_date	datetime	Зберігання дати та часу.	Дата тренування клієнта

Таблиця “ p\_workout ” зберігає інформацію про тренувальну програму клієнта (таблиця 2.6).

Таблиця 2.6 – Опис стовпців таблиці «p\_workout»

Назва стовпця	Тип даних	Визначення типу	Опис
id	int	Ціле число	Унікальний ідентифікатор програми тренування, первинний ключ.
user_id	int	Ціле число	Унікальний ідентифікатор клієнта, зовнішній ключ від таблиці user.
w_program	text	текст (до 64 Кбайт)	Зберігає інформацію про план тренування клієнта.
w_date	datetime	Зберігання дати та часу.	Дата написання програми тренування

Таблиця “ask” призначена для збереження питань від клієнтів та відповідей тренера на поставлені питання (таблиця 2.7).

Таблиця 2.7 – Опис стовпців таблиці «ask»

Назва стовпця	Тип даних	Визначення типу	Опис
id_ask	int	Ціле число	Унікальний ідентифікатор, первинний ключ.
question	text	текст (до 64 Кбайт)	Містить питання клієнтів до тренера.
answer	text	текст (до 64 Кбайт)	Містить відповіді тренерів на питання клієнтів.
date	datetime	Зберігання дати та часу.	Містить дату заданого питання
user_id	int	Ціле число	Унікальний ідентифікатор клієнта, зовнішній ключ від таблиці user.

На рис. 2.1 представлена схема зв'язку таблиць бази даних.

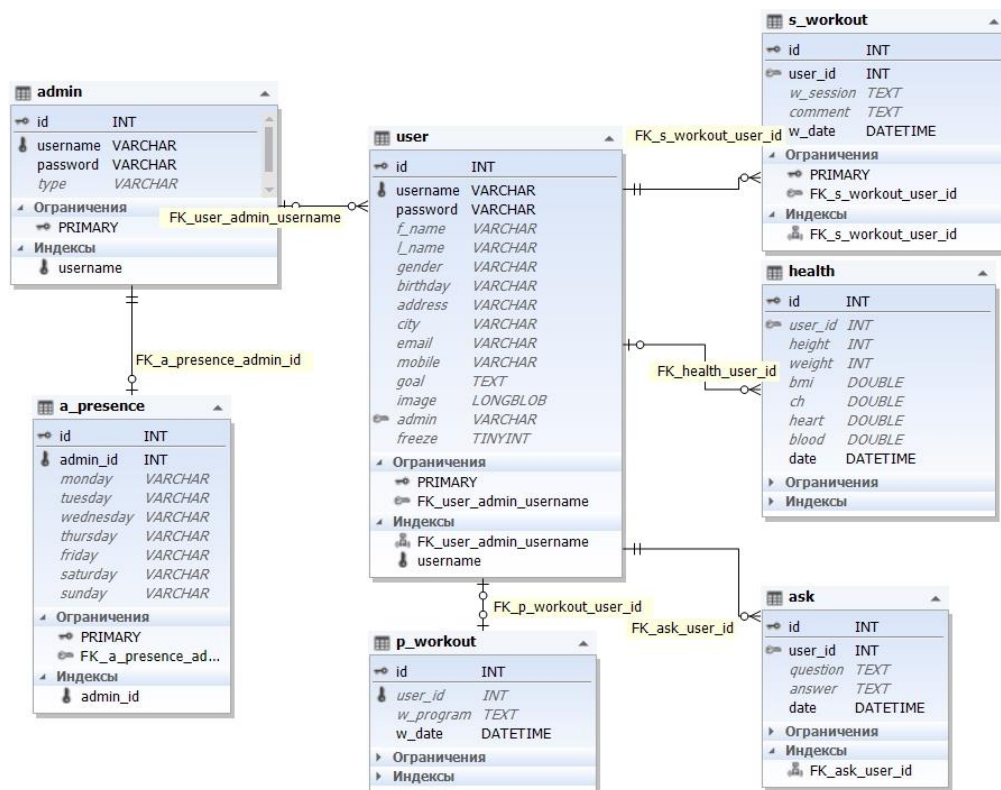


Рисунок 2.1- Схема даних

Між таблицями «admin», «a\_presence», таблицею програми тренувань «p\_workout» та «user» встановлений зв'язок «один до одного».

Між таблицями «user», «s\_workout», «health», «ask» встановлений зв'язок «один до багатьох».

Створені зв'язки виконують дії каскадного оновлення і видалення даних, що виключає появу у базі даних зайвої інформації та логічних помилок.



## Висновки до другого розділу

У другому розділі було визначено основні етапи роботи над проектом, а також розроблено базу даних сайту.

База даних містить у собі 7 таблиць, а саме:

- «admin» – містить інформацію про адміністраторів та тренерів;
- «user» – містить інформацію про клієнтів;
- «health» – містить інформацію про показники здоров'я клієнта;
- «a\_presence» – містить інформацію про вільні години роботи тренера;
- «s\_workout» – містить інформацію про виконане тренування клієнтом та коментарі тренера;
- «p\_workout» – містить інформацію з програмою тренування для клієнтів;
- «ask» – містить інформацію з питаннями від клієнтів та відповідями тренера.

Також у розділі детально описані поля таблиць та вказано зв'язки між ними.

## 3 РОЗРОБКА СТРУКТУРИ ТА ОПИС ДОДАТКУ

### 3.1 Розробка структури додатку

Структура сайту фітнес клубу була розроблена відповідно до необхідного функціоналу.

Сайт фітнес-клубу передбачає можливість надання прав доступу для користувачів. Основними користувачами сайту є: адміністратор, тренер та клієнт.

Головна сторінка надає можливість ознайомитись з персоналом клубу та виконати авторизацію як клієнт та персонал.

Авторизація клієнта має доступ до наступних розділів сайту:

— меню клієнта:

- 1) профіль, де користувач може переглянути власні дані;
- 2) редагування особистих даних;
- 3) графік роботи тренера;
- 4) програма тренувань;
- 5) питання для тренера, де клієнт може отримати відповіді на питання стосовно тренувального процесу ;
- 6) показники стану здоров'я;
- 7) тренування, де клієнт може додати записи про нове тренування, переглянути відгук тренера та видалити тренування.

Авторизація персоналу має розмежування доступу для адміністратора та тренера.

Адміністратор має можливість працювати з наступними розділами сайту:

— меню адміністратора:

1) керування персоналом, де надається доступ до створення особистого кабінету для персоналу, редагування даних, перегляд графіку роботи для тренерів, а також видалення персоналу.

2) керування клієнтами, де надається доступ до створення особистого кабінету клієнта, редагування даних про нього, можливість призначення тренера та видалення клієнта.

Тренер має можливість працювати з наступними розділами сайту:

— меню тренера:

- 1) робота з клієнтами, де тренеру надається можливість додавати нову та редагувати дійсну програму тренувань;
- 2) керування графіком роботи.

На рис. 3.1 зображено структуру веб-сайту:

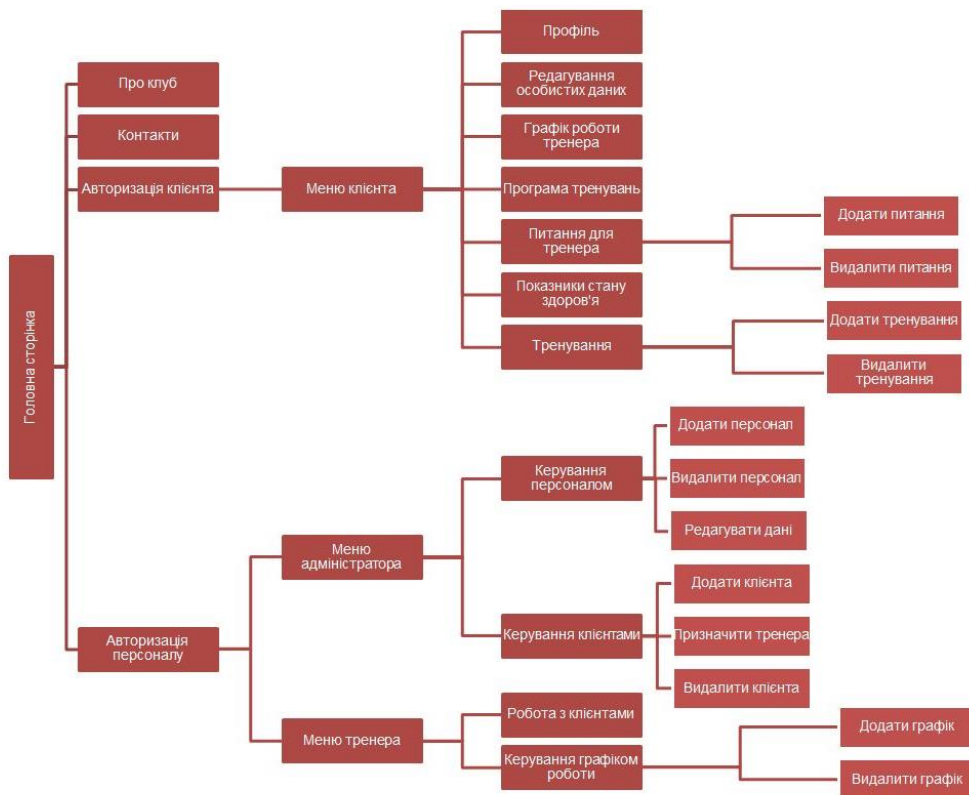


Рисунок 3.1 – Структура веб-сайту

## 3.2 Середовище роботи клієнта

Для того щоб увійти у систему користувач повинен натиснути на головному меню пункт “Особистий кабінет”.

Після натиснення пункту “Особистий кабінет” завантажується сторінка авторизації (рис. 3.2).

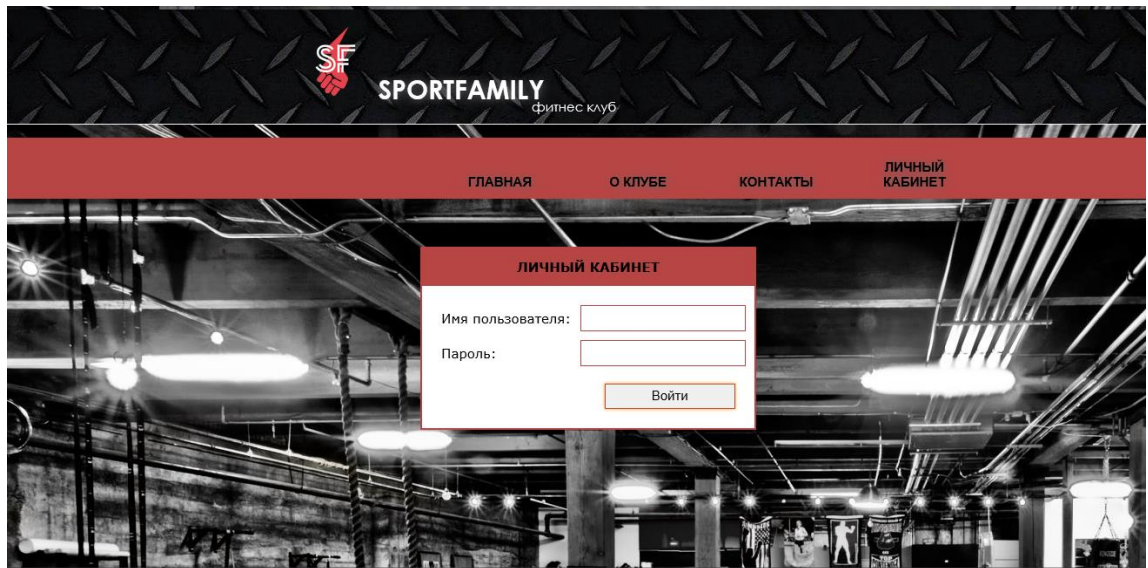


Рисунок 3.2 - Форма авторизації у системі

При введенні імені користувача та пароля форма відправляє дані на обробку.

Код обробки форми наведено нижче:

```
<?php
$fitname = "";
if (isset($_POST['submit'])) { $needfulfield =
array("username", "password");
confirm_availability($needfulfield);
if (empty($erfatal)) {
$fitname = $_POST["username"];
$password = $_POST["password"];
$detectedus = entry_try($fitname, $password);
if ($detectedus) {
$_SESSION["id"] = $detectedus["id"];
$_SESSION["username"] = $detectedus["username"];
header("location: user_content.php");
exit;} else { $_SESSION["inform"] = "Неверно введен пароль
или имя пользователя!"; }
```

Функція `confirm_availability()` в представленому коді виконує перевірку порожніх полів форми.

Код наведено нижче:

```
function confirm_availability ($needfulfield) {
    global $erfatal;
    foreach ($need_fields as $poleim) {
        $value = trim($_POST[$poleim]);
        if (!h_value($value)) {
            $erfatal[$poleim] = tfield($poleim) . " не может быть
пустым";}}}
```

У приведеному коді функція `h_value` приймає значення як аргумент та повертає значення типу `bool`. Якщо значення задане та воно не дорівнює рядку, означає, що воно присутнє —“true”, інакше —“false”.

Код функції має наступний вигляд:

```
function h_value($value) {
    return isset($value) && $value !== "";
```

Отримані з форми дані порівнюються з даними бази даних, якщо пара значень відповідає один одному, то користувач вважається зареєстрованим і значення `id` передаються у сесію. Такі важливі дані більш доцільно буде розмістити саме у сесії.

Сесія використовується для того, щоб в змінній запам'ятати `id` користувача та його `username` для подальшої роботи у системі після перезавантаження сторінки.

При використанні сесії користувачу не будуть доступні дані, що передаються, адже сесія знаходиться на серверній стороні, тому її використання більш безпечно в порівнянні з `cookies`. Якщо ж данні користувача не співпадають, на екрані відображується відповідне повідомлення (рис. 3.3).

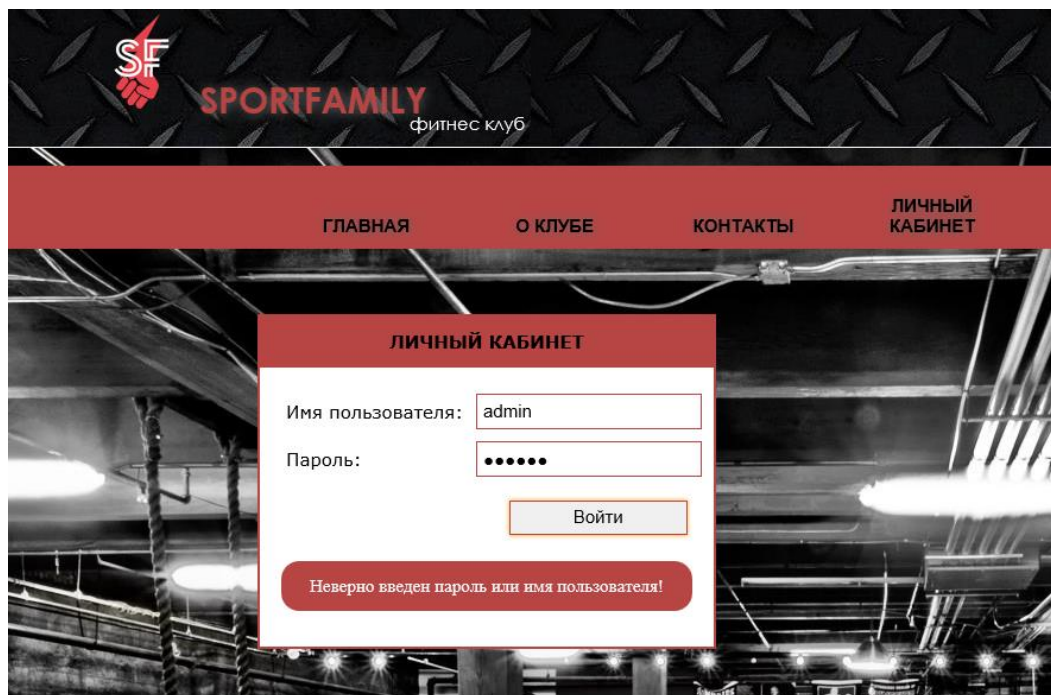


Рисунок 3.3 – Проверка введенных данных

У випадку відправки порожнього поля форми на екрані відображується повідомлення про помилки (рис. 3.4).

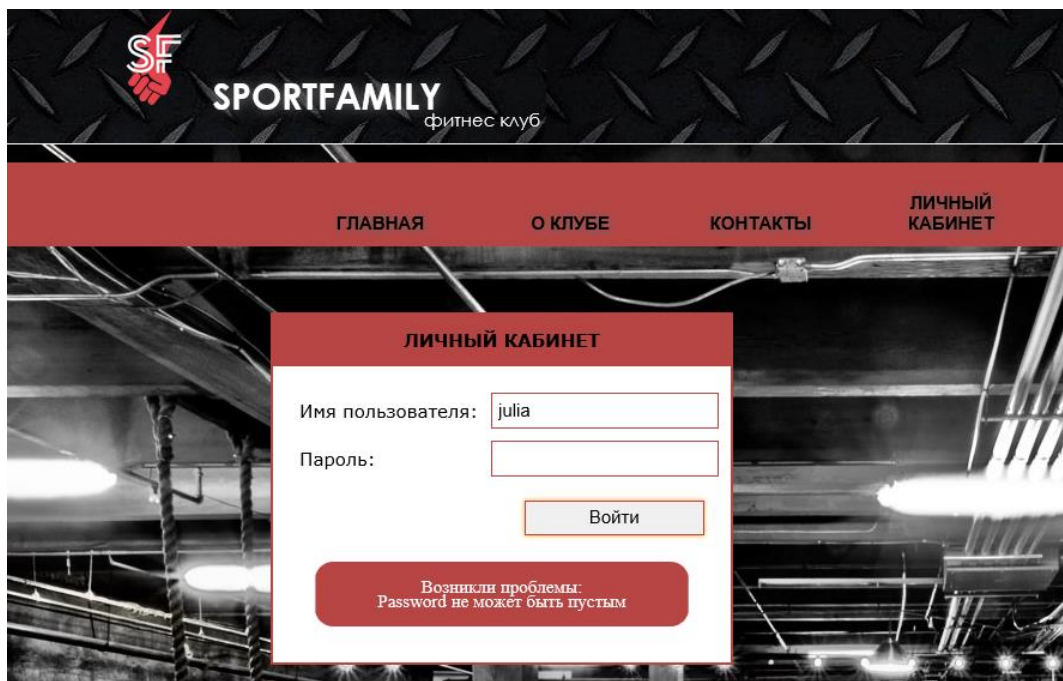


Рисунок 3.4– Пример відправки порожньої форми входу

Зареєстрований користувач при успішному вході потрапляє на особисту сторінку (рис. 3.5)

Головна сторінка власного кабінету відображає ім'я персонального тренера та містить посилання на графік роботи тренера.

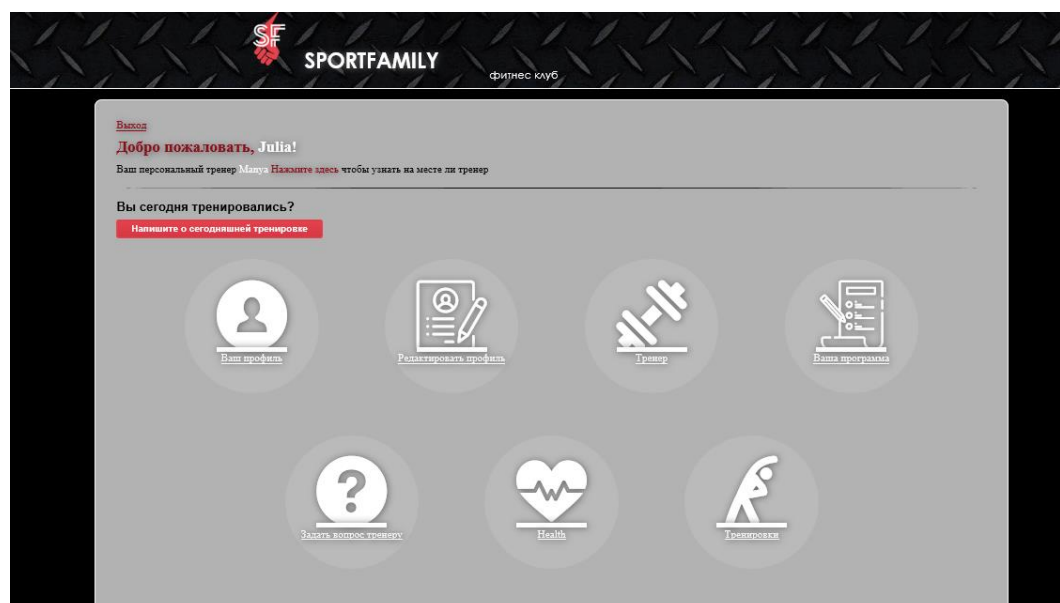
Для пошуку даних клієнта було використано наступну функцію, яка отримує дані з відповідної таблиці БД:

```
function poisk_fitclient($id) { //пошук всіх клієнтів у
таблиці user
global $globalconnect;
$query = "SELECT * FROM user WHERE id = {$id}";
$fit = mysqli_query($globalconnect, $query);
confirm_query($fit); return $fit;}
```

Для пошуку відповідного тренера за іменем клієнта було застосовано наступну функцію яка отримує дані з таблиці admin:

```
function poisk_trainer($fitname) {global
$globalconnect;$query = "SELECT * FROM admin WHERE username =
'{$fitname}'";
$allstaff = mysqli_query($globalconnect, $query);
confirm_query($allstaff); return $allstaff;}
```

На головній сторінці знаходиться головне меню, що містить блоки «Особистий профіль», «Редагувати профіль», «Графік роботи тренера», «Програма тренування», «Health», «Тренування» та посилання «Вихід».



### Рисунок 3.5– Головна сторінка користувача

На сторінці редагування профілю користувач може переглянути інформацію про себе та змінити за допомогою кнопки «Редагувати» такі дані як : ім'я користувача, власне ім'я та прізвище, дату народження, адресу, місто, електронну пошту, контактний номер, мету, яку треба досягти під час тренувань у клубі, а також завантажити власну фотокартку (рис. 3.6).

Ваш профиль  
**Редактировать профиль: julia**

Имя пользователя:

Имя:

Фамилия:

Дата рождения:

Адрес:

Город:

Email:

Контактный номер:

Цели:

Изображение **В настоящее время у вас есть изображение профиля. [Нажмите здесь](#), чтобы заменить его.**

Рисунок 3.6 – Сторінка для редагування інформації профілю

Сторінка «Ваша програма тренувань» відображає інформацію про тренувальну програму клієнта призначену власним персональним тренером.

Щоб відобразити дані сторінки виконувана сесія зберігається до змінної \$user\_id.

Наступним кроком виконується пошук необхідної програми з таблиці (рис. 3.7).





Рисунок 3.7 – сторінка «Програма тренувань»

У випадку відсутності програми у базі даних на сторінці буде відображена відповідна інформація.

Ділянка коду, що вказує на відсутність програми тренування наведена нижче:

```

<div id="pbody">
<p><a href="user_content.php">Профиль</a></p>
<?php echo inform(); ?>
<?php $wprog = mysqli_fetch_assoc($allprog) ?>
<h2>Ваша программа:</h2>
<?php if (empty($wprog["w_program"])) { ?>
<span style="color:red; font-weight: bold;">У вас еще нет
программы! Это может быть потому, что:<br /><br /></span>
<ul style="color: red;">
<li>у Вас нет тренера</li><li>у Вас нет цели</li><li> Ваш
личный тренер не написал вам программу тренировок</li></ul>
<?php } else { ?>
<div id="box"><div id="formbox">
<div id="workout">
<?php echo nl2br( htmlentities($wprog["w_program"])); ?><br
/>
<div style="float: right; font-weight: bold; margin-
top: 6px;">
Дата и время:
<?php echo htmlentities($wprog["w_date"]); ?></div> </div>
<?php } ?></br><h3>Ваш персональный тренер:
<?php if (empty($fitus["admin"])) {
echo '<span> у Вас нет персонального тренера. Обратитесь к
администратору. </span>';} else {echo '<span style="color:
white;">';echo ucfirst(htmlentities($fitus["admin"]));
echo '</span>';} ?></h3> </div></div></div></div>

```

Якщо змінна \$wprog порожня, то на сторінці розміщується інформація про те, чому відсутня програма тренування. Інакше дані змінної завантажуються у блок <div id="workout">.

На сторінці «Питання та відповіді» (рис. 3.8) клієнт може задавати питання щодо тренування своєму тренеру та отримати на них відповіді. Для того щоб додати нове питання на сторінці міститься відповідна кнопка.



Рисунок 3.8– сторінка «Питання для тренера»

При натисненні кнопки «Додати нове питання» завантажувється сторінка з формою на якій міститься поле для введення питання та кнопка «Задати питання». Після відправки форми на сторінці з'являється повідомлення про успішне виконання чи невдачу.

Відображення повідомлення виконується за допомогою виклику функції `inform()`.

Код приведено нижче:

```
session_start();
function inform() { if (isset($_SESSION["inform"])) {
    $output = "<div class=\"inform\">"; $output .=
htmlentities($_SESSION["inform"]); $output .= "</div>";
    $_SESSION["inform"] = null;
    //видалити повідомлення після використання
```

```
return $output; }}
```

Клієнт може також відстежувати дані, щодо стану свого здоров'я за допомогою сторінки «Показники стану здоров'я». Ці данні доступні користувачеві для перегляду, а дозвіл вносити показники до бази даних надається лише тренеру (рис. 3.9).



Профіль

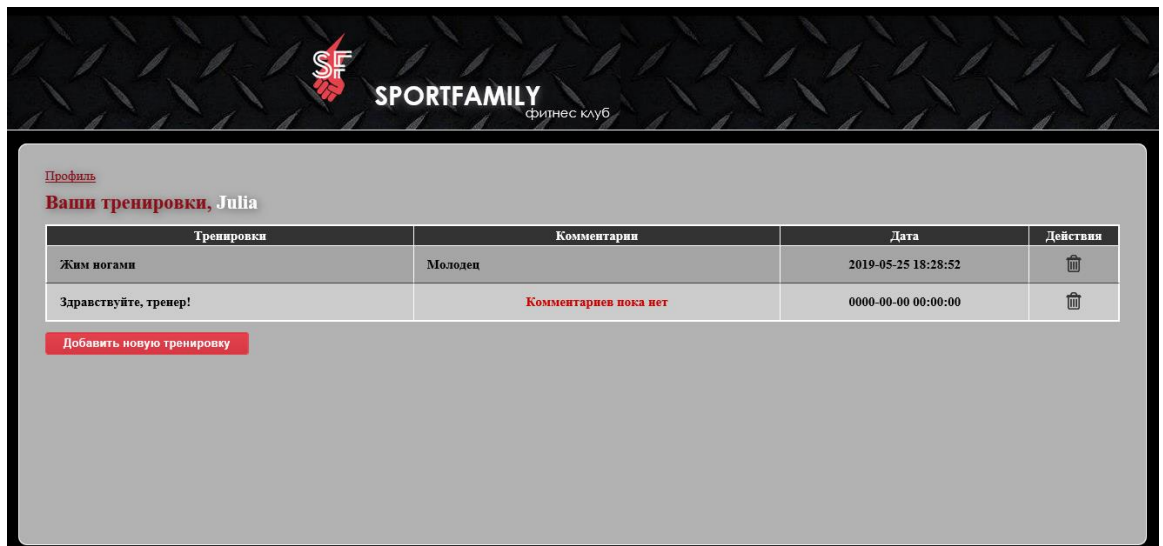
**Ваши показатели Julia:**

Рост	Вес	BMI	Холестерин	Частота серцебиття	Пульс	Дата
168 cm	52 kg	16.00 bmi	4.00 mmol/L	156.00 rpm	15.00 bpm	2019-05-22 08:06:00
168 cm	54 kg	16.20 bmi	3.00 mmol/L	125.00 rpm	12.00 bpm	2019-05-18 18:00:00
168 cm	52 kg	16.00 bmi	2.00 mmol/L	156.00 rpm	15.00 bpm	0000-00-00 00:00:00
168 cm	56 kg	21.00 bmi	5.00 mmol/L	123.00 rpm	24.00 bpm	0000-00-00 00:00:00

Рисунок 3.9 – сторінка «Показники стану здоров'я користувача»

На сторінці тренувань клієнт може переглядати та додавати записи про своє тренування, в якості звіту для тренера, та отримувати коментарі від нього, що до тренувань.

Інформація на сторінці завантажується у вигляді таблиці, де містяться відповідні стовпці: «Тренування», «Коментарі», «Дата» та «Дії» (рис. 3.10).



«Рисунок 3.10 – сторінка «Тренування користувача»

Додавання нового тренування здійснюється за допомогою кнопки «Додати нове тренування».

Після натиснення відповідної кнопки завантажувється форма для внесення повідомлення (рис. 3.11).

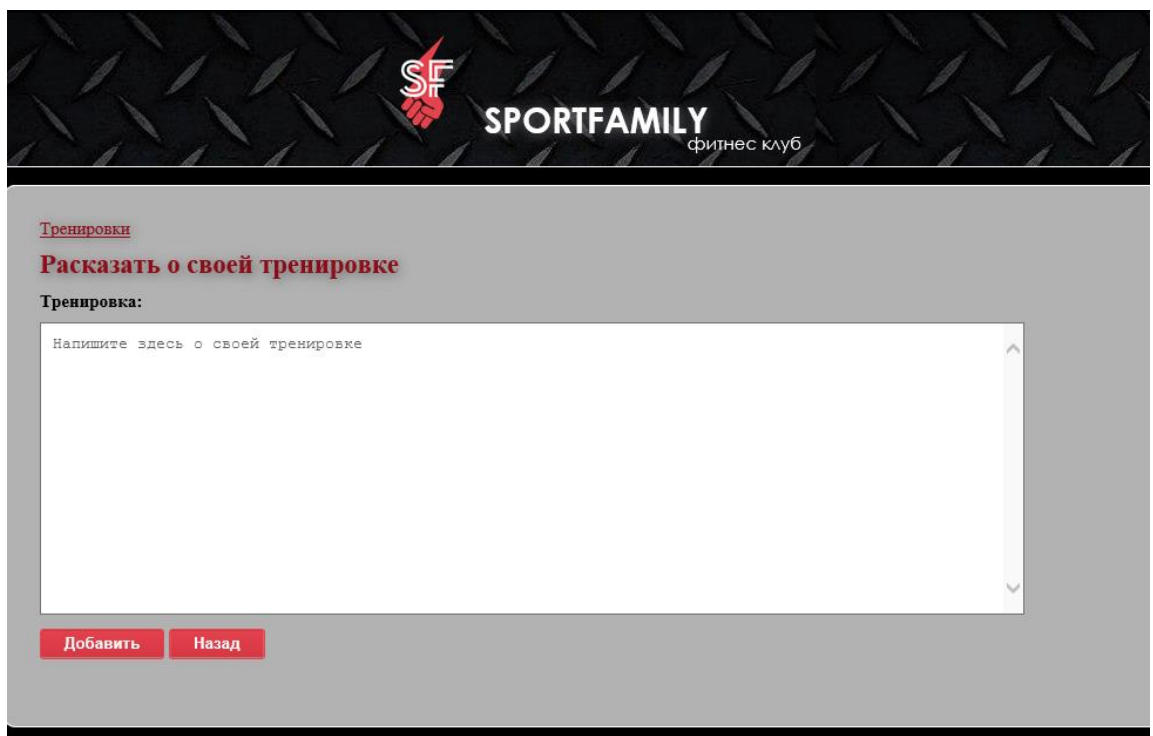


Рисунок 3.11 – сторінка «Додати тренування»

Нижче наведено код обробки форми додавання нового тренування:

```
<?php if (isset($_POST['submit'])) {
    $needfulfield = array("w_session");
    confirm_availability($needfulfield);
    if (empty($erfatal)) {
        $workout =
        $_POST["w_session"]; $workout =
        mysqli_real_escape_string($globalconnect, $workout);
        $query = "INSERT INTO s_workout (user_id, w_session) VALUES
        ('{$user_id}', '{$workout}')";
        $resultworkout = mysqli_query($globalconnect,
        $query);
        if ($resultworkout) { $_SESSION["inform"] = "Треніровка
        успешно добавлена!"; header("location:workout.php");
        } else { $_SESSION["inform"] = "Не удалось создать
        тренировку"; } } }
```

При натисненні на кнопку дані зберігаються до асоціативного масиву `$needfulfield`, наступним кроком виконується перевірка порожніх полів за допомогою раніше використаної функції `confirm_availability()`.

Функція полягає у наступному. Асоціативний масив `$needfulfield` запускається у циклі викликаючи для кожного значення функцію `h_value()`, що повертає значення `true` або `false`. Якщо присутності немає, то буде відображена помилка. Якщо помилок не було виявлено дані `$workout` екрануються та додаються до таблиці тренувань `s_workout`.

Після визначення запиту на додавання, він виконується за допомогою функції `mysql_query()`. Функція `mysql_query ()` повертає у змінну `$result` значення `true`, в разі успішного запиту, і `false` в разі помилки. Тому якщо `$result` відповідає значенню `true`, на екрані фіксується повідомлення про успішне внесення даних, інакше - з'являється повідомлення про невдачу.

### 3.3 Середовище роботи персоналу

#### 3.3.1 Робота тренера

До персоналу системи відносяться адміністратори та тренери. Для того щоб персонал мав доступ до необхідних даних, необхідно увійти у систему за допомогою посилання на головній сторінці «Увійти як персонал».

Після натиснення відповідного посилання завантажується сторінка авторизації. Процес авторизації відбувається аналогічно авторизації клієнта.

Якщо користувач відправляє форму, то її необхідно обробити. Першим кроком виконується валідація, де задаються обов'язкові поля для заповнення. Якщо не виникає жодних помилок, то виконується спроба входу. Для подальшої роботи виконується перевірка даних відправлених за допомогою методу `post`, та відшуковуються поля з іменем `username` та `password`, які копіюють зміст цих полів у відповідні змінні `$fitname` та `$password`. Наступним кроком виконується пошук типу користувача за його `username`. Отримані дані витягуються до результуючого ряду у вигляді асоціативного масиву до змінної `$tstaff`. Після чого виконується функція в якій виконується пошук імені користувача, його типу та паролю.

Для перевірки паролю використовується функція `pass_test()`, яка використовує існуючий хеш.

Код наведено нижче:

```
function pass_test($password, $nowh) {
    $hash = crypt($password, $nowh);
    if ($hash === $nowh) {
        return true;
    } else {
        return false;}}

```

Так як існуючий хеш зберігається у базі даних, то процес складається з двох етапів. Спочатку відшукується користувач і якщо, його знайдено, існуючий хеш порівнюється з його паролем. Якщо пароль не знайдено, то очевидно що виконати вхід не вдалось.

У випадку успішного збігу, тобто значення `$fatrain` було знайдено, то в глобальній змінній `$SESSION` зберігаються `id` користувача, його `username` та тип.

Алгоритм авторизації наведено на рис.3.12

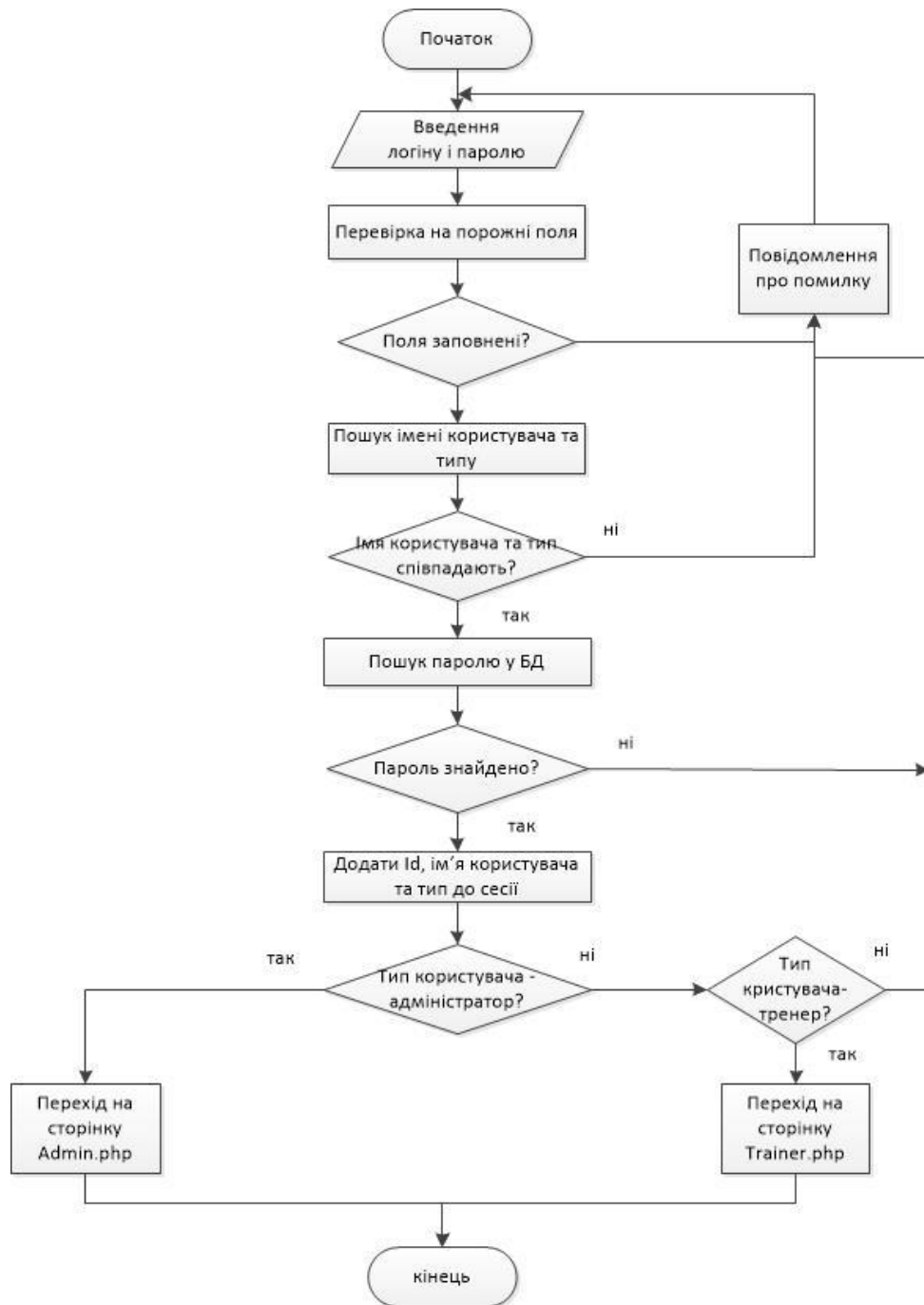


Рисунок 3.12 – Алгоритм авторизації персоналу

Для визначення сторінки переходу в залежності від типу користувача виконується наступний код:

```

if ($found_admin) {
    $_SESSION["admin_id"] = $fatrain["id"];
    $_SESSION["username"] = $fatrain ["username"];
    $_SESSION["type"] = $fatrain["type"];
    if ($tstaff["type"] === "Admin") {
header("location:admin.php");
exit;}
}

```

```

elseif ($tstaff["type"] === "Trainer") {
header("location:trainer.php");
exit;}
} else {$_SESSION["inform"] = "Не правильно введено имя
пользователя или пароль"; } }

```

Якщо користувач увійшов у систему як «Тренер», то завантажується сторінка «Меню тренера» (рис. 3.13). На сторінці розміщено блоки «Мої клієнти» та «Графік роботи».



Рисунок 3.13 – Головна сторінка тренера

Сторінка «Мої клієнти» що відображена на рисунку 3.14 містить таблицю з даними про клієнтів тренера. Вона містить такі назви стовпців: ім'я, прізвище, дату народження клієнта, номер телефону, мету та стовпець дії, де тренер може додавати нову та редагувати вже існуючу тренувальну програму клієнта, переглядати звітність тренувань, додавати показники стану здоров'я клієнта після тренування та відповідати на питання.





Рисунок 3.14 – Сторінка «Клієнти»

Сторінка «Редагування» програми має наступний вигляд (рис. 3.15):

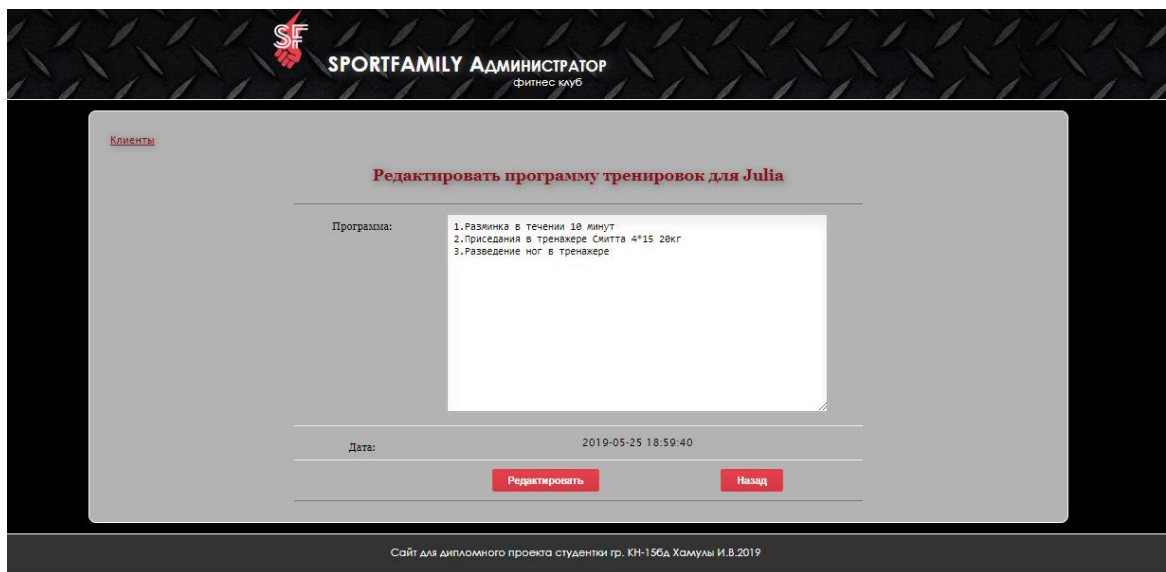


Рисунок 3.15 – Сторінка для редагування тренувальної програми

Обробка сторінки виконується за допомогою наступного коду:

```
<?php
    $program_id= programuser($_GET["id"]);
    //Пошук програми для редагування
    $fit = id_user($_GET["id"]);
    // пошук імені користувача для відображення у заголовку
сторінки
```

```

        if (!$program_id || $fit["admin"] !== $_SESSION["username"])
    {
        //якщо не знайдено програми тренування або не вказано
        тренера, то перехід на сторінку меню
        header("location:mclients_trainer.php");
        exit;}}>
        <?php if (isset($_POST['submit'])) { if (empty($erfatal))
    {$fitus = $wprogid["user_id"];
        // получает программу из текстового поля и устанавливает ее
    в столбец программы тренировок mysql
    $wprog= $_POST["w_program"];
    $wprog= mysqli_real_escape_string($globalconnect,$wprog);
    $query = "UPDATE p_workout SET w_program = '{ $wprog}' WHERE
    user_id = {$fitus} LIMIT 1";
    $resultwprog = mysqli_query($globalconnect, $query);
    if ($resultwprog && mysqli_affected_rows($globalconnect) == 1)
    {$SESSION["inform"] = "Программа обновлена";
    header("location:mclients_trainer.php");
    exit;}
    else {
    $_SESSION["inform"] = "Не удалось обновить программу";}}

```

У змінну `$program_id` за допомогою створеної функції `programuser` методом `$_GET` виконується пошук `id` програми, що необхідно відредагувати. У змінну `$fit` за допомогою функції `id_user()` виконується пошук імені користувача для відображення його у заголовку.

Якщо `id` програми відсутній, недійсний або ж не знайдений у базі даних, не обрано тренера або ж тренер користувача не відповідає дійсному тренерові виконується перехід до сторінки `mclients_trainer.php`.

Якщо було натиснуто кнопку відправки виконується обробка форми, у випадку відсутності помилок. Наступним кроком виконується оновлення. Змінна `$wprog` отримує тренувальну програму з `textarea` та встановлює її у стовпець `w_program` бази даних `mysql`.

Функція `mysqli_affected_rows` повертає кількість записів, які були використані для оновлення за поточним з'єднанням. Якщо результат є успішним, то повертається 1 запис. Якщо жодного запису не було змінено, це означає те, що або оновлення не виконалось або жоден запис не збігся з умовою `WHERE user_id`.

На сторінці «Здоров'я» (рис. 3.16) тренер може переглядати минулі показники стану здоров'я клієнта (зріст, вага, індекс маси тіла, холестерин, частоту серцебиття, пульс) та додавати нові.

**Характеристики здоров'я для Julia**

Рост:  см  
 Вес:  kg  
 Индекс массы тела:  bmi  
 Холестерин:  mmol/L  
 Частота сердцебиения:  rpm  
 Кровяное давление:  bpm

[Отмена](#)

---

**Старые характеристики здоровья:**

Рост	Вес	Индекс массы тела:	Холестерин:	Частота сердцебиения:	Кровяное давление:	Дата и время	Действия
168 cm	52 kg	16.00 bmi	4.00 mmol/L	156.00 rpm	15.00 bpm	2019-05-22 08:06:00	
168 cm	54 kg	16.20 bmi	3.00 mmol/L	125.00 rpm	12.00 bpm	2019-05-18 18:00:00	
168 cm	52 kg	16.00 bmi	2.00 mmol/L	156.00 rpm	15.00 bpm	0000-00-00 00:00:00	
168 cm	56 kg	21.00 bmi	5.00 mmol/L	123.00 rpm	24.00 bpm	0000-00-00 00:00:00	

Сайт для дипломного проекта студентки гр. КН-156д Хамулы И.В.2019

Рисунок 3.16 – Сторінка для внесення показників здоров'я клієнта

Сторінка з питаннями містить таблицю, що містить питання від клієнта, поле для відповіді тренера та час і дату (рис. 3.17).

**SPORTFAMILY АДМИНИСТРАТОР**  
фитнес клуб

[Профиль](#)

**Вопросы от Julia :**

Вопросы	Ответы	Дата и время
Сколько выполнять подходов?	необходимо выполнять 4 подхода по 20 раз <input type="button" value="Ответ"/>	2019-06-06 06:43:52
Сколько упражнений выполнять?	Необходимо выполнять по 5-6 упражнений для каждой группы мышц <input type="button" value="Ответ"/>	2019-06-06 06:50:38

Рисунок 3.17 – Сторінка для перегляду питань від клієнта

Обравши блок «Графік роботи» на головній сторінці тренера надається можливість призначення та перегляду графіку годин його роботи у відповідний день тижня, а також відобразити графік у особистих кабінетах його клієнтів. Тільки тренер може додавати поточний графік роботи. Для того щоб додати новий графік спочатку треба видалити попередній (рис. 3.18).



Рисунок 3.18 –Сторінка «Графік роботи»

Додати графік можна за допомогою кнопки «Додати». Після чого завантажиться сторінка, де необхідно обрати потрібні години роботи (рис. 3.19).

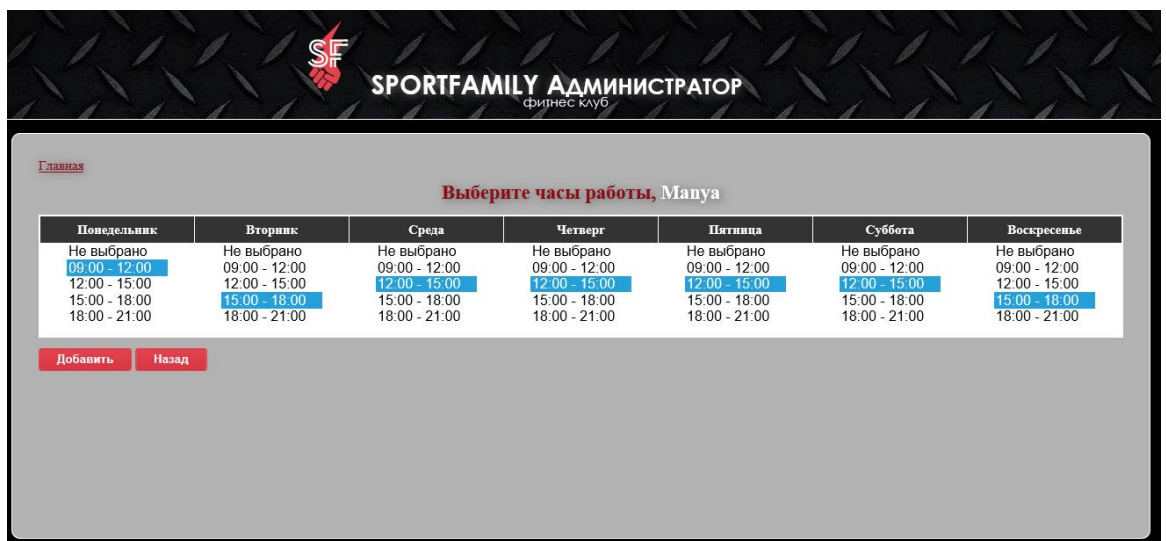


Рисунок 3.19 –Сторінка з вибором часу роботи

Фрагмент форми внесення даних має наступний вигляд:

```
<form          action="trainer_board.php?id=<?php          echo
urlencode($atrain ["id"]); ?>" method="POST">
  <table class="calendar" >
  <thead>
  <tr><th>Понедельник</th><th>Вторник</th>
  <th>Среда</th><th>Четверг</th>
  <th>Пятница</th><th>Суббота</th>
  <th>Воскресенье</th></tr></thead><tbody><tr>
  <td style="text-align: center;">
  <select class="graphic" name="mon []" multiple="multiple"
  required> <option value="Не выбрано!<br />">Не выбрано</option>
  <option value="09:00 - 12:00<br /><br />">09:00 - 12:00</option>
  <option value="12:00 - 15:00<br /><br />">12:00 - 15:00</option>
  <option value="15:00 - 18:00<br /><br />">15:00 -
  18:00</option><option value="18:00 - 21:00<br /><br />">18:00 -
  21:00</option></select></td>
```

### 3.3.2 Робота адміністратора

Якщо користувач увійшов у систему як «Адміністратор», то завантажується сторінка «Меню адміністратора», що зображена на рис. 3.20.

Адміністратор має можливість керувати контентом сайту, адміністраторами та клієнтами.

На сторінці «Меню адміністратора» відображено зазначені блоки.



Рисунок 3.20 – Головна сторінка кабінету адміністратора

На сторінці керування персоналом адміністратору надається таблиця яка містить наступні стовпці: ім'я користувача, тип, графік роботи для

тренера та дії (рис. 3.21). Адміністратору надається можливість редагування інформації про тренерів та адміністраторів а, також видалення даних про персонал з бази даних.



Рисунок 3.21 – Головна сторінка кабінету адміністратора

Сторінка з редагуванням даних про адміністратора відображена на рис. 3.22.

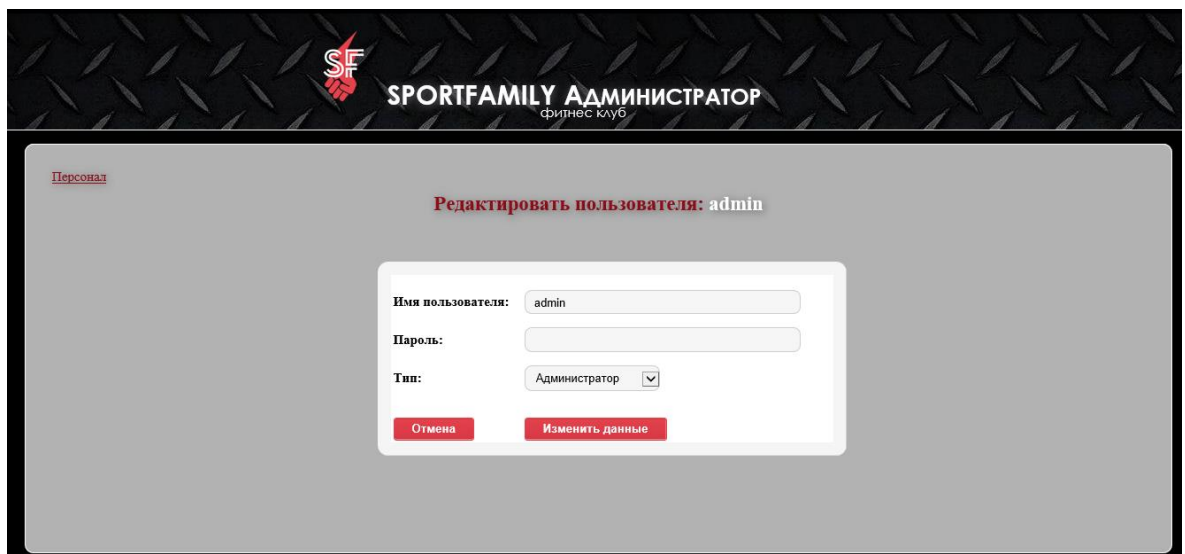


Рисунок 3.22 – Сторінка «Редагувати дані персоналу»

Адміністратору також надається можливість введення даних про новий персонал. Для цього у меню сторінки знаходиться відповідне гіперпосилання. Після переходу на сторінку завантажується форма для реєстрації нового користувача (рис. 3.23).

**Создать пользователя**

Имя:

Пароль:

Тип персонала:

Рисунок 3.23 – Сторінка «Створити нового користувача»

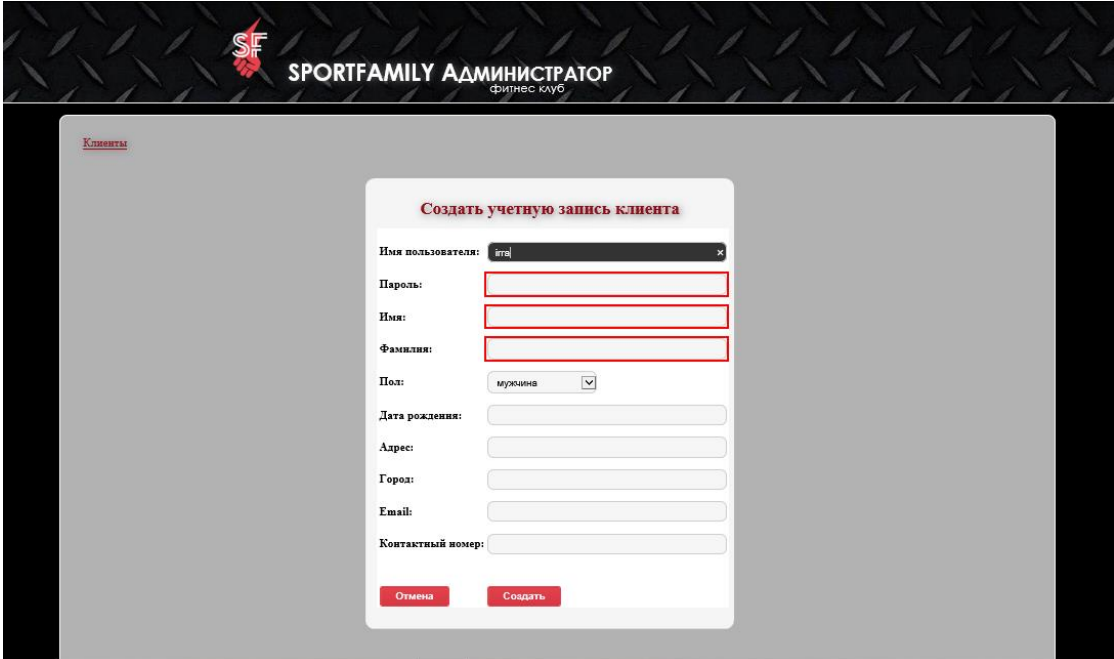
На сторінці управління клієнтами розміщена таблиця з анкетними даними клієнтів рис. 3.24. Адміністратору надається можливість призначити тренера клієнтам, редагувати дані про них та видаляти клієнта з системи.

**Управление клиентами**

Имя пользователя	Фамилия	Имя	Дата рождения	Адрес	Город	Email	Контактный номер	Цель	Действия
dima	Дмитрий	Цуканов	1997-10-20		Северодонецк		06456876456	Набрать 5кг мышечной массы	Назначить тренера
julka	Юлия	Брилко	1985-12-12	ул.Богдана Хмельницкого,22	Рубежное	julibrilko@gmail.com	380990373226	Набрать мышечную массу	Назначить тренера
leva	Мария	Евгеньева	1992-04-12					Нет цели!	Назначить тренера
roina	Полина	Подкуйко	1993-02-12	ул.Набережная,12	Северодонецк	simbapolina@gmail.com	0990373326	Подготовиться к осеннему сезону	Назначить тренера
tatyana	Татьяна	Дахно	1998-02-15	пер.Отакара Яроша 10 кв.32	Харьков			Нет цели!	Назначить тренера

Рисунок 3.24 – Сторінка «Керування клієнтами»

Також адміністратор має можливість реєструвати нових клієнтів клубу у системі (рис. 3.25).



The screenshot displays the 'SPORTFAMILY АДМИНИСТРАТОР' (SPORTFAMILY ADMINISTRATOR) interface for a fitness club. The main heading is 'Создать учетную запись клиента' (Create client account). The form includes the following fields: 'Имя пользователя:' (Username) with a dropdown menu showing 'itaj'; 'Пароль:' (Password) with a red-bordered input field; 'Имя:' (Name) with a red-bordered input field; 'Фамилия:' (Surname) with a red-bordered input field; 'Пол:' (Gender) with a dropdown menu set to 'мужчина' (male); 'Дата рождения:' (Date of birth) with a date picker; 'Адрес:' (Address) with a text input; 'Город:' (City) with a text input; 'Email:' with a text input; and 'Контактный номер:' (Contact number) with a text input. At the bottom of the form are two buttons: 'Отмена' (Cancel) and 'Создать' (Create).

Рисунок 3.25 – Сторінка «Реєстрація нового клієнта»

Алгоритм реєстрації нового клієнта наведено на рис. 3.26



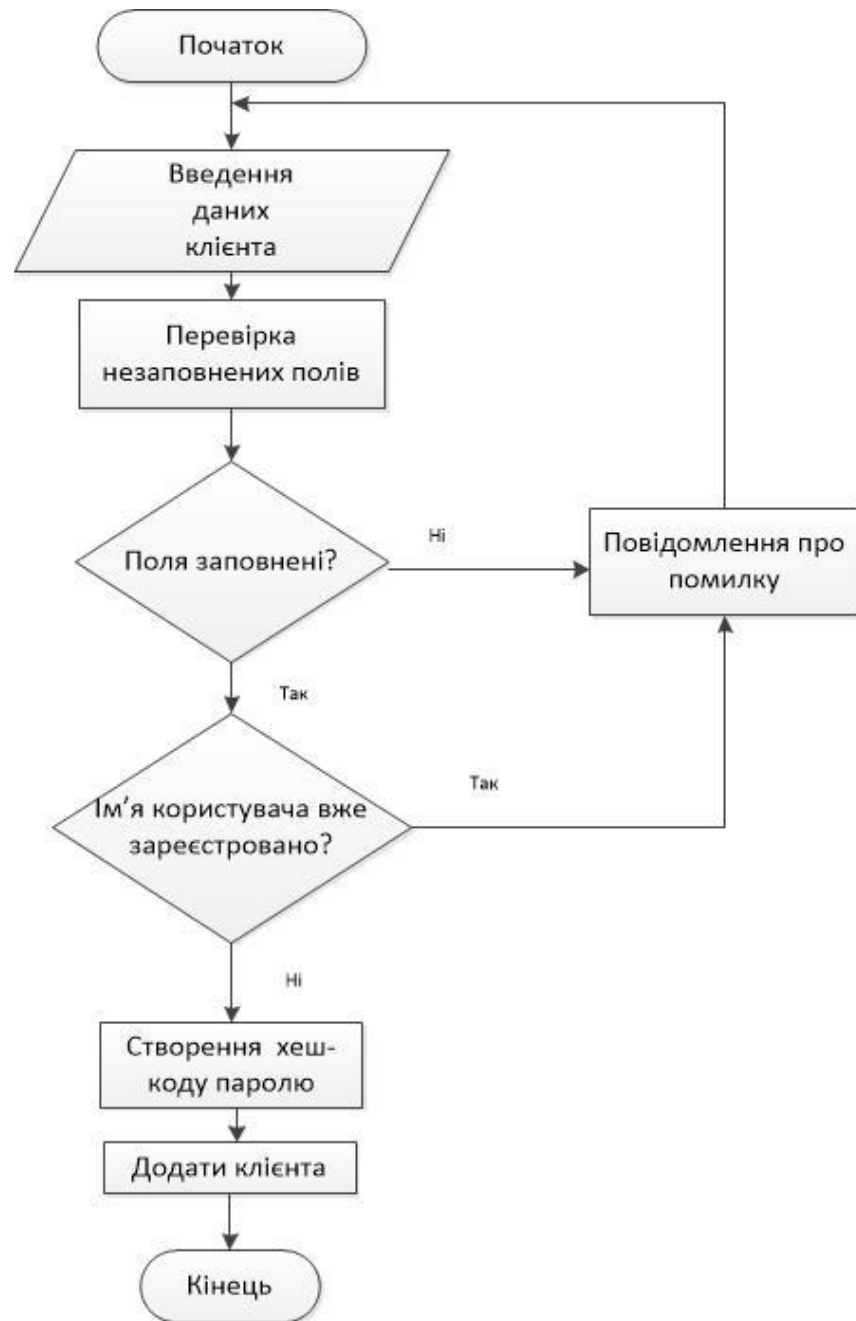


Рисунок 3.26 – Алгоритм реєстрації нового клієнта

## **Висновки до третього розділу**

У третьому розділі було виконано структуру та опис роботи розробленого сайту.

Він охоплює:

- розробку структури сайту;
- середовище роботи клієнта, де розглянуто основні сторінки з якими працює клієнт;
- середовище роботи персоналу, а саме роботу тренера та роботу адміністратора.

Розділ містить фрагменти коду виконуваних сторінок.

Також у розділі наведено структурну схему сайту.

## 4 ОХОРОНА ПРАЦІ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра було розробити web-додаток для роботи фітнес-клубу, і в результаті створено додаток, що задовольняє всі зазначені вимоги. Так як в процесі проектування використовувався персональний комп'ютер, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблятися та використовуватися розроблена система.

### 4.1 Аналіз стану умов праці

#### 4.1.1 Вимоги до приміщення

Таблиця 4.1 - Розміри приміщення.

Найменування	Значення
Довжина, м	3,2
Ширина, м	2,9
Висота, м	2,5
Площа, м <sup>2</sup>	9,28
Об'єм, м <sup>3</sup>	23,2

Згідно з ДСН 3.3.6.042-99 [6] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності.

Для забезпечення потрібного рівного освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

#### 4.1.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 [7] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне Значення	Нормативне Значення
Висота робочої поверхні, мм	700	680 ÷ 800
Висота простору для ніг, мм	650	не менше 600
Ширина простору для ніг, мм	540	не менше 500
Глибина простору для ніг, мм	660	не менше 650
Висота поверхні сидіння, мм	420	400 ÷ 500
Ширина сидіння, мм	410	не менше 400
Глибина сидіння, мм	420	не менше 400
Висота поверхні спинки, мм	500	не менше 300
Ширина опорної поверхні спинки, мм	400	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷ 800

## 4.2 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

### 4.2.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-1.28-10 [11], яке встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга  $U=+220\text{В} \pm 5\%$ ;
- робочий струм  $I=2\text{А}$ ;
- споживана потужність  $P=350\text{ Вт}$ .

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна Оцінка	Нормативні Документи
1	2	3	4
<b>Фізичні:</b>			
підвищена або	-//-	3	[7]

знижена вологість повітря			
підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	3	[9] [10]
<b>Психофізіологічні:</b>			
1	2	3	4
нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- формулювання теми; - пошук інформації про предметну область; - пошук інформації про наявні аналоги; - проектування структур та алгоритмів; - виконання роботи; - оформлення записки.	4	[10] [11]
фізичні (статичне - сидіння)	порушення умов організації робочого часу (безперервна робота)	2	[8] [11]

Робочі місця в обов'язковому порядку повинні відповідати вимогам до санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, що затверджені постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [7].

#### 4.2.2 Пожежна безпека

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80...100°C). При проходженні електричного струму по провідниках і деталей виділяється

тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини, електроди).

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- 1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 °С;
- 2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С;
- 3) склотекстоліт ДЦ - матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- 4) пластикат кабельний №489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;
- 5) деревина - будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 °С, температура самозаймання 399 °С.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходиться пожежонебезпечні речовини і матеріали відповідно до ДСТУ Б В.1.1-36:2016 [12] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та

важкозаймисті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, надається II ступінь вогнестійкості.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки устаткування;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [13].

### 4.2.3 Електробезпека

Виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних



розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

### **4.3 Гігієнічні вимоги до параметрів виробничого середовища**

#### **4.3.1 Мікроклімат**

Мікроклімат робочих приміщень - це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт 1а. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [7] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період Року	Категорія Робіт	Температура С <sup>0</sup>	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1

#### **4.3.2 Освітлення**

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових - 1/10:

$$S_b = \left(\frac{1}{5} / \frac{1}{10}\right) * S_n \quad (4.1)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;

$S_n$  – площа підлоги,  $m^2$  .

$$S_n = a \cdot b = 4,4 \cdot 2,8 = 12,32 \text{ м}^2 ,$$

$$S = 1/10 \cdot 25 = 1,232 \text{ м}^2 .$$

Приймаємо 1 вікно площею  $S=1,6 \text{ м}^2$ .

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 4,4 м, ширина 2,8 м, світильниками ЛПО2П, оснащеними лампою типа ЛБ (одна - 80 Вт) з світловим потоком 5400 лм. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $n$  виробляється по формулі (4.2):

$$n = \frac{E * S * Z * K}{F * U * M} \quad (4.2)$$

де  $E$  - нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  - освітлювана площа,  $m^2$  ;  $S = 12,32 \text{ м}^2$  ;

$Z$  - поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;

$K$  - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U - коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п.

- 0,575 M - число люмінесцентних ламп в світильнику - 1;

F - світловий потік лампи - 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 * 12,32 * 1,15 * 1,5}{5400 * 0,575 * 1} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, оснащених лампами типа ЛБ (одна - 80 Вт) зі світловим потоком 5400 лм.

#### **4.4 Вентилювання**

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення > 40 м<sup>3</sup> на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

Для підтримки в приміщенні оптимального температурного режиму відповідно до вимоги ДБН В.2.5-67:2013 [15] є централізоване опалювання і вентиляція. У теплий період року використовується кондиціонування.

#### **4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій**

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря:

а) якщо об'єм приміщення  $20 \text{ м}^3$ , то потрібно подати не менш як  $30 \text{ м}^3$  /год повітря;

б) якщо об'єм приміщення у межах від  $20$  до  $40 \text{ м}^3$ , то потрібно подати не менш як  $20 \text{ м}^3$ /год повітря;

в) якщо об'єм приміщення становить понад  $40 \text{ м}^3$ , допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;

#### 4.5.1 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [14], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{шт.з.} = \frac{R_{\partial} * R_{пр.з.}}{R_{пр.з.} - R_{\partial}} \quad (4.3)$$

де  $R_{пр.з.}$  - опір природних заземлювачів;

$R_{\partial}$  - допустимий опір заземлення.

Якщо природні заземлювачі відсутні, то  $R_{шт.з.} = R_{\partial}$ .

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з.} = \frac{4 * 40}{40 - 4} \approx 40 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho = 40$  Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту,  $R_{розр.}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $R_{розр.в.}$ , і горизонтальних  $R_{розр.г.}$ , Ом·м за формулою:

$$R_{розр.} = \Psi * \rho \quad (4.4)$$

де  $\Psi$  - коефіцієнт сезонності для вертикальних заземлювачів І кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів  $R_{розр.в.} = 1,7$  і горизонтальних  $R_{розр.г.} = 5,5$  Ом·м

$$P_{\text{озр.в}} = 1,7 * 40 = 68 \text{ Ом/м}$$

$$P_{\text{озр.г}} = 0,55 * 40 = 220 \text{ Ом/м}$$

4) Розраховується опір розтікання струму вертикального заземлювача  $R_B$ , Ом, за (4.5).

$$R_B = \frac{P_{\text{розр.в}}}{2 * \pi * l_B} * \left( \ln \frac{2 * l_B}{d_{\text{ст}}} + \frac{1}{2} * \ln \frac{4 * t + l_B}{4 * t - l_B} \right) \quad (4.5)$$

де  $l_B$  - довжина вертикального заземлювача (для труб - 2–3 м;  $l_B=3$  м);

$d_{\text{ст}}$  - діаметр стержня (для труб - 0,03–0,05 м;  $d_{\text{ст}}=0,05$  м);

$t$  - відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_B + \frac{l_B}{2} \quad (4.6)$$

де  $h_B$  - глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_B = \frac{68}{2 * \pi * 3} * \left( \ln \frac{2 * 3}{0,05} + \frac{1}{2} * \ln \frac{4 * 2,3 + 3}{4 * 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_B$ :

$$n = \frac{2 * R_B}{R_0} = \frac{2 * 18,5}{4} = 9,25 \quad (4.7)$$

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $n_B$ , шт:

$$n_B = \frac{2 * R_B}{R_0 * \eta_B} = \frac{2 * 18,5}{4 * 0,57} = 16,2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м:

$$l_c = 1,05 * L_B * (n_B - 1) \quad (4.9)$$

де  $L_B$  - відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3$  м);

$n_B$  - необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 * 3 * (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)  $R_\Gamma$ , Ом:

$$R_\Gamma = \frac{P_{розр.г}}{2 * \pi * l_c} * \ln \frac{2 * l_c^2}{d_{см} * h_\Gamma} \quad (4.10)$$

де  $d_{см}$  - еквівалентний діаметр смуги шириною  $b$ ,  $d_{см} = 0,95b$ ,  $b = 0,15$  м;

$h_\Gamma$  - глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  - довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_\Gamma = \frac{220}{2 * \pi * 48} * \ln \frac{2 * 48^2}{0,95 * 0,15 * 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$ . відповідно до необхідної кількості вертикальних заземлювачів  $n_B$ . Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$  (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заз} = \frac{R_B * R_\Gamma}{R_B * \eta_c + R_\Gamma * n_B * \eta_B} \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{заз}$  Ом, а саме:

$$R_{заз} = \frac{18,5 * 8,1}{18,5 * 0,3 + 8,1 * 16 * 0,57} = 1,9 \leq R_\delta$$

### Висновки до четвертого розділу

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які

заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.



## ВИСНОВКИ

У результаті дипломної роботи було розроблено веб-додаток, що автоматизує роботу фітнес клубу, а саме адміністраторів, тренерів та клієнтів.

Додаток має зручний інтерфейс, що не потребує додаткового навчання для роботи з ним.

Створений додаток дозволяє достовірно та швидко виконувати потрібні операції з даними, а тому дозволяє швидко та якісно виконувати свою роботу тренерам, адміністраторам та клієнтам при сумісній праці.

В результаті виконання роботи було визначено необхідність розробки, а також побудовано схеми структури сайту та бази даних.

В повному обсязі було проведено аналіз технічного завдання та обрано необхідне програмне забезпечення для розробки додатку.

Фітнес в нас час має досить великий попит, а тому персоналу галузі фітнесу доводиться працювати з великою кількістю клієнтів. Тому розроблений додаток буде гарним доповненням невеликим фітнес-клубам та тренажерним залам, адже значно підвищить ефективність роботи.

**ПЕРЕЛІК ДЖЕРЕЛ І ПОСИЛАНЬ**

- 1.CSS [Електронний ресурс] // Вікіпедія: вільна енциклопедія.//URL: <https://uk.wikipedia.org/wiki/CSS> // (дата звернення:14.06.2019)
- 2.PHP [Електронний ресурс] // Вікіпедія: вільна енциклопедія.//URL:[https://uk.wikipedia.org/wiki/PHP#cite\\_note-gutmans-2](https://uk.wikipedia.org/wiki/PHP#cite_note-gutmans-2) (дата звернення:14.06.2019)
3. Африна Е. И. Школы, компьютеры, сеть, учителя и ученики / Е. И. Африна // Народное образование. – 2001. – №8. – С. 138-143.
4. Грэхем И. Объектно-ориентированные методы. Принципы и практика. [Пер. с англ.] / И.Грэхем – М.: Издательский дом "Вильямс". – 2004. – 880 с.
5. HTTP Cookies [Електронний ресурс] //URL: <http://um.co.ua/11/11-2/11-29333.html> // (дата звернення:15.06.2019)
6. Астахова И.Ф., Толстобров А.П., Мельников В.М. SQL в примерах и задачах. Воронеж, 2001. 160с.
7. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. Міністерство охорони здоров'я України (МОЗ). Постанова № 42 від 01.12.1999
8. ДСанПІН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Міністерство охорони здоров'я України (МОЗ).
9. ГОСТ 12.1.030-81 ССБТ. Електробезпека. Захисне заземлення. Занулення.
10. ГОСТ 13109-97. Норми якості електричної енергії в системах електропостачання загального призначення.
11. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями.

12. ДСТУ Б В.1.1-36:2016. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою
13. ГОСТ 12.1.044-89. Система стандартів безпеки праці. Вогнестійкість. Номенклатура показників і методи їх визначення (ІСО 4589-84).
14. НПАОП 40.1-1.01-97. Правила безпечної експлуатації електроустановок.
15. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування.

## Додаток А

### Лістинг програми

#### Fitclubdb.php

```
<?php
define("DB_SERVER", "localhost");
define("DB_USER", "root");
define("DB_PASS", "root");
define("DB_NAME", "sportfamily");
// Создать соединение с базой данных
$globalconnect = mysqli_connect(DB_SERVER, DB_USER,
DB_PASS, DB_NAME);

if (mysqli_connect_errno()) {
    die("Ошибка подключения к базе данных: " .
        mysqli_connect_error() . " (" .
mysqli_connect_errno() . ")");}
?>
```

#### Func.php

```
<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php
$erfatal = array();
function tfield($poleim) {
    $poleim = str_replace("_", " ", $poleim);
    $poleim = ucfirst($poleim);
    return $poleim;
}
function h_value($value) {
    return isset($value) && $value !== "";
}
function confirm_availability($needfulfield) { //перевірка
на порожне поле
    foreach ($needfulfield as $poleim) {
        $value = trim($_POST[$poleim]);
        if (!h_value($value)) {
            $erfatal[$poleim] = tfield($poleim) . " не может быть
пустым"; } }
}
function has_inclusion_in($value, $set) {
    return in_array($value, $set);
}
function confirm_query($result_set) {
    if (!$result_set) {
die("Запрос не выполнен");}
}
```

```

function errorsset($erfatal = array()) { //набір помилок
    $output = "";
    if (!empty($erfatal)) {
        $output .= "<div class=\"error\">";
        $output .= "Возникли проблеми:";
        $output .= " ";
        foreach ($erfatal as $key => $erfatal) {
            $output .= "<p> ";
            $output .= htmlentities($erfatal);
            $output .= "</p>";
        }
        $output .= " ";
        $output .= "</div>";
    }
    return $output;
}

function id_question($question_id, $public = true) { // пошук
питання за номером
    global $globalconnect;
    $sqnom = mysqli_real_escape_string($globalconnect,
    $question_id);
    $query = "SELECT * FROM ask WHERE id = {$sqnom} LIMIT
    1";

    $setq = mysqli_query($globalconnect, $query);
    confirm_query($setq);
    if ($question = mysqli_fetch_assoc($setq)) {return
    $question;
    } else {
        return null;
    }
}

function id_health($health_id, $public = true) { // пошук
даних про здоров'я за номером
    global $globalconnect;
    $hnom= mysqli_real_escape_string($globalconnect,
    $health_id);
    $query = "SELECT * FROM health WHERE id = {$hnom} LIMIT
    1";

    $hset = mysqli_query($globalconnect, $query);
    confirm_query($hset);
    if ($health = mysqli_fetch_assoc($hset)) { return
    $health;
    } else { return null;
    }
}

function all_staff() { //пошук всіх користувачів у таблиці
адмін
    global $globalconnect;

    $query = "SELECT * FROM admin ORDER BY username ASC";

```

```

    $allstaff = mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    return $allstaff;
}

function all_trainers() { //пошук всіх тренерів
    global $globalconnect;
    $query = "SELECT * FROM admin WHERE type = 'Trainer' ";
    $allstaff = mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    return $allstaff;
}

function poisk_fitclient($id) { // пошук всіх клієнтів у
таблиці user
    global $globalconnect;
    $query = "SELECT * FROM user WHERE id = {$id}";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function poisk_program($id) { //пошук програми тенування
    global $globalconnect;
    $query = "SELECT * FROM p_workout WHERE user_id =
{$id}";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function idquestion($id) { // пошук питання за номером
    global $globalconnect;
    $query = "SELECT * FROM ask WHERE user_id = {$id}";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function workoutuser($id) { //пошук тренування за номером
    global $globalconnect;
    $query = "SELECT * FROM s_workout WHERE user_id =
{$id}";
    $wset= mysqli_query($globalconnect, $query);
    confirm_query($wset);
    return $wset;
}

function huser($id) { // пошук даних про стан здоров'я
клієнта
    global $globalconnect;

```

```

    $query = "SELECT * FROM health WHERE user_id = {$id} ORDER
BY date DESC";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function all_clients() { //пошук всіх клієнтів з таблиці
клієнтів
    global $globalconnect;
    $query = "SELECT * FROM user ORDER BY username ASC";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function ustrain($atrain) { //пошук тренера у таблиці
клієнтів
    global $globalconnect;

    $query = "SELECT * FROM user WHERE admin = {$atrain}";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function tgraph($atrain) { //
global $globalconnect;
    $query = "SELECT * FROM a_presence WHERE admin_id =
{$atrain}";
    $allstaff = mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    return $allstaff;
}

function id_admin($admin_id) { //
    global $globalconnect;
    $spers= mysqli_real_escape_string($globalconnect,
$admin_id);
    $query = "SELECT * FROM admin WHERE id = '{$spers}' ";

    $allstaff= mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    if ($atrain= mysqli_fetch_assoc($allstaff)) {
        return $atrain;
    } else {return null;
    }
}
}

```

```

function id_user($user_id) {
    global $globalconnect;
    $sus      =      mysqli_real_escape_string($globalconnect,
$user_id);
    $query = "SELECT * FROM user WHERE id = '{$sus}' LIMIT
1";

    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    if ($fitus = mysqli_fetch_assoc($fit)) {
        return $fitus;
    } else {
        return null;
    }
}

function poisk_trainer($fitname) {
    global $globalconnect;
    $query = "SELECT * FROM admin WHERE username =
'{$fitname}'";
    $allstaff = mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    return $allstaff;
}

function adname($fitname) {
    global $globalconnect;
    $query = "SELECT * FROM admin WHERE username =
'{$fitname}'";
    $allstaff = mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    return $allstaff;
}

function adminusername($fitname, $fitty) {
    global $globalconnect;
    $sus      =      mysqli_real_escape_string($globalconnect,
$fitname);
    $stype    =      mysqli_real_escape_string($globalconnect,
$fitty);
    $query = "SELECT * FROM admin WHERE username = '{$sus}'
AND type = '{$stype}' LIMIT 1";
    $allstaff = mysqli_query($globalconnect, $query);
    confirm_query($allstaff);
    if ($atrain = mysqli_fetch_assoc($allstaff)) {
        return $atrain;
    } else {
        return null;
    }
}

function poiskuser($fitname) {
    global $globalconnect;

```



```

        $sus      =      mysqli_real_escape_string($globalconnect,
$fitname);
        $query = "SELECT * FROM user WHERE username = '{$sus}'
LIMIT 1";
        $fit = mysqli_query($globalconnect, $query);
        confirm_query($fit);
        if ($fitus = mysqli_fetch_assoc($fit)) {
            return $fitus;
        } else {
            return null;
        }
    }

    function id_workout($idworkout) {
        global $globalconnect;
        $sworkout = mysqli_real_escape_string($globalconnect,
$idworkout);
        $query = "SELECT * FROM s_workout WHERE id =
{$sworkout} LIMIT 1";
        $allworkout= mysqli_query($globalconnect, $query);
        confirm_query($allworkout);
        if ($workout = mysqli_fetch_assoc($allworkout)) {
            return $workout;
        } else {
            return null;
        }
    }

    function idgraphic($calendar) {
        global $globalconnect;
        $scalendar = mysqli_real_escape_string($globalconnect,
$scalendar);
        $query = "SELECT * FROM a_presence WHERE id =
{$scalendar} LIMIT 1";
        $allday = mysqli_query($globalconnect, $query);
        confirm_query($allday);
        if ($calendar = mysqli_fetch_assoc($allday)) {
            return $calendar;} else { return null;
        }
    }

    function programuser($program_id) {
        global $globalconnect;
        $sprog=      mysqli_real_escape_string($globalconnect,
$program_id);
        $query = "SELECT * FROM p_workout WHERE user_id =
{$sprog} LIMIT 1";
        $allprog = mysqli_query($globalconnect, $query);
        confirm_query($allprog);
        if ($wprog=      mysqli_fetch_assoc($allprog)) {return
$wprog;
        } else { return null;
    }

```

```

    }
}

function wprogus($user_id) {
    global $globalconnect;
    $query = "SELECT * FROM p_workout WHERE user_id =
{$user_id}";
    $fit = mysqli_query($globalconnect, $query);
    confirm_query($fit);
    return $fit;
}

function selectwprog($public = false) { //поточна програма
тренивання
    global $user_id;
    global $nowwp;

    if (isset($_GET["user"])) {
        $user_id = id_user($_GET["user"]);
        $nowwp = null;
    } elseif (isset($_GET["program"])) {
        $user_id = null;
        $nowwp = wprogus($_GET["program"], $public);
    } else {
        $user_id = null;
        $nowwp = null;
    }
}

function selectquest($public = false) { //поточне питання
    global $nowclient;
    global $nowquest;

    if (isset($_GET["user"])) {
        $nowclient= id_user($_GET["user"]);
        $nowquest = null;
    } elseif (isset($_GET["question"])) {
        $nowclient= null;
        $nowquest      =      id_question($_GET["question"],
$public);
    } else {
        $nowclient= null;
        $nowquest = null;
    }
}

function selectworkout($public = false) { //поточне
тренивання
    global $nowclient;
    global $nowworkout;

    if (isset($_GET["user"])) {

```

```

        $nowclient = id_user($_GET["user"]);
        $nowworkout = null;
    } elseif (isset($_GET["session"])) {
        $nowclient = null;
        $nowworkout = id_workout($_GET["session"],
$public);
    } else {
        $nowclient = null;
        $nowworkout = null;
    }
}

function code($password) {
    $hash_format = "$2y$10$";
    $salt_length = 22;
    $salt = generate_salt($salt_length);
    $format_and_salt = $hash_format . $salt;
    $hash = crypt($password, $format_and_salt);
    return $hash;
}

function generate_salt($length) {
    $unique_random_string = md5(uniqid(mt_rand(), true));
    $base64_string = base64_encode($unique_random_string);
    $modified_base64_string = str_replace('+', '.',
$base64_string);
    $salt = substr($modified_base64_string, 0, $length);
    return $salt;
}

function pass_test($password, $nowh) { //перевірка паролю
    $hash = crypt($password, $nowh); // існуючий хеш містить
формат і сіль при запуску
    if ($hash === $nowh) {
        return true;} else {
        return false;
    }
}

function loginsys($fitname, $fitty, $password) { //спроба
входу користувача.перевірка імені користувача типу та пароль в
базі даних
    $atrain= adminusername($fitname, $fitty);
    if ($atrain) {
        // після знайденого адміністратора виконується
перевірка паролю
        if (pass_test($password,$atrain["password"])) {
            return $atrain;// якщо пароль відповідає
імені користувача
        } else {
            return false;// якщо пароль не знайдено
        }
    }
}

```

```

    } else {
        // якщо не знайдено адміна
        return false;
    }
}

function entry_try($fitname, $password) { //спроба входу
персоналу.перевірка паролю та імені користувача в бд
    $fitus = poiskuser($fitname);
    if ($fitus) {
        if (pass_test($password, $fitus["password"])) {
            return $fitus;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

function logged_in() {
    return isset($_SESSION['admin_id']);
}

function login_confirmation() {
    if (!logged_in()) {
        header("location: error.php");
    }
}

function signed_in() {
    return isset($_SESSION['id']);
}

function confirmation() {
    if (!signed_in()) {
        header("location:effor.php");
        exit;
    }
}

function confirm_trainer() {
    if ($_SESSION["type"] !== "Trainer") {
        header("location: error.php");
        exit;
    }
}

function admin_type() {
    if ($_SESSION["type"] !== "Admin") {
        header("location: error.php");
    }
}

```

```

        exit;
    }
}

Index.php
<?php
    error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php include("header.php"); ?>
<?php include("menu.php"); ?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>SPORTFAMILY</title>
        <meta charset="utf-8" />
        <link          rel="stylesheet"          type="text/css"
href="winmodal.css" />

        <style>

        .trainers:after {
            content: "";
            display: table;
            clear: both;
        }

        .trainers > div {
            top: 15px;
            width: 20%;
            float: left;
            background: rgba(0,0,0,0.3);
            margin-left:5%;
            text-align: center;
            position: relative;
            padding-bottom: 31.7%;
            cursor: pointer;
            -webkit-transition:all 0.35s linear;
            transition:all 0.35s linear;
        }

        .trainers > div:first-child {
            margin-left:5px;
        }

        /*Тень снизу у блока при наведении*/
        .trainers > div:hover {
            -webkit-box-shadow: 0 35px 35px -35px #000000;
            -moz-box-shadow: 0 35px 35px -35px #000000;
            box-shadow: 0 35px 35px -35px #000000;
        }
        .bullet-item span{

```

```

    line-height: 25px;
    font-size: 12.5em;
    color:#fff;
    position: absolute;
    top:70px;
    left:0;
    display: block;
    width: 100%;
    -webkit-transition:all 0.35s ease;
    transition:all 0.35s ease;
}

.trainers > div:hover span.iconBullet{
    line-height: 1.35em;
    font-size: 10.625em;
    top:-2.7%;
}

.titleBullet, .hideText {
    font-family:"BebasNeue", arial, verdana, sans-serif;
    position: absolute;
    left:0;
    bottom:0;
    text-align: center;
    width: 100%;
    margin: 0;
    color: #fff;
    -webkit-transition:all 0.35s ease;
    transition:all 0.35s ease;
}

.titleBullet {
    line-height: 2.5em;
    font-size: 30px;
    background:#3B3B3B;
    z-index: 10;
}

.hideText {
    line-height: 25px;
    font-size: 1.25em;
    padding: 1em 0;
    background:#3B3B3B;
    font-weight: 300;
    z-index: 8;
    height: 50px;
    -webkit-transition:all 0.35s linear 0.2s;
    transition:all 0.35s linear 0.2s;
}

/*Анимация при наведении видимого текстового блока*/
.trainers > div:hover p.titleBullet{

```

```

        bottom:80px;
    }

    /*Анимация при наведении скрытого текстового блока*/
    .trainers > div:hover p.hideText{
        background:#454545;
    }</style>

</head>
<body>

    <div id="base">
        <p><a style="right: 20%;" href="login_admin.php">
Admin</a></p>
        <center><h2 style="color: #fff; font-
family:BebasNeue,arial,verdana, sans-serif; ">НАША
КОМАНДА</h2></center>
        <div class="trainers">
        <div class="bullet-item">
            <span></span>
            <p class="titleBullet">Инна</p>
            <p class="hideText">Персональный <br> тренер</p>

        </div>
        <div class="bullet-item">
            <span></span>
            <p class="titleBullet">Никита</p>
            <p class="hideText">Персональный тренер</p>
        </div>

        <div class="bullet-item">
            <span></span>
            <p class="titleBullet">Валерия</p>
            <p class="hideText">Персональный тренер</p>
        </div>
            <div class="bullet-item">
                <span></span>
                <p class="titleBullet">Дмитрий</p>
                <p class="hideText">Персональный тренер</p>
            </div>
        </div> </div>

    </div>
</div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.m
in.js"></script>

```

```

    <script src="glavn.js"></script>
</body>
</html>
<?php include("bottom.php"); ?>

Login_admin.php
<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php
require_once("test.php");
require_once("fitclubdb.php");
require_once("func.php");
$fitname= "";

    if (isset($_POST['submit'])) {

        $needfulfield = array("username", "password");
        confirm_availability($needfulfield);

        if (empty($erfatal)) {

            $fitname = $_POST["username"];
            $password = $_POST["password"];

            $alltype = adname($fitname);
            $tstaff = mysqli_fetch_assoc($alltype);
            $found_admin =loginsys($fitname, $tstaff["type"],
$password);

            if ($found_admin) {

                $_SESSION["admin_id"] = $found_admin["id"];
                $_SESSION["username"] = $found_admin["username"];
                $_SESSION["type"] = $found_admin["type"];

                if ($tstaff["type"] === "Admin") {
                    header("location:admin.php");
                    exit;
                } elseif ($tstaff["type"] === "Trainer") {
                    header("location:trainer.php");
                    exit;
                }
            } else {

                $_SESSION["inform"] = "Не правильно введено
имя пользователя или пароль";
            }
        } else {

```



```

    }
    ?>
<?php $headline = "public"; ?>
<?php include("header.php"); ?>
<?php include ("menu.php"); ?>
<body style="background-image: URL(images/dd.jpg)">
    <div style="text-align: center;">
    <div style="box-sizing: border-box;
    display: inline-block;
    width: auto;
    max-width: 480px;
    background-color: #FFFFFF;
    border: 2px solid #b74544;
    margin: 50px auto auto;">

        <div style="background: #b74544; padding: 15px;">
            <span style="font-family: verdana,arial; font-
            size: 1.00em; font-weight:bold;">ПЕРСОНАЛ</span></div>
            <div style="padding: 15px">

                <style type="text/css" scoped>
                    td { text-align:left; padding:
                    5px;font-family: verdana,arial; color: #000000; font-size:
                    1.00em; font-weight: normal;}
                    input { border: 1px solid #b74544;
                    display: inline-block; font-size: 1.00em; padding: 5px; width:
                    100%; }
                    input[type="button"],
                    input[type="reset"], input[type="submit"] { height: auto; width:
                    auto; cursor: pointer; box-shadow: 0px 0px 5px #F90; float:
                    right; margin-top: 10px; }
                    table.center { margin-left:auto;
                    margin-right:auto; }
                </style>

                <form method="POST"
                action="login_admin.php" name="login_admin" onsubmit="return
                validateSignIn();">
                    <table class='center'>
                        <tr>
                            <td>Имя пользователя:</td>
                            <td><input type="text"
                            name="username" value="<?php echo htmlentities($fitname);
                            ?>"></td>
                        </tr>
                        <tr>
                            <td>Пароль:</td>
                            <td><input type="password"
                            name="password" value="" ></td>
                        </tr>
                    </table>
                </form>
            </div>
        </div>
    </div>
</body>

```

```

                <tr>
                    <td>&nbsp;</td>
                    <td><input type="submit"
name="submit" value="Войти"></td>
                </tr>
            </table>
        </form>
        <?php echo inform(); ?> <?php echo
errorsset($erfatal); ?>
    </div>
</div>

```

```

    </div>
</div>
</body>

```

```

Login_client.php
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php
    $fitname = "";

    if (isset($_POST['submit'])) {
        $needfulfield = array("username", "password");
        confirm_availability($needfulfield);

        if (empty($erfatal)) {
            $fitname = $_POST["username"];
            $password = $_POST["password"];
            $detectedus = entry_try($fitname, $password);
            if ($detectedus) {

                $_SESSION["id"] = $detectedus["id"];
                $_SESSION["username"] =
$detectedus["username"];
                header("location: user_content.php");
                exit;
            } else {
                $_SESSION["inform"] = "Неверно введен пароль
или имя пользователя!";
            }
        }
    } else {
    }
?>
<?php include("header.php"); ?>
<?php include ("menu.php"); ?>
<body style="background-image: URL(images/dd.jpg)">
    <div style="text-align: center;">
    <div style="box-sizing: border-box;

```

```

display: inline-block;
width: auto;
max-width: 480px;
background-color: #FFFFFF;
border: 2px solid #b74544;
margin: 50px auto auto;"/>
<div style="background: #b74544; padding: 15px;">
    <span style="font-family: verdana,arial; font-
size: 1.00em; font-weight:bold;">ЛИЧНЫЙ КАБИНЕТ</span></div>
    <div style="padding: 15px">

        <style type="text/css" scoped>
            td { text-align:left; height: 50px;
padding: 7px; font-family: Verdana; color: #000000; font-size:
15px;; font-weight: normal;}
            input{ border: 1px solid #b74544;
display: inline-block; font-size: 1.00em; padding: 5px; width:
100%; }

            button[type="button"],
button[type="reset"], button[type="submit"] { font-family:
"Roboto", sans-serif; text-transform: uppercase; outline: 0;
background: #4CAF50;
width: 100%;
border: 0;
padding: 15px;
color: #FFFFFF;
font-size: 14px;
-webkit-transition: all 0.3 ease;
transition: all 0.3 ease;
cursor: pointer;}

            table.login { margin-left:auto;
margin-right:auto; }
        </style>

        <form method="POST"
action="login_client.php" name="login_client" onsubmit="return
validateSignIn();">
            <table class="login">
                <tr>
                    <td><label>Имя
пользователя:</label></td>
                    <td><input type="text"
name="username" value="<?php echo htmlentities($fitname);
?>"></td>
                </tr>
                <tr>
                    <td><label>Пароль:</label></td>
                    <td><input type="password"
name="password" value="" ></td>
            </table>

```

```

                </tr>
                <tr>
                    <td><button      class="submit"
name="submit" value="Войти">Войти</button></td>
                </tr>
            </table>
        </form>
        <?php echo inform();?>
        <?php echo errorsset($erfatal);?>
    </div>
</div>
</div>
</body>
Mclients_admin.php
<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php login_confirmation(); ?>
<?php admin_type(); ?>

<?php $fit = all_clients(); ?>
<?php $headline = "admin"; ?>

<?php include("header.php"); ?>
<style>
    tr:nth-child(odd) { background-color: #cccccc; }
    tr:nth-child(even) { background-color: #999999; }
    td { border-right: 1px solid white; }
</style>
<div id="base">
    <div id="pbody">
        <a href="admin.php">Главное меню</a>
        <br /><br />
        <u><b><a                      class="link"
href="add_client.php">Добавить      нового      клиента</a></b></u><br
/><br />
        <?php echo inform(); ?>
        <h2>Управление клиентами</h2>
        <table style="text-align: center;">
            <tr>
                <th
                    style="width:          75px;">Имя
пользователя</th>
                <th style="width: 80px;">Имя</th>
                <th style="width: 100px;">Фамилия</th>
                <th style="width: 80px;">Дата рождения</th>
                <th style="width: 100px;">Адрес</th>
                <th style="width: 70px;">Город</th>

```

```

        <th style="width: 100px;">Email</th>
        <th style="width: 70px;">Контактный
номер</th>
        <th style="width: 200px;">Цель</th>
        <th colspan="3" style="width:
300px;">Действия</th>
    </tr>
    <?php while ($fitus = mysqli_fetch_assoc($fit))
    { ?>
        <tr>
            <th style="height: 20px;"><?php echo
htmlentities($fitus["username"]); ?></th>
            <td><?php echo
htmlentities($fitus["f_name"]); ?></td>
            <td><?php echo
htmlentities($fitus["l_name"]); ?></td>
            <td><?php echo
htmlentities($fitus["birthday"]); ?></td>
            <td><?php echo
htmlentities($fitus["address"]); ?></td>
            <td><?php echo
htmlentities($fitus["city"]); ?></td>
            <td><?php echo
htmlentities($fitus["email"]); ?></td>
            <td><?php echo
htmlentities($fitus["mobile"]); ?></td>
            <td><?php if (empty($fitus["goal"])) {
?>
                <center style="color:#f00;">Нет
цели!</center>
                <?php } else {
                echo htmlentities($fitus["goal"]);
                }
                ?>
            </td>
            <td style="width: 100px;"><a
class="manage" href="add_new_trainer.php?id=<?php echo
urlencode($fitus["id"]); ?>">Назначить тренера</a></td>
            <td style="width: 60px;" ><a
class="manage" title="Редактировать пользователя"
href="edit_fitnesuser.php?id=<?php echo urlencode($fitus["id"]);
?>">
                
                </a>
            </td>
            <td style="width: 60px;"> <a
class="manage" title="Удалить пользователя"
href="delete_client.php?id=<?php echo urlencode($fitus["id"]);
?>" onclick="return confirm(' Вы действительно желаете удалить
клиента? '); ">

```

```

                                
                                </a>
                                <b></b>
                                </td>
                                </tr>
                                <?php } ?>
                                </table>
                                </div>
</div>
</div>
</div>
<?php include("bottom.php"); ?>

Mclients_trainer.php
<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php login_confirmation(); ?>
<?php confirm_trainer();?>

<?php
    $admius    =    $_SESSION["username"];          $fit    =
ustrain($admius);?>
<?php $headline = "admin"; ?>
<?php include("header.php"); ?>
<style>
    tr:nth-child(odd) { background-color:#f9f9f9;
border-bottom:#a5a5a5 2px solid; }
    tr:nth-child(even) {
        background-color: #f9f9f9;
        border-bottom: #a5a5a5 2px solid;
    }
    body{font-family:Arial; color:#5d5d5d;}
    .container{display:block;position:relative;    margin:0
auto; width:100%;}
    .table_overflow{
        float:left;
        width:100%;
        font-size:12px;
    }
    .table_overflow thead{
        background-color:#f9f9f9;
        border-bottom:#a5a5a5 2px solid;
    }

```



```

        </tr>
    </thead>
    <tbody>

        <?php while ($fitus =
mysqli_fetch_assoc($fit)) { ?>
            <tr>
                <td style="width: 120px;" <?php if
(empty($fitus["image"])) { ?>
                    <?php } else { ?>
                        
                    <?php } ?></td>
                    <td style="width: 90px;" ><?php
echo htmlentities($fitus["f_name"]); ?></td>
                    <td style="width: 100px;" ><?php
echo htmlentities($fitus["l_name"]); ?></td>
                    <td style="width: 100px;"><?php
echo htmlentities($fitus["birthday"]); ?></td>
                    <td style="width: 70px;"><?php
                    if
(htmlentities($fitus["email"]) == NULL) {
                        echo
'<center>_____</center>';
                    } else {
                        echo
htmlentities($fitus["email"]); }
                    ?></td>
                    <td style="width: 100px;"><?php
                    if
(htmlentities($fitus["mobile"]) == NULL) {
                        echo
'<center>_____</center>';
                    } else {
                        echo
htmlentities($fitus["mobile"]);
                    }
                    ?></td>
                    <td style="width: 100px;"><?php
                    if
(htmlentities($fitus["goal"]) == NULL) {
                        echo
'<center
style="color: red;"> Пока нет цели!</center>';
                    } else {
                        echo
htmlentities($fitus["goal"]);
                    }
                    ?>
                </td>
            <?php
                $allprog = wprogus($fitus["id"]);

```



```

mysql_fetch_assoc($allprog);
    $wprog =
    ?>
    <td>
        <?php
            if
                (!($wprog["user_id"])) { ?>
                    <a
                        class="link"
                    href="newprogram.php?id=<?php echo urlencode($fitus["id"]);
                    ?>">Добавить программу</a>
                    <?php } else { ?>
                        <a
                            class="link"
                        href="editprogram.php?id=<?php echo urlencode($fitus["id"]);
                        ?>">Редактировать программу</a>
                    </td>
                <?php } ?>
                <td><a
                    class="link"
                href="exercise.php?id=<?php echo urlencode($fitus["id"]);
                ?>">История тренировок</a></td>
                <td><a
                    class="link"
                href="new_health_profile.php?id=<?php echo
                urlencode($fitus["id"]); ?>">Здоровье</a></td>
                <td><a
                    class="link"
                href="answ_for_client.php?id=<?php echo urlencode($fitus["id"]);
                ?>">Вопросы клиента</a></td>
            </tr>
        <?php } ?>
    </tbody>
</table>

    <?php } ?>
</div>
</div>
</div>
</div>
<?php include("bottom.php"); ?>

Add_client.php
<?php
    error_reporting(-1);
    ini_set('display_errors', 'on');
    ?>
    <?php require_once "test.php"; ?>
    <?php require_once "fitclubdb.php"; ?>
    <?php require_once "func.php"; ?>
    <?php login_confirmation(); ?>
    <?php admin_type(); ?>

<?php
    if (isset($_POST['submit'])) {
        $needfulfield = array("username", "password");
        confirm_availability($needfulfield);
    }
}

```

```

        if (empty($erfatal)) {
            //приведение полученных значений к единому
стандарту sql
            $fitus=
mysqli_real_escape_string($globalconnect,$_POST["username"]);
            $password = code($_POST["password"]);
            $f_name =
mysqli_real_escape_string($globalconnect,$_POST["f_name"]);
            $l_name =
mysqli_real_escape_string($globalconnect,$_POST["l_name"]);
            $gender =
mysqli_real_escape_string($globalconnect,$_POST["gender"]);
            $birthday =
mysqli_real_escape_string($globalconnect,$_POST["birthday"]);
            $address =
mysqli_real_escape_string($globalconnect,$_POST["address"]);
            $city =
mysqli_real_escape_string($globalconnect,$_POST["city"]);
            $email =
mysqli_real_escape_string($globalconnect,$_POST["email"]);
            $mobile =
mysqli_real_escape_string($globalconnect,$_POST["mobile"]);

            $query = "INSERT INTO user(username, password,
f_name, l_name, sex, birthday, address, city, email, mobile )
VALUES ( '{$username}', '{$password}', '{$f_name}', '{$l_name}',
'{$gender}', '{$birthday}', '{$address}', '{$city}', '{$email}',
'{$mobile}');"
            $resultclient = mysqli_query($globalconnect,
$query);

            if ($resultclient) { // Если результат успешен
                $_SESSION["inform"] = "Клиент добавлен";
                header("location:mclients.php");
                exit;
            } else { // Иначе
                $_SESSION["inform"] = "Не удалось добавить
клиента. Возможно Вы указали имя пользователя, которое уже
зарегистрировано. ";
            }
        }
    }
    ?>
    <?php $headline = "admin"; ?>
    <?php include "header.php"; ?>
    <div id="base">

        <div id="pbody">
            <a href="manage_clients.php">Клиенты</a><br />
            <?php echo inform(); ?> <?php echo
errorsset($erfatal); ?>
            <div id="box">

```

```

        <div id="formbox">
            <h2 style="text-align: center;">Создать учетную
запись клиента</h2>

            <div id="add" style="margin-top: 10px">
                <form action="add_client.php" method="post">
                    <div>
                        <label for="username">Имя
пользователя:</label>
                        <input type="text" name="username" required
value="">
                    </div>
                    <div>
                        <label for="password">Пароль:</label>
                        <input type="text" name="password" required
value="">
                    </div>
                    <div>
                        <label >Имя:</label>
                        <input type="text" name="f_name" required
value="">
                    </div>
                    <div>
                        <label >Фамилия:</label>
                        <input type="text" name="l_name" required
value="">
                    </div>
                    <div>
                        <label >Пол:</label>
                        <select style="width: 300px;" name="gender"
style="width: 11em;">
                            <option
value="мужчина">мужчина</option>
                            <option
value="женщина">женщина</option>
                        </select>
                    </div>
                    <div>
                        <label >Дата рождения:</label>
                        <input type="date" name="birthday" value="">
                    </div>
                    <div>
                        <label>Адрес:</label>
                        <input type="text" name="address" value="">
                    </div>
                    <div>
                        <label>Город:</label>
                        <input type="text" name="city" value="">
                    </div>
                    <div>
                        <label>Email</label>
                        <input type="email" name="email" value="">
                    </div>
                </form>
            </div>
        </div>

```

```

        </div>
        <div>
            <label>Контактный номер:</label>
            <input type="tel" name="mobile" value="">
        </div>
        <div>
            <button class="submit" name="submit"
value="Назад" type="button"
onclick="history.back()">Отмена</button>
            <button class="submit" name="submit"
value="Создать" >Создать</button>
        </div>

    </form>
    <br /> </div>
</div>
    </div>
</div>
<?php include("bottom.php"); ?>

```

Add\_new\_trainer.php

```

<?php require_once "test.php"; ?>
<?php require_once "fitclubdb.php"; // з'єднання з БД ?>
<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once"func.php"; ?>
<?php login_confirmation();?>
<?php admin_type(); //пошук типу?>

<?php
    $fitus = id_user($_GET["id"]);
    $fittrainer = all_trainers();
    if (!$fitus) {
header("location:mclients_admin.php");
exit;
    }
?>
<?php
    if (isset($_POST['submit'])) {

        if (empty($erfatal)) {

            $id = $fitus["id"];
            $atrain
mysql_real_escape_string($globalconnect,$_POST["admin"]);
            $query = "UPDATE user SET ";
            if ($atrain === 'NULL') {

```

```

        $query .= "admin = {$atrain} ";
    } else {
        $query .= "admin = '{$atrain}' ";
    }
    $query .= "WHERE id = {$id} ";
    $query .= "LIMIT 1";
    $resultuser = mysqli_query($globalconnect,
$query);

    if ($resultuser &&
mysqli_affected_rows($globalconnect) == 1) {

        $_SESSION["inform"] = "Запись обновлена";
        header("location:mclients_admin.php");
        exit;
    } else {

        $_SESSION["inform"] = "Не удалось обновить
запись о тренере"; }}} else { }
?>
<?php $headline = "admin"; ?>
<?php include("header.php"); ?>
<style>

<div id="base">
    <div id="pbody">
        <a href="manage_clients.php">Клиенты</a><br /><br
/>
        <?php echo inform(); ?> <?php echo
errorsset($rfatal); ?>
        <h2 style="text-align: center">Назначить тренера
для:<b style="color: #fff;"> <?php echo
htmlentities($fitus["username"]); ?></b></h2>
        <div id="box">
            <div id="formbox">
                <form style="text-align:
center;"action="add_new_trainer.php?id=<?php echo
urlencode($fitus["id"]); ?>" method="post">

                    <table style="margin-top: 25px; width: 450px;
background:inherit; padding: 10px;">
                        <tr>
                            <td style="text-align:
center;">Тренер:</td>
                            <td>
                                <select class="button" name="admin">

                                    <?php while ($strainer =
mysqli_fetch_assoc($fittrainer)) { ?>

```

```

                                <option       style="width:
350px" value="<?php echo htmlentities($trainer["username"]); ?>"
                                <?php
                                if ($fitus["admin"] ==
$trainer["username"]) { echo 'sps';
                                } ?> >
                                <?php       echo
htmlentities($trainer["username"]); ?>
                                </option>
                                <?php } ?>
                                <?php
                                if (empty($fitus["admin"])) {
sps>Нет тренера</option>';
                                echo ' <option value="NULL"
                                } else {
                                echo ' <option
value="NULL">Нет тренера</option>';
                                }
                                ?>
                                </select>
                                </td>
                                </tr>
                                <tr>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                                </tr>
                                <tr>
                                <td><button class="submit" name="submit"
value="Назад" type="button"
onclick="history.back()">Отмена</button></td>
                                <td><button
                                class="submit"
name="submit">Назначить</button></td>
                                </tr>
                                </table>
                                </form>
                                </div>
                                </div>
                                </div>
                                <?php include("bottom.php"); ?>

```

Add\_staff.php

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>

```

```

<?php require_once("func.php"); ?>

<?php login_confirmation(); ?>
<?php admin_type(); ?>

<?php
    if (isset($_POST['submit'])) {

        $needfulfield = array("username", "password","type");
        confirm_availability($needfulfield);

        if (empty($erfatal)) {

            $fitname= $_POST["username"];
            $fitname
mysqlqli_real_escape_string($globalconnect,$fitname);
            $password = code($_POST["password"]);
            $fitty = $_POST["type"];
            $fitty
mysqlqli_real_escape_string($globalconnect,$fitty);

            $query = "INSERT INTO admin ( username, password,
type) VALUES ('$fitname', '$password', '$fitty')";
            $resultstaff      =      mysqlqli_query($globalconnect,
$query);

            if ($resultstaff) {

                $_SESSION["inform"] = "Пользователь создан.";
                header("location:mclients_admin.php");
                exit;
            } else {

                $_SESSION["inform"] = "Не удалось создать
пользователя";
            }
        }
    } else {

    }

?>
<?php $headline= "admin"; ?>
<?php include("header.php"); ?>
<div id="base">
    <style>
        td, tr { padding: 5px; } input { padding: 2px; }
    #formbox{

background-color:#f5f5f5;
padding:15px;
width: 450px;

```

```

-moz-border-radius:12px;
-khtml-border-radius: 12px;
-webkit-border-radius: 12px;
border-radius:12px;
}
        td{
white-space:nowrap;
padding-top: 15px;
}

        input {
width: 300px;
font-size: 13px;
padding: 6px 0 4px 10px;

border: 1px solid #cecece;
background: #F6F6f6;
border-radius: 8px;
}

        select{

font-size: 13px;
padding: 6px 0 4px 10px;

border: 1px solid #cecece;
background: #F6F6f6;
border-radius: 8px;
        }

box{

width:500px;
margin:40px auto;
}

</style>

<div id="pbody">
    <a href="mclients_admins.php">Персонал</a><br />
<?php echo inform();?>
    <?php echo errorsset($erfatal); ?>
    <h2 style="text-align: center;">Создать
пользователя</h2>

    <div id="formbox">
<form action="add_staff.php" method="post">
    <table>
        <tr>
            <td>Имя:</td>
            <td><input type="text" name="username"
required value=""></td>
        </tr>

```



```

        <tr>
            <td>Пароль:</td>
            <td><input
name="password" required value=""></td>
            type="password"
        </tr>
        <tr>
            <td>Тип персонала:</td>
            <td><select name="type">
                <option
value="Admin">Администратор</option>
                <option
value="Trainer">Тренер</option>
            </select>
            </td>
        <tr>
            <td><button class="submit" name="submit"
value="Назад" type="button"
onclick="history.back()">Отмена</button></td>
            <td><button class="submit" name="submit"
value="Создать" >Создать</button></td>
        </tr>
    </table>
</form>
<br />
</div>
</div>
</div>
<?php include("bottom.php"); ?>

```

### **Admin.php**

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); // ?>
<?php require_once("func.php"); // ?>
<?php login_confirmation(); // подтвердить, если
администратор вошел в систему ?>
<?php admin_type(); // подтвердите, если тип является
администратор?>
<?php $headline = "admin"; // показать заголовок плюс АДМИН
после него ?>
<?php include("header.php"); ?>

<div id="base">
<style>
@font-face {
font-family: 'icomoon';

```

```
        src:url('fonts/icomoon.eot');
        src:url('fonts/icomoon.eot?#iefix')      format('embedded-
opentype'),
        url('fonts/icomoon.woff') format('woff'),
        url('fonts/icomoon..ttf') format('truetype'),
        url('fonts/icomoon.svg#icomoon') format('svg');
font-weight: normal;
font-style: normal;
}

.ir-icon {
display: inline-block;
font-size: 15px;
cursor: pointer;
margin: 25px 45px;
width: 190px;
height: 190px;
border-radius: 50%;
text-align: center;
position: relative;
z-index: 1;
color: #fff;
}

.ir-icon:after {
pointer-events: none;
position: absolute;
width: 100%;
height: 100%;
border-radius: 50%;
content: '';
-webkit-box-sizing: content-box;
-moz-box-sizing: content-box;
box-sizing: content-box;
}

.ir-icon:before {
font-family: 'icomoon';
margin-top: 25px;
speak: none;
font-size: 100px;
line-height: 110px;
font-style: normal;
font-weight: normal;
font-variant: normal;
text-transform: none;
display: block;
-webkit-font-smoothing: antialiased;
}

.ir-icon-news:before {
content: "\e942";
}
```

```

.ir-icon-staff:before {
    content: "\e941";
}

.ir-icon-person:before {
    content: "\e940";
}
/* Effect 1 */
.ir-icon-effect-1 .ir-icon {
    background: rgba(255,255,255,0.1);
    -webkit-transition: background 0.2s, color 0.2s;
    -moz-transition: background 0.2s, color 0.2s;
    transition: background 0.2s, color 0.2s;
}
.ir-icon-effect-1 .ir-icon:after {
    top: -7px;
    left: -7px;
    padding: 7px;
    box-shadow: 0 0 0 4px #fff;
    -webkit-transition: -webkit-transform 0.2s, opacity
0.2s;
    -webkit-transform: scale(.8);
    -moz-transition: -moz-transform 0.2s, opacity 0.2s;
    -moz-transform: scale(.8);
    -ms-transform: scale(.8);
    transition: transform 0.2s, opacity 0.2s;
    transform: scale(.8);
    opacity: 0;
}

.ir-icon-effect-1a .ir-icon:hover {
    background: rgba(255,255,255,1);
    color: #e5434e;
}

.ir-icon-effect-1a .ir-icon:hover:after {
    -webkit-transform: scale(1);
    -moz-transform: scale(1);
    -ms-transform: scale(1);
    transform: scale(1);
    opacity: 1;
}
</style>

<div id="pbody" class="animated fadeIn">
    <a href="logout.php"><b>Выход</b></a><br /><br />
    <center><h2>Меню администратора</h2></center>
    <div style="padding-left: 170px;"><h3>Добро пожаловать,
<b
        style="color:#fff;"><?php
        echo
ucfirst(htmlentities($_SESSION["username"]));
        ?>!</b></h3><br
/></div>

```

```

    <div style="text-align: center;"class="ir-icon-wrap ir-
icon-effect-1 ir-icon-effect-1a">
        <a href="mstaff.php" class="ir-icon ir-icon-
staff"><p>Управление персоналом</p></a>
        <a href="mclients_admin.php" class="ir-icon ir-icon-
person"><p>Управление клиентами</p></a>
    </div>
</div>
</div>
<?php include("bottom.php");?>

```

### Bottom.php

```

div id="bottom">Сайт для дипломного проекта студентки гр.
КН-15бд Хамулы И.В.<?php echo date("Y"); ?></div>

```

```

</body>
</html>
<?php
    if (isset($globalconnect)) {
        mysqli_close($globalconnect);
    }
?>

```

### Delete\_admin.php

```

<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php login_confirmation(); ?>
<?php admin_type(); ?>

<?php $vses = $_SESSION["admin_id"];?>
<?php
    $atrain = id_admin($_GET["id"]);
    if (!$atrain) {
        header("location:mstaff.php");
        exit;
    }
    $id = $atrain["id"];
    if ($vses === $id) {
        $_SESSION["inform"] = "Вы не можете удалить себя";
        header("location:mstaff.php");
        exit;
    } else {
        $query = "delte FROM admin WHERE id = {$id} LIMIT 1";
        $resultstaff = mysqli_query($globalconnect, $query);
        if ($resultstaff
mysql_affected_rows($globalconnect) == 1) {
            $_SESSION["inform"] = "Запись удалена";
            header("location:mstaff.php");

```

```

        exit;
    } else {

        $_SESSION["inform"] = "Запись не удалена";
        header("location:mstaff.php");
        exit;
    }
}
?>

```

#### Delete\_client.php

```

<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php login_confirmation(); ?>
<?php admin_type(); ?>
<?php
    $fitus = id_user($_GET["id"]);
    if (!$fitus) {// идентификатор участника отсутствует или
недействителен или клиент не найден в базе данных
        header("location:mclients_admin.php");
        exit;}
    $id = $fitus["id"];
    $query = "DELETE FROM user WHERE id = {$id} LIMIT 1";
    $resultuser = mysqli_query($globalconnect, $query);
    if ($resultuser && mysqli_affected_rows($globalconnect) ==
1) {
        $_SESSION["inform"] = "Клиент удален";
        header("location:manage_client.php");
        exit;
    } else {
        $_SESSION["inform"] = "Не удалось удалить запись";
        header("location:manage_client.php");
        exit;
    }
}
?>

```

#### Editfituser.php

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php login_confirmation(); ?>
<?php admin_type(); ?>

<?php

```

```

$fitus = id_user($_GET["id"]);
if (!$fitus) {
    header("location:mclients_trainer.php");
    exit;
}
?>
<?php
    if (isset($_POST['submit'])) {

        //проверка заполненности поля имя пользователя
        $needfulfield = array("username");
        confirm_availability($needfulfield);
        if (empty($erfatal)) {
            // если нет ошибок, то выполняется обновление
записи в бд
            $fitus=
mysqli_real_escape_string($globalconnect,$_POST["username"]);
            $password = code($_POST["password"]);
            $f_name =
mysqli_real_escape_string($globalconnect,$_POST["f_name"]);
            $l_name =
mysqli_real_escape_string($globalconnect,$_POST["l_name"]);
            $gender =
mysqli_real_escape_string($globalconnect,$_POST["gender"]);
            $birthday =
mysqli_real_escape_string($globalconnect,$_POST["birthday"]);
            $address =
mysqli_real_escape_string($globalconnect,$_POST["address"]);
            $city =
mysqli_real_escape_string($globalconnect,$_POST["city"]);
            $email =
mysqli_real_escape_string($globalconnect,$_POST["email"]);
            $mobile =
mysqli_real_escape_string($globalconnect,$_POST["mobile"]);

            $cold = (int) $_POST["freeze"];
            $query = "UPDATE user SET username = '{$fitname}',
f_name = '{$f_name}', l_name = '{$l_name}', gender=
 '{$gender}',birthday = '{$birthday}', address = '{$address}',
city = '{$city}', email = '{$email}', mobile = '{$mobile}',
freeze = '{$cold}' WHERE id = {$id} LIMIT 1";
            $resultuser = mysqli_query($globalconnect,
$query);

            if ($resultuser == 1) &&
mysqli_affected_rows($globalconnect) == 1) {
                $_SESSION["inform"] = "Запись обновлена";
                header("location:mclients_trainer.php");
                exit;
            } else {

```

```

        $_SESSION["inform"] = "Данные о клиенте не
удалось обновить";
    }
}
} else {

}
?>
<?php $headline = "admin"; ?>
<?php include("header.php"); ?>
<style type="text/css">td {min-width: 110px; } </style>
<div id="base">
    <style type="text/css">td {min-width: 110px; } </style>
<div id="base">

    <a href="mclients_admin.php"></a><br /><br />
<div id="pbody">

    <style>

        .boxx {
margin:0 auto;
width:700px;
}

input, textarea {
width:200px;
height:15px;
background:#efefef;
border:1px solid #dedede;
padding:2px;
margin-top:2px;
font-size:0.9em;
color:#3a3a3a;
}

        label {
display:block;
margin-top:25px;
letter-spacing:2px;
}

        textarea {
            height: 213px;
        }

input, textarea {
width:400px;
height:20px;
background:#efefef;
border:1px solid #dedede;
padding:10px;

```

```

margin-top:3px;
font-size:15px;
color:#3a3a3a;
-moz-border-radius:5px;
-webkit-border-radius:5px;
border-radius:5px;
}
    select {
    width:420px;
    height:43px;
padding: 10px;
background: #efefef;
border: none;
border-radius: 3px;
padding: 1em 2em 1em 1em;
font-size: 1em;
}
    #radio{
        width: 10px;
    }
</style>

<?php echo inform(); ?>
    <?php echo errorsset($erfatal); ?>
    <div class="boxx">
        <a href="mclients_admin.php"></a><br /><br />
        <h2>Редактировать:<b style="color:#fff "><?php echo
ucfirst(htmlentities($fitus["username"])); ?></b></h2>
        <form action="edit_member.php?id=<?php echo
urlencode($fitus["id"]); ?>" method="post">

            <?php if (empty($fitus["image"])) { ?>
                <span class="clientimg" style="padding: 50px
5px;">Нет изображения</span>
            <?php } else { ?>
                
            <?php } ?>
            <table style="border-bottom-color: #efefef" >
                <tr>
                    <td style="min-width: 100px;">Имя
пользователя:</td>
                    <td><input type="text" name="username"
value="<?php echo htmlentities($fitus["username"]); ?>"></td>
                </tr>
                <tr>
                    <td style="min-width: 100px;">Имя:</td>
                    <td><input type="text" name="f_name"
value="<?php echo htmlentities($fitus["f_name"]); ?>"></td>

```



```

        </tr>
        <tr>
            <td style="min-width:
100px;">Фамилия:</td>
            <td><input type="text" name="l_name"
value="<?php echo htmlentities($fitus["l_name"]); ?>"></td>
        </tr>
        <tr>
            <td style="min-width: 100px;">Пол:</td>
            <td>
                <select name="gender" >
                    <?php
                    echo
htmlentities($fitus["gender"]);
                    if ($fitus["gender"] ===
"Female") {
                        ?>
                        <option value="Female"
selected>:женский</option>
                        <option
value="Male">мужской</option>
                    <?php } else { ?>
                        <option value="Male"
selected>мужской</option>
                        <option
value="Female">женский</option>
                    <?php }
                    ?>
                </select>
            </td>
        </tr>
        <tr>
            <td style="min-width: 100px;">Дата
рождения:</td>
            <td><input type="date" name="birthday"
value="<?php echo $fitus["birthday"]; ?>"></td>
        </tr>
        <tr>
            <td style="min-width:
100px;">Адрес:</td>
            <td><input type="text" name="address"
value="<?php echo htmlentities($fitus["address"]); ?>"></td>
        </tr>
        <tr>
            <td style="min-width:
100px;">Город:</td>
            <td><input type="text" name="city"
value="<?php echo htmlentities($fitus["city"]); ?>"></td>
        </tr>
        <tr>

```

```

                                <td style="min-width:
100px;">Email:</td>
                                <td><input type="email" name="email"
value="<?php echo htmlentities($fitus["email"]); ?>"></td>
                                </tr>
                                <tr>
                                <td style="min-width: 100px;">Контактный
номер:</td>
                                <td><input type="tel" name="mobile"
value="<?php echo htmlentities($fitus["mobile"]); ?>"></td>
                                </tr>
                                <tr>
                                <td>Тренер:</td>
                                <td><?php if (empty($fitus["admin"])) {
?>
                                <span style="color: #f00;">Нет
персонального тренера</span>
                                <?php
                                } else {
                                echo
htmlentities($fitus["admin"]);
                                }
                                ?>
                                </td>
                                </tr>
                                <tr>
                                <td>Статус:</td>
                                <td><label>Активировать</label>
                                <input type="radio" name="freeze"
value="1" onclick="return confirm('Вы действительно хотите
активировать аккаунт клиента?');"><?php
                                if ($fitus["freeze"] == 1) {
                                echo "checked";
                                }
                                ?> />
                                <label>Заблокировать</label>
                                <input type="radio" name="freeze"
value="0" onclick="return confirm('Вы действительно хотите
заблокировать пользователя?');"><?php
                                if ($fitus["freeze"] == 0) {
                                echo "checked";
                                }
                                ?> />
                                </td>
                                </tr>
                                <tr>
                                <td>&nbsp;</td>
                                <td>&nbsp;</td>
                                </tr>
                                <tr>

```

```

                                <td><button class="submit" name="submit"
value="Назад"                                type="button"
onclick="history.back()">Отмена</button></td>
                                <td><button                                class="submit"
name="submit">Редактировать</button></td>
                                </tr>
                                </table>
                                </form>
                                </div>
                                </div>
</div>
<?php include("bottom.php"); ?>

```

### **Edit\_profile.php**

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php confirmation(); ?>

<?php
$user_id = $_SESSION["id"];
$fit = id_user($user_id);
$fitus = mysqli_fetch_assoc($fit);
$trainer_set = poisk_trainer($fitus["admin"]);
$trainer = mysqli_fetch_assoc($trainer_set);
?>

<?php $headline= "public"; ?>

<div id="base">
    <div id="pbody">
        <a href="user_content.php"> Назад</a>
    </div>
    <div id="page"><?php echo inform(); ?>
        <h2>Ваш            профиль            <a            class="link"
href="edit_user.php?id=<?php            echo            urlencode($fitus["id"]);
?>">Редактировать</a></h2>

        <table class="user_table">

            <?php if (empty($fitus["image"])) { ?>
                <span class="profile" style="padding: 60px
5px;">Нет изображения</span>
            <?php } else { ?>
                

```

```

<?php } ?>
    <tr>
        <td class="profile">Имя пользователя:</td>
        <td><?php echo
htmlentities($fitus["username"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Имя:</td>
        <td><?php echo
htmlentities($fitus["f_name"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Фамилия:</td>
        <td><?php echo
htmlentities($fitus["l_name"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Пол:</td>
        <td><?php echo
htmlentities($fitus["gender"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Дата рождения:</td>
        <td class="add"><?php echo
htmlentities($fitus["birthday"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Адрес:</td>
        <td><?php echo
htmlentities($fitus["address"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Город:</td>
        <td><?php echo htmlentities($fitus["city"]);
?></td>
    </tr>
    <tr>
        <td class="profile">Email:</td>
        <td><?php echo
htmlentities($fitus["email"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Контактный номер:</td>
        <td><?php echo
htmlentities($fitus["mobile"]); ?></td>
    </tr>
    <tr>
        <td class="profile">Ваша цель:</td>
        <td>
            <?php if ($fitus["goal"] == "") { ?>
                <span style="color:#cc0000; font-
weight: bold;">У вас сейчас нет целей! Чтобы добавить цель

```

```

просто      <a          href="edit_user.php?id=<?php          echo
urlencode($fitus["id"]); ?>">нажмите здесь</a>.</span>
              <?php } ?>
              <?php echo htmlentities($fitus["goal"]);
?>
              </td>
            </tr>
            <tr>
              <td class="profile">Ваш тренер:</td>
              <td><?php if (empty($fitus["admin"])) { ?>
                  <span style="color: #cc0000; font-
weight: bold;">У Вас нет персонального тренера! Обратитесь за
помощью к администратору</span>
                  <?php } else { ?>
                      <span style = "color: blue;"><?php
echo htmlentities($fitus["admin"]); ?></span>
                      <br />
                      <a
href="view_availability.php?id=<?php          echo
urlencode($trainer["id"]); ?>">Занятость тренера</a>
                      <?php }
                  ?>
              </td>
            </tr>
            <tr>
              <td class="profile">&nbsp;</td>
              <td>
                  <button class="buttons" type="button"
value="edit" style="width: 150px; height: 30px;"
onclick="window.location.href = 'edit_user.php?id=<?php echo
urlencode($fitus["id"]); ?>
';">Редактировать</button>
              </td>
            </tr>
        </table>
    </div>
</div>

```

### **Edit\_staff.php**

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); // ?>
<?php login_confirmation();?>
<?php admin_type(); ?>

<?php

```

```

$atrain = id_admin($_GET["id"]);

if (!$atrain) {
    header("location:mclients_admin.php");
    exit;
}
?>
<?php
    if (isset($_POST['submit'])) {

        $needfulfield = array("username", "password");
        confirm_availability($needfulfield);

        if (empty($erfatal)) {

            $sid = $fitperson["id"];
            $fitname = $_POST["username"];
            $fitname =
mysqli_real_escape_string($globalconnect,$fitname);
            $password = code($_POST["password"]);
            $fitty = $_POST["type"];
            $fitty =
mysqli_real_escape_string($globalconnect,$fitty);

            $query = "UPDATE admin SET username ='$fitname',
password ='$password', type = '$fitty' WHERE id = $id ";
            $resultstaff =
mysqli_query($globalconnect,
$query);

            if ($resultstaff &&
mysqli_affected_rows($globalconnect) == 1) {
                $_SESSION["inform"] = "Запись обновлена.";
                header("location:mclients_admin.php");
                exit;
            } else {

                $_SESSION["inform"] = "Запись не обновлена.";
            }
        }
    } else {

    }

?>
<?php $headline= "admin";?>
<?php include("header.php"); ?>
<div id="base">
    <div id="pbody">
        <a href="mstaff.php">Персонал</a><br />

```

```

                <center><h2>Редактировать      пользователя:      <b
style="color: #fff;"><?php echo urlencode($atrain["username"]);
?></b></center></h2>
                <?php echo inform(); ?>
                <?php echo errorsset($erfatal); ?>
                <style>
                    #formbox{
background-color:#f5f5f5;

padding:15px;
width: 500px;
-moz-border-radius:12px;
-khtml-border-radius: 12px;
-webkit-border-radius: 12px;
border-radius:12px;
}

                    td{
white-space:nowrap;
padding-top: 15px;
}

                    input {
width: 300px;
font-size: 13px;
padding: 6px 0 4px 10px;

border: 1px solid #cecece;
background: #F6F6f6;
border-radius: 8px;
}

                    select{

font-size: 13px;
padding: 6px 0 4px 10px;

border: 1px solid #cecece;
background: #F6F6f6;
border-radius: 8px;
                    }
                    #box{
width:500px;
margin:40px auto;
}

                </style>
                <div id="box">
                <div id="formbox">
                <form      action="edit_staff.php?id=<?php      echo
urlencode($atrain["id"]); ?>" method="post">
                <table width="100%" border="0" cellpadding="0"
cellpadding="5">
                    <tr>
                        <td><p>Имя пользователя:</p></td>

```

```

                                <td><input type="text" name="username"
value="<?php echo htmlentities($atrain["username"]); ?>"></td>
                                </tr>
                                <tr>
                                <td>Пароль:</td>
                                <td><input type="password"
name="password" value=""></td>
                                </tr>
                                <tr>
                                <td>Тип:</td>
                                <td><select name="type" style="width:
153px;">
                                <?php
                                echo
                                htmlentities($atrain["type"]);
                                if ($atrain["type"] ==
                                "Admin") {
                                ?>
                                <option value="Admin"
                                selected>Администратор</option>
                                <option
                                value="Trainer">Тренер</option>
                                <?php } else { ?>
                                <option value="Trainer"
                                selected>Тренер</option>
                                <option
                                value="Admin">Администратор</option>
                                <?php }
                                ?>
                                </select></td>
                                </tr>
                                <tr>
                                <td><button class="submit" name="submit"
                                value="Назад" type="button"
                                onclick="history.back()">Отмена</button></td>
                                <td><button class="submit"
                                name="submit">Изменить</button></td>
                                </tr>
                                </table>
                                </form>
                                </div>
                                </div></div>
                                </div>
                                <?php include("bottom.php"); ?>

Exercise.php

?php
    error_reporting(-1);
    ini_set('display_errors', 'on');

```





```

                                <button style="float: right;"
type="submit" value="Comment" onclick="window.location.href =
'comment.php?id=?php echo urlencode($w_session["id"]);
?>'>Комментировать</button>
                                </td>
                                <td><center><?php echo
htmlentities($workout["w_date"]); ?></center></td>
                                </tr>
                                <?php } ?>
                                </table>
                                <?php } ?>
                                </div>
</div>
<?php include("bottom.php"); ?>

```

### Mstaff.php

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php login_confirmation(); ?>
<?php admin_type(); ?>

<?php $allstaff = all_staff(); ?>
<?php $headline= "admin"; ?>
<?php include("header.php"); ?>
<style>
    tr:nth-child(odd) { background-color: #cccccc; }
    tr:nth-child(even) { background-color: #999999; }
    td { border-right: 1px solid white; }
</style>
<div id="base">
    <div id="pbody">
        <a href="admin.php">Главное меню</a>
        </br>
        <a href="add_staff.php">Добавить новый персонал</a>
        <link rel="shortcut icon" href="../favicon.ico">
        <div class="wrapper">
            <div class="inner">
                <script type="text/javascript">
function adminDelete() {

                    if (confirm("Вы действительно хотите удалить запись?
")) {
                        return true;
                    } else {
                        return false;
                    }
                }

```

```

    }
</script>
    <?php echo inform(); ?>
    <center><h2>Управление персоналом</h2></center>
    <table style="width: 100%;" border="1">
        <tr>
            <th style="width: 200px;">Имя
пользователя</th>
            <th style="width: 400px;">Тип</th>
            <th style="width: 400px;">График работы</th>
            <th style="width:
110px;" colspan="2">Действия</th>
        </tr>
        <?php while ($atrain =
mysqli_fetch_assoc($allstaff)) { ?>
            <tr>
                <td style="text-align: center;"><?php echo
htmlentities($atrain["username"]); ?></td>
                <td style="text-align: center;"><?php echo
htmlentities($atrain["type"]); ?></td>
                <td style="text-align: center;">
                    <?php if ($atrain["type"] ===
"Trainer") { ?>
                        <a class="link"
href="graphic.php?id=<?php echo urlencode($atrain["id"]);
?>">График работы</a>
                    <?php } else { ?>
                        <hr style="width: 50%; margin-left: auto; margin-right: auto;" />
                    <?php } ?>
                </td>
                <td>
                    <a href="edit_staff.php?id=<?php
echo urlencode($atrain["id"]); ?>">
                        <center></center>
                    </a>
                </td>
                <td>
                    <a href="delete_admin.php?id=<?php
echo urlencode($atrain["id"]); ?>" onclick="return
adminDelete();">
                        <center> </center>
                    </a>
                </td>
            </tr>
        <?php } ?>
    </table>
</div>
</div>
</div>
</div>

```

```

<?php include("bottom.php"); ?>

Question_and_answers.php

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("fitclubdb.php"); ?>
<?php require_once("func.php"); ?>
<?php confirmation(); ?>
<?php
    $user_id = $_SESSION["id"];
    $setq = idquestion($user_id);
    $fit = id_user($user_id);
    ?>
<?php include("header.php"); ?>
    <?php $headline = "public"; ?>
<div id="base">
    <div id="pbody">
        <p><a href="user_content.php">Профиль</a></p>
        <?php echo inform(); ?>
        <h2>
            Вопросы для тренера от Вас, <b style="color:
white"><?php echo
ucfirst(htmlentities($_SESSION["username"]));?></b>
        </h2>
        <?php if (empty($fit["admin"])) { ?>
            <p style="color: #f00;">Вы не можете задавать
вопросы, потому что у вас пока нет личного тренера.</p>
            <p style="color: #f00; font-weight: bold;">Чтобы
получить личного тренера, пожалуйста, поговорите с
администрацией.</p>
            <?php } else { ?>
                <table style="width: 100% border="1"
style="border-collapse: collapse;">
                    <tr>
                        <th style="width: 400px;">Ваши
вопросы</th>
                        <th style="width: 500px;">Ответы</th>
                        <th >Дата</th>
                        <th >Действия</th>
                    </tr>
                    <?php while ($fitus =
mysqli_fetch_assoc($setq)) { ?>
                        <tr>
                            <td style="padding: 15px;"><?php echo
htmlentities($fitus["question"]); ?></td>
                            <td style="padding: 15px;">
                                <?php

```

```

//якщо $fitus порожня, то відобразити текст "В ожидании
ответа ...";
        if (empty($fitus["answer"])) {
            echo "В ожидании ответа ...";
            echo "<center>";
            echo "</center>";
        } else {
            echo
htmlentities($fitus["answer"]);
        }
        ?>
        </td>
        <td><center><?php
htmlentities($fitus["date"]); ?></center></td>
        <td><center><a
class="manage"
title="Удалить"
href="delquest.php?id=?php
urlencode($fitus["id"]);
?>"></a>
        </center></td>
        </tr>
        <?php } ?>
    </table> <br />
    <button
class="submit"
onclick="window.location.href = 'add_question.php';">Добавить
новый вопрос</button>
    <?php } ?>
</div>
</div>
</div>
<?php include("bottom.php"); ?>

```

### Test.php

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');

session_start();

function inform() {
    if (isset($_SESSION["inform"])) {
        $output = "<div class=\"inform\">";
        $output .= htmlentities($_SESSION["inform"]);
        $output .= "</div>";

        // clear message after use
        $_SESSION["inform"] = null;

        return $output;
    }
}

```

```

}

function erfatal() {
    if (isset($_SESSION["erfatal"])) {
        $erfatal = $_SESSION["erfatal"];

        $_SESSION["erfatal"] = null;

        return $erfatal;    } }
?>

```

Trainer.php

```

<?php
error_reporting(-1);
ini_set('display_errors', 'on');
?>
<?php require_once("test.php"); ?>
<?php require_once("func.php"); ?>
<?php include("header.php"); ?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title></title>
        <link rel="shortcut icon" href="../favicon.ico">
        <style>

@font-face {
    font-family: 'icomoon';
    src:url('fonts/icomoon.eot');
    src:url('fonts/icomoon.eot?#iefix') format('embedded-
opentype'),
        url('fonts/icomoon.woff') format('woff'),
        url('fonts/icomoon..ttf') format('truetype'),
        url('fonts/icomoon.svg#icomoon') format('svg');
    font-weight: normal;
    font-style: normal;
}

.irmenu {

    font-size: 15px;
    cursor: pointer;
    margin: 25px 45px;
    width: 190px;
    display: inline-block;
    height: 190px;
    border-radius: 50%;
    text-align: center;
    position: relative;

```

```

        z-index: 1;
        color: #fff;
    }

    .irmenu:after {
        pointer-events: none;
        position: absolute;
        width: 100%;
        height: 100%;
        content: '';
        -webkit-box-sizing: content-box;
        -moz-box-sizing: content-box;
        box-sizing: content-box;
    }

    .irmenu:before {
        font-family: 'icomoon';
        margin-top: 25px;
        speak: none;
        font-size: 100px;
        line-height: 110px;
        font-style: normal;
        font-weight: normal;
        font-variant: normal;
        text-transform: none;
        display: block;
        -webkit-font-smoothing: antialiased;
    }

    .irmenu-client:before {
        content: "\e900";
    }

    .irmenu-screen:before {
        content: "\e901";
    }

    .irmenu-effect-1 .irmenu{
        background: rgba (white, 1);
        -webkit-transition: background 0.2s, color 0.2s;
        -moz-transition: background 0.2s, color 0.2s;
        transition: background 0.2s, color 0.2s;
    }

    .irmenu-effect-1 .irmenu:after {
        top: -7px;
        left: -7px;
        padding: 7px;
        box-shadow: 0 0 0 4px #fff;
        -webkit-transition: -webkit-transform 0.2s, opacity
0.2s;
        -webkit-transform: scale(.8);
        -moz-transition: -moz-transform 0.2s, opacity 0.2s;
    }

```

```

        -moz-transform: scale(.8);
        -ms-transform: scale(.8);
        transition: transform 0.2s, opacity 0.2s;
        transform: scale(.8);
        opacity: 0;
    }
    .irmenu-effect-1a .irmenu:hover {
        background: rgba (white, 1);
        color: #e5434e;
    }
    .irmenu-effect-1a .irmenu:hover:after {
        -webkit-transform: scale(1);
        -moz-transform: scale(1);
        -ms-transform: scale(1);
        transform: scale(1);
        opacity: 1;
    }
    </style>
</head>

<body>
    <div id="base">
    <div id="pbody" class="effects">
        <a href="logout.php"><b>Выйти</b></a>
        <h2>Меню тренера</h2>
        <h2>Добро пожаловать в тренерскую, <b style="color:
#fff;"><?php echo ucfirst(htmlentities($_SESSION["username"]));
?>!</b></h2><br />
        <div style="text-align: center;" class="irmenu-wrap irmenu-
effect-1 irmenu-effect-1a">
            <a href="mclients_trainer.php" class="irmenu irmenu-
client"><p>Мои клиенты</p></a>
            <a href="week.php" class="irmenu irmenu-
screen"><p>График работы</p></a>
        </div>

    </div>
    </div>
        <?php include("bottom.php"); ?>
    </body>
</html>

```



## Додаток Б

### Комп'ютерна презентація



Рисунок Б.1–Титульний слайд

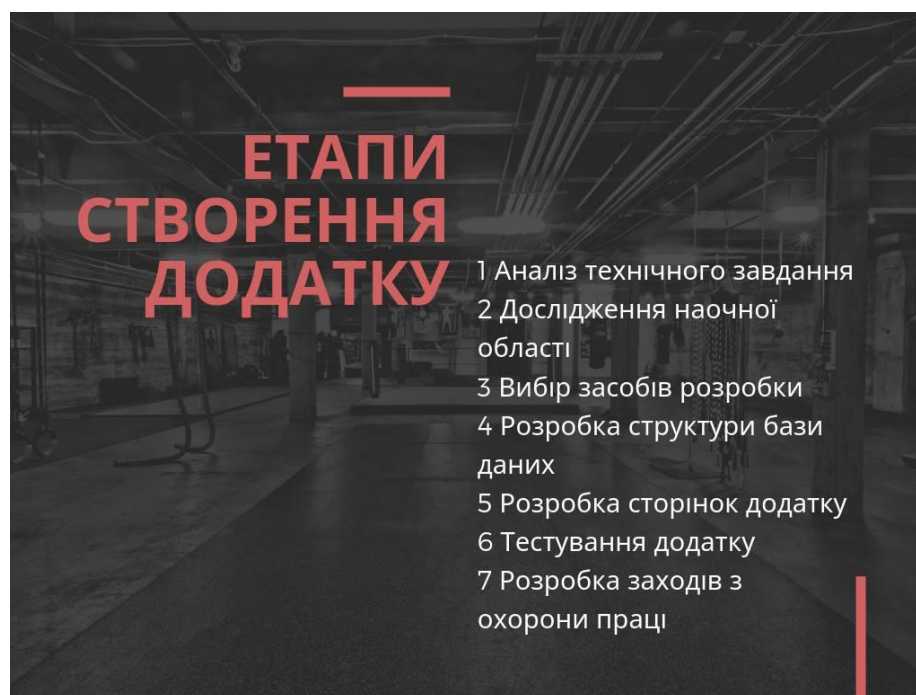


Рисунок Б.2–Етапи створення додатку



Рисунок Б.3 –Схема бази даних



Рисунок Б.4 –Схема сайту

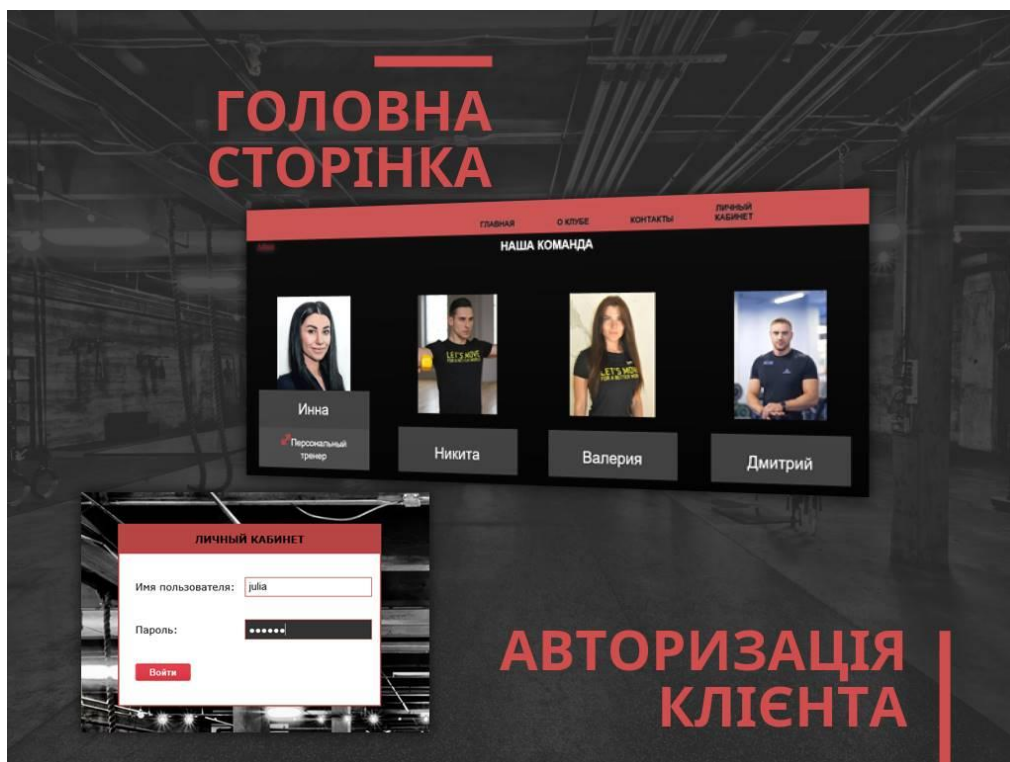


Рисунок Б.5 –Сторінки «Головна сторінка» та «Авторизація»

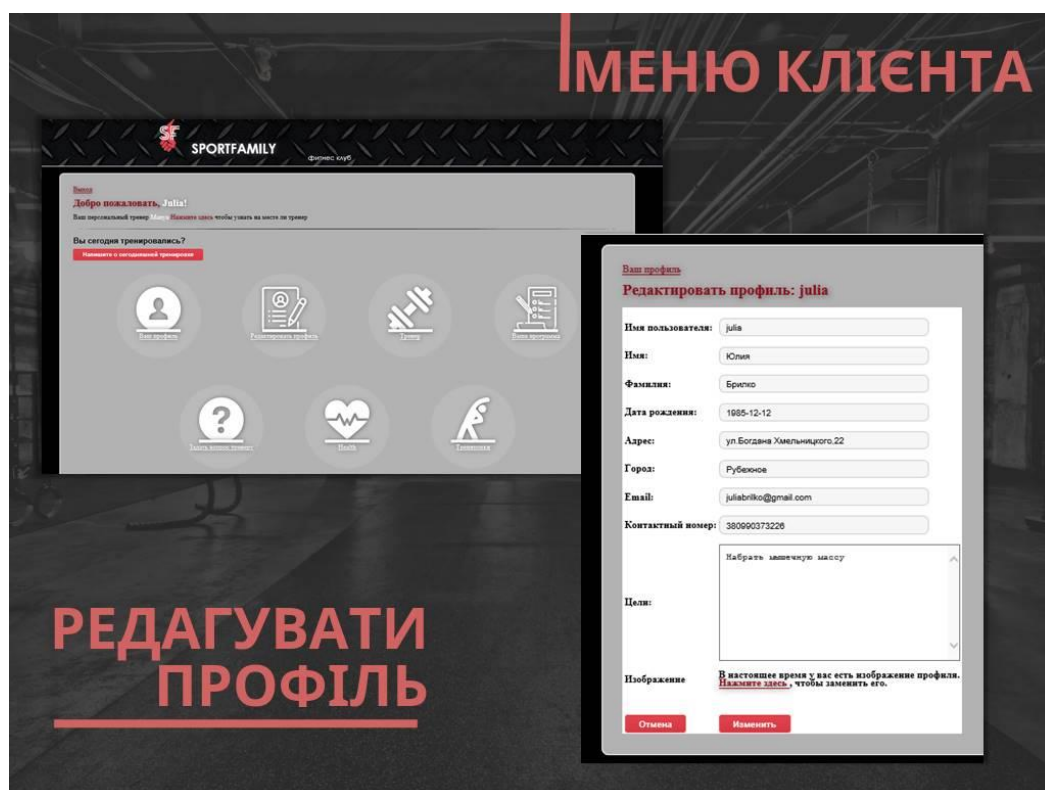


Рисунок Б.6 –Сторінки «Меню клієнта» та «Редагування профілю»

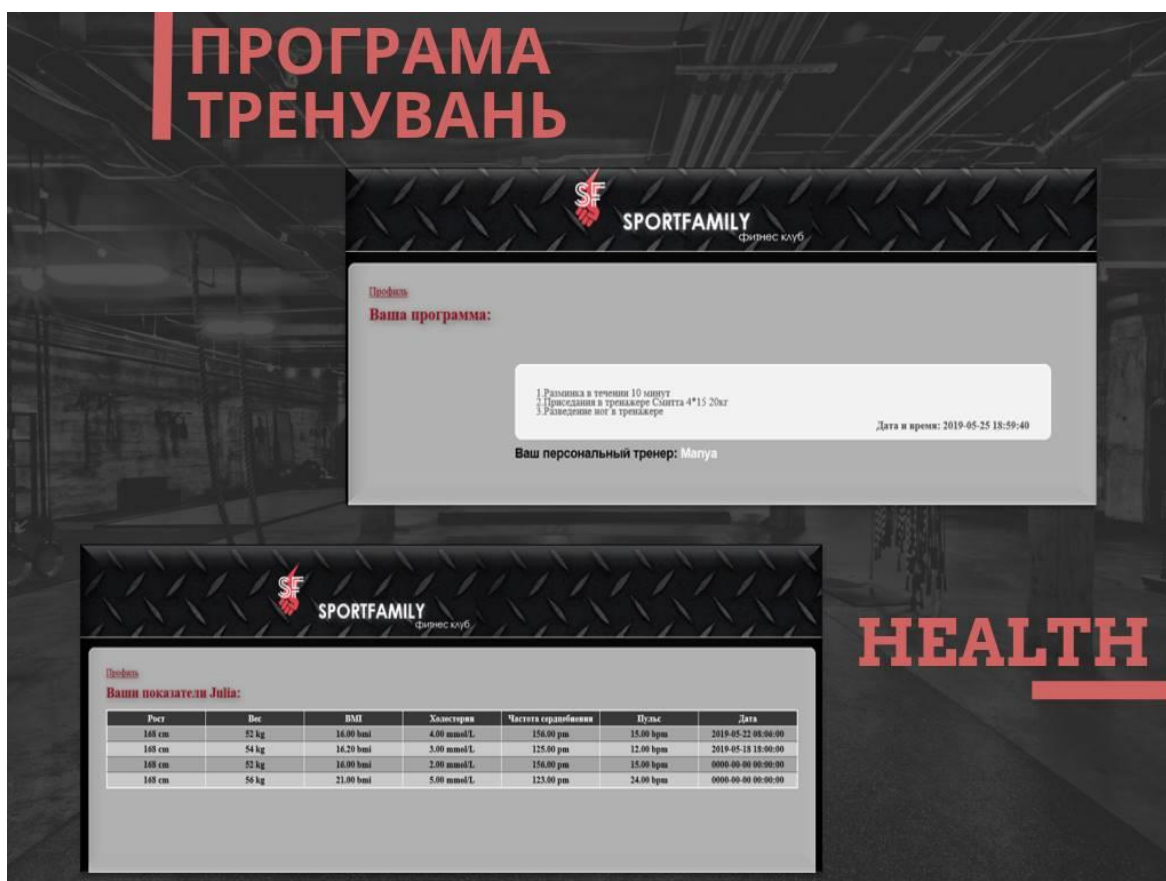


Рисунок Б.7 –Сторінки «Програма тренувань» та «HEALTH»

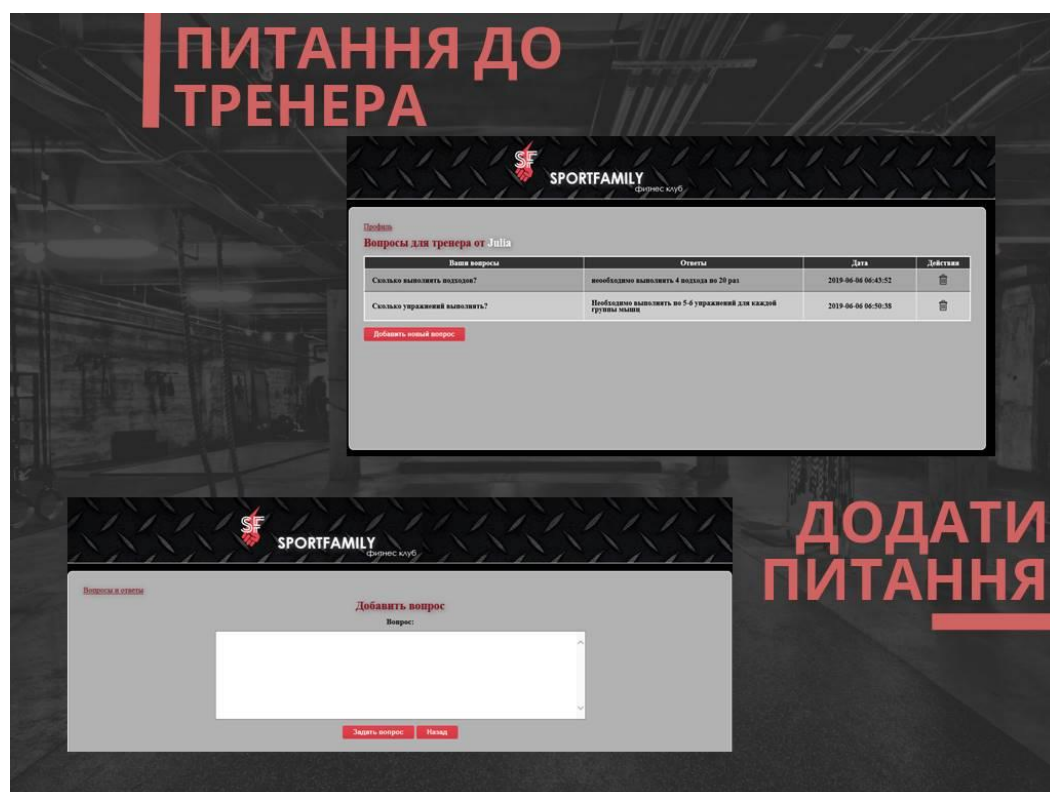


Рисунок Б.8 –Сторінки «Питання до тренера» та «Додати питання»

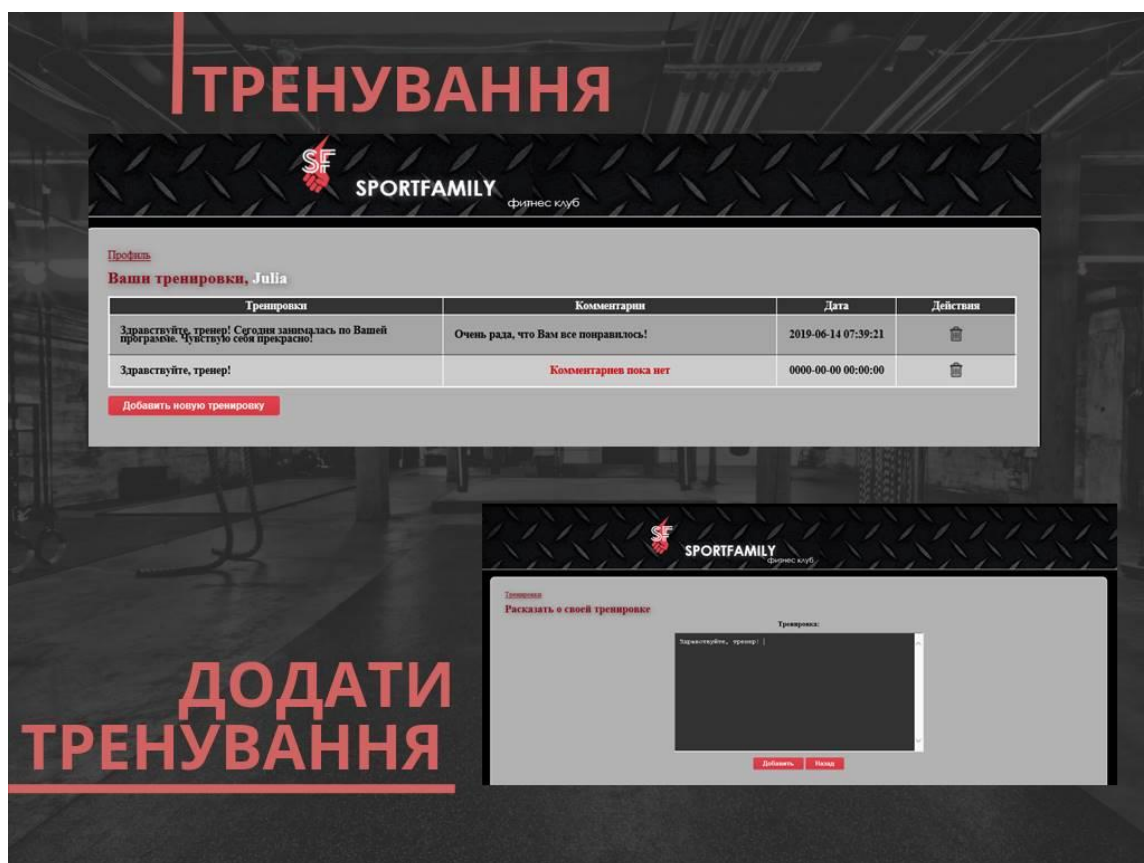


Рисунок Б.9 –Сторінки «Тренування» та «Додати тренування»

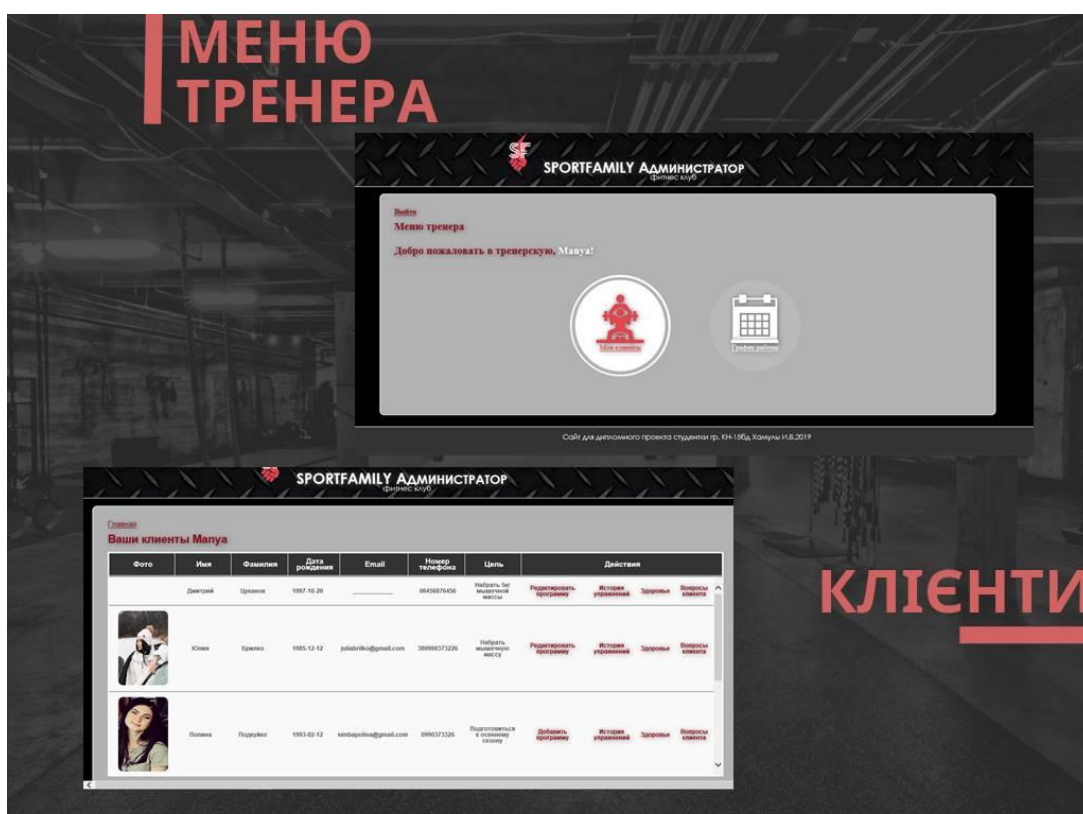


Рисунок Б.10 –Сторінки «Меню тренера» та «Клієнти»

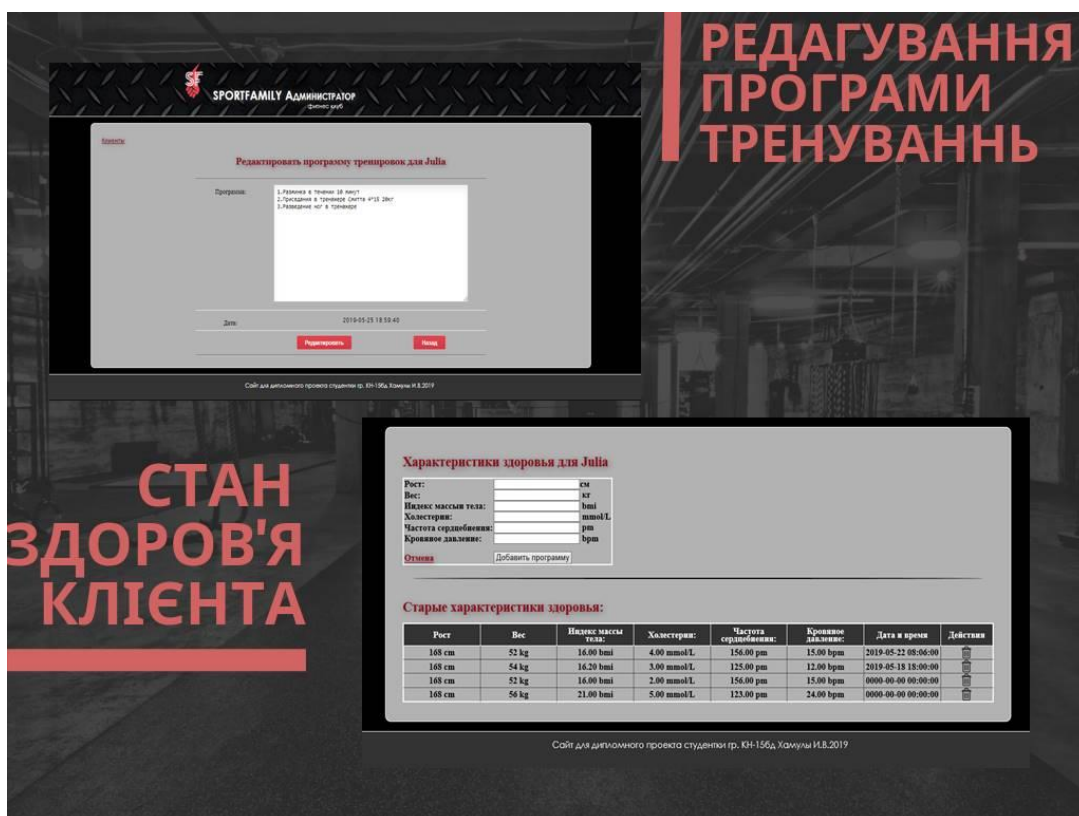


Рисунок Б.11 –Сторінки «Стан здоров'я клієнта» та «Редагування програми тренувань»

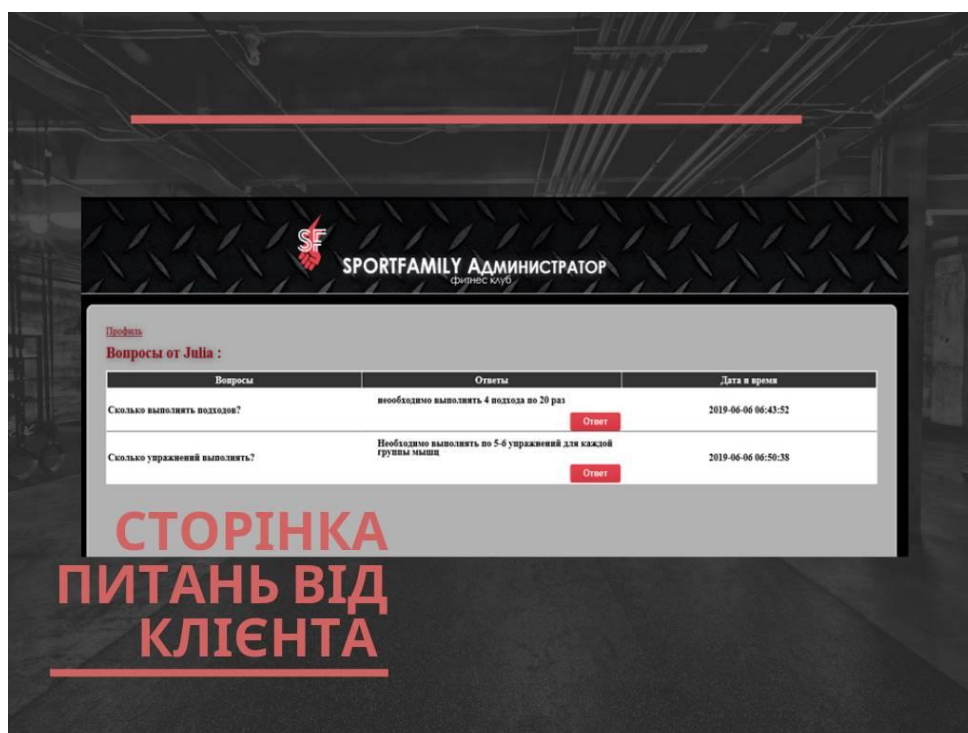


Рисунок Б.12 –Сторінка «Питань клієнта»

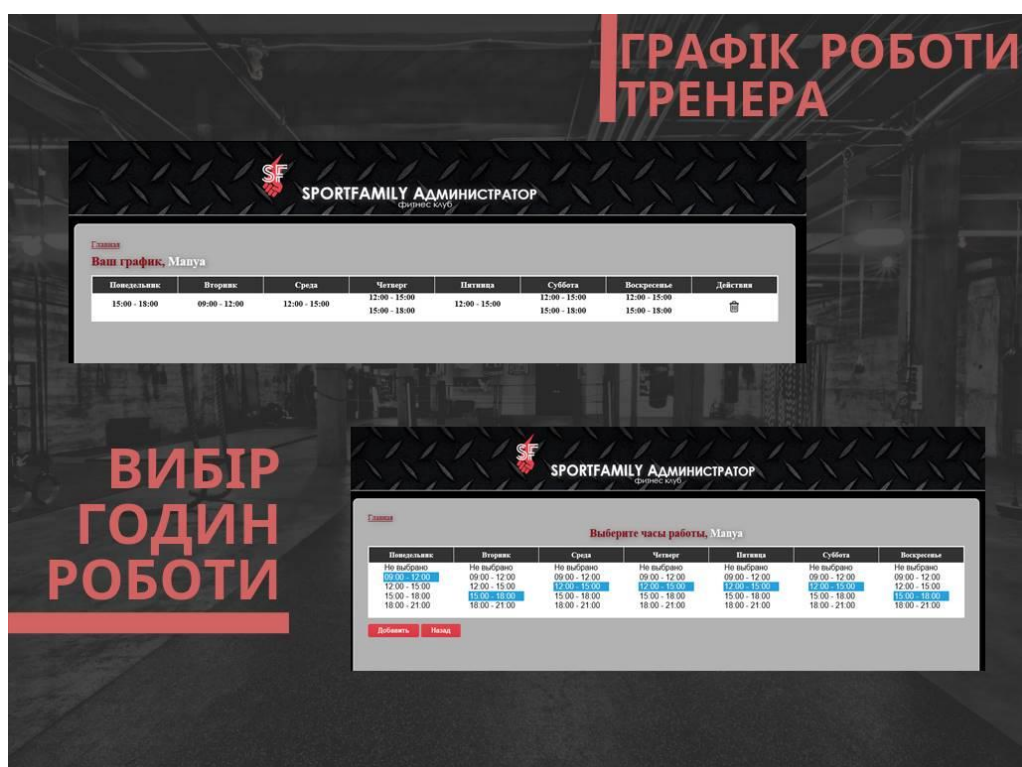


Рисунок Б.13 –Сторінки «Вибір годин роботи» та «Графік роботи тренера»

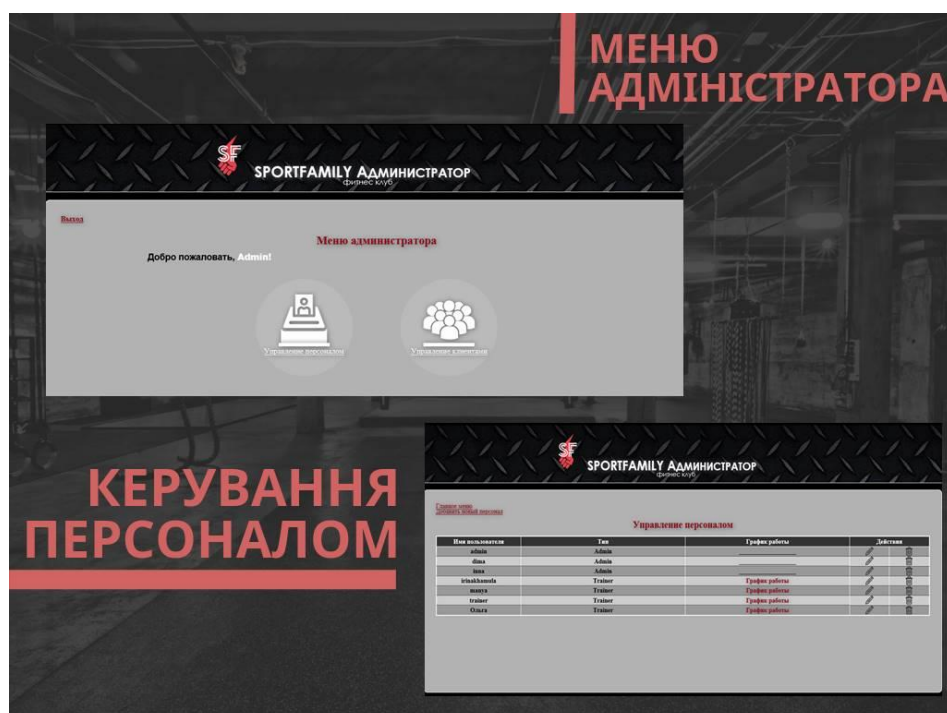


Рисунок Б.14 –Сторінки «Керування персоналом» та «Меню адміністратора»

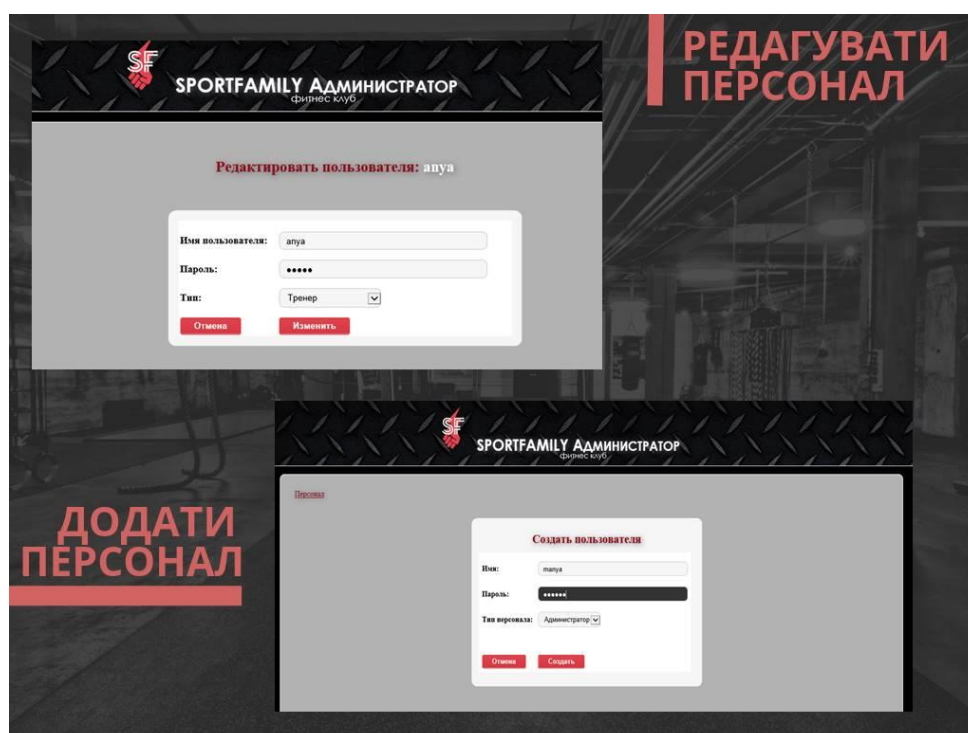


Рисунок Б.15–Сторінки «Додати персонал» та «Редагувати персонал»

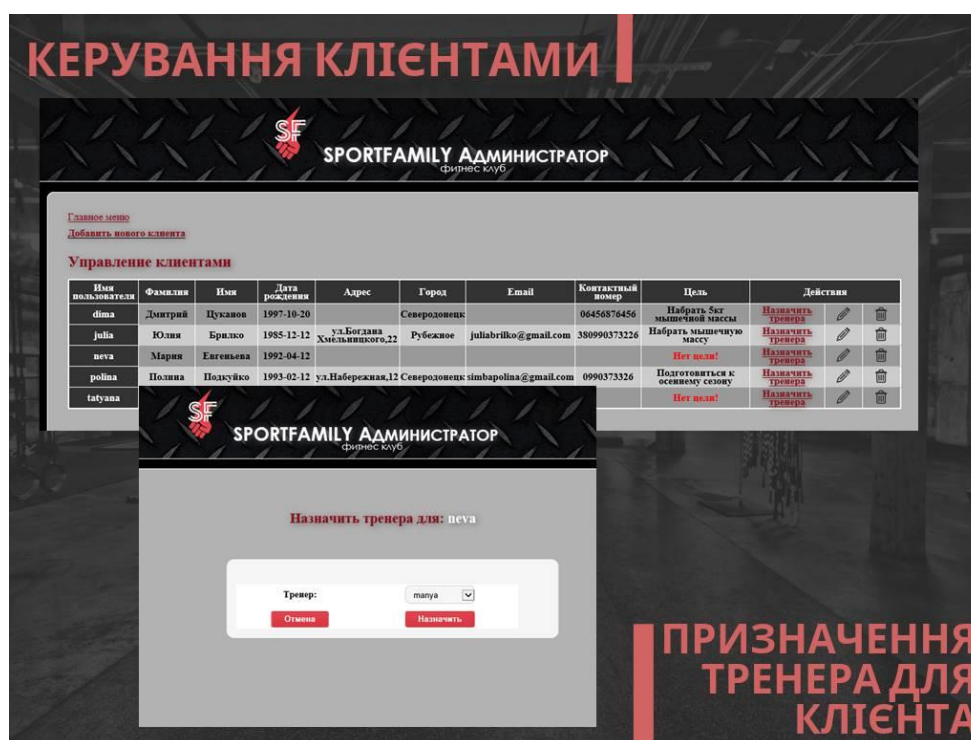


Рисунок Б.16 – Сторінки «Керування клієнтами» та «Призначення тренера для клієнта»



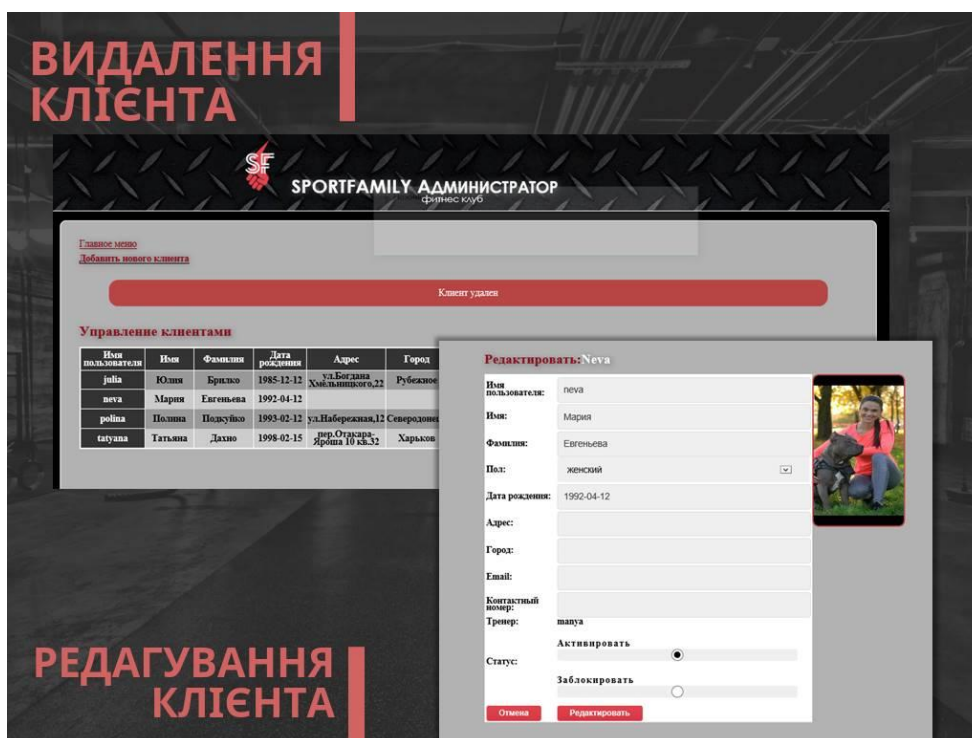


Рисунок Б.17 –Сторінки «Видалення клієнта» та «Редагування клієнта»

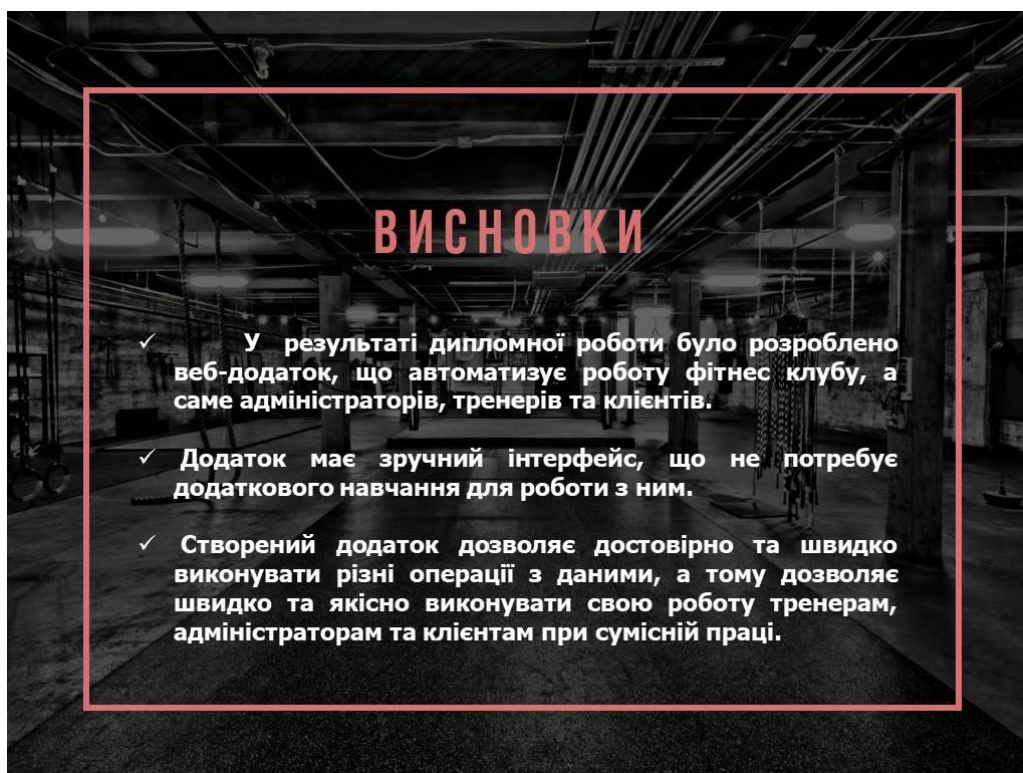


Рисунок Б.18 –Сторінка «Висновки»