НАВЧАЛЬНО–НАУКОВИЙ КОМПЛЕКС

“ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”

НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Кафедра Математичних методів системного аналізу

КУРСОВА РОБОТА

з дисципліни

Системи баз даних

на тему: “Соціальна мережа для фотографів”

Студента III курсу групи КА–75

Шевчук О.С.

Спеціальність: 122 Комп’ютерні науки

Керівник: асист. Афанасьєва І. В.

Національна оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

**Анотація**

У даній роботі представлена розробка сайту для спортивного залу гуртожитка №7 Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”.

Останнім часом серед підлітків та молоді набуває все більшої популярності заняття спортом. Таким видом занять захоплюються люди, які люблять конкурувати з іншими та боротися проти самого себе, досягати нових вершин та долати перепони. Але спорт потребує контролю та осмисленого підходу, адже без цього неможливо досягнути результату. Ентузіасти читають літературу, дивляться відео, пишуть нотатки і все для того, щоб побити свої рекорди, поставлені раніше.

У даній роботі представлено сайт, який спрощує життя таким спортсменам. Зараз у веб-додатку користувач може зареєструватися, увійти у систему та вийти з неї, а також створювати власні тренування (з урахуванням можливостей спортивного залу гуртожитку №7) і додати результати до них.

**Abstract**

The development of web-page of gym of hostel №7 of National technical university of Ukraine Kyiv polytechnic institute is presented in this work.

Today, sport becomes more popular among teenagers. This hobby is liked by people, which like to compete with other and to fight with themselves, to achieve new goals and to overcome the obstacles. But sport require a control and strategic approach, because it is impossible to get a result without it. Enthusiasts read an information, watch videos, write notes in order to beat their records, which were achieved earlier.

The web-site, which simplifies the life of these sportsmen, is presented in this work. At this moment, user has an ability to sign in, log in system and log out from it, and also to create his own trainings (taking into account the abilities of hostel №7) and add results to them.

[Вступ](#_3uc6gd4tw147) 5

[РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ САЙТУ СОЦІАЛЬНА МЕРЕЖА ДЛЯ ФОТОГРАФІВ](#_asf3dzdmfpva)  7

[Обґрунтування доцільності розробки](#_73w27ifo42j5) 7

[Обґрунтування вибору інструментів розробки](#_wqo6j99le697) 8

[Постановка задачі проектування](#_yl3sr62amdjg) 10

[РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ](#_om4qhfslvk5o) 12

[Концептуальна модель інформаційної системи. Діаграми потоків даних](#_djbukmsa747o) 12

[Інфологічна модель бази даних. Діаграми «сутність – зв’язок»](#_l0i3v5dv67is) 14

[РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ](#_s2dg5a9hl12b) 16

[Реалізація взаємодії прикладної програми зі спроектованою базою даних](#_ffu2629b3m6r) 16

[Інструкція користувача](#_cv6ycc3e4jwh) 18

[ВИСНОВКИ](#_ows6gixk5w0i) 24

[Список використаних джерел](#_2zdqijg3hxg4) 27

[ДОДАТОК А. Лістинг код](#_dq3hf1fu0698) 28

[ДОДАТОК Б. ДІАГРАМА ПОТОКІВ ДАНИХ](#_q49tiejg8uza) 30

# **Вступ**

У даній роботі представлена розробка сайту для спортивного залу гуртожитка №7 Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”.

Дана тема була вибрана по причині зацікавленості розробника спортом, а конкретно заняттям у спортивному залі. Вибір цієї теми роботи стимулював розробника зробити якомога якісніший продукт як на всіх етапах його розробки від фронтенда до бекенда та баз даних, так і кінцевому результаті в цілому. Даний проект допоміг розробнику отримати навички моделювання системи, розбиття задачі на частини та поетапної розробки продукту.

Дана робота в повній мірі демонструє знання розробника про системи баз даних, адже вони активно використовуються для коректного зберігання даних користувачів.

У ході виконання роботи були використані різноманітні технології розробки програм:

* Node.js runtime environment;
* Express framework;
* Passport.js library;
* Cryptr library;
* HTML і CSS;
* JavaScript programming language;
* JQuery library;
* Bootstrap 4 library;
* MySQL DB;
* MySQL Workbench.

# **РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ САЙТУ СПОРТИВНОГО ЗАЛУ**

## Обґрунтування доцільності розробки

Ми живем у такий період, коли все більше і більше молоді та підлітків залучається до спорту. Ця тенденція є дуже гарною для сучасного суспільства, адже спорт зміцнює людське здоров’я. Зараз люди самі залучаються до заняття спортом, адже це є дуже популярним, але можна підсилити цей ефект. Дайте суспільству можливість, покажіть як і воно саме почне займатись цим. Тому зараз саме той момент коли кожен з нас має шанс зробити той крок до життя у здоровому, гарному, процвітаючому суспільстві.

Наразі серед спортсменів-ентузіастів існує великий попит на додатки з можливість створення своїх тренувань та запису та зберігання результатів цих тренувань, адже вини змушені запам’ятовувати усю цю інформацію або записувати вручну. Дана розробка якраз покриває потреби таких людей, адже надає можливість створювати тренування та записувати результати і зберігати в електронному вигляді. Користувачу даного додатку більше буде не потрібно шукати свій записничок, щоб подивитись свої результати на попередньому тренуванні, а буде достатньо взяти мобільний телефон та знайти все потрібне там.

Крім цього треба зазначити, що розроблений сайт моделювався для студентів КПІ, що проживають у гуртожитку №7, тому дана розробка базувалась можливостях місцевого спортзалу. Такий підхід має як свої переваги, так і недоліки. Проект не матиме такої універсальності, як мають подібні розробки на ринку спортивних додатків, проте для мешканців гуртожитку дана розробка буде ідеальним варіантом, адже вона зачіпає певні особливі аспекти спортивного залу. На додачу, на ринку гуртожитку подібних проектів немає, тому проект має свою цільову аудиторію і буде вигідним для розробки.

## Обґрунтування вибору інструментів розробки

Для реалізації фронтенду було використано широкий спектр технологій. Основними з них є засоби розробки HTML [1], CSS [2] та JS [3]. HTML – це гіпертекстова мова розмітки, що потрібна для створення скелету нашого сайту. Вона є сірим, негарним фундаментом веб-додатку. CSS – це каскадні таблиці стилю, що використовуються з метою стилізації сайту та налаштування його зовнішнього вигляду. JS (JavaScript) – мова програмування, що використовується для надання сайту динамічності, а також зв’язку клієнтської та серверної частин. Крім вище зазначеного, у фронтенд частині проекту було використані наступні допоміжні елементи:

* Bootstrap 4 library [4];
* JQuery library [5].

Ці бібліотеки допомагають стилізувати та розташовувати елементи сайту. Вони не є необхідними для розробки, проте значно спрощують життя веб-розробника. Варто зазначити, що ці дві технології є дуже популярними у області прикладного веб-програмування. Це підтверджується статистикою, адже на 2019 рік JQuery використовується у 74% фронтенд частинах веб-додатків, а Bootstrap 4 library 20%[6].

Для реалізації бекенду була використана технологія Node.js [7] та її бібліотеки і фреймворки:

* Express framework;
* Session library;
* Passport.js library;
* Cryptr library.

Використання Node.js обумовлено тим, що вона є нескладною в упануванні. Оскільки технологія використовує мову JavaScript, яка душе швидко розвивається, модифікується та покращується, то вона є досить перспективною для її подальшого розвитку. Фреймворк Express використовується для обробки запитів з клієнтської частини програми. Бібліотека Session потрібна для створення сесії користувача, а Passport.js для створення паспорта користувача, що зберігає інформацію про нього і дає можливість автоматичної авторизації та аутентифікації користувача. Бібліотека Cryptr була використана для шифрування паролю користувача, при записі в базу даних. Це певною мірою захищає користувача від можливого розповсюдження приватної інформації.

Для зберігання інформації зазвичай використовують різноманітні види баз даних. У даному випадку було використано реляційну базу даних MySQL [9]. Цей вибір обумовлюється тим, що дана технологія займає друге місце за популярністю після PostgreSQL. Крім цього саме на прикладі цієї бази даних викладався курс систем баз даних, тому освоїти цю технологію було легше.

На останок слід зазначити, що усі вище згадані технології лежать у відкритому доступі. З цього слідує, що кожен розробник може отримати доступ до стартового коду бібліотек і фреймворків та налаштувати їх під себе.

## Постановка задачі проектування

Важливим етапом розробки будь-якого додатку є його проектування. Після вибору інструментів розробки якраз буде доцільно визначитись з тим, як кінцевий продукт має виглядати. У даному випадку сайт задумувався багатосторінковим, з можливістю авторизації та аутентифікації і основними функціями були надавання інформації та створення записів про тренування та їх результати.

Складаючи план розробки, було отримано наступний результат:

1. Створення макету сайту мовою HTML;
2. Створення візуальної складової сайту мовою CSS;
3. Створення основних елементів бази даних таких, як таблиці з урахуванням потрібних типів даних;
4. Написання JS функція з метою надання сайту динамічної поведінки;
5. Створення серверної частини програми та налаштування зв’язку між нею та базою даних;
6. Налаштування запитів від клієнтської частин и до серверної та від серверної до бази даних;
7. Тестування коректності роботи кінцевої програми;
8. Завантаження програми в онлайн сховище.

Зазначений вище план в достатній мірі описує етапи розробки програми, але треба враховувати, що у процесі розробки стаються різні ситуації і можливі певні зміни у плані.

# 

# **РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ**

## Концептуальна модель інформаційної системи. Діаграми потоків даних

Відповідно до обраної теми курсового проекту, було проведено аналіз і створено діаграма потоків даних. Процес запису даних у базу даних зображено на рис. 2.1. Процес виведення даних з бази даних зображено на рис. 2.2.

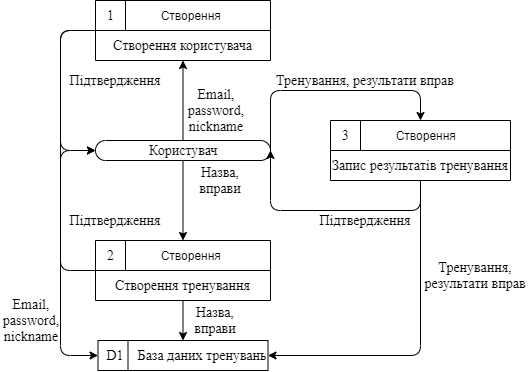


Рис. 2.1 Діаграма потоків даних процесу запису даних

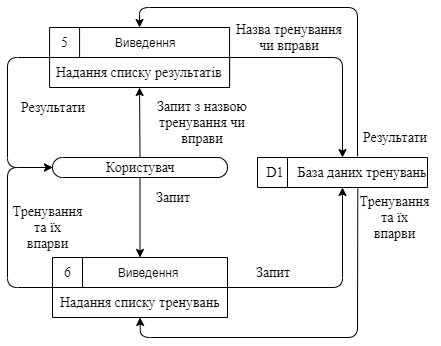


Рис. 2.2 Діаграма потоків даних процесу виведення даних

## Інфологічна модель бази даних. Діаграми «сутність – зв’язок»

На основі діаграми потоків даних побудуємо діаграму сутність–зв’язок (рис. 2.3.).

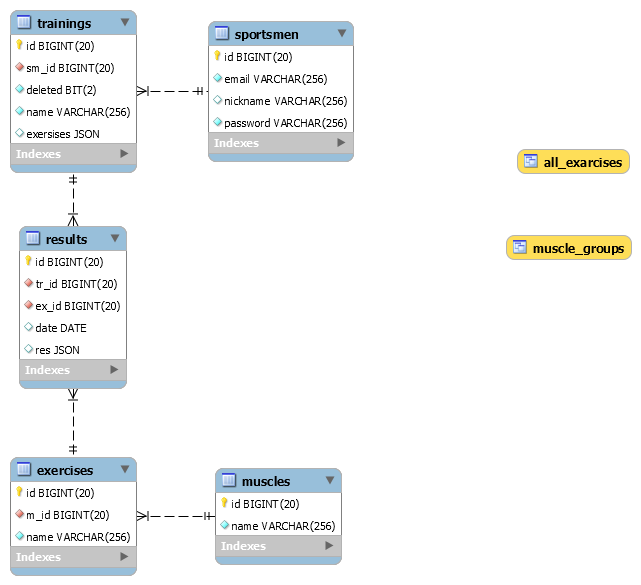


Рис. 2.3 Діаграма “сутність-зв’язок” для бази даних

Оскільки у даній базі даних всі атрибути є простими, всі домени містять скалярні значення і немає повторів кортежів у відношенні, то відношення знаходиться у першій нормальній формі (1НФ) [10]

Так як кожен неключовий атрибут незвідно залежить від первинного ключа, то відношення знаходяться у 2НФ [10].

Оскільки всі неключові поля, вміст яких може бути віднесений до декількох записів таблиці винесені у окремі таблиці, то усі відношення знаходяться у 3НФ [10].

Так як всі багатозначні функціональні залежності фактично є функціональними залежностями від ключів, то БД знаходиться у 4НФ [10].

Оскільки відсутні складні залежні з’єднання між атрибутами, то відношення знаходиться у 5НФ [10].

База даних не містить жодних інших обмежень крім обмежень доменів і обмежень ключів, тобто БД знаходиться у доменно–ключовій нормальній формі.

# **РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

## Реалізація взаємодії прикладної програми зі спроектованою базою даних

Для роботи з БД була використана СУБД MySQL 5.7. В свою чергу для роботи з MySQL було використано Node.js. Для творення підключення між базою даних та серверною частиною потрібно було створити спеціальний модуль налаштування взаємозв’язку між елементами. У цьому модулі указувались різні параметри серед яких є тип бази даних з якою створюється зв’язок, хост, до якого буде звертатися Node.js і порт, по якому буде іти підключення. Серед інших параметрів слід зазначати користувача в даній MySQL базі даних, його пароль та таблиця, з якою буде проводитись робота. Треба зауважити, що особливу увагу слід надати до вибору користувача, адже від цього буде залежати, чи буде він мати доступ до тих чи інших функцій, запитів, методів та ін. Тут потрібно добре розуміти які різновиди користувачів системи ви маєте (адмін чи звичайний користувач) і відповідно створювати різні можливості зв’язків Node.js та бази даних.

## Інструкція користувача

Користувач має можливість ознайомитися з сайтом. На головній сторінці він може прочитати інформацію та переглянути зображення інтер’єру та обладнання спортивного залу (рис. 3.1).

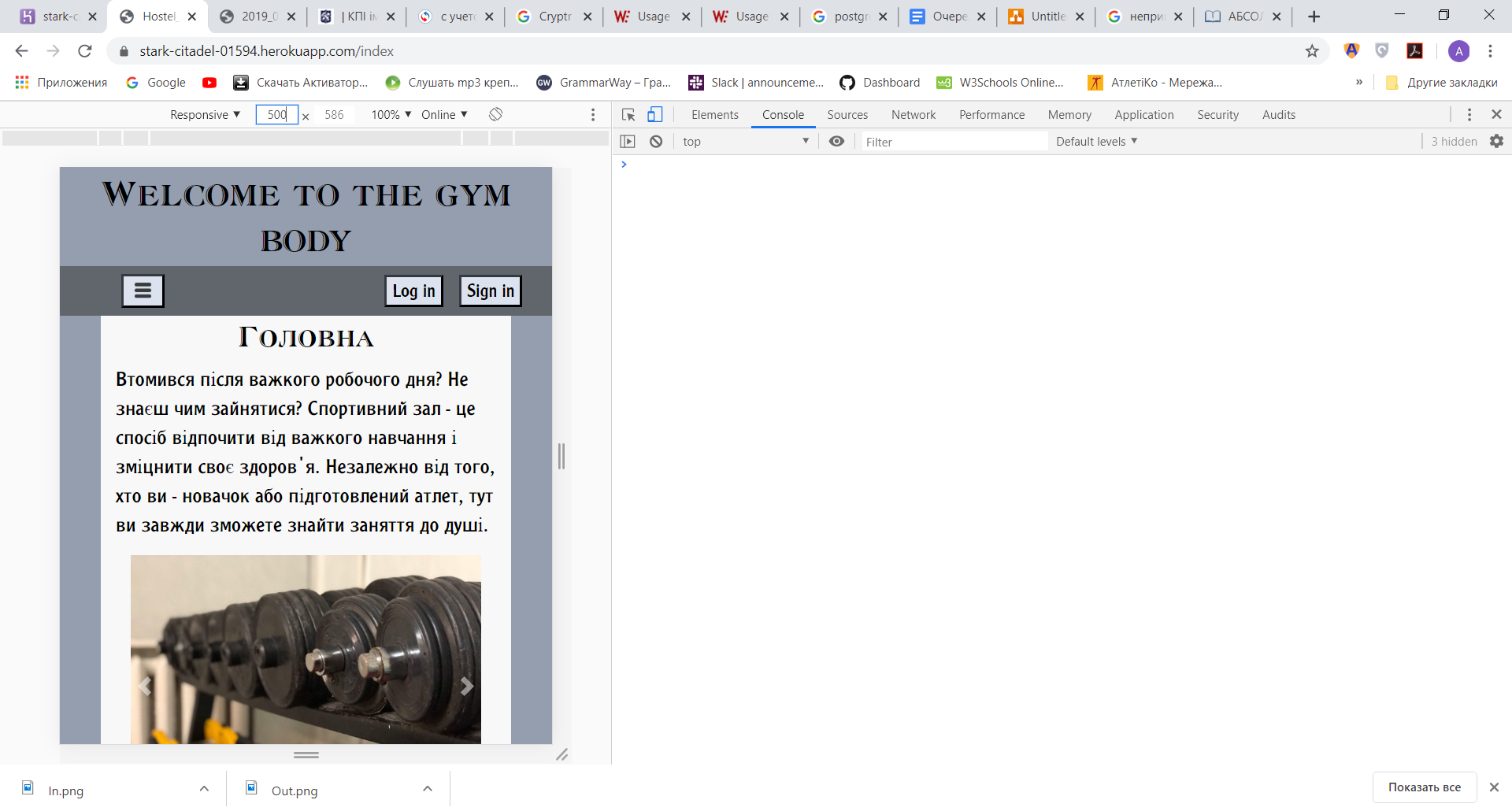


Рис. 3.1 Екран головної сторінки

Він має доступ до усіх сторінок, але при цьому його можливості обмежені. Користувач не бачить частину вмісту сайту, доки не аутентифікується. Він це може зробити натиснувши на кнопку Log in (рис. 3.1). При на тиканні зазначеної кнопки, у користувача на екрані з’явиться форма для заповнення, у якій він матиме можливість підтвердити свою особистість (рис. 3.2). У разі гарного перебігу подій, сторінка перезавантажиться і користувач увійде до аккаунту. Якщо певні дані були введені некоректно, то користувач побачить інформацію про це на екрані монітора.

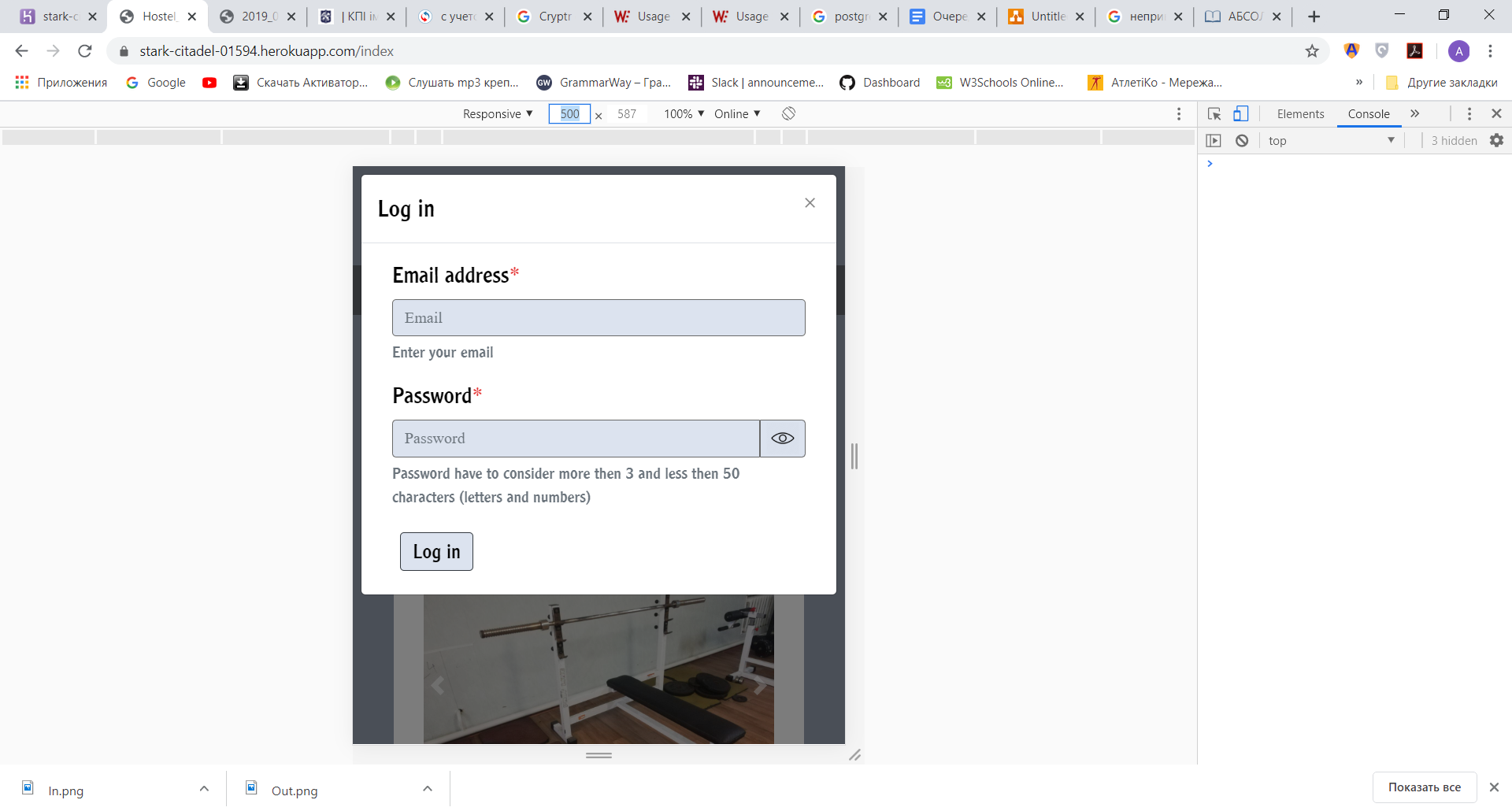


Рис. 3.2 Екран аутентифікації

У випадку коли користувач не має аккаунту на даному веб-сайті, він має можливість зареєструватись натиснувши на кнопку Sign in (рис. 3.1). У цьому випадку, як і попереднього разу перед очима користувача з’явиться форма для заповнення (рис. 3.3). Заповнивши коректно усі поля форми, користувач створить акаунт і отримає сповіщення про це, інакше отримає сповіщення про невдачу створення.

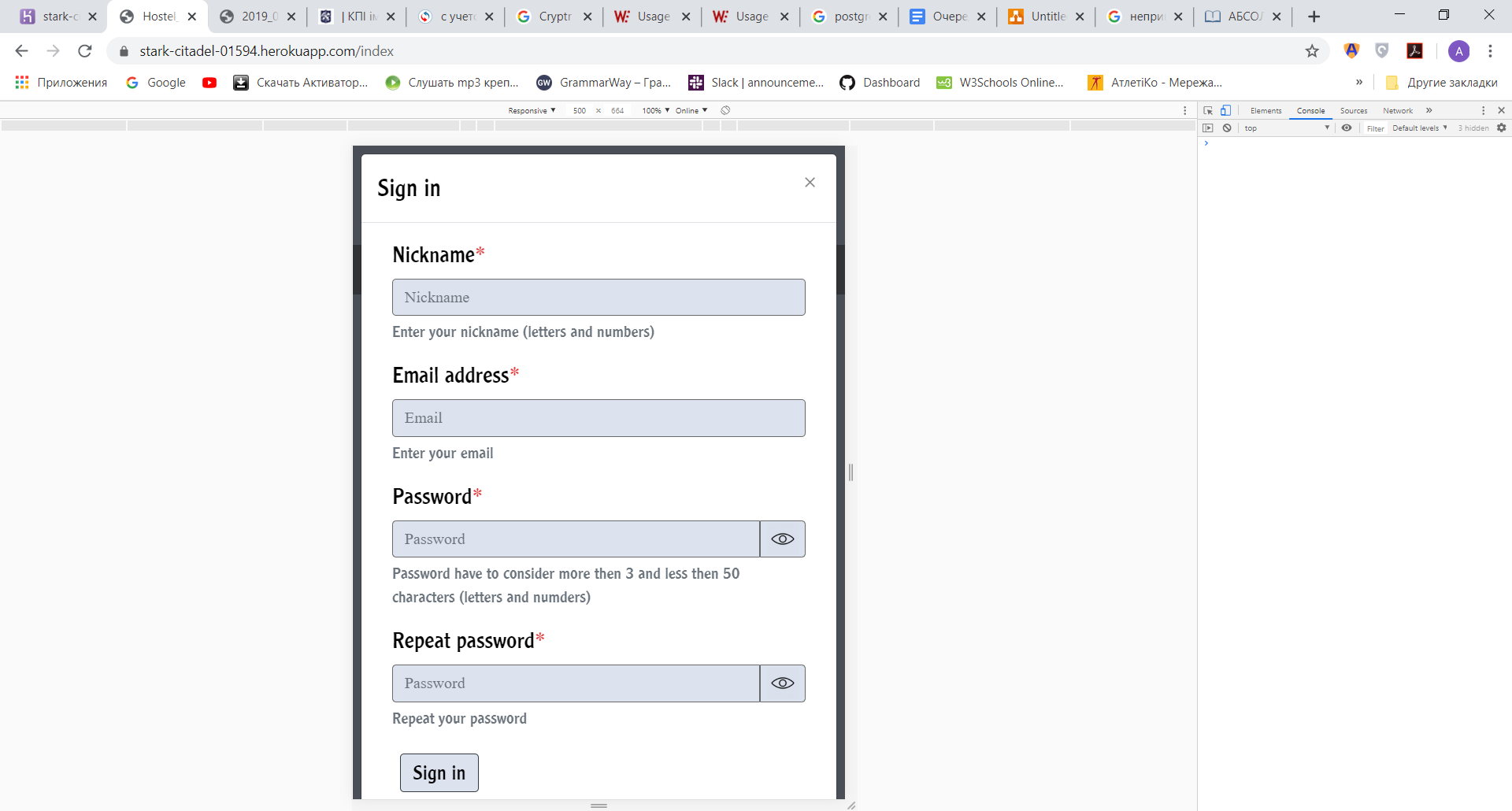


Рис. 3.3 Екран реєстрації

Увійшовши до аккаунту користувач отримує можливість скористатися усіма можливостями даного проекту. Після входу користувач біля головного меню сайту буде бачити свій ім’я користувача (рис. 3.4). Це буде ще одним підтвердженням того, що він увійшов до аккаунту.

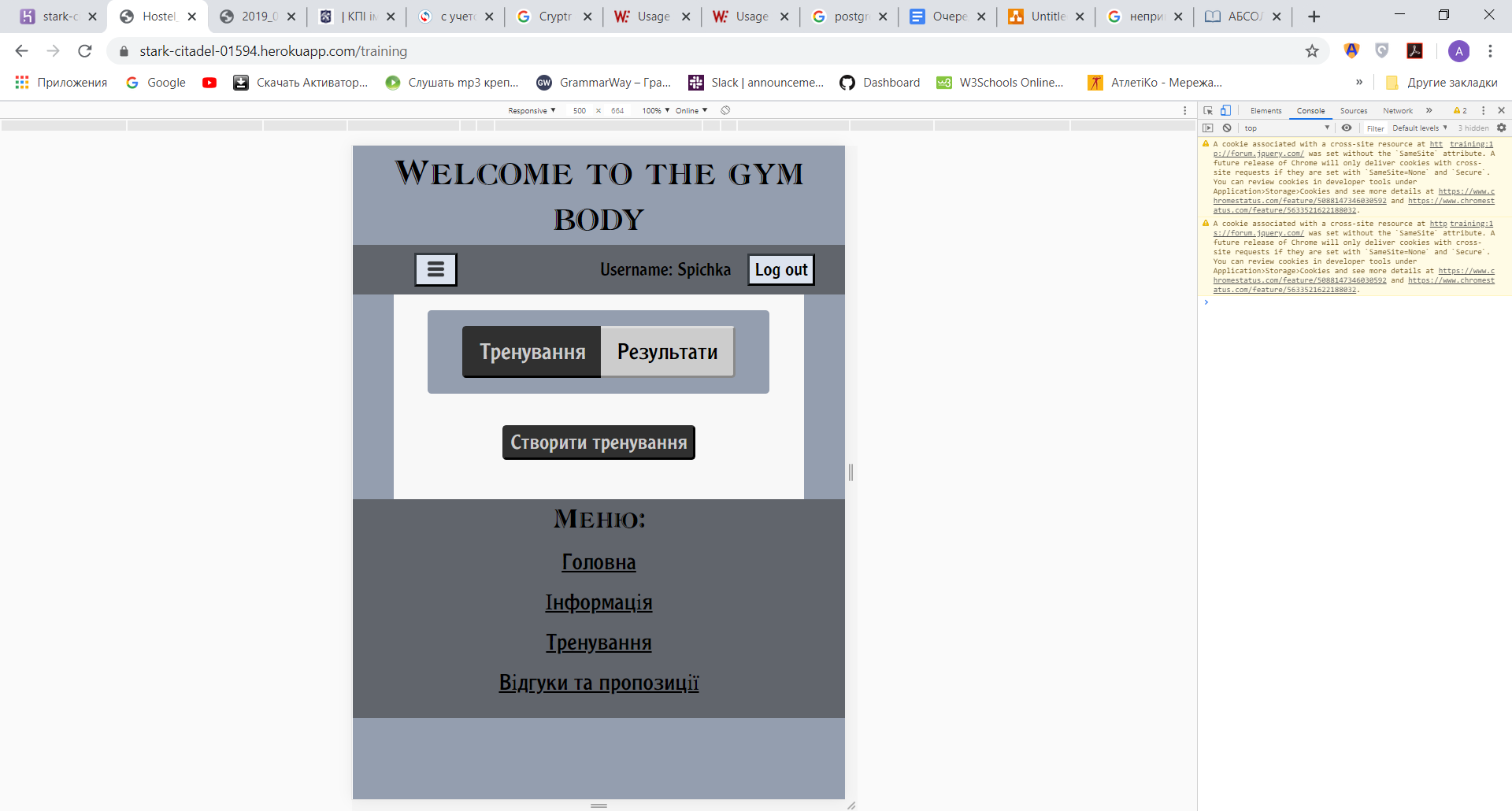


Рис. 3.4 Екран сторінки тренувань

Після підтвердження своєї особистості новий користувач матиме можливість створити своє тренування. Для цього потрібно натиснути кнопку “Створити тренування” і отримати нову форму для вводу даних (рис. 3.5). У цій формі потрібно заповнити поле “Назва тренування” та обрати потрібні вправи. Для фільтрації можна обирати різні групи м’язів. Це пришвидшить пошук вправ. Якщо користувач захоче додати ще тренування, йому потрібно натиснути кнопку “Додати” (рис. 3.5). Для відправлення запиту на створення тренування в базі даних користувач повинен натиснути кнопку “Створити” (рис. 3.5).

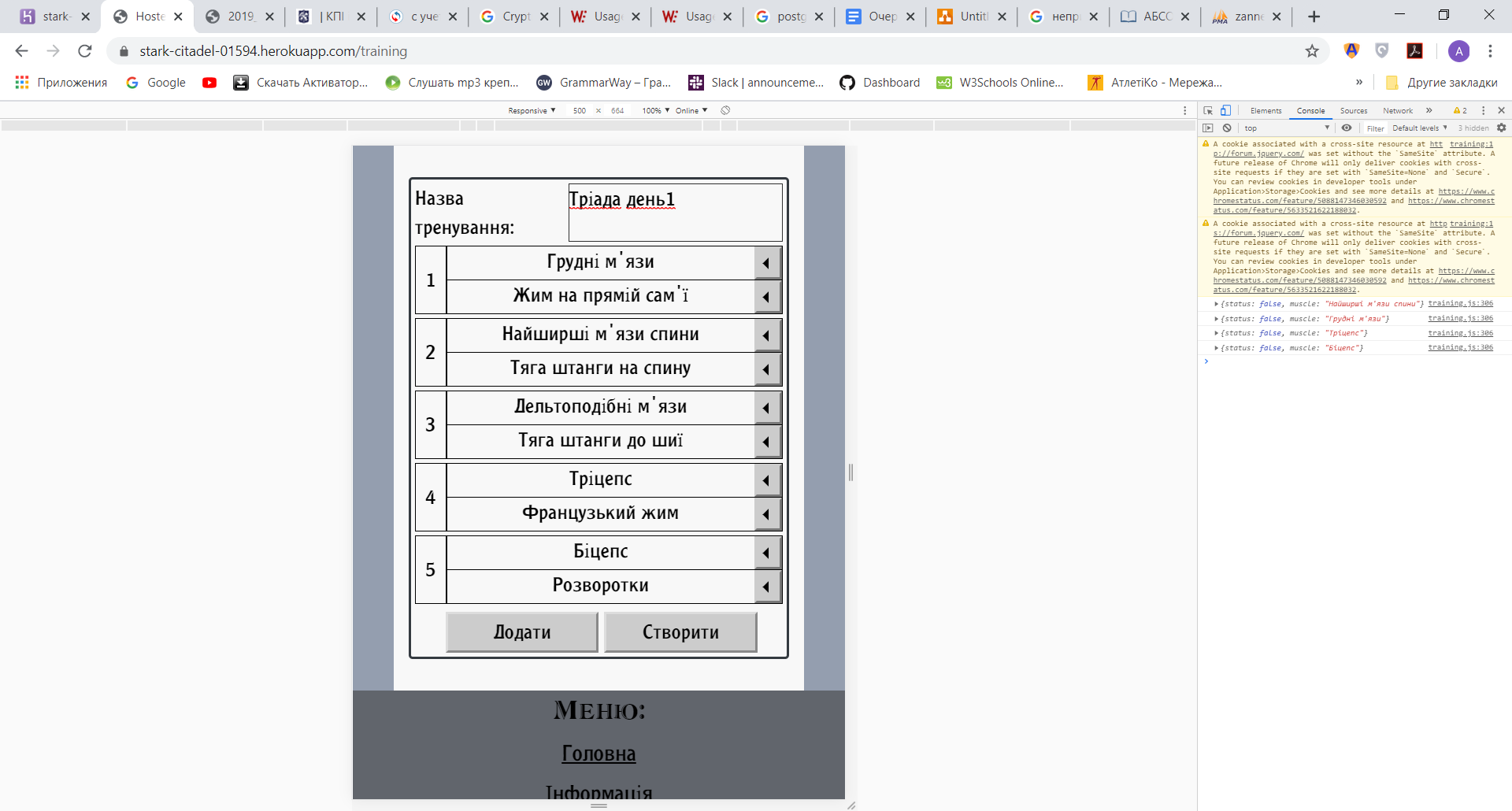


Рис. 3.5 Екран створення тренування

Після натискання клавіші створити сторінка сайту буде перезавантажена і користувач побачить результат виконаної роботи (рис. 3.6). Якщо у користувача є створені тренування він також має певні можливості взаємодіяти з ними. Переглянути список своїх тренувань він може на сторінці, що містить редактор створення тренування. Перш за все користувач може переглянути вправи, що входять у дане тренування, натиснувши на клавішу випадаючого меню (рис. 3.6). На цю ж саму кнопку він може сховати список вправ.

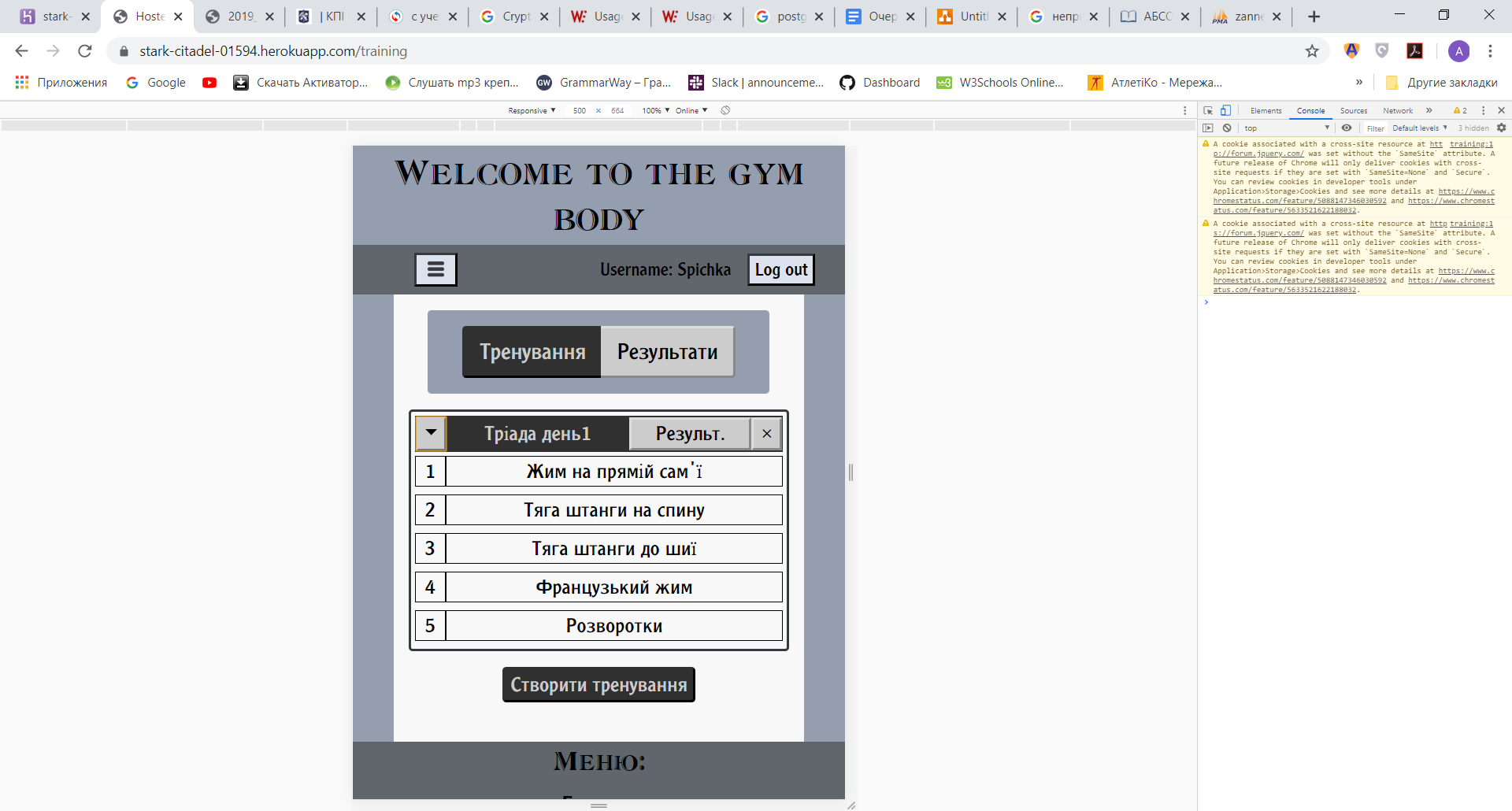


Рис 3.6 Екран користувача з тренуваннями

Другою можливістю є видалення даного тренування. При цьому інформація про результати по вправах, що містились у цьому тренуванні збережеться. При видаленні тренування зникає лише можливість перегляду результатів при сортуванні по тренуванню і воно більше не буде з’являтися в списку тренувань. Для видалення потрібно натиснути хрестик навпроти тренування (рис. 3.6).

Наступною сторінкою користувача є сторінка з результатами (рис. 3.7). На даній сторінці користувач має можливість переглянути результати своєї роботи у залі. При цьому у нього є можливість обрати, за якими критеріями він хоче переглядати ці результати. Є можливість переглянути результати відсортувавши за тренуваннями та за вправами (рис. 3.7).

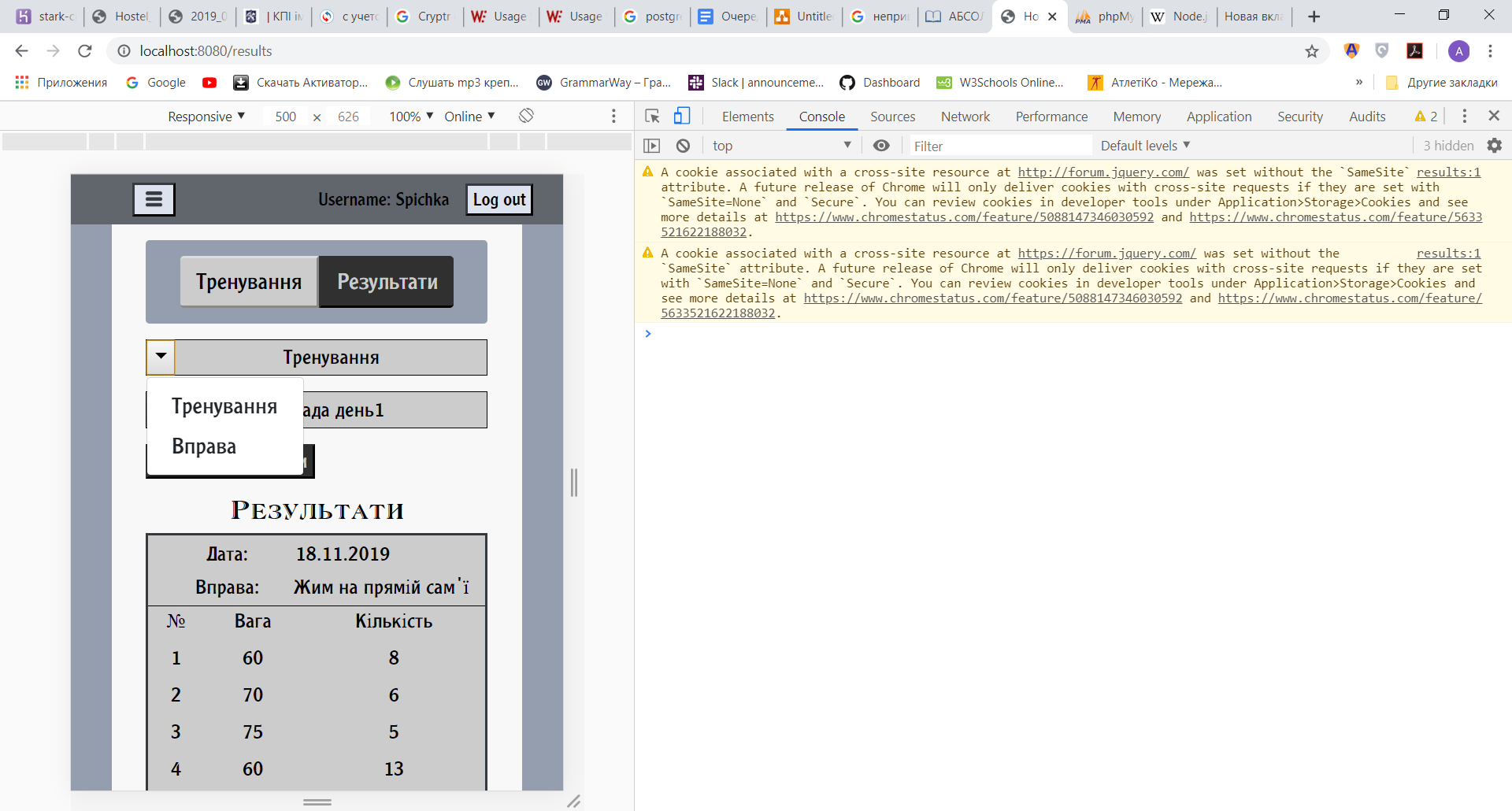


Рис. 3.7 Екран перегляду результатів

Для додавання результатів тренування потрібно обов’язково відсортувати результати за тренуваннями. Лише у цьому випадку на екрані з’явиться кнопка “Додати результати” (рис. 3.8). Після натискання даної клавіші перед користувачем з’явиться форма з полями для заповнення (рис. 3.8). Також разом з полями з’являться кнопка з підписом “Додати” (рис. 3.8). При натисканні даної кнопки до вправи, що знаходиться вище будуть додаватися додаткові поля вводу.

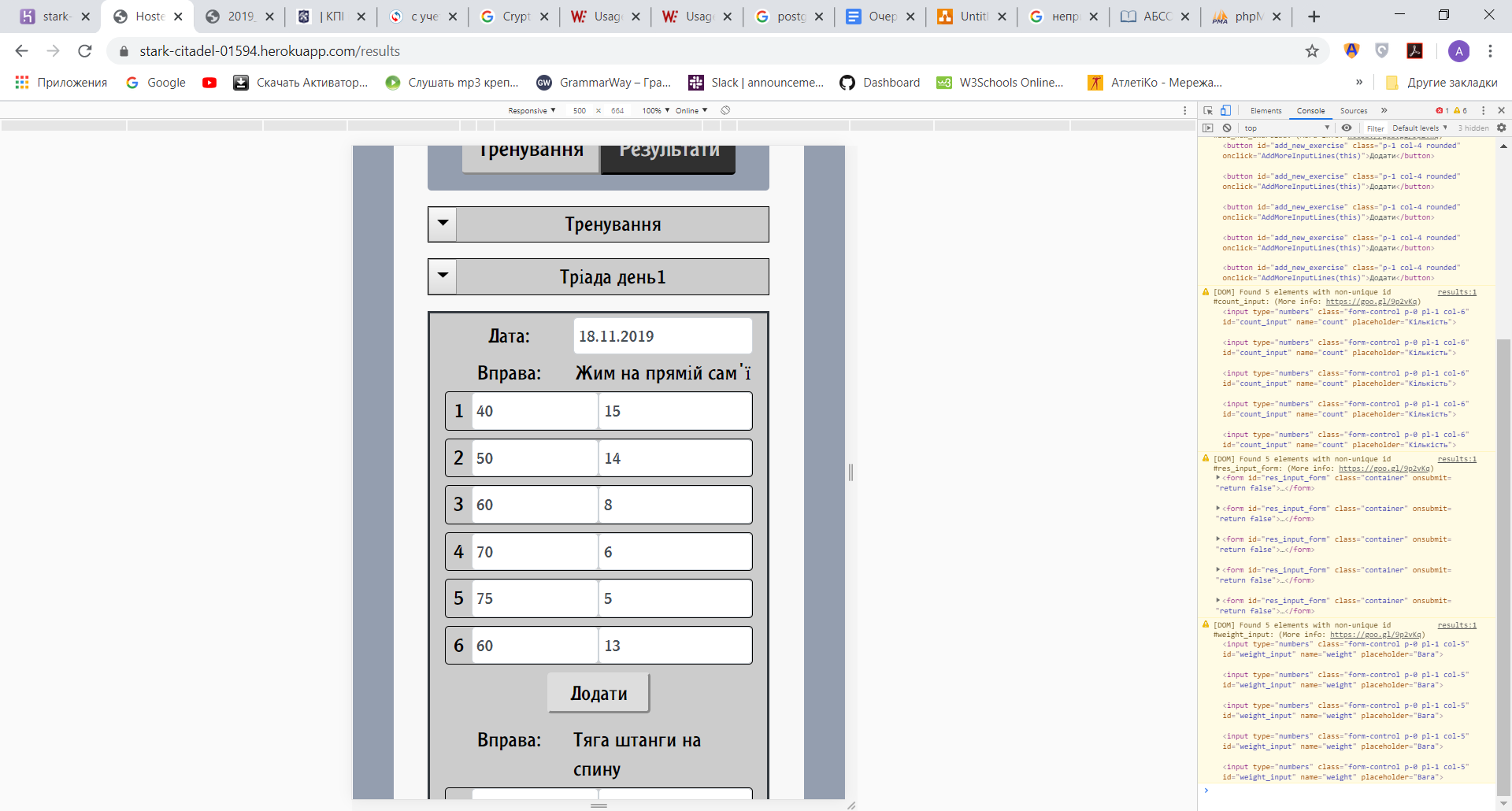


Рис. 3.8 Екран вводу результатів

Якщо користувач закінчив вводити результати, він може відправити результати натиснувши кнопку “Створити” (рис. 3.9). Після натискання даної клавіші сторінка користувача буде перезавантажена. У разі некоректного вводу даних користувач буде повідомлений про це.

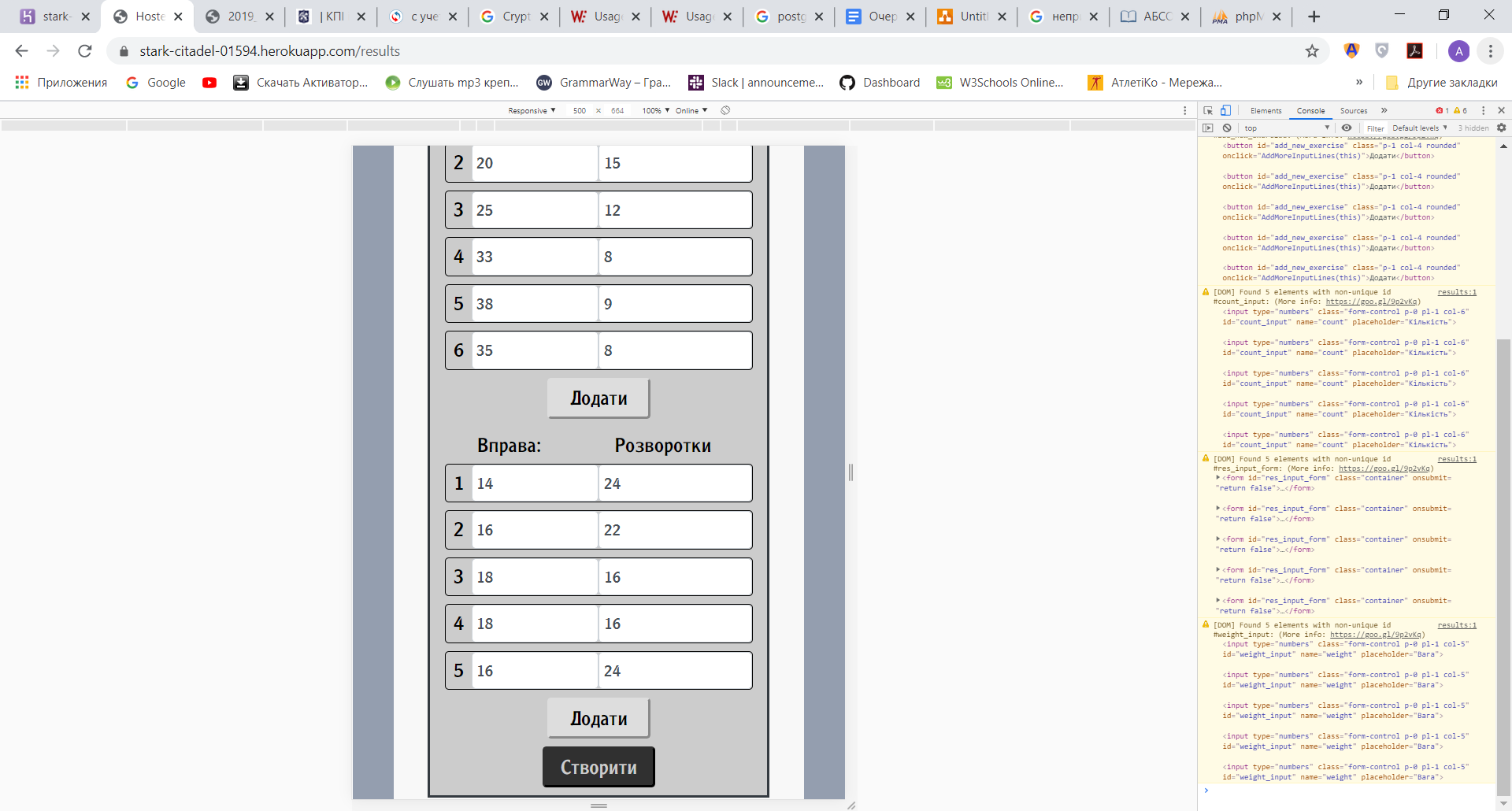


Рис. 3.9 Екран створення введених результатів

# **ВИСНОВКИ**

У результаті виконання даного проекту було створено веб-сайт на тему: “Спортивний зал гуртожитку №7”. Було створено базу даних та наповнено її всіма можливими видами вправ, що можна виконувати у спортивному залі гуртожитку, а також заповнено групи м’язів на які ці вправи можуть впливати. Зазначена вище база даних була прив’язана до дружнього графічного інтерфейсу, що дає можливість людям не знайомим з програмуванням користуватися даною базою.

У процесі виконання були освоєні навички розробки веб-додатків, їх фронтенд та бекенд частин, а також створення баз даних Серед навичок розробки баз даних них слід відзначити:

* концептуальна модель бази даних: діаграми потоків даних;
* інфологічна модель бази даних. Діаграма “сутність–зв’язок”;
* даталогічна модель бази даних. Схеми функціональних залежностей. Нормалізація відношень;
* фізична модель бази даних. Засоби забезпечення цілісності;

Крім роботи з базами даних проводилась робота з величезною кількістю інших технологій, таких як:

* Node.js runtime environment – середовище розробки, для створення високопродуктивних мережевих застосунків, використовуються для прийому запитів від клієнтської частини продукту;
* HTML і CSS – базові засоби розробки будь-якої веб-сторінки, що використовуються для розмітки та стилізації сайтів;
* Express framework та його Password.js, Session бібліотеки – засоби розробки, що використовується для створення постійної сесії між сервером та користувачем та зберігання його паспорту – даних, допомагають автоматично проводити аутентифікацію користувача;

В цілому було створено готовий кінцевий продукт, що з легкістю може іти у використання.

Провівши кінцевий аналіз продукту можна визначити його позитивні та негативні сторони. Плюсами проекту можна назвати:

* систему авторизації;
* легкість та зручність інтерфейсу;
* величину попиту на ринку;
* наявність цільової аудиторії;
* великі можливості розвитку проекту.

Негативними елементам даного кінцевого продукту є:

* поганий дизайн сайту;
* відсутність деякого функціоналу інтерфейсу;
* недостатньо якісна валідація вхідних даних;
* відсутність можливості зміни результатів тренувань;
* відсутність сторінки адміністратора сайту.

Хоча в програмі й існує певна кількість недоліків, я вважаю що вони усі досить легко можуть бути вирішені, оскільки у ході розробки проекту були отримані ґрунтовні навички створення великих веб-застосунків з усіма можливими їх елементами. Тому проект є дуже перспективним та цікавим, якщо продовжити розробку у майбутньому.

# 

# **Список використаних джерел**

1. HTML documentation [Електроний ресурс] - <https://devdocs.io/html/>
2. CSS documentation [Електроний ресурс] - <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. Usage of JavaScript libraries for websites [Електронний ресурс] – Режим доступу до ресурсу: <https://w3techs.com/technologies/overview/javascript_library>.
4. Bootstrap documentation [Електроний ресурс] - <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
5. JQuery documentation [Електроний ресурс] - <https://api.jquery.com/>
6. Express documentation [Електроний ресурс] - <http://expressjs.com/uk/api.html>
7. Usage of JavaScript libraries for websites [Електроний ресурс] - <https://w3techs.com/technologies/overview/javascript_library>
8. Node,js documentation [Електроний ресурс] - <https://nodejs.org/ru/docs/>
9. MySQL Documentaion [Електроний ресурс] - <https://dev.mysql.com/>
10. Теория и практика построение баз данных. 8 изд. / Д. Кренке - СПб: Питер, 2003. - 800 с.

# **ДОДАТОК А. Лістинг код**

CREATE TABLE `sportsmen` (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`email` varchar(256) NOT NULL,

`nickname` varchar(256),

`password` varchar(256) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY unique\_email (email)

)

ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;;

CREATE TABLE `muscles`(

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`name` varchar(256) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY unique\_name (name)

)

ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;;

CREATE TABLE `exercises`(

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`m\_id` bigint(20) unsigned NOT NULL,

`name` varchar(256) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY unique\_name (name)

)

ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;;

CREATE TABLE `trainings`(

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`sm\_id` bigint(20) unsigned NOT NULL,

`deleted` bit(2) NOT NULL,

`name` varchar(100) NOT NULL,

`exersises`JSON,

PRIMARY KEY (`id`)

)

ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;;

CREATE TABLE `results`(

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`tr\_id` bigint(20) unsigned NOT NULL,

`ex\_id` bigint(20) unsigned NOT NULL,

`date` Date,

`res` JSON,

PRIMARY KEY (`id`))

ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;;

ALTER TABLE `exercises`

ADD FOREIGN KEY (`m\_id`) REFERENCES `muscles` (`id`);;

ALTER TABLE `trainings`

ADD FOREIGN KEY (`sm\_id`) REFERENCES `sportsmen` (`id`);;

ALTER TABLE `results`

ADD FOREIGN KEY (`tr\_id`) REFERENCES `trainings` (`id`);;

ALTER TABLE `results`

ADD FOREIGN KEY (`ex\_id`) REFERENCES `exercises` (`id`);;

CREATE PROCEDURE `SPORTSMAN\_INSERT`(

IN `email` varchar(256),

IN `nickname` varchar(256) charset utf8 COLLATE utf8\_unicode\_ci,

IN `password` varchar(256) charset utf8 COLLATE utf8\_unicode\_ci

)

BEGIN

INSERT INTO sportsmen(`email`, `nickname`, `password`)

VALUES(email, nickname, password);

END;;

CREATE PROCEDURE `MUSCLE\_INSERT`(

IN `name` varchar(256) charset utf8 COLLATE utf8\_unicode\_ci

)

BEGIN

INSERT INTO `muscles`(`name`)

VALUES(name);

END;;

CREATE PROCEDURE `EXERCISE\_INSERT`(

IN `m\_id` bigint(20),

IN `name` varchar(256) charset utf8 COLLATE utf8\_unicode\_ci

)

BEGIN

INSERT INTO `exercises`(`m\_id` ,`name`)

VALUES(m\_id, name);

END;;

CREATE PROCEDURE `TRAINING\_INSERT`(

IN `sm\_id` bigint(20),

in `deleted` bit(1),

IN `name` varchar(256) charset utf8 COLLATE utf8\_unicode\_ci,

IN `exercises` JSON

)

BEGIN

INSERT INTO `trainings`(`sm\_id` ,`deleted` , `name`, `exersises`)

VALUES(sm\_id, deleted, name, exercises);

END;;

CREATE PROCEDURE `DELETE\_TRAINING`(

tr nvarchar(256) COLLATE utf8\_unicode\_ci,

mail varchar(256))

BEGIN

UPDATE `trainings` t

LEFT JOIN `sportsmen` s ON (s.id = t.sm\_id)

SET t.`deleted` = 1

where (s.`email` = mail) && (t.`name` = tr);

END;;

CREATE PROCEDURE `ADD\_RESULTS`(

tr\_id bigint(20),

ex\_id bigint(20),

dt date,

res json)

BEGIN

INSERT INTO `results`(`tr\_id`, `ex\_id`, `date`, `res`)

VALUES (tr\_id, ex\_id, dt, res);

END;;

CREATE FUNCTION `FIND\_SPORTSMAN\_BY\_EMAIL`

(mail varchar(256))

RETURNS bit(1)

NOT DETERMINISTIC

BEGIN

if(SELECT COUNT(`email`) from `sportsmen`

where (`email` = mail)) != 0

then

RETURN 1;

else

RETURN 0;

end if;

END;;

CREATE FUNCTION `GET\_USER\_PASSWORD`

(mail varchar(256))

RETURNS varchar(256)

NOT DETERMINISTIC

BEGIN

RETURN (SELECT `password` from `sportsmen`

where (`email` = mail));

END;;

CREATE FUNCTION `GET\_USER\_NICKNAME`

(mail varchar(256))

RETURNS varchar(256)

NOT DETERMINISTIC

BEGIN

RETURN (SELECT `nickname` from `sportsmen`

where (`email` = mail));

END;;

CREATE FUNCTION `GET\_EXERSISE\_MUSCLES`

(ex nvarchar(256) COLLATE utf8\_unicode\_ci)

RETURNS nvarchar(256) COLLATE utf8\_unicode\_ci

NOT DETERMINISTIC

BEGIN

RETURN (SELECT m.`name`

from `muscles` m

LEFT JOIN `exercises` e ON (m.id = e.m\_id)

where (e.`name` = ex));

END;;

CREATE FUNCTION `GET\_SPORTSMAN\_ID\_BY\_EMAIL`

(mail varchar(256))

RETURNS bigint(20)

NOT DETERMINISTIC

BEGIN

DECLARE res bigint(20);

SET res = (SELECT `id` from `sportsmen`

where (`email` = mail));

RETURN res;

END;;

CREATE FUNCTION `CHECK\_TRAINING\_NAME\_UNIQUENESS`

(mail varchar(256),

`exname` nvarchar(256) COLLATE utf8\_unicode\_ci)

RETURNS bit(1)

NOT DETERMINISTIC

BEGIN

RETURN (SELECT Count(t.`name`) from `trainings` t

LEFT JOIN `sportsmen` sm ON (sm.id = t.sm\_id)

where (sm.`email` = mail) && (t.`name` = exname));

END;;

CREATE FUNCTION `GET\_TRAINING\_ID\_BY\_NAME\_EMAIL`

(mail varchar(256),

`name` nvarchar(256) COLLATE utf8\_unicode\_ci)

RETURNS bigint(20)

NOT DETERMINISTIC

BEGIN

RETURN(SELECT t.`id` from `trainings` t

LEFT JOIN `sportsmen` sm ON (sm.id = t.sm\_id)

where (sm.`email` = mail) && (t.`name` = `name`));

END;;

CREATE FUNCTION `GET\_EXERCISES\_ID`

(name nvarchar(256) COLLATE utf8\_unicode\_ci)

RETURNS bigint(20)

NOT DETERMINISTIC

BEGIN

RETURN (SELECT e.`id`

from `exercises` e

where (e.`name` = name));

END;;

SELECT `nickname` as name, `password` as password

from `sportsmen`

where (`email` = ?);

SELECT t.`name` from `trainings` t

LEFT JOIN `sportsmen` sm ON (sm.id = t.sm\_id)

where sm.`email` = ?

SELECT t.`exersises` from `trainings` t

LEFT JOIN `sportsmen` sm ON (sm.id = t.sm\_id)

where (sm.`email` = 'sasa@gmail.com') && (t.`name` = 'Sasa')

SELECT DATE\_FORMAT(r.date, '%d.%m.%Y') as date, r.res, e.name from `results` r

LEFT JOIN `exercises` e ON (e.id = r.ex\_id)

LEFT JOIN `trainings` t ON (r.tr\_id = t.id)

LEFT JOIN `sportsmen` sm ON (sm.id = t.sm\_id)

where (t.`name` = ? && sm.`email` = ?)

ORDER BY date desc

SELECT DATE\_FORMAT(r.date, '%d.%m.%Y') as date, r.res e from `results` r

LEFT JOIN `exercises` e ON (e.id = r.ex\_id)

LEFT JOIN `trainings` t ON (r.tr\_id = t.id)

LEFT JOIN `sportsmen` sm ON (sm.id = t.sm\_id)

where (e.`name` = ? && sm.`email` = ?)

ORDER BY date desc

create view muscle\_groups as

Select `name` from `muscles`;

create view all\_exarcises as

Select `name` from `exercises`;