

ДИПЛОМНА РОБОТА

ДР.Пс – 11.00.000 ПЗ

Група Пс-2015

Лилик М. М.

2019

Кафедра Інформаційних технологій та програмної інженерії

УДК 004.4

## ДИПЛОМНА РОБОТА

Тема *Розробка веб-сервісу для організації спортивних турнірів за круговою системою*

Напрямок підготовки *6.050103 – «Програмна інженерія»*  
(код і назва спеціальності)

## ПОЯСНЮВАЛЬНА ЗАПИСКА

ДР.Пс – 11.00.000 ПЗ  
(позначення)

Студент

*Лилик М.М.*  
(підпис) (дата) (розшифрування підпису)

Керівник проекту

*к.т.н., доц. Мануляк І.З.*  
(посада) (підпис) (дата) (розшифрування підпису)

Нормоконтроль

*к.т.н., доц. Мануляк І.З.*  
(посада) (підпис) (дата) (розшифрування підпису)

Допускається до захисту

Завідувач кафедри

*д.т.н., доц. Мельничук С.І.*  
(посада) (підпис) (дата) (розшифрування підпису)

# ПВНЗ УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет Інформаційних технологій

Кафедра Інформаційних технологій та програмної інженерії

Напрямок підготовки 6.050103 – «Програмна інженерія»

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри ІТПП

С.І. Мельничук

“ ” 2019 р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Студенту Лилику Миколі Миколайовичу

1. Тема проекту (роботи) Розробка веб-сервісу для організації спортивних турнів за круговою системою.

Затверджена наказом ректора Університету Короля Данила від 15.11.2018 р. № 20/4

2. Термін задачі студентом закінченого проекту (роботи) 10.06.2019 р.

3. Вихідні дані до проекту (роботи) Метод організації кругової системи в спортивних турнірах, вихідні дані про спортсменів, веб-орієнтовані мови програмування HTML, CSS, Java Script, PHP, MySQL .

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)  
1. Опис наявних аналогів, їх алгоритми роботи та опис використаних технологій. 2. Структура панелі адміністратора, інтерфейсу користувача та розробка бази даних. 3. Розробка алгоритмів для функціонування а також стилістика.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Постановка завдання, аналоги веб-сервісу, структура бази даних, авторизація, загальний вигляд адміністративної панелі, загальний вигляд інтерфейсу користувача, функції веб-сервісу, структура коду.

6. Консультанти з проекту (роботи), із зазначенням розділів проекту, що стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 30.10.2018

Керівник Мануляк І.З.

Завдання прийняв до виконання Лирик М. М.

### КАЛЕНДАРНИЙ ПЛАН

Пор №	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Опис наявних сайтів аналогів, їх алгоритми роботи та опис використаних технологій	30.11.2018	
2.	Розробка алгоритмів та дизайну кругової системи проведення турнірів	25.02.2019	
3.	Розробка допоміжних функцій та інтерфейсів для кругової системи проведення турнірів	22.03.2019	
4.	Оформлення пояснювальної записки	06.04.2019	
5.	Оформлення графічного матеріалу та підготовка до захисту роботи	10.05.2019	

Студент-дипломник Лирик М. М.  
(підпис) (розшифровка підпису)

Керівник проекту Мануляк І.З.  
(підпис) (розшифровка підпису)

## АНОТАЦІЯ

Даний дипломний проект присвячений розробці веб-сервісу для організації проведення турнірів за круговою системою. Метою роботи є розробка сервісу, яка дозволяє частково замінити людину в організації спортивних турнірів за круговою системою та забезпечити інформацією користувача. Розроблений проект реалізований засобами HTML, CSS, Javascript, PHP, база даних MySQL, Bootstrap.

Ключові слова: веб-сервіс, турніри, матчі, учасники, адміністратори, кругова система.

## **ABSTRACT**

This project is devoted to the development of a web service for organizing tournaments in a circular system. The purpose of the work is to develop a service that allows partially to replace a person in the organization of sports tournaments on a circular system and provide user information. The developed project is implemented by means such as: HTML, CSS, Javascript, PHP, MySQL database, Bootstrap.

Keywords: web-service, tournaments, matches, participants, administrators, round robin.

## РЕФЕРАТ

Розрахунково-пояснювальна записка: 82 сторінки, 54 малюнків, 2 додатки.

**Об'єктом дослідження** є методи організації змагань на основі кругової системи.

**Метою роботи** є розробка системи, яка дозволяє проводити турніри від їх створення до їх завершення за круговою системою.

Дипломний проект присвячений розробці веб-сервісу для організації спортивних турнірів за круговою системою.

Перший розділ складається з аналізу та пояснень, а також які засоби розробки застосовувались в дипломному проекті.

В другому розділі описано структуру бази даних та зв'язки між таблицями.

Третій розділ містить опис розробленої функціональної частини дипломного проекту.

ТУРНІРИ, МАТЧІ, УЧАСНИКИ, АДМІНІСТРАТОРИ, HTML, CSS, JAVASCRIPT, PHP, MYSQL.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	7
ВСТУП .....	8
1 АНАЛІЗ СЕРВІСІВ СПОРТИВНИХ ЗМАГАНЬ ТА ДОСЛІДЖЕННЯ КРУГОВОГО МЕТОДУ .....	9
1.1 Структура та особливості застосування кругового методу при організації спортивних турнірів .....	9
1.2 Аналіз інформаційних сервісів підтримки спортивних змагань.....	13
1.3 Вимоги до веб-сервісу, обґрунтування середовища розробки та постанова задачі .....	16
2 РОЗРОБКА СТРУКТУРНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-СЕРВІСУ .....	19
2.1 Загальна структура веб-сервісу .....	19
2.2 Структура бази даних .....	20
2.3 Структура панелі адміністратора та інтерфесу користувача .....	25
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-СЕРВІСУ ЗАСОБАМИ PHP, MYSQL, JS.....	28
3.1 Розробка інтерфейсу користувача .....	28
3.2 Вхід в панель адміністратора.....	29
3.3 Функції панелі адміністратора.....	32
3.4 Проектування інтерфейсу користувача .....	49
ВИСНОВОК.....	54
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	56
Додаток А.....	58
Додаток Б .....	70

						<b>ДР.Пс – 11.00.000 ПЗ</b>		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>						
<i>Розроб.</i>		<i>Лилик М. М.</i>			<i>Розробка веб-сервісу для організації спортивних турнірів за круговою системою</i>	<i>Літ.</i>	<i>Ар.</i>	<i>Акрушіє</i>
<i>Перевір.</i>		<i>Мануляк І.З.</i>					6	82
<i>Реценз.</i>						<i>УКД, Пс – 2015</i>		
<i>Н. Контр.</i>		<i>Мануляк І.З.</i>						
<i>Затверд.</i>		<i>Мельничук С.І.</i>						



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

JS – Java Script

PHP – Hypertext Preprocessor

MySQL – open-source relational database management system

SEO – search engine optimization (Пошукова оптимізація сервісу)

БД – база даних

					<b>ДР.ІІс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		7

## ВСТУП

Інтернет є невід'ємною частиною сучасної людини, адже всесвітня павутина перевернула наше життя з ніг на голову. Кожного дня в мережі сидять мільярди людей які мають спільні інтереси, розваги, або просто спілкуються між собою без границь. Технології та системи з кожним роком все більше дивують людей.

Оскільки останнім часом все більше людей почали займатись та цікавитись спортом, потрібно провести дослідження і на його основі розробити максимально зручний і корисний сервіс для проведення спортивних турнірів, визначити необхідні функції, яких потребуватиме проект. Тому **актуальність теми** зумовлена розробкою веб-сервісу для спортивних турнірів, що значно покращить організацію та спростить процес проведення змагань, зменшить затрати часу на підрахунки результатів, а також забезпечить інформацією користувачів.

Кругова система – це система розіграшів у спортивних змаганнях згідно з якою кожен учасник турніру грає з усіма іншими учасниками (кожен грає з кожним).

Найбільшу популярність здобула у футболі, хокеї, волейболі та баскетболі.

Ця система вважається найбільш справедливою, але при цьому порівняно з іншими турнірними системами, потребує великої кількості ігор, відповідно і часу затраченого на увесь турнір чи раунд.

					ДР.Шс – 11.00.000 ПЗ	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		8

# 1 АНАЛІЗ СЕРВІСІВ СПОРТИВНИХ ЗМАГАНЬ ТА ДОСЛІДЖЕННЯ КРУГОВОГО МЕТОДУ

## 1.1 Структура та особливості застосування кругового методу при організації спортивних турнірів

При розігрішці за круговою системою незалежно від результату змагань всі учасники грають до тих пір, поки кожен не зустрінеться з усіма іншими.

В результаті цих змагань переможцем вважається той з учасників, який набере більшу суму очок за всі змагання. Решта учасників займають наступні по порядку місця в залежності від кількості очок, набраних під час розігрішу. Точне встановлення зайнятих місць по силам учасників при застосуванні кругової системи є її позитивною стороною. Недоліком є те, що розігріш відбувається тривалий період часу (рис. 1.1) [1, 2].

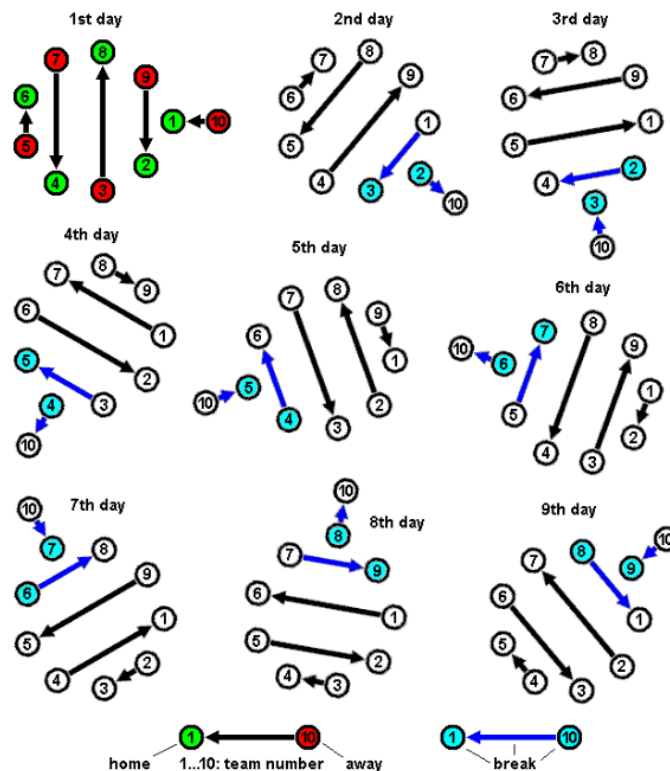


Рисунок 1.1 – Приклад турніру з 10 учасниками

Порядок зустрічей супротивників один з одним при круговій системі не має великого значення. Порядок зустрічей призначають так: привласнюють кожному з  $N$  учасників змагань порядкові номери, від 1 до  $N$ . Якщо число учасників парне, тобто  $N = 2K$ , то записують в правому стовпці зверху вниз номери від 1 до  $K$  зверху вниз, а в лівому стовпці номери від  $K + 1$  до  $N$  знизу вгору. Учасники, номери яких написані навпроти один одного, зустрічаються в першому турі. Номер 1 залишають на місці, а всі інші номери пересувають проти годинникової стрілки, щоразу на один крок. У зв'язку з останньою процедурою колова система проведення змагань, власне, і називається коловою. Якщо число учасників непарне, то додають номер  $X$ , і для отриманого набору номерів, кількість яких тепер парно, виконують вище зазначені процедури. Учасник, проти номера якого в будь-якому турі стоїть число  $X$ , в цьому турі вільний, не грає [3 — 6].

Схема календаря ігор при проведенні змагань за круговою системою на п'ять команд, буде наступною:

1-й день	2-й день	3-й день	4-й день	5й день
1 – X	1 – 5	1 – 4	1 – 3	1 – 2
2 – 5	4 – X	5 – 3	4 – 2	3 – X
3 – 4	2 – 3	2 – X	5 – X	4 – 5

Результат, досягнутий командою, тобто, перемога чи поразка, у змаганнях у спортивних іграх оцінюється очками. Тільки цифрові показники як об'єктивні дані можуть визначити переможця. Вони характеризують досягнення кожної команди, як по окремих зустрічах, так і по всьому розіграшу [7].

Виявлення переможця розіграшу визначається за допомогою підсумовування кількості очок, отриманих командою за відповідні результати кожної зустрічі. За загальною кількістю очок визначається місце, яке команда посіла у розіграші. Команда, що набрала найбільшу суму очок, отримує перше місце, наступна за нею – друге і т. д.

Підрахунок очок проводиться за такою системою: за перемогу команда-переможець отримує 3 очка, яка програла – 0 очок; при нічії обидві команди отримують по 1 очку. Систему підрахунків очок може змінюватись відносно виду спорту.

Записується підсумкова кількість очок, отриманих командою за весь час розіграшу, а на основі цього виводиться місце, зайняте командою у розіграші, і записується в останній графі. При запису кількості партій, виграних і програних командою, першою при виграші зустрічі пишеться цифра виграних, а другим – програних [8].

При однаковій кількості очок місця команд за умовою визначаються по зустрічі між супротивниками або ж різницею голів.

#### Переваги:

- максимальна справедливість розіграшу, так як усі зіграють з усіма;
- справедливо визначаються місця зайняті всіма учасниками турніру;
- навіть найслабший суперник має шанс зіграти проти найсильнішого;
- немає додаткових умов щодо кількості учасників (у швейцарській системі кількість учасників завжди має бути парною);

- система є стійкою до ситуації, коли один чи декілька учасників знімаються з розіграшу. Достатньо лише викреслити тих, що вибули, з турнірної таблиці та анулювати результати тих ігор, які вони уже зіграли. В результаті вийде, ніби вони взагалі не брали участі у змаганнях. В інших системах доводиться присуджувати технічні поразки, але це створює нерівні умови для учасників у подальших стадіях (коли зустрічається учасник, що відіграв важкий поєдинок у попередньому раунді, проти учасника, що відпочивав через неявку суперника).

#### Недоліки:

- необхідна велика кількість зустрічей, відповідно і часу на їх проведення. Це найтриваліша система розіграшів серед усіх інших. Кількість зустрічей росте квадратично до кількості учасників. На практиці максимально можливою кількістю учасників для кругової системи є 20-30 учасників (для 30 потрібно 29

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		11

турів, тобто майже місяць ігрового часу, якщо проводити по одній зустрічі щодня). Через цей недолік, у змаганнях з великою кількістю учасників кругова система майже не використовується;

– якщо протягом турніру один з учасників набирає найбільшу кількість очок і відривається від решти суперників, результат турніру стає передбачуваним через, що втрачається зацікавленість до змагань;

– щодо видовищності, якщо різниця в силі суперників занадто велика, турніри за круговою системою програють іншим турнірам з динамічнішими системами розіграшу. Багато зустрічей закінчуються з передбачуваним результатом;

– виникає проблема договірних зустрічей. Суперникам які досягли бажаного результату вигідніше домовитись про нічию в іграх між собою, ніж боротись за перемогу ризикуючи втратити важливі очки;

– одним з варіантів вирішення цього недоліку – заборона нічиїх;

– велика ймовірність виникнення однакових результатів у рівних за силою учасників. Для визначення переможця доводиться використовувати додаткові показники: кількість перемог, кількість перемог в особистих зустрічах, кількість очок в партіях; в шахах додаткові коефіцієнти (Бухгольца, Бергера) тощо [9].

Місця команд в турнірі визначаються за наступними показниками:

– найбільша кількість набраних балів;

– різниця забитих та пропущених м'ячів у всіх матчах турніру;

– кількість забитих голів у всіх матчах турніру.

Якщо дві чи більше команд будуть мати рівну кількість балів по трьох вказаних вище показниках, їх місця в таблиці будуть визначатись за наступним чином:

– найбільша кількість набраних балів в матчах турніру між цими командами;

– різниця забитих та пропущених м'ячів в матчах турніру між цими командами;

– найбільша кількість забитих голів в матчах турніру між цими командами;

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
						12
Змн.	Арк.	№ докум.	Підп.	Дата		

- найбільше число балів fair play(жовта картка = -1, непряма червона картка = -3, пряма червона картка = -4, жовта і пряма червона картка = -5, при цьому кожен учасник може отримати тільки один раз з вказаних штрафів);
- жеребкування, організоване організатором турніру.

## 1.2 Аналіз інформаційних сервісів підтримки спортивних змагань

Для порівняння було проаналізовано аналоги інформаційних сервісів спортивних змагань. Прямих аналогів, які б були подібними до розробленого веб-ервісу знайти важко. Та було переглянуто подібні сервіси, які пов'язані з проведенням турнірів.

Перший аналог є генератором матчів по круговій системі, та він спрямований більше для шахматних турнірів, оскільки є й інші функції для гри в шахмати(Калькулятор рейтингу шахматистів). На рис. 1.2 можна розглянути перший аналог веб-сервісу [10].

Рисунок 1.2 – Перший аналог веб-сервісу

Перевагами аналогу є:

- доступність(не потребує входу);
- простий інтерфейс;

- швидка побудова календаря ігор;
- можливість друку таблиці.

Недоліки:

- немає функції заповнення результатів;
- немає можливості перегляду результатів ігор;
- можливість проведення двох та більше кіл;
- застарілий дизайн.

Для складення календаря потрібно вказати назву турніру та прелік учасників через кому (рис. 1.3).

Рисунок 1.3 – Заповнені дані в форму аналога

Результат після відправлення даних зображено на рис. 1.4.

Тестовый турнир	
Таблица кругового турнира для печати	
Учасник3 — Учасник1	1 тур
Учасник4 — Учасник2	
Учасник5 — свободен	
Учасник1 — Учасник4	2 тур
Учасник5 — Учасник3	
Учасник2 — свободен	
Учасник4 — Учасник5	3 тур
Учасник2 — Учасник1	
Учасник3 — свободен	
Учасник5 — Учасник2	4 тур
Учасник3 — Учасник4	
Учасник1 — свободен	
Учасник2 — Учасник3	5 тур
Учасник1 — Учасник5	
Учасник4 — свободен	

Рисунок 1.4 – Результат після відправлення



Другий аналог створений для генерації турнірної сітки за різними системами та для різними видами турнірів(кібертурнірів, більярд, дартс). Та з його допомогою й можна створити календар ігор для спортивних турнірів. Сторінку для генерування турнірної сітки можна переглянути на рис. 1.5 [11].

Рисунок 1.5 – Другий аналог веб-сервісу

Перевагами аналогу є:

- реєстрація та авторизація;
- можливість створювати свої турніри та зберігати їх;
- багата функціональність;
- приємний інтерфейс;
- створення різних турнірних сіток для зареєстрованих та не зареєстрованих користувачів;
- заповнення результатів;
- утворення таблиць результатів;
- вибір дати проведення матчів;
- можливість друку календаря ігор та таблиці результатів.

Недоліки:

- створений більше для проведення кібертурнірів, ніж спортивних;
- є можливість змінити мову сайту, але немає української.

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		15

– можливість введення дробових та від’ємних чисел в результати ігор.

### **1.3 Вимоги до веб-сервісу, обґрунтування середовища розробки та постанова задачі**

Для організації проведення турнірів потрібні не тільки кошти та учасники, а ще й можливість запису результатів. При проведенні турнірів, їх організатори створюють сайти без автономних таблиць та алгоритмів, які б ці таблиці заповнювали. Багато часу втрачається на підрахування результатів, перебудування таблиці результатів та багато іншого, що б міг зробити штучний інтелект. Для економії часу та для пришвидшення організації спортивних турнірів, потрібно створити веб-сервіс, який частково замінить роботу організаторів та допоможе з підрахування результатів і побудовою турнірної сітки.

Веб-сервіс має бути зручним у користуванні (з простим зрозумілим інтерфейсом для кожного), швидким і повинен працювати коректно (без помилок та збоїв).

Серверна частина повинна коректно і надійно опрацьовувати запити адміністратора, зберігати сесії адміністраторів, і всю інформацію, що вводиться, бути стійкою до можливих атак з боку злоумисників. Також адміністративна панель повинна бути зручною у користуванні і мати приємний інтерфейс з вдало підбраною палітрою кольорів.

Головними вимогами до розробки даного проекту є:

- зручний інтерфейс адмін-панелі, який надасть можливість керувати системою;
- максимальна швидкість обробки даних, для забезпечення оперативної обробки вхідних і вихідних запитів;
- коректна робота для відображення чітких результатів результатів;
- забезпечення стабільної роботи системи.

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		16

Основні функціональні можливості – панель адміністратора, для додавання, редагування та видалення турнірів, матчів, учасників, а також результатів.

Створення або розробка сервісу починається зі складання технічного завдання, порядку наповнення web-вузла. На початку ставиться мета та тематика, з'ясовується, на яку аудиторію розраховувати.

Структура, дизайн в технічному завданні виділяється особисто. Тут вказуються підрозділи, категорії, пункти, переходи за посиланням. Проста і доступна навігація дозволить користувачу швидко орієнтуватися, знаходячи потрібну інформацію.

Наповнення вкладок web-розробки повинно бути грамотно відображеним, інформативним. Професійні тексти (SEO) гарантують якісне розкриття тематики статей. Чим більше корисних, цікавих, можливо, історично важливих або доведених складених даних, тим вищий рейтинг сервісу [12].

Враховуючи поставлені вище завдання, можна казати, що найбільш оптимальним засобом для розробки такого роду електронної комерції – HTML, CSS, JavaScript, PHP, MySQL та Bootstrap.

Необхідна розвинена гіпертекстова структура – HTML створена саме для цього. Необхідно забезпечити гнучку структуру управління матеріалом – управляти HTML-вмістом зручно за допомогою PHP + MySQL. Саме ця зв'язка дозволяє генерувати динамічні HTML-сторінки; необхідно реалізувати варіативність представлення матеріалу – в цьому випадку використання HTML+CSS більш ніж обгрунтовано. Саме ця комбінація засобів дозволяє створювати мультимедійні документи [13, 14]. Для взаємодії з користувачем використовувався JavaScript [15].

Для економії часу і на допомогу в роботі взято інструменти Bootstrap, які містять в собі HTML та CSS шаблони, оформлення та інші можливості які полегшать розробку сайту [16].

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		17

Для іконок використовувався сервіс Font Awesome – це шрифт з піктограмами, які можуть добавлятися до різних елементів веб-сторінки, щоб збільшити їх наочність та покращити дизайн [17].

Середовищем для розробки було обрано Brackets – текстовий редактор від компанії Adobe, призначений для редагування JavaScript, HTML, CSS, PHP. Серцевий код Brackets написаний з використанням веб-технологій (JavaScript, HTML CSS) і поширюється під ліцензією MIT [18].

Під час розробки серверною платформою було обрано Open Server.

Open Server – програмний комплекс, який має великий набір серверного програмного забезпечення який містить багатofункціональний та зручний інтерфейс. Дана платформа широко використовується з метою надання веб-сервісів в локальних мережах. Open Server дуже зручний у використанні, та має всі необхідні компоненти для розробки [19].

Для розробки було вибрано мову програмування PHP, оскільки метою даного проекту була доступність для всіх, а також зручність використання [20].

Вибір бази даних був одним із найважчих кроків створення системи, адже все будується кругом даних, які у ній зберігаються. Існує багато баз даних, але для даного проекту було обрано SQL [21].

Для адміністрування бази даних MySQL використовувався phpMyAdmin. Але більшість запитів відправляється із системи. Основною перевагою даної програми є те, що вона дозволяє керувати базами даних без вводу SQL команд через дружній інтерфейс і з будь якого комп'ютера без необхідності встановлення додаткового програмного забезпечення.

На основі проведеного аналізу необхідно розробити та забезпечити:

- 1) Зручний інтерфейс панелі адміністратора та інтерфейс користувача
- 2) Функціонал, для можливості створення та проведення турнірів
- 3) Покращення швидкодії при опрацюванні даних
- 4) Коректну роботу для відображення чітких результатів
- 5) Можливість редагування та видалення даних
- 6) Можливість під'єднання до сайтів

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		18

## 2 РОЗРОБКА СТРУКТУРНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-СЕРВІСУ

### 2.1 Загальна структура веб-сервісу

Для якісного функціонування веб-сервісу необхідно знати найоптимальніші інструментальні засоби для його розробки, реалізувати реляційну базу даних, створити інтерфейс сторінки зрозумілим, будь якому користувачу. Розробка загальної структури сайту починається побудови блок-схеми з переходами, які будуть реалізовані (рис. 2.1).

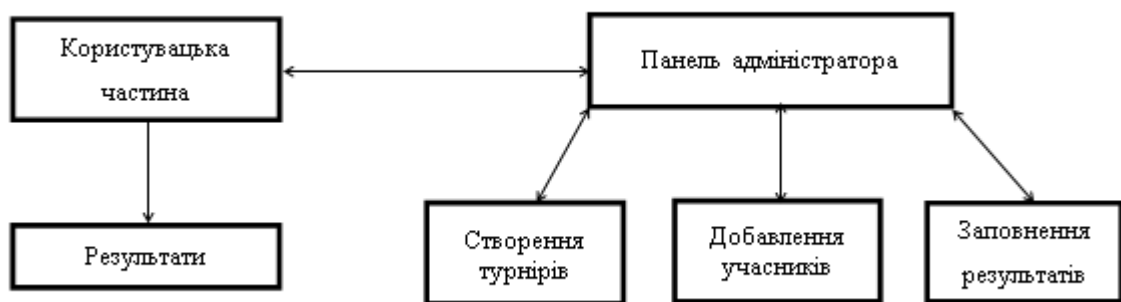


Рисунок 2.1 – Структурна схема сайту головної сторінки

В основній директорії розташована директорія “admin” яка містить папку з сторінками, доступні тільки для адміністратора, папку із зображеннями та інші файли потрібні для входу та адміністрування веб-сервісу. Також основна директорія містить файли для користувачької частини та стилі для всього сервісу, а ще файл з створенням таблиць бази даних. Структура файлів та папок виглядає так як зображено на рис. 2.2.

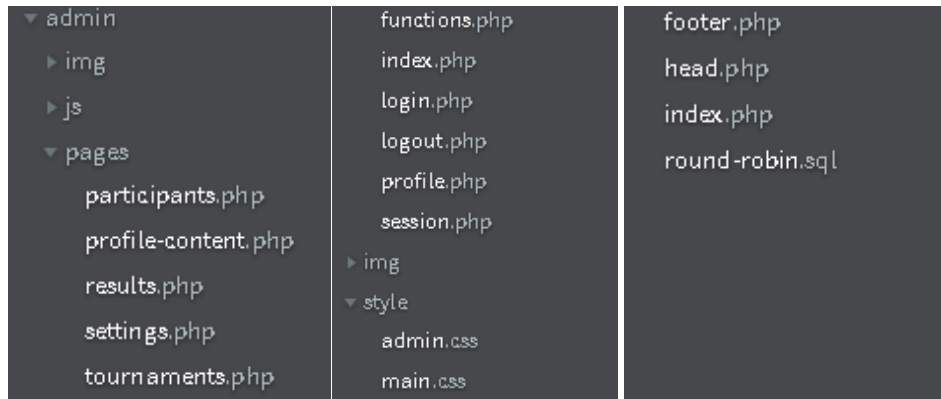


Рисунок 2.2 – Структура файлів та папок

Також основна директорія містить файли для користувацької частини та стилі для всього сервісу, а ще файл з створенням таблиць бази даних.

## 2.2 Структура бази даних

Ретельно продумана і спроектована база даних є запорукою успішної подальшої розробки якісного сервісу і правильної роботи програми.

База даних є основою для створення і розвитку всіх необхідних функцій веб-сайту. Також важливо розробити базу даних, яка буде захищена від можливих атак, збережені дані повинні бути добре зашифровані.

Основні функціональні можливості:

- авторизація;
- особистий кабінет зареєстрованого адміністратора;
- адміністративна панель, для додавання нових турнірів, учасників, результатів;
- можливість редагування та видалення даних;
- витягнення результатів з бази даних та їх відображення.

Для реалізації даних функцій необхідно створити таблиці бази даних, які повинні бути побудовані таким чином, щоб можна було з'єднати їх між собою і, за потреби, звернутись до будь-якої таблиці і отримати необхідні дані за допомогою спеціальних запитів. Для з'єднання таблиць між собою повинні бути

створені ключові поля. PhpMyAdmin ідеально підходить для створення бази даних, а також для подальшої роботи з ними.

База даних складається із 4 таблиць (рис. 2.3):

- admins(адміністратори);
- matches(матчі);
- participants(учасники);
- tournaments(турніри).

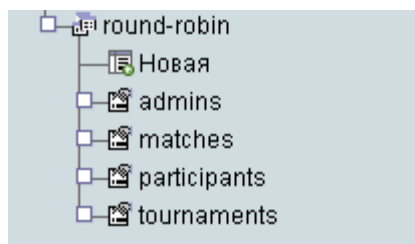


Рисунок 2.3 – Структура бази даних веб-сервісу

Кожна таблиця містить поле “id”, яке слугує ідентифікатором для кожної таблиці. Значенням поля “id” завжди є автоматично згенерованим унікальним ключем. Значення поля id ніколи не повторюється. Перша таблиця “admins” буде слугувати для збереження даних про зареєстрованих адміністраторів та для порівнянням з даними введеними користувачем при вході в панель адміністратора. Структура таблиці “admins” зображена на рис. 2.4.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	<b>id</b> 🔑	int(11)			Нет	Нет		AUTO_INCREMENT
2	<b>adminname</b>	varchar(255)	utf8_general_ci		Нет	Нет		
3	<b>password</b>	varchar(255)	utf8_general_ci		Нет	Нет		

Рисунок 2.4 – Структура таблиці “admins”

Кожен рядок таблиці з даними про адміністраторів буде складатись з таких полів:

- adminname – логін або ім’я адміністратора;

– password – пароль, який шифрується за допомогою алгоритму md5 (рис. 2.5).

id	adminname	password
1	kolyalylyk	5d65ccee830d4671c4a46808033ad968

Рисунок 2.5 – Приклад шифрування пароля

Наступна таблиця “matches” створена для зберігання даних про матчі та їх результати. Структура даної таблиці зображена на рис. 2.6.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🗝️	int(11)			Нет	Нет		AUTO_INCREMENT
2	tournament_id 🗝️	int(11)			Нет	Нет		
3	f_participant_id 🗝️	int(11)			Да	NULL		
4	f_participant_score	int(3)			Да	NULL		
5	s_participant_id 🗝️	int(11)			Да	NULL		
6	s_participant_score	int(3)			Да	NULL		

Рисунок 2.6 – Структура таблиці “matches”

Кожен рядок таблиці з даними матчів буде складатись з наступних полів:

- tournament\_id – ідентифікатор турніру;
- f\_participant\_id – ідентифікатор першого учасника;
- f\_participant\_score – результат першого учасника;
- s\_participant\_id – ідентифікатор другого учасника;
- s\_participant\_score – результат другого учасника.

Структуру таблички “participants” подано на рис. 2.7. Вона створена для зберігання учасників та даних про них:

- tournament id – ідентифікатор турніру;
- name – ім’я учасника;
- games – кількість ігор;
- victories – кількість вигравів;
- draws – кількість нічиїх;
- defeats – кількість програшів;



- scored\_goals – забиті голи, шайби і т.д;
- missed\_goals – пропущені голи, шайби і т.д;
- points – бали.

Таблиця результатів утворюється відносно заповнених даних. В створену таблицю розставляють по місцях відносно критеріїв вказаних в теоретичній частині. В наступній версії веб-сервісу заплановано створити ще одне поле для таблиці “participants”, яка буде зберігати логотип команди.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🗄	int(11)			Нет	Нет		AUTO_INCREMENT
2	tournament_id 🗄	int(11)			Нет	Нет		
3	name	varchar(255)	utf8_general_ci		Нет	Нет		
4	games	tinyint(4)			Нет	0		
5	victories	tinyint(4)			Нет	0		
6	draws	tinyint(4)			Нет	0		
7	defeats	tinyint(4)			Нет	0		
8	scored_goals	smallint(6)			Нет	0		
9	missed_goals	smallint(6)			Нет	0		
10	points	smallint(6)			Нет	0		

Рисунок 2.7 – Структура таблиці “participants”

Наступна таблиця створена для збереження турнірів,. Дана таблиця називається “tournaments”.

Структурний вигляд рядків в цій таблиці зображено на рис. 2.8.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🗄	int(11)			Нет	Нет		AUTO_INCREMENT
2	tournament_name	varchar(100)	utf8_general_ci		Нет	Нет		
3	teams	tinyint(2)			Нет	Нет		
4	status	varchar(20)	utf8_general_ci		Нет	Нет		
5	admin_id 🗄	int(11)			Нет	Нет		

Рисунок 2.8 – Структура таблиці “tournaments”

Таблиця складається з 5 полів.

- id – обов’язкове унікальне поле, яке є увсіх вище описаних таблицях;
- tournament\_name – назва турніру, яка буде відображатися користувачам при виборі турніру;

- teams – кількість команд які беруть участь в турнірі;
- status – статус турніру, який може бути активним та не активним.

При створенні бази даних використовуються окремі таблиці для різних категорій, наприклад: турніри, учасники, матчі. Але також потрібно мати зв'язки між цими таблицями. Наприклад, адміністратор створює новий турнір, а в турнірі беруть участь учасники у яких багато даних. Ці відносини повинні бути представлені в базі даних. Крім того, при завантаженні даних з SQL потрібно використовувати певні типи запитів JOIN щоб отримати те, що потрібно.

Існує кілька типів зв'язків із базою даних:

- один до одного;
- один до багатьох;
- багато до багатьох.

На рис. 2.9 можна побачити зв'язок адміністраторів та турнірів, турніри мають зв'язки з матчами і учасниками, матчі з'єднуються з учасниками. Тобто один адміністратор може мати багато турнірів[22].

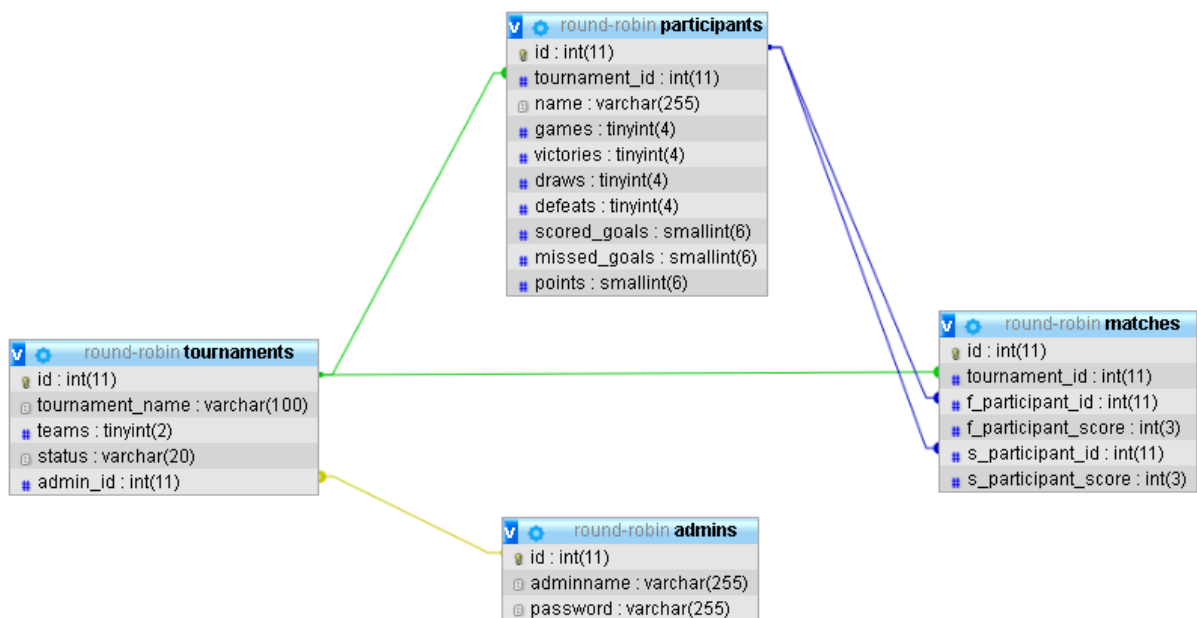


Рисунок 2.9 – Зв'язки таблиць бази даних

Турнір в свою чергу може мати багато учасників. Таблиця “matches” є зв’язною таблицею. Це полегшує створення матчів та їх відображення в користувацькій частині.

Всі зв’язки створені з допомогою унікального ключа та індекса. Наприклад, у таблиці “admins” унікальним ключем є поле “id”, воно з’єднується з таблицю “tournaments” по полю “admin\_id”, який є індексом.

### 2.3 Структура панелі адміністратора та інтерфесу користувача

Панель адміністратора – це “серце” сайту. Із неї адміністратор проводить усі дії: створює турніри, додає учасників, заповнює та відправляє результати. В цілому панель адміністратора заключає у собі всі настройки і елементи, які забезпечують стабільну роботу сервісу.

Потрібно проаналізувати всі функції адміністратора та розмістити їх таким чином, щоб кожен мав можливість зрозуміти сервіс, та використовувати його для проведення турнірів. Структура панелі адміністратора представлено на рис. 2.10.

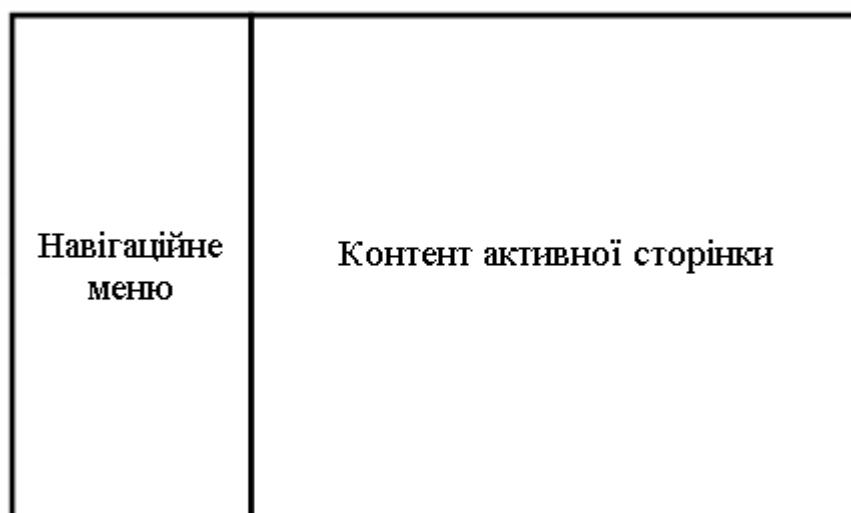


Рисунок 2.10 – Структура панелі адміністратора

Розділимо її на окремі блоки. Щоб адміністратор міг швидко переключатись між процесами та точно заповнювати дані, створено навігаційну панель. Вона ділиться на “шапку” панелі та пункти для переходу між сторінками.

“Шапка” навігаційної панелі містить:

- привітання (Привіт, “adminname”);
- перейти на сайт (Посилання на сторінку користувача).

Пункти для переходу між сторінками:

- профіль – сторінка профілю;
- турніри – сторінка для створення редагування та видалення турнірів;
- учасники – сторінка для створення редагування та видалення учасників;
- результати – сторінка, де заповнюються результати;
- налаштування – сторінка для налаштувань;
- вихід – вихід із панелі адміністратора.

Розглянемо структуру та роль сторінок у веб-сервісі.

Сторінка профілю тимчасово містить логотип сервісу. Її створено для майбутньої реалізації більш функціональнішої системи, ніж вона зараз. Планується додати аватар адміністратора та інформацію про список турнірів які він веде і т.д.

Сторінка “Турніри” складається з списку турнірів, які обслуговує адміністратор у вигляді таблиці. Біля кожного турніру знаходяться функціональні кнопки. Кнопка “Додати турнір” відкриває модальне вікно з формою.

Для сторінки учасників, крім головної навігації, відкривається додаткова навігація зі списком турнірів. Після вибору турніра відображається табличка з учасниками та кнопка, яка відкриває модальне вікно з формою.

Щоб заповнити результати на сторінці “Результати”, відкривається додаткова навігація, так як і в таблиці учасників, тільки замість таблиці з учасниками відображаються турнірні стадії (тури) з матчами та кнопкою відправлення результатів.

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		26

Так як і сторінка профілю, сторінка з налаштуваннями буде оновлюватись, будуть добавлятись нові функції, такі як: зміна кольорів адміністративної панелі, багато різних елементів, які зможуть контролювати поведження елементів на сайті. На даний момент сторінка “Налаштування” містить табличку з адміністраторами та кнопки “Добавити адміністратора” та кнопка видалення для кожного адміністратора.

Структура користувацької частини зображена на рис. 2.11.

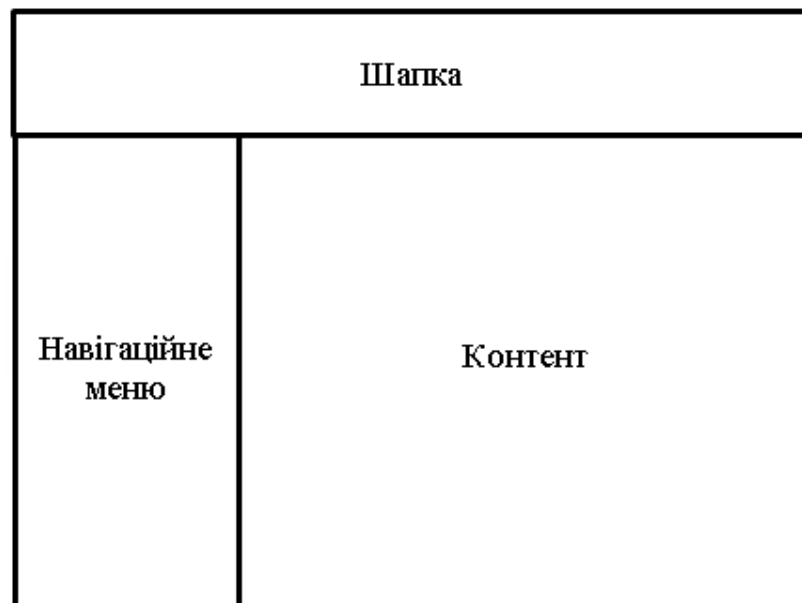


Рисунок 2.11 – Структура користувацької частини

Користувацька частина складається з шапки, навігаційного меню та контенту. У шапці розташований логотип та кнопка входу в панель адміністратора. Навігаційне меню складається з списку турнірів, натиснувши на один із яких в контенті буде відображатися турнірна таблиця та результати матчів.

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-СЕРВІСУ ЗАСОБАМИ PHP, MYSQL, JS

### 3.1 Розробка інтерфейсу користувача

Вибирати колір і співвідношення кольорів досить важко. На перший погляд це здається простою звичайною справою, але це не так, підібрати колір для дизайну сайту або сервісу достатньо тяжке завдання. Найтяжчою задачею вважається вибір правильної кольорової схеми, яка несе відповідальність за сприйняття, і створює те, що робить їх унікальними. Вчені підкреслюють такий факт, що кольорова палітра може суттєво впливати на зорове сприйняття.

При виборі колірної схеми для сайту потрібно відштовхуватися не від власних смаків, а від того, як обрані кольори вплинуть на сприйняття сервісу і які емоції можуть викликати у користувачів. Як правило, у якісному дизайні не використовують більше трьох основних кольорів. Для акцентування уваги на тих чи інших речах можна застосовувати різні їхні відтінки, але тут діє правило – чим менше, тим краще.

В інтернеті є багато сервісів для автоматично генерування палітри кольорів. Для сервісу було підібрано палітру кольорів за допомогою генератора палітр кольорів [coolors.co](http://coolors.co)[23]. Палітру кольорів можна розглянути на рис. 3.1.



Рисунок 3.1 – Палітра кольорів для веб-сервісу

Кожна деталь у дизайні має велике значення, і нікому заздалегідь не може бути відомо, що стане об'єктом пильної уваги з боку потенційного користувача. Саме тому необхідно професійно підходити до опрацювання усіх питань,

					ДР.Шс – 11.00.000 ПЗ	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		28

пов'язаних із дизайном. Розуміння того, що колір – важлива складова успішного проекту, допоможе прийняти правильні рішення.

### 3.2 Вхід в панель адміністратора

Адміністративна панель є важливою складовою кожного функціонального веб-сервісу. Вона полегшує рутинну роботу з кодом. Для людей, які не вміють писати код, це легкий шлях до змін веб-сервісу. Але без захисту адміністративна панель не буде такою корисною, адже всі зможуть заходити і робити все, що їм хочеться. Тому для того, щоб таке не сталося розроблений вхід, який підтверджує, те що користувач дійсно являється адміністратором.

Увійти до форми входу в панель адміністратора можна двома шляхами: в користувацькій частині натиснути кнопку “Увійти” або ж дописати до адреси сайту “/admin/” щоб повна адреса виглядала так “/ground-robin/admin/”. Після загрузки сторінки буде запропонована форма з полями для входу(рис. 3.2).

Рисунок 3.2 – Форма авторизації

Вибране поле збільшується та міняє колір тексту та границю знизу для того, щоб зрозуміти яке поле заповнюється. При знятті виділення, кольори повертаються (рис. 3.3).

					ДР.Шс – 11.00.000 ПЗ	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		29

Рисунок 3.3 – Введене та вибране поля для вводу

При невірному вводі логіна або пароля ,або пароль та логін не співпадають під кнопкою відправки форми появляється відповідна помилка, а поля очищаються (рис. 3.4).

Рисунок 3.4 – Помилка відправлення форми з невірними даними

Якщо відправлено пусту форму, тоді поля очищаються і з'являється помилка “Введіть логін та пароль”, яку зображено на рис. 3.5.



Рисунок 3.5 – Помилка відправлення пустої форми

Перевірка форми здійснюється за допомогою файла login.php. Код реалізації можна побачити у Додатку А :

Код перевіряє введені дані з форми авторизації та виконує перевірку на правильність вводу. До бази даних відправляється запит та перевіряються всі дані зареєстрованих адміністраторів. Якщо знайдено збіги – запускається сесія, код якої записаний у файлі session.php[24].

```

$connection = mysql_connect("localhost", "root", "");
$db = mysql_select_db("round-robin", $connection);
session_start();
$login_session = $_SESSION['logged_admin'];
if(!isset($login_session)){
mysql_close($connection);
header('Location: index.php');
}

```

Веб-сервер не підтримує постійного зв'язку з клієнтом, і кожний запит обробляється як новий, без зв'язку з попередніми.

Тобто, неможливо відслідкувати запити від одного і того ж відвідувача, ні зберегти для нього змінні між переглядом окремих сторінок. Для вирішення цієї проблеми створені сесії.

Сесії – це механізм, який дозволяє однозначно ідентифікувати браузер і створений для цього браузера файл на сервері, в якому зберігаються змінні сеанса.

У файлі login.php у сесію записується ім'я адміністратора:

```
$_SESSION['logged_admin']=$adminname;
```

А у файлі session.php ім'я записується вже в змінну:

```
$login_session = $_SESSION['logged_admin'];
```

Після всіх цих операцій адміністратора перекидає на профіль у панелі адміністратора, яка має вигляд як на рис. 3.6.



Рисунок 3.6 – Вигляд панелі адміністратора при вході

Як можна побачити на рис. 3.6, все дуже зручно. З лівого боку навігаційне меню, справа контент з інформацією.

### 3.3 Функції панелі адміністратора

Для переходу між сторінками використовується навігаційне меню. Воно складається з шапки та навігаційних пунктів (рис. 3.7).

					ДР.Шс – 11.00.000 ПЗ	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		32

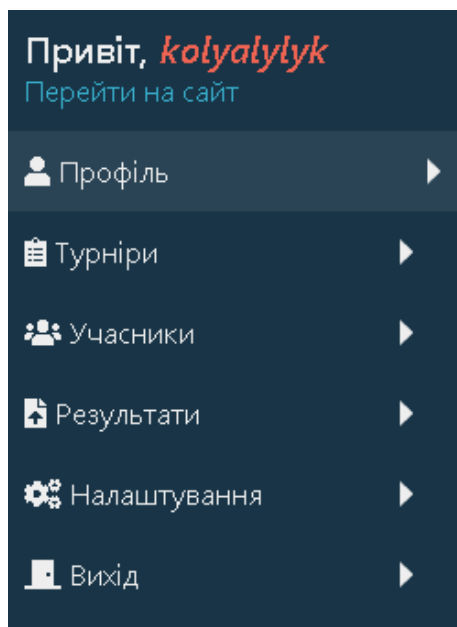


Рисунок 3.7 – Вигляд навігаційного меню

Вибравши один із пунктів навігаційного меню підгружається контент відповідно до вибраного пункта. Тобто загружаються не окремі сторінки, а до сторінки профілю підеднуються блоки з інформацією.

Вибравши пункт “Турніри” з навігаційного меню, підгружається сторінка, яка дає можливість переглянути список з турнірами, створювати нові турніри, редагувати їх, а також видаляти. Контент сторінки виглядає наступним чином(рис. 3.8).

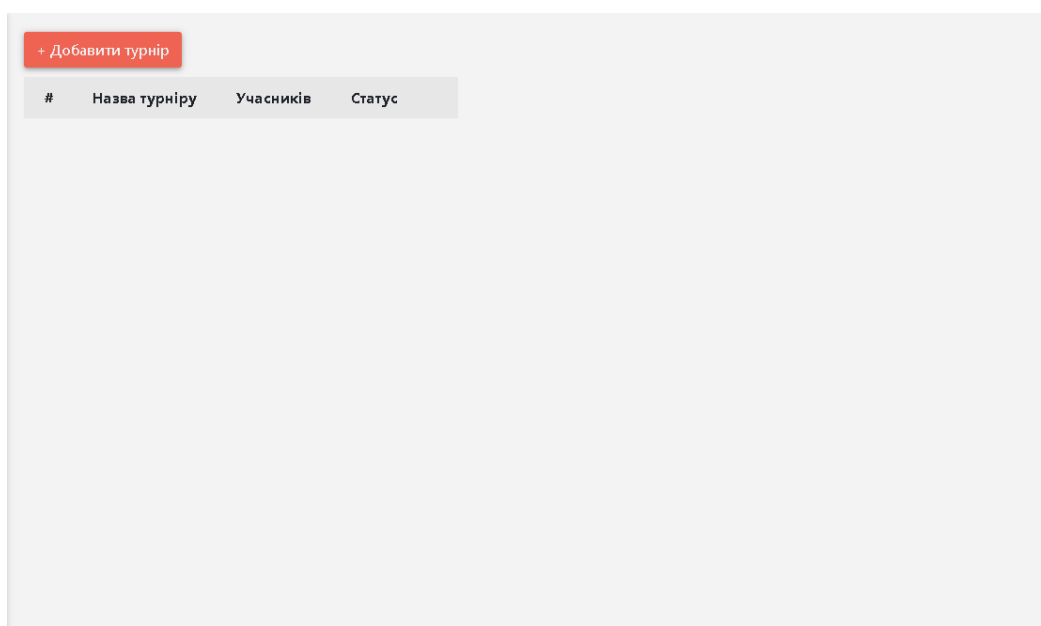


Рисунок 3.8 – Контент сторінки з турнірами

На рис. 3.8 подано таблицю з турнірами (на рис. 3.8 таблиця пуста) та навігаційну кнопку “Добавити турнір”. При натисненні на кнопку “Добавити турнір” з’являється модальне вікно з полями для заповнень (рис. 3.9), адміністратору потрібно ввести: назву турніру, кількість учасників та його статус[25].

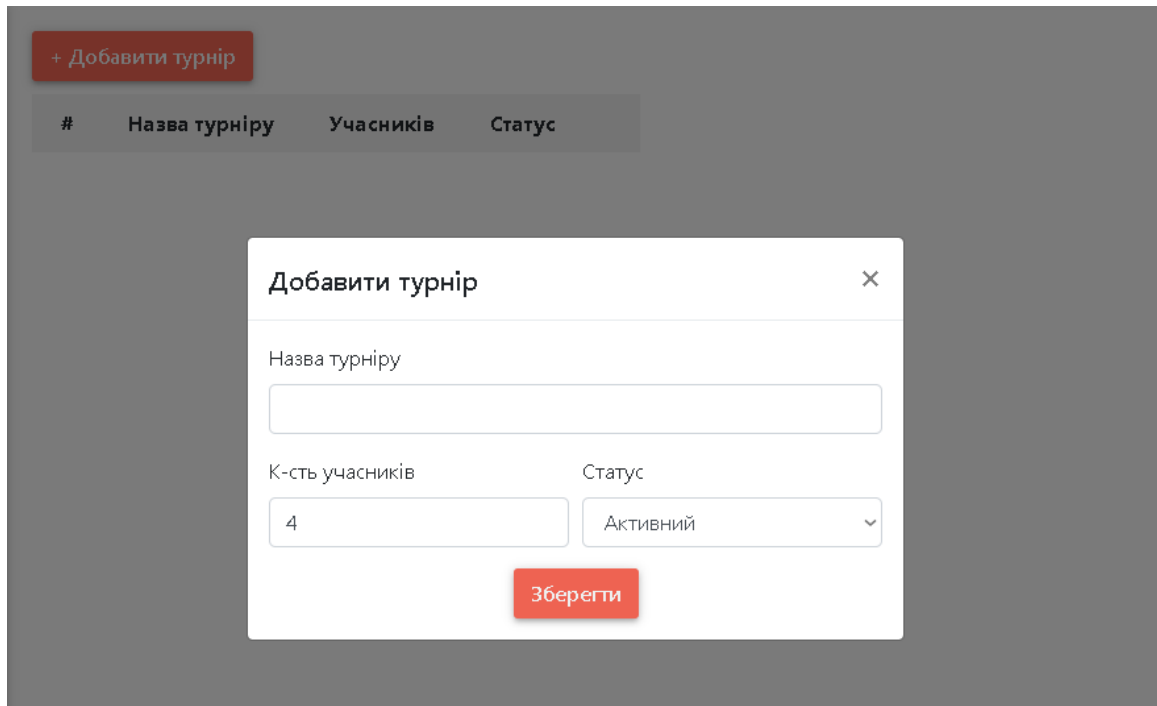


Рисунок 3.9 – Модальне вікно з формою

Більшість функцій знаходяться у файлі functions.php. Код функції додавання турніру:

```
if (isset($_POST['save-tournament'])) {
    header('Location: '.$pageUrl.'');
    $tournamentName = $_POST['tournament-name'];
    $teamsQuantity = $_POST['teams-quantity'];
    $tournamentStatus = $_POST['tournament-status'];
    mysql_query("INSERT INTO `tournaments`(`id`, `tournament_name`, `teams`,
    `status`, `admin_id`) VALUES ('', '{$tournamentName}', '{$teamsQuantity}',
    '{$tournamentStatus}', '{$adminId}')");
}
```

Після відправки форми у таблиці з’явиться турнір з назвою, яку ви вказали у формі, з кількістю учасників статус турніру та навігаційними кнопками: редагувати, зберегти, видалити. Для наглядності створено кілька турнірів з різною кількістю учасників(рис. 3.10).

**+ Додати турнір**









#	Назва турніру	Учасників	Статус			
1	Ліга Чемпіонів	16	Активний			
2	Українська Прем'єр Ліга	9	Активний			
3	Відбір Євро 2020 - Група F	6	Активний			

Рисунок 3.10 – Список із створеними турнірами

Натиснувши кнопку “Редагувати” (іконка з синім олівцем) включається режим редагування(рис. 3.11). Ця функція реалізується JS скриптом, його можна буде переглянути у Додатку А.











#	Назва турніру	Учасників	Статус			
1	<input type="text" value="Ліга Чемпіонів"/>	<input type="text" value="16"/>	Активний 			
2	Українська Прем'єр Ліга	9	Активний Не активний			
3	Відбір Євро 2020 - Група F	6	Активний			

Рисунок 3.11 – Режим редагування турніра

Код функції “Зберегти” (іконка з зеленою дискетою) та код функції видалення турніру представлено в Додатку А, файл functions.php:

Видалити турнір не просто. Спочатку потрібно видалити всі матчі цього турніру (це робиться автоматично запитом).

```
mysql_query("DELETE FROM `matches` WHERE `tournament_id`='{ $id}'");
```

Після видалення матчів, потрібно видалити всіх учасників цього турніру (це можна зробити автоматично), код реалізації нижче.

```
mysql_query("DELETE FROM `participants` WHERE `tournament_id`='{ $id}'");
```

Після попередніх операцій наступний запит видаляє сам турнір.

```
mysql_query("DELETE FROM `tournaments` WHERE `id`='{ $id}'");
```

Після створення турніру, до нього потрібно додати учасників. Тому потрібно перейти на вкладку з учасниками (рис. 3.12).

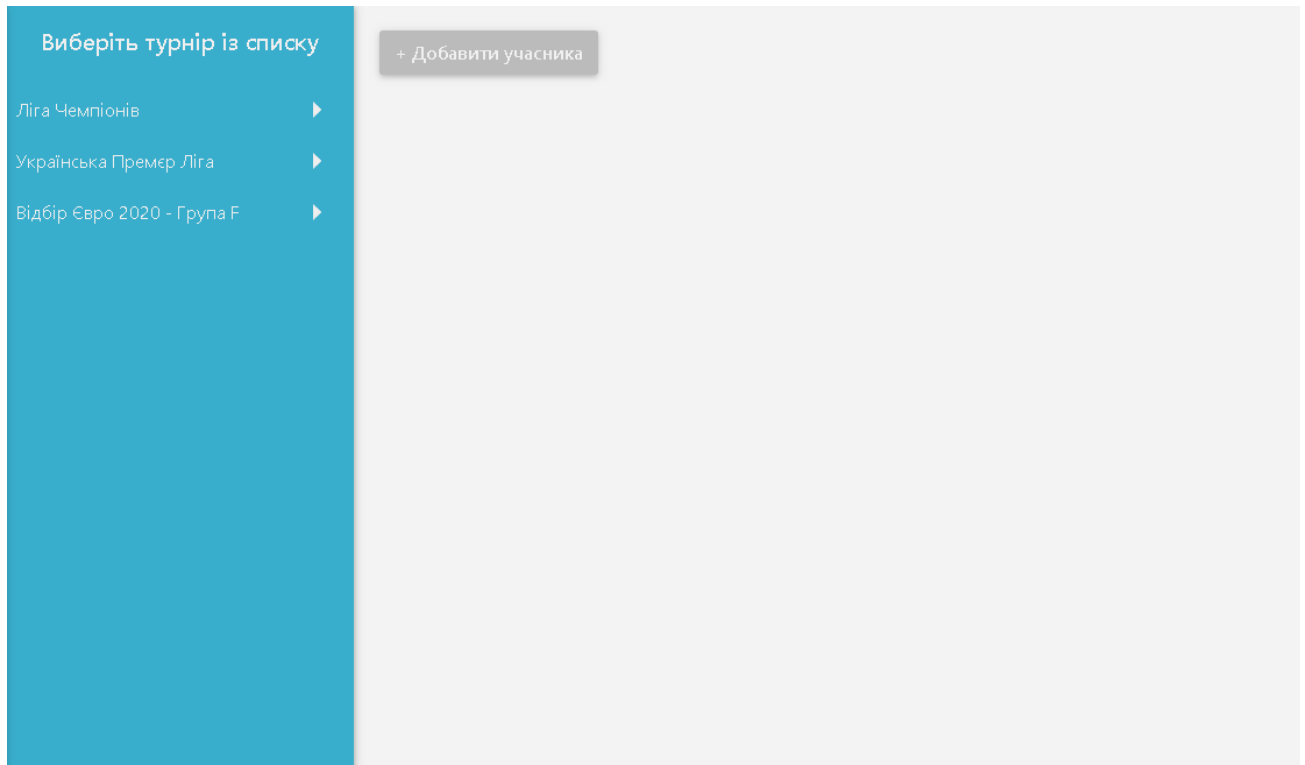


Рисунок 3.12 – Контент сторінки учасників

При виборі пункту “Учасники” з навігаційного меню, з’являється додаткове меню для вибору турніру в який потрібно додати учасників та заблокована кнопка “Додати учасника”. Вибравши турнір адміністраторові відкриється список учасників (який звісно що пустий), а кнопка розблоковується (рис. 3.13).

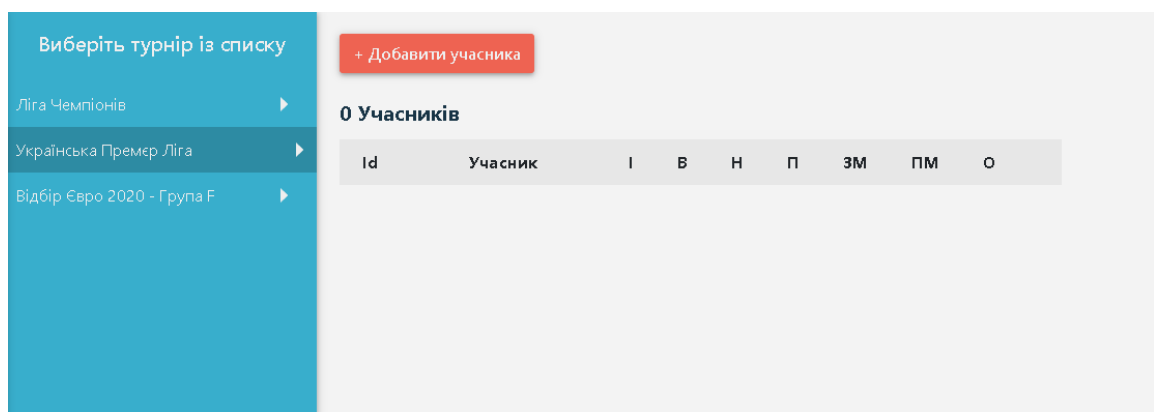


Рисунок 3.13 – Контент вибраного турніру

Як видно на рис. 3.13 активний пункт меню відрізняється від інших пунктів. Також бачимо над табличкою кількість добавлених учасників. Натиснувши кнопку “Добавити учасника” відкривається модальне вікно з полем для вводу імені учасника(рис. 3.14).

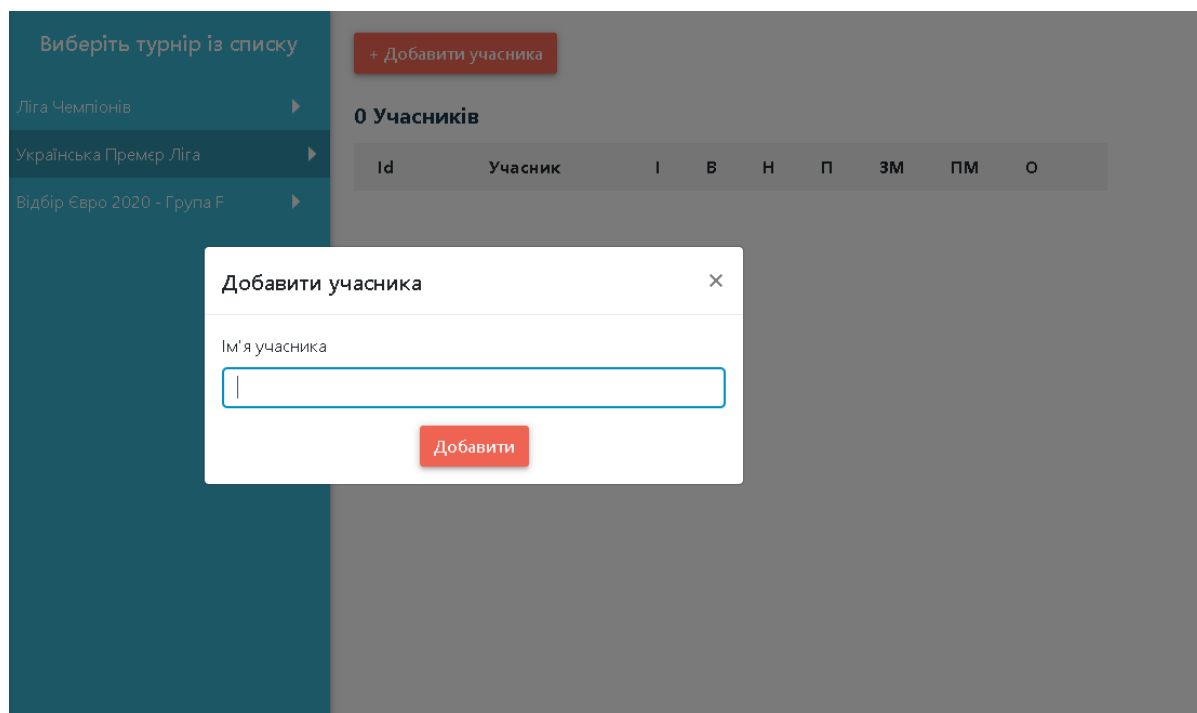


Рисунок 3.14 – Модальне вікно для добавлення учасника

Кількість учасників вказано у вкладці “Турніри”, тому при добавленій вказаній кількості учасників, кнопка “Добавити учасника” блокується, код реалізації в Додатку А. Над табличкою пише кількість учасників. Над табличкою пише кількість учасників. Вигляд таблиці після добавлення учасників на рис. 3.15

+ Додати учасника

**9 Учасників**

Id	Учасник	І	В	Н	П	ЗМ	ПМ	О			
1	Шахтар	0	0	0	0	0	0	0			
2	Динамо	0	0	0	0	0	0	0			
3	Олександрія	0	0	0	0	0	0	0			
4	Маріуполь	0	0	0	0	0	0	0			
5	Зоря	0	0	0	0	0	0	0			
6	Львів	0	0	0	0	0	0	0			
7	Ворскла	0	0	0	0	0	0	0			
8	Десна	0	0	0	0	0	0	0			
9	Олімпік	0	0	0	0	0	0	0			

Рисунок 3.15 – Вигляд таблиці після додавання учасників

Натиснувши кнопку реагувати, відкривається можливість редагувати назву команди (рис. 3.16).

**9 Учасників**

Id	Учасник	І	В	Н	П	ЗМ	ПМ	О			
1	<input type="text" value="Шахтар"/>	0	0	0	0	0	0	0			

Рисунок 3.16 – Режим редагування учасника

Функція редагування виконується JS скриптом.

```
function editParticipant(elem) {
var line = elem.parentElement;
var tdList = line.childNodes;
elem.removeAttribute("onclick");
tdList[2].innerHTML = '<input style="width: 110px" type="text" value="" +
tdList[2].textContent + '" name="updatedParticipantName' +
tdList[1].textContent + '" form="updateParticipantForm' +
tdList[1].textContent + '" autocomplete="off"/>';
tdList[11].childNodes[0].disabled = false;
}
```



Код збереження учасника та код видалення учасника знаходиться у Додатку А, файл functions.php

Видалення учасників аналогічне видаленню турнірів. Спочатку потрібно видалити всі матчі турніру в якому бере участь команда яку видаляють, це робиться для перебудови матчів (це робиться автоматично).

```
mysql_query("DELETE FROM `matches` WHERE `tournament_id`  
='{ $tournamentId }'");
```

Після запиту який видаляє матчі слідує запит видалення учасника

```
mysql_query("DELETE FROM `participants` WHERE `id`='{ $id }'");
```

Після стадії додавання учасників іде стадія генерування матчів. Як відомо потрібно розробити алгоритм, який буде створювати матчі так, щоб кожен учасник зіграв з іншими. Спочатку було розроблену схему цього алгоритму.

Алгоритм кругової системи можна зробити по різному. Можна спочатку згенерувати всі можливі матчі та вибирати з них по одному матчеві щоб всі зіграли, а можна так як зробив автор проекту.

З бази даних витягаються учасники вибраного турніру та записуються у масив. Другим кроком учасники діляться на два рівні масиви.

Якщо у турнірі парна кількість учасників, то получится два рівні масиви з якими можна робити подальші дії. Принцип дії алгоритма такий:

- перший елемент першого масива фіксується;
- з першого масива забирається останній елемент та добавляється в кінець другого масива;
- з другого масиву забирається перший елемент масива та добавляється після фіксованого елемента;
- останнім кроком записується матч, для цього по одному елементу записуються супротивники.

При непарній кількості команд до меншого масива добавляється “пустий учасник”, щоб спрацьовував той самий алгоритм що і з парною кількістю учасниками. На рис. 3.17 можна побачити схематичну дію алгоритма.

Тип 1.(1 грає з 14, 2 грає з 13, ...)

1	2	3	4	5	6	7
14	13	12	11	10	9	8

Тип 2.(1 грає з 13, 14 грає з 12, ...)

1	14	2	3	4	5	6
13	12	11	10	9	8	7

Тип 2.(1 грає з 12, 13 грає з 11, ...)

1	13	14	2	3	4	5
12	11	10	9	8	7	6

Алгоритм продовжується поки не вернеться в початкову позицію

Тип 13.(1 грає з 2, 3 грає з 14, ...)

1	3	4	5	6	7	8
2	14	13	12	11	10	9

Рисунок 3.17 – Схематична дія алгоритма

Код реалізації генерування та запису матчів в базу даних при парній та непарній кількості команд наведений нижче.

```

$tournamentId = $_GET['tournament'];
if($_GET['tournament']){
    $participantsQuery = mysql_query("SELECT * FROM `participants` WHERE
    `tournament_id` = $tournamentId", $connection);
    $resultMatches = mysql_query("SELECT * FROM `matches` WHERE
    `tournament_id`= $tournamentId", $connection);
    while($row = mysql_fetch_assoc($resultMatches)){
        if(isset($_POST['send-results'])){
            header('Location: '.$pageUrl.'');
            if($_POST['f-participant-score'].$row['id']] != '' && $_POST['s-participant-
            score'].$row['id']] != ''){
                mysql_query("UPDATE `matches` SET `f_participant_score`= {$_POST['f-
                participant-score'].$row['id']}, `s_participant_score`= {$_POST['s-
                participant-score'].$row['id']} WHERE `id` = {$_POST['f-participant-
                score'].$row['id']}");
                mysql_query("UPDATE `participants` SET `games`=games+1,
                `scored_goals`=scored_goals+{$_POST['f-participant-score'].$row['id']},
                `missed_goals`=missed_goals+{$_POST['s-participant-score'].$row['id']}
                WHERE `id` = {$_POST['f-participant-score'].$row['id']}");
                mysql_query("UPDATE `participants` SET `games`=games+1,
                `scored_goals`=scored_goals+{$_POST['s-participant-score'].$row['id']},
                `missed_goals`=missed_goals+{$_POST['f-participant-score'].$row['id']}
                WHERE `id` = {$_POST['s-participant-score'].$row['id']}");
                if ($_POST['f-participant-score'].$row['id']] > $_POST['s-participant-
                score'].$row['id']) {
                    mysql_query("UPDATE `participants` SET `victories`=victories+1,
                    `points`=points+3 WHERE `id` = {$_POST['f-participant-
                    score'].$row['id']}");
                    mysql_query("UPDATE `participants` SET `defeats`=defeats+1 WHERE `id` =
                    {$_POST['s-participant-score'].$row['id']}");
                }
            }
        }
    }
}

```

```

}else if ($_POST['f-participant-score'].$row['id']] < $_POST['s-participant-
score'].$row['id'])) {
mysql_query("UPDATE `participants` SET `victories`=victories+1,
`points`=points+3 WHERE `id` = {$row['s_participant_id']}");
mysql_query("UPDATE `participants` SET `defeats`=defeats+1 WHERE `id` =
{$row['f_participant_id']}");
}else if ($_POST['f-participant-score'].$row['id']] == $_POST['s-
participant-score'].$row['id'])) {
mysql_query("UPDATE `participants` SET `draws`=draws+1, `points`=points+1
WHERE `id` = {$row['f_participant_id']}");
mysql_query("UPDATE `participants` SET `draws`=draws+1, `points`=points+1
WHERE `id` = {$row['s_participant_id']}");
}}}}

```

Оскільки перший елемент фіксується, він завжди буде першим, щоб цього уникнути, кожного парного туру, матчі відзеркалюються. Після виконання коду в таблиці бази даних “matches” появляються матчі (рис. 3.18).

	id	tournament_id	f_participant_id	f_participant_score	s_participant_id	s_participant_score
<input type="checkbox"/>	1	2	1	NULL	5	NULL
<input type="checkbox"/>	2	2	2	NULL	6	NULL
<input type="checkbox"/>	3	2	3	NULL	7	NULL
<input type="checkbox"/>	4	2	4	NULL	8	NULL
<input type="checkbox"/>	5	2	9	NULL	NULL	NULL
<input type="checkbox"/>	6	2	6	NULL	1	NULL
<input type="checkbox"/>	7	2	7	NULL	5	NULL
<input type="checkbox"/>	8	2	8	NULL	2	NULL
<input type="checkbox"/>	9	2	NULL	NULL	3	NULL
<input type="checkbox"/>	10	2	9	NULL	4	NULL
<input type="checkbox"/>	11	2	1	NULL	7	NULL
<input type="checkbox"/>	12	2	6	NULL	8	NULL
<input type="checkbox"/>	13	2	5	NULL	NULL	NULL
<input type="checkbox"/>	14	2	2	NULL	9	NULL
<input type="checkbox"/>	15	2	3	NULL	4	NULL
<input type="checkbox"/>	16	2	8	NULL	1	NULL
<input type="checkbox"/>	17	2	NULL	NULL	7	NULL
<input type="checkbox"/>	18	2	9	NULL	6	NULL
<input type="checkbox"/>	19	2	4	NULL	5	NULL
<input type="checkbox"/>	20	2	3	NULL	2	NULL
<input type="checkbox"/>	21	2	1	NULL	NULL	NULL
<input type="checkbox"/>	22	2	8	NULL	9	NULL
<input type="checkbox"/>	23	2	7	NULL	4	NULL
<input type="checkbox"/>	24	2	6	NULL	3	NULL
<input type="checkbox"/>	25	2	5	NULL	2	NULL
<input type="checkbox"/>	26	2	9	NULL	1	NULL
<input type="checkbox"/>	27	2	4	NULL	NULL	NULL

Рисунок 3.18 – Табличка “matches” після виконання алгоритма

Всього згенерувало 45 матчів для одного кола. Як можна побачити на рис. 3.18 в полях “f\_participant\_id” та “s\_participant\_id” стоять ідентифікатори учасників. Навівши на ідентифікатор в таблиці можна переглянути що це за учасник. Та можна побачити що в деяких полях замість цифри стоїть NULL. Це

не баг, просто в згенерованому турнірі дев'ять учасників (непарна кількість учасників), це і є “пустий учасник” з яким би мав зіграти інший суперник. В такому випадку учасник просто пропускає тур (протягом кола всі команди пропускають тур). Матч записується, але результат в нього заповнити не можна.

Щоб алгоритм спрацював двічі (два кола), потрібно тільки відзеркалити згенеровані матчі. Щоб супротивник міг зіграти вдома та в гостях.

Згенерувавши матчі, після їх проведення заповнюються результати. Це можна зробити на сторінці “Результати”. Для цього в додатковому меню, яке відкриється після вибору в навігаціоному меню пункту “Результати”, вибрати турнір. Перед цим потрібно додати всіх учасників. Один тур – це одна форма для відправки, де в кожному матчеві для кожного учасника є своє поле для введення результату. Матч де є “пустий учасник” має заблоковані поля, але він одно записується у тур, щоб наглядно бачити яка команда і в якому турі залишилася без пари. В усіх полях є атрибут “placeholder” з значенням “0”, щоб адміністратор знав куди заповнювати рахунок. На рис. 3.19 зображено сторінку з турами та форми з полями для введення результатів, а також для кожного туру виведена кнопка відправлення форми з результатами “Відправити результат”.

Виберіть турнір із списку

- Ліга Чемпіонів
- Українська Прем'єр Ліга
- Відбір Євро 2020 - Група F

**ТУР 1**

Шахтар	0	-	0	Зоря
Динамо	0	-	0	Львів
Олександрія	0	-	0	Ворскла
Маріуполь	0	-	0	Десна
Олімпік	0	-	0	

Відправити результат

**ТУР 2**

Львів	0	-	0	Шахтар
Ворскла	0	-	0	Зоря
Десна	0	-	0	Динамо
	0	-	0	Олександрія
Олімпік	0	-	0	Маріуполь

Відправити результат

Рисунок 3.19 – Сторінка для заповнення результатів

Поля перевіряються на введення цілих чисел від 0. Приклад помилки коли в поле ввести дробове число (рис. 3.20).

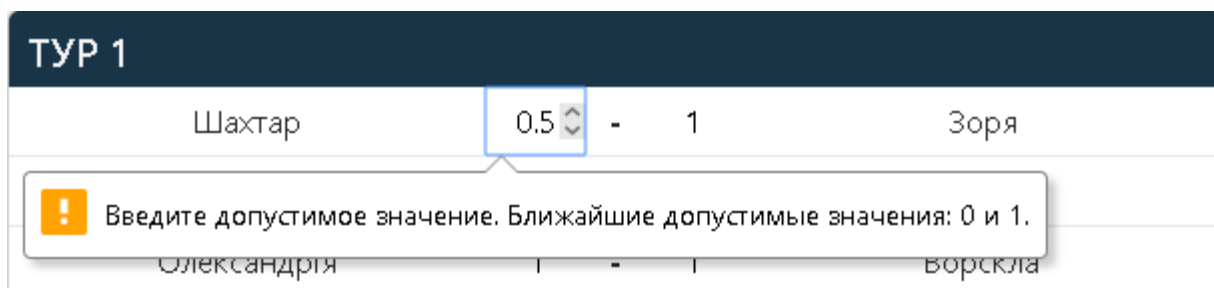


Рисунок 3.20 – Помилка при введенні дробового числа.

Також на рис. 3.21 зображено помилку при введенні від’ємного числа.

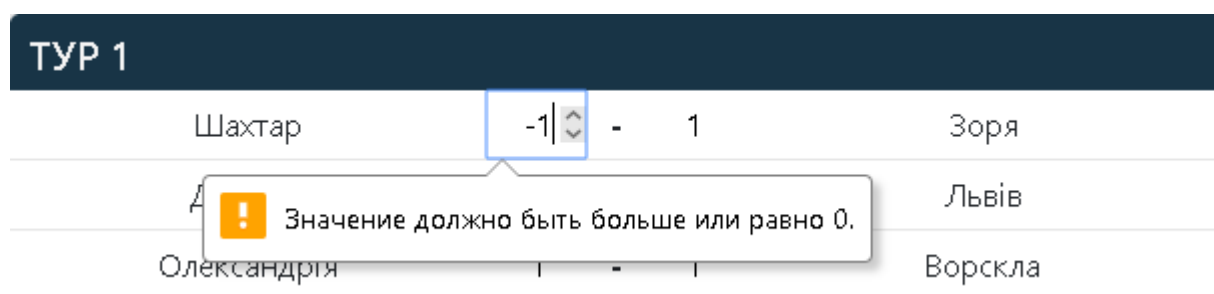


Рисунок 3.21 – Помилка при введенні від’ємного числа.

Для того щоб таблиця результатів велась правильно, без лишніх заповнених даних, поля для заповнення результатів зроблені обов’язковими. Тобто можна відправляти кожен тур окремо, але тільки в випадку коли всі матчі заповнено (крім того матчу де команда не має пари).

У випадку коли учасник не зіграв, результат заповнюється тоді коли учасник зіграє, після цього відправляється. Послідовність відправлення турів не має значення.

Після відправлення форми з результатами в полях залишаються введені значення, кнопка “Відправити результат” блокується (рис. 3.22).

ТУР 1				
Шахтар	1	-	3	Зоря
Динамо	1	-	2	Львів
Олександрія	3	-	2	Ворскла
Маріуполь	1	-	1	Десна
Олімпік	0	-	0	

Відправити результат

Рисунок 3.22 – Вигляд відправленого туру.

Після відправлення результати відображаються також і в користувацькій частині. А також утворюється табличка місць учасників.

Структура сторінки “Налаштувань” не є остаточною. Зараз в налаштуваннях є тільки функція додавання та видалення адміністраторів.

Сторінка має вигляд як на рис. 3.23.

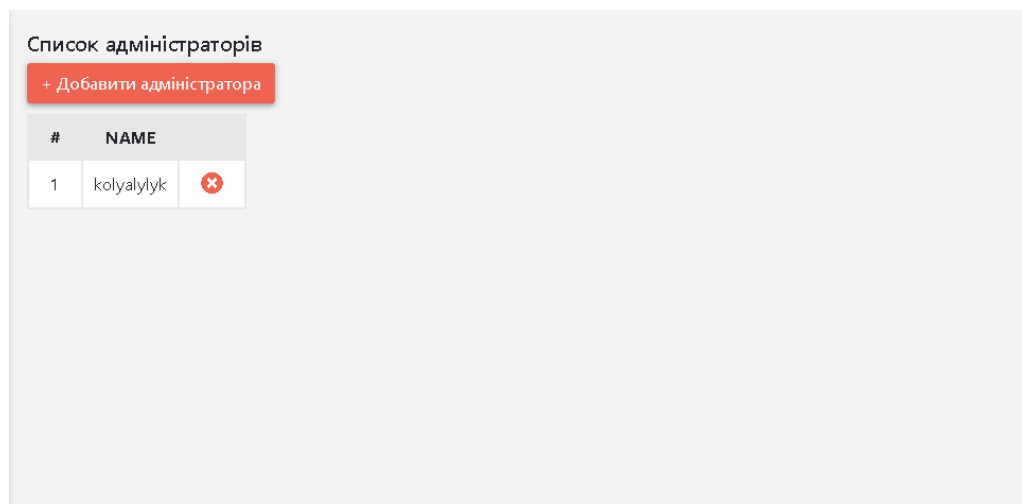


Рисунок 3.23 – Вигляд сторінки “Налаштування”

На сторінці розміщено табличку із списком адміністраторів та кнопка “Додати адміністратора”, яка відкриває модальне вікно, що містить форму з полями для вводу (рис. 3.24). Тут головний адміністратор(1 в списку) може додати інших адміністраторів, або ж це можливо зробити через базу даних.

В цій версії сервісу кожен адміністратор має свої турніри, якими може керувати. В наступній версії сервісу, задумано додати більше налаштувань та можливість додавання кількох адміністраторів до однакових турнірів.

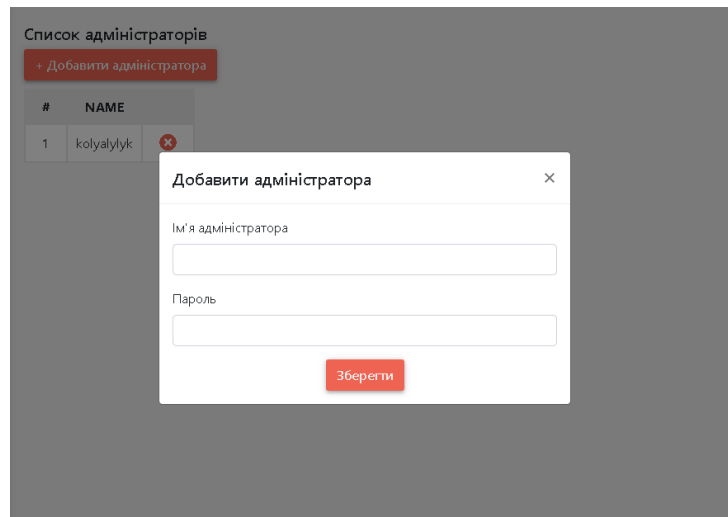


Рисунок 3.24 – Модальне вікно для додавання адміністратора

Для наглядності на рис. 3.25 додано тестового адміністратора (Ім'я адміністратора: admin, Пароль: admin).

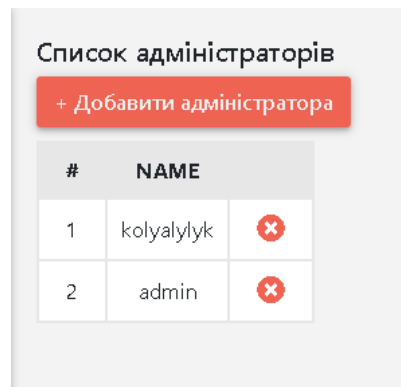


Рисунок 3.25 – Список з доданим адміністратором

Також для форми додавання адміністратора буде додано поле “Підтвердження пароля”, в якому треба буде ввести пароль повторно. І якщо два паролі зходяться тоді додаться новий адміністратор.

Код додавання адміністратора в базу даних виглядає наступним чином.

```

if (isset($_POST['save-new-admin'])) {
header('Location: '.$pageUrl.'');
$newAdminName = $_POST['new-admin-name'];
$newAdminPassword = $_POST['new-admin-password'];
mysql_query("INSERT INTO `admins`(`id`, `adminname`, `password`) VALUES
('', '{$newAdminName}', md5('{$newAdminPassword}'))");
}

```

Для прикладу потрібно зайти на сторінку нового адміністратора. Перед цим потрібно вийти натиснувши на пункт “Вийти” в навігаційному меню(рис. 3.26).

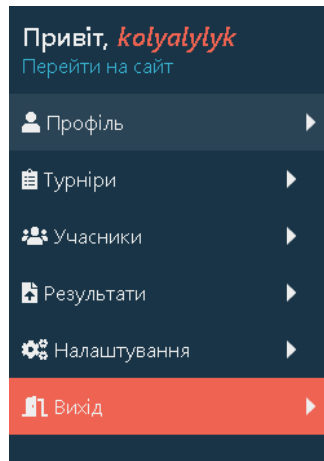


Рисунок 3.26 – Пункт “Вихід” в навігаційній панелі

При натисненні на кнопку йде переадресація на файл `logout.php`. Код файлу вказано нижче.

```

<?php
session_start();
if(session_destroy()) // Destroying All Sessions
{
header("Location: index.php"); // Redirecting To Home Page
}
?>

```

Файл `logout.php` переадресує на форму авторизації. Потрібно ввести логін та пароль створеного адміністратора(рис. 3.27).

Рисунок 3.27 – Введені дані нового адміністратора

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		46



На рис. 3.28 можна замітити, що привітання в навігаційному меню змінилось з “Привіт, kolyalyuk” на “Привіт, admin”.



Рисунок 3.28 – Сторінка при вході в адміністративну панель нового адміністратора

Для наглядності можна переглянути турніри нового адміністратора (рис. 3.28).

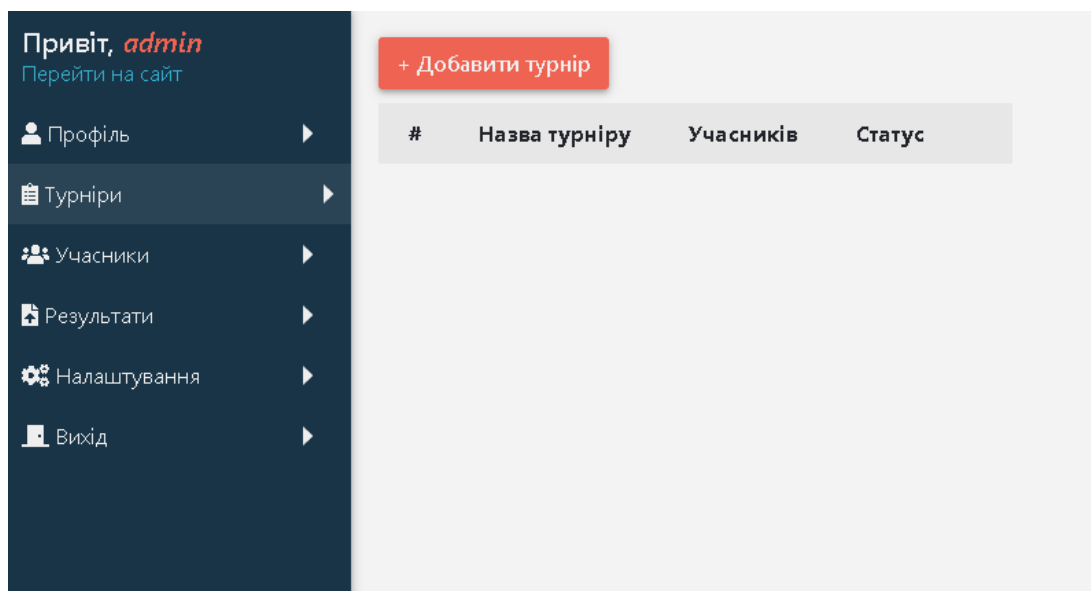


Рисунок 3.29 – Турніри нового адміністратора

Як можна побачити на рис. 3.29, турнірів в нового адміністратора ще не існує. Щоб дійсно переконатися в тому, що кожен адміністратор керує своїми турнірами, створено турнір(рис. 3.30).

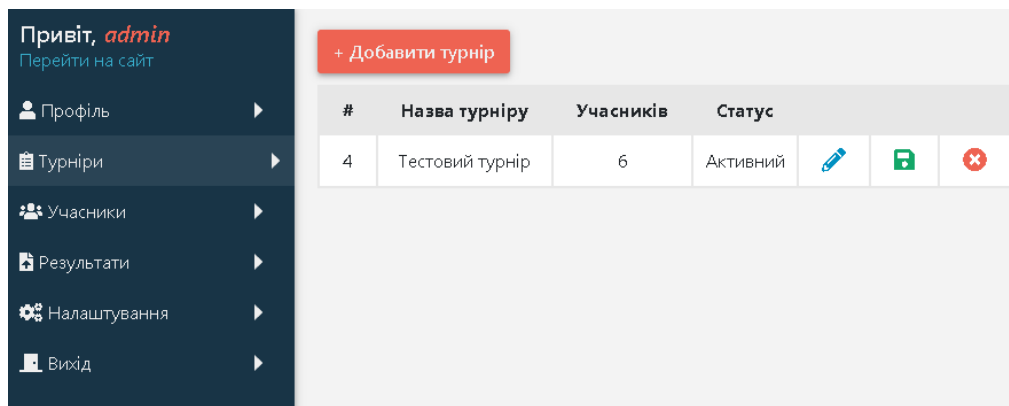


Рисунок 3.30 – Список з новим турніром

В базі даних турнір буде виглядати так як на рис. 3.31.

id	tournament_name	teams	status	admin_id
1	Ліга Чемпіонів	16	Активний	1
2	Українська Прем'єр Ліга	9	Активний	1
3	Відбір Євро 2020 - Група F	6	Активний	1
4	Тестовий турнір	6	Активний	2

Рисунок 3.31 – Створені турніри всіма адміністраторами

Як можна побачити записано три турніри з значенням поля “admin\_id” 1 – це ідентифікатор адміністратора kolyalyuk, та один турнір створений з значенням 2 – це адміністратор admin. Якщо знову зайти на сторінку адміністратора kolyalyuk, то ми будемо бачити тільки ті турніри де значення “admin\_id” буде дорівнювати 1.

Для видалення адміністратора потрібно видалити всі турніри які створені ним, після цього можна видаляти адміністратора. Код видалення адміністратора з бази даних представлено кодом нижче.

```
if (isset($_POST['delAdmin'])) {
    $query = mysql_query("SELECT `id` FROM `admins` WHERE
    `adminname`='".$_login_session'", $connection);
    $row = mysql_fetch_assoc($query);
    $id = $_POST['delAdmin'];
    if($id == $row['id']){
```

```

header('Location: ../admin/logout.php');
}else{
header('Location: '.$pageUrl.'');
}
mysql_query("DELETE FROM `tournaments` WHERE `admin_id`='{ $id}'");
mysql_query("DELETE FROM `admins` WHERE `id`='{ $id}'");
}

```

### 3.4 Проектування інтерфейсу користувача

Інтерфейс користувача розділяється на три основні блоки:

- “шапку” з логотипом та кнопкою входу в панель адміністратор;
- меню для вибору турнірів;
- контент який відображається відносно вибраного турніру.

Загальний вигляд користувацької частини можна розглянути на рис. 3.32.

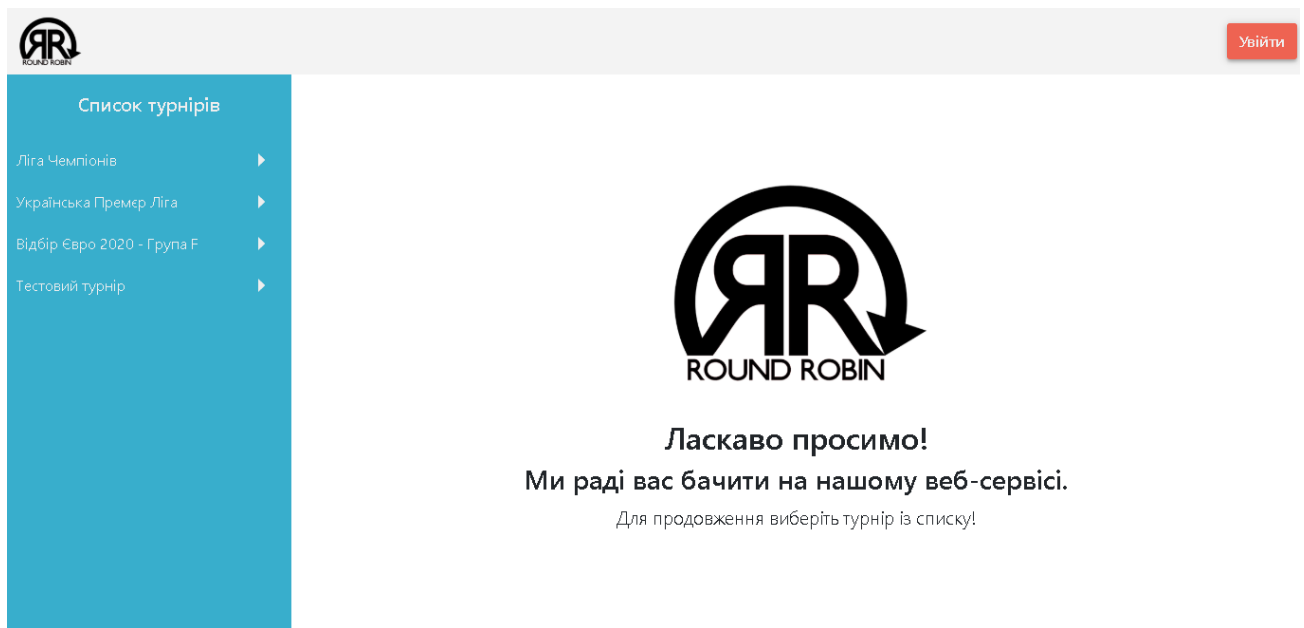


Рисунок 3.32 – Загальний вигляд користувацької частини

У навігаційному меню відображаються пункти з назвами турнірів всіх адміністраторів(рис. 3.33).

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		49

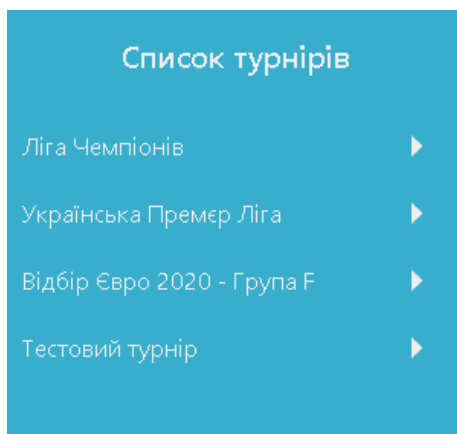


Рисунок 3.33 – Список всіх турнірів

У адмінській частині при редагуванні турнірів, можна вказати його статус (рис. 3.34).

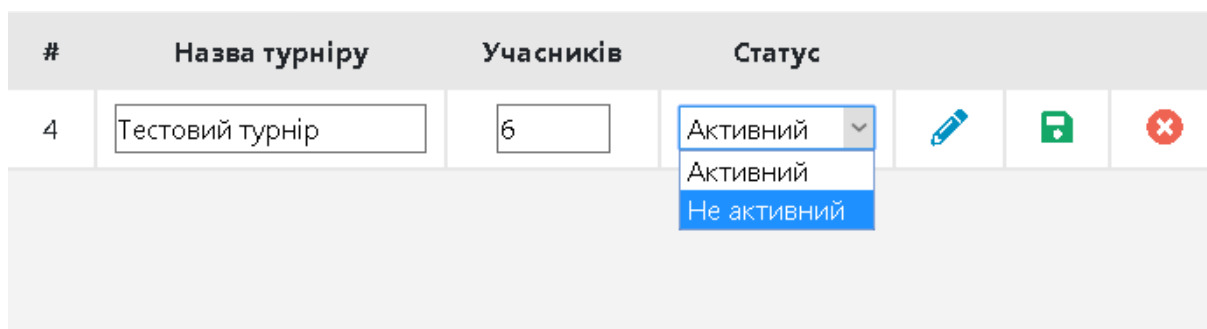


Рисунок 3.34 – Функція для встановлення статусу турніру

Якщо вказати статус “Не активний”, результатом буде зникнення цього турніру із списку доступних турнірів для користувача. Для прикладу турніру “Тестовий турнір”, який є у списку на рис. 3.34 було задано статус “Не активний”. Після цієї дії список турнірів доступний для користувачів буде виглядати як на рис. 3.35.

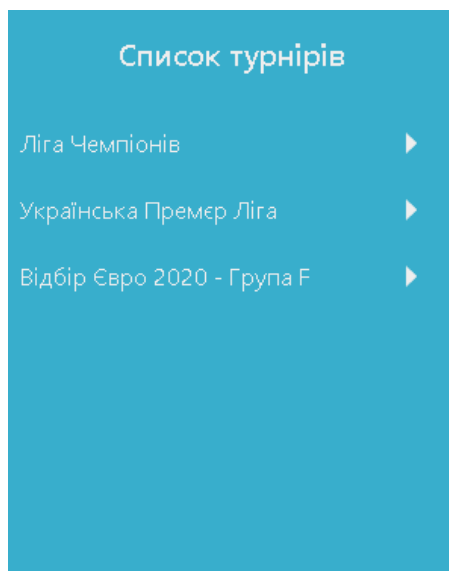


Рисунок 3.35 – Функція для встановлення статусу турніру

Як можна побачити на рис. 3.35, турнір зник та недоступний для користувача. Ця функція є дуже зручною якщо в турнірі є помилка та адміністратор хоче виправити її, також вона корисна, коли турнір планується провести ще раз і з тими ж командами. Для того, щоб не видаляти та не створювати знову, можна приховати його від користувачів та обнулити результати та даними учасників SQL запитом в базі даних.

Щоб переглянути турнірну таблицю та тури з результатами, користувачеві потрібно вибрати турнір із списку. Для прикладу вибраний турнір в який заповнювалися учасники та результати матчів (рис. 3.36).

#	Учасник	І	В	Н	П	ЗМ	ПМ	О
1	Зоря	1	1	0	0	3	1	3
2	Олександрія	1	1	0	0	3	2	3
3	Львів	1	1	0	0	2	1	3
4	Маріуполь	1	0	1	0	1	1	1
5	Десна	1	0	1	0	1	1	1
6	Ворскла	1	0	0	1	2	3	0
7	Динамо	1	0	0	1	1	2	0
8	Шахтар	1	0	0	1	1	3	0
9	Олімпік	0	0	0	0	0	0	0

Рисунок 3.36 – Контент вибраного турніру

Згідно із відправленими результатами першого туру (рис. 3.37) заповнюється таблиця з учасниками.

ТУР 1				
Шахтар	1	-	3	Зоря
Динамо	1	-	2	Львів
Олександрія	3	-	2	Ворскла
Маріуполь	1	-	1	Десна
Олімпік	-	-	-	

Рисунок 3.37 – Відправлені результати

Згідно відправленого результату та закінченого першого туру є три гри з виграшом однієї команди та нічия. Після того як результати відправлені кожному учаснику крім того, хто не має пари добавляється по одній зіграній грі. Відносно того, який учасник виграв, який програв чи нічия й додається одиниця в поле виграшу нічії чи програшу. В даному прикладі переможці матчів учасники “Зоря”, “Львів” та “Олександрія”. Кожному переможцю в таблицю добавиться одиничка до зіграних ігор та виграшу, також забиті та пропущені м’ячі, шайби тощо, і три очки. Програвшим командам: “Шахтар”, “Динамо”,

“Ворскла” добавиться теж по одній зіграній грі, програшу, забиті та пропущені м’ячі і нуль очок. В випадку нічиєї, двом командам під назвою “Маріуполь” та “Десна” добавиться по одній зіграній грі, по одній нічії, забиті та пропущені м’ячі та по одному очку кожному.

Після всіх підрахунків таблиця виглядає так як на рис. 3.38.

#	Учасник	І	В	Н	П	ЗМ	ПМ	О
1	Зоря	1	1	0	0	3	1	3
2	Олександрія	1	1	0	0	3	2	3
3	Львів	1	1	0	0	2	1	3
4	Маріуполь	1	0	1	0	1	1	1
5	Десна	1	0	1	0	1	1	1
6	Ворскла	1	0	0	1	2	3	0
7	Динамо	1	0	0	1	1	2	0
8	Шахтар	1	0	0	1	1	3	0
9	Олімпік	0	0	0	0	0	0	0

Рисунок 3.38 – Таблиця результатів після першого туру

Алгоритм який виконує підрахунки запускається при відправленні результатів туру. Код алгоритму представлено в Додатку А, файл functions.php.

## ВИСНОВОК

Під час виконання даного дипломного проекту було розроблено веб-сервіс для організації спортивних турнірів за круговою системою під назвою “Round Robin”.

Веб-додаток реалізований за допомогою технологій: HTML, CSS, JS, PHP, MySQL. Проект розроблений з розрахунком на подальше вдосконалення та додавання нових функціональних можливостей.

Проект може бути не тільки окремо, як веб-сервіс. Для потреби його можна переробити під кожен сайт.

Результатом роботи є повністю функціональний, протестований і готовий до використання веб-сервіс.

Тестування сервісу відбувалося в два етапи. Перший етап полягав в тестуванні додатку в різних браузерах (Google Chrome, Mozilla, Opera, Internet Explorer), мобільна версія не розроблялася. Необхідність такого тестування полягає в тому, щоб виявити і виправити помилки, допущені під час розробки. Другий етап тестування необхідний для опрацювання всіх можливих сценаріїв дій користувача, для того, щоб переконатися, що все працює правильно і ніяких помилок в проектуванні не допущено.

Веб-сервіс дозволяє адміністратору керувати (додавати, редагувати, видаляти) всіма даними, які відображаються в проекті, відвідувати особистий кабінет адміністратора, щоб організувати та керувати турнірами.

На основі поставлених задач:

- розроблено зручний інтерфейс панелі адміністратора та інтерфейс користувача;
- розроблено функціонал, щоб була можливість створювати та проводити турніри;
- покращено швидкодію при опрацюванні даних та відображення чітких результатів;

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		54



- розроблено можливість редагування та видалення даних
- розроблено можливість під'єднання до сайтів.

На відміну від приведених до прикладу аналогів, розроблений веб-сервіс має свої переваги та недоліки.

**Переваги:**

- зручний інтерфейс.
- можливість авторизації;
- один з перших веб-сервісів в Україні(українських аналогів не було знайдено);
- можливість організувати турніри від початку до кінця;
- користувач може переглянути результати ігор та турнірну таблицю;
- може бути використаний в якості панелі адміністратора для сайту спортивних команд з подальшою підтримкою.

**Недоліки:**

- немає можливості зареєструватись;
- немає можливості керувати турніром кількома адміністраторами;

					<b>ДР.ІІс – 11.00.000 ІЗ</b>	Арк.А
						55
Змн.	Арк.	№ докум.	Підп.	Дата		

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Round-robin tournament – Wikipedia. URL: [https://en.wikipedia.org/wiki/Round-robin\\_tournament](https://en.wikipedia.org/wiki/Round-robin_tournament) (дата звернення 15.01.2019)
2. Схеми проведення змагань. URL: [https://studopedia.su/4\\_31935\\_shemi-provedennya-zmagan.html](https://studopedia.su/4_31935_shemi-provedennya-zmagan.html) (дата звернення 21.01.2019)
3. Каневец Г.М. Организационно-методические основы проведения спортивно-массовых мероприятий. Лекция. Л.: ГДО-ИФК, 1984. 24 с.
4. Борисов И.А. Организация, проведение и судейство спортивных соревнований. Лекция. М. 1974. 21 с.
5. Переверзин И.И. Менеджмент спортивной организации: Учебное пособие. 2-изд., пере раб. и доп. М.: „СпортАкадемПресс”, 2002. 244 с.
6. Системы проведения соревнований. URL: <https://studfiles.net/preview/3219486/page:4/> (дата звернення 01.02.2019)
7. Андросчук О. І., Соболенко І. М. Футбол, Міні-футбол та методика його викладання. Умань, 2013. 118 с. – URL: <https://studopedia.org/7-19273.html> (дата звернення 18.02.2019)
8. Кругова система – Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Кругова\\_система](https://uk.wikipedia.org/wiki/Кругова_система) (дата звернення 04.03.2019)
9. Кругова система – Howling Pixel. URL: [https://howlingpixel.com/i-uk/Кругова\\_система](https://howlingpixel.com/i-uk/Кругова_система) (дата звернення 11.03.2019)
10. Таблицы и расписания круговых турниров по шахматах. URL: <http://chess.sainfo.ru/table.html> (дата звернення 14.03.2019)
11. Генератор турнирной сетки – Challonge . URL: [https://challonge.com/ru/tournament/bracket\\_generator](https://challonge.com/ru/tournament/bracket_generator) (дата звернення 20.03.2019)
12. Основні вимоги до створення сайтів в інтернеті. URL: <https://recommerce.com.ua/vimogi-do-stvorennya-saitu> (дата звернення 25.03.2019)

					ДР.Шс – 11.00.000 ПЗ	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		56

13. Лабберс П., Олберс Б., Салим Ф. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. : Пер. с англ. М. : ООО “И.Д. Вильямс”, 2011. 272 с.

14. Шмитт К. CSS. Рецепты программирования. URL: <http://static.ozone.ru/multimedia/book-file/1005871779.pdf> (дата звернения 03.04.2019)

15. Кантор И. Современный учебник JavaScript. 2007. – URL: <https://learn.javascript.ru> (дата звернения 08.04.2019)

16. Что такое Bootstrap. URL: <http://blogwork.ru/что-такое-bootstrap/> (дата звернения 11.04.2019)

17. Font Awesome. URL: <https://fontawesome.com> (дата звернения 15.01.2019)

18. Brackets – A modern, open source text editor that understands web design. URL: <http://brackets.io> (дата звернения 22.04.2019)

19. Open Server Panel – Локальный веб-сервер для Windows. URL: <https://ospanel.io> (дата звернения 26.04.2019)

20. Котеров Д., Симдянов И. PHP 7. Наиболее полное руководство. 2016.

21. Дюбуа П. MySQL. Сборник рецептов. Пер. с англ. СПб: Символ-Плюс, 2006. 1056 с.

22. SQL для начинающих. Часть 3. URL: <http://jtest.ru/bazyi-dannyix/sql-dlya-nachinayushhix-chast-3.html> (дата звернения 08.05.2019)

23. Colors.co – The super fast color schemes generator. URL: <https://colors.co> (дата звернения 14.05.2019)

24. PHP: Руководство по PHP – Manual. URL: <https://php.net/manual/en/> (дата звернения 23.05.2019)

25. Bootstrap – The most popular HTML, CSS and JS library in the world. URL: <https://getbootstrap.com> (дата звернения 27.05.2019)

					<b>ДР.Шс – 11.00.000 ПЗ</b>	Арк.А
Змн.	Арк.	№ докум.	Підп.	Дата		57

## ДОДАТОК А

### Файл /round-robin/admin/login.php

```
<?php
session_start(); // Starting Session
$error=''; // Variable To Store Error Message
if (isset($_POST['submit'])) {
if (empty($_POST['adminname']) || empty($_POST['password'])) {
$error = "Введіть логін та пароль!";
}else{
// Define $adminname and $password
$adminname=$_POST['adminname'];
$password=MD5($_POST['password']);
// Establishing Connection with Server by passing server_name, user_id
and password as a parameter
$connection = mysql_connect("localhost", "root", "");
// To protect MySQL injection for Security purpose
$adminname = stripslashes($adminname);
$password = stripslashes($password);
$adminname = mysql_real_escape_string($adminname);
$password = mysql_real_escape_string($password);
// Selecting Database
$db = mysql_select_db("round-robin", $connection);
// SQL query to fetch information of registered users and finds user
match.
$query = mysql_query("SELECT * FROM admins WHERE
adminname='$adminname' AND password='$password'", $connection);
$rows = mysql_num_rows($query);
if ($rows == 1) {
```

```

$_SESSION['logged_admin']=$adminname;
header("location: profile.php?profile-content");
} else {
$error = "Логін або пароль введено неправильно!";
}
mysql_close($connection); // Closing Connection
}
}
?>

```

### Файл /round-robin/admin/functions.php

```

<?php
    include('session.php');
    /*
    Підключення до БД*/
    $connection = mysql_connect("localhost", "root", "");
    /*Вибір ДБ*/
    $db = mysql_select_db("round-robin", $connection);
    /*Змінна отримує ім'я сесії*/
    $adminName = $login_session;
    /*Вибірка з бази даних*/
    $query = mysql_query("SELECT `id` FROM `admins` WHERE
`adminname`='".$adminName'");
    /*Прохід по рядках*/
    $adminInfo = mysql_fetch_row($query);
    /*Змінна отримує id з БД*/
    $adminId = $adminInfo[0];

    /*Функція приймає дані з форми та відправляє запит в БД*/

```

```

if (isset($_POST['save-tournament'])) {
    header('Location: '.$pageUrl.'');
    $tournamentName = $_POST['tournament-name'];
    $teamsQuantity = $_POST['teams-quantity'];
    $tournamentStatus = $_POST['tournament-status'];

    mysql_query("INSERT INTO `tournaments` (`id`,
`tournament_name`, `teams`, `status`, `admin_id`) VALUES ('',
'{$tournamentName}', '{$teamsQuantity}', '{$tournamentStatus}',
'{$adminId}')");
}

/*Функція приймає дані з форми та перезаписує дані в БД*/
if (isset($_POST['updateTournament'])) {
    header('Location: '.$pageUrl.'');
    $updatedTournamentId = $_POST['updateTournament'];
    $updatedTournamentName =
$_POST['updatedTournamentName'.$updatedTournamentId.''];
    $updatedTournamentTeams =
$_POST['updatedTournamentTeams'.$updatedTournamentId.''];
    $updatedTournamentStatus =
$_POST['updatedTournamentStatus'.$updatedTournamentId.''];

    mysql_query("UPDATE `tournaments` SET
`id`='{$updatedTournamentId}', `tournament_name`='{$updatedTournamentName}',
`teams`='{$updatedTournamentTeams}', `status`='{$updatedTournamentStatus}'
WHERE `tournaments`.`id`='{$updatedTournamentId}");
}

/*Функція видалення турніру*/
if (isset($_POST['delTournament'])) {
    header('Location: '.$pageUrl.'');
    $id = $_POST['delTournament'];

    mysql_query("DELETE FROM `matches` WHERE
`tournament_id`='{$id}");
}

```

```

        mysql_query("DELETE FROM `participants` WHERE
`tournament_id`='{ $id}'");

        mysql_query("DELETE FROM `tournaments` WHERE `id`='{ $id}'");

    }

/*Функція додавання нового адміна в БД*/
if (isset($_POST['save-new-admin'])) {
    header('Location: '.$pageUrl.'');
    $newAdminName = $_POST['new-admin-name'];
    $newAdminPassword = $_POST['new-admin-password'];
    mysql_query("INSERT INTO `admins`(`id`, `adminname`,
`password`) VALUES ('', '{$newAdminName}',
md5('{$newAdminPassword}'))");
}

/*Функція видалення адміна*/
if (isset($_POST['delAdmin'])) {
    $query = mysql_query("SELECT `id` FROM `admins` WHERE
`adminname`='{$login_session'", $connection);
    $row = mysql_fetch_assoc($query);
    $id = $_POST['delAdmin'];
    if($id == $row['id']){
        header('Location: ../admin/logout.php');
    }else{
        header('Location: '.$pageUrl.'');
    }
    mysql_query("DELETE FROM `tournaments` WHERE
`admin_id`='{ $id}'");
    mysql_query("DELETE FROM `admins` WHERE `id`='{ $id}'");
}

```

```

/*Функція додавання учасника*/
if(isset($_POST['save-participant'])) {
    header('Location: '.$pageUrl.'');
    $tournamentId = $_GET['tournament'];
    $participantName = $_POST['participant-name'];
    mysql_query("INSERT INTO `participants` (`tournament_id`,
`name`) VALUES ('{$tournamentId}', '{$participantName}')");
}

/*Функція перезапису даних учасника*/
if (isset($_POST['updateParticipant'])) {
    header('Location: '.$pageUrl.'');
    $updatedParticipantId = $_POST['updateParticipant'];
    $updatedParticipantName =
$_POST['updatedParticipantName'].$updatedParticipantId.'';
    mysql_query("UPDATE `participants` SET
`name`='{$updatedParticipantName}' WHERE
`participants`.`id`='{$updatedParticipantId}'");
}

/*Функція видалення учасника*/
if (isset($_POST['delParticipant'])) {
    header('Location: '.$pageUrl.'');
    $id = $_POST['delParticipant'];
    $tournamentId = $_GET['tournament'];
    mysql_query("DELETE FROM `matches` WHERE `tournament_id`
='{$tournamentId}'");
    mysql_query("DELETE FROM `participants` WHERE `id`='{$id}'");
}

/*Функція відправлення результатів матчів та оновлення інформації
учасників в відповідні таблиці*/
$tournamentId = $_GET['tournament'];

```



```

if($_GET['tournament']){

    $participantsQuery = mysql_query("SELECT * FROM `participants`
WHERE `tournament_id` = $tournamentId", $connection);

    $resultMatches = mysql_query("SELECT * FROM `matches` WHERE
`tournament_id`= $tournamentId", $connection);

    while($row = mysql_fetch_assoc($resultMatches)){

        if(isset($_POST['send-results'])){

            header('Location: '.$pageUrl.'');

            if($_POST['f-participant-score'].$row['id']] != '' &&
$_POST['s-participant-score'].$row['id']] != ''){

                mysql_query("UPDATE `matches` SET
`f_participant_score`= {$_POST['f-participant-score'].$row['id']},
`s_participant_score`= {$_POST['s-participant-score'].$row['id']}
WHERE `id` = {$row['id']}", $connection);

                mysql_query("UPDATE `participants` SET
`games`=games+1, `scored_goals`=scored_goals+{$_POST['f-participant-
score'].$row['id']}, `missed_goals`=missed_goals+{$_POST['s-
participant-score'].$row['id']} WHERE `id` =
{$row['f_participant_id']}");

                mysql_query("UPDATE `participants` SET
`games`=games+1, `scored_goals`=scored_goals+{$_POST['s-participant-
score'].$row['id']}, `missed_goals`=missed_goals+{$_POST['f-
participant-score'].$row['id']} WHERE `id` =
{$row['s_participant_id']}");

                if ($_POST['f-participant-score'].$row['id']] >
$_POST['s-participant-score'].$row['id']) {

                    mysql_query("UPDATE `participants` SET
`victories`=victories+1, `points`=points+3 WHERE `id` =
{$row['f_participant_id']}");

                    mysql_query("UPDATE `participants` SET
`defeats`=defeats+1 WHERE `id` = {$row['s_participant_id']}");

                }else if ($_POST['f-participant-score'].$row['id']]
< $_POST['s-participant-score'].$row['id']) {

                    mysql_query("UPDATE `participants` SET
`victories`=victories+1, `points`=points+3 WHERE `id` =
{$row['s_participant_id']}");

```

```

        mysql_query("UPDATE `participants` SET
`defeats`=defeats+1 WHERE `id` = {$row['f_participant_id']}");

        }else if ($_POST['f-participant-score'].$row['id']]
== $_POST['s-participant-score'].$row['id'])) {

            mysql_query("UPDATE `participants` SET
`draws`=draws+1, `points`=points+1 WHERE `id` =
{$row['f_participant_id']}");

            mysql_query("UPDATE `participants` SET
`draws`=draws+1, `points`=points+1 WHERE `id` =
{$row['s_participant_id']}");

        }

    }

}

}

?>

```

### Файл /round-robin/admin/pages/results.php

```

<div class="right-block results-block">
<div class="text-center">
    <?php
        $tournamentId = $_GET['tournament'];

        if($_GET['tournament']){

            $availableMatches = mysql_query("SELECT `id` FROM `matches`
WHERE `tournament_id`='{ $tournamentId }'", $connection);

            $availableQuantity = mysql_num_rows($availableMatches);

            $query = mysql_query("SELECT `id`, `name` FROM `participants`
WHERE `tournament_id`='{ $tournamentId }'", $connection);

            $rows = mysql_num_rows($query);

            $teams = array();

            if($rows == 0){

```

```

        return 'Немає команд';
    }

    while ($row = mysql_fetch_array($query)) {
        array_push($teams, $row[0]);
    }

    $teams = array_chunk($teams, $rows/2);
    $teams1 = $teams[0];
    $teams2 = $teams[1];
    $noTeam = 'NULL';
    if($rows % 2 == 1){
        array_push($teams1, $teams[2][0]);
    }

    if ($rows % 2 == 0) {
        for ($i = 0; $i < ($rows - 1); $i++) {
            for ($j = 0; $j < count($teams1); $j++) {
                if($availableQuantity == 0){
                    if(($i+1) % 2 == 1){
                        mysql_query("INSERT INTO
`matches`(`tournament_id`, `f_participant_id`, `f_participant_score`,
`s_participant_id`, `s_participant_score`) VALUES ($tournamentId,
$teams1[$j], NULL, $teams2[$j], NULL)");
                    }else{
                        mysql_query("INSERT INTO
`matches`(`tournament_id`, `f_participant_id`, `f_participant_score`,
`s_participant_id`, `s_participant_score`) VALUES ($tournamentId,
$teams2[$j], NULL, $teams1[$j], NULL)");
                    }
                }
            }
        }

        $fixedElem = $teams1[0];
        $elem1 = array_pop($teams1);
        $elem2 = array_shift($teams2);
    }

```

```

        array_splice($teams1, 1, 0, $elem2 );
        array_push($teams2, $elem1);
    }
} else {
    if (count($teams1) > count($teams2)) {
        array_push($teams2, $noTeam);
    } else if (count($teams1) < count($teams2)) {
        array_push($teams1, $noTeam);
    }

    for ($i = 0; $i < $rows; $i++) {
        if($availableQuantity == 0){
            if(($i+1) % 2 == 1){
                mysql_query("INSERT INTO
`matches`(`tournament_id`, `f_participant_id`, `f_participant_score`,
`s_participant_id`, `s_participant_score`) VALUES ($tournamentId,
$teams1[$j], NULL, $teams2[$j], NULL)");
            }else{
                mysql_query("INSERT INTO
`matches`(`tournament_id`, `f_participant_id`, `f_participant_score`,
`s_participant_id`, `s_participant_score`) VALUES ($tournamentId,
$teams2[$j], NULL, $teams1[$j], NULL)");
            }
        }
    }

    $fixedElem = $teams1[0];
    $elem1 = array_pop($teams1);
    $elem2 = array_shift($teams2);
    array_splice($teams1, 1, 0, $elem2 );
    array_push($teams2, $elem1);
}
}

```

```

    }

?>

</div>

<div id="matchList">

    <?php

        if(!$_GET['tournament']){

            echo

            '<script>document.getElementById(\'matchList\').style.display =
            "none"</script>';

        }

    ?>

    <?php

        $matchesquery = mysql_query("SELECT `matches`.`id`,
        `matches`.`tournament_id`, `matches`.`f_participant_id`,
        `matches`.`f_participant_score`, `matches`.`s_participant_id`,
        `matches`.`s_participant_score`, `participants`.`name` as
        f_participant_id, t3.`name` as s_participant_id FROM `matches` LEFT
        JOIN `participants` ON `matches`.`f_participant_id` =
        `participants`.`id` left join `participants` as t3 on
        `matches`.`s_participant_id` = t3.`id` WHERE `matches`.`tournament_id`
        = $tournamentId ORDER BY `matches`.`id`", $connection);

        $matches = mysql_num_rows($matchesquery);

        $teamsQuantity = mysql_query("SELECT `teams` FROM
        `tournaments` WHERE `id`= $tournamentId", $connection);

        while ($row = mysql_fetch_assoc($teamsQuantity)) {

            $quantity['teams'] = $row['teams'];

        }

        $quantity = intval($quantity['teams']) ;

        $teamsHalf = $quantity / 2;

        $matchesInTour = round($teamsHalf);

        $i=0;

        $j=1;

        $m = false;

        while ($row = mysql_fetch_assoc($matchesquery)) {

```

```

        if($i % $matchesInTour == 0){

            $m = false;

            echo '<form action="" method="post"
style="background: #fff; box-shadow: 0 1px 8px 0 rgba(0, 0, 0, .2), 0
3px 4px 0 rgba(0, 0, 0, .14), 0 3px 3px -2px rgba(0, 0, 0, .12);
border-radius: 5px;"><h5 style="background: #183446; color: #f3f3f3;
margin: 0; padding: 7px; padding-left: 10px; text-align: left;
border-radius: 5px 5px 0 0;">TYP '.$j.'</h5>';

            $j++;

        }

        if($row['f_participant_id'] == NULL ||
$row['s_participant_id'] == NULL){

            echo '<div style="display: flex; justify-content:
center; margin: 0; border-bottom: 1px solid #e7e7e7;">

                <div style="width: 250px; background: #fff;
display: flex; align-items: center; justify-content: center; font-
size: 16px;">'.$row['f_participant_id'].'</div>';

                echo '<input class="text-center" style="width:
50px; padding-left: 15px; font-size: 16px; border: none;"
type="number" value="'.$row['f_participant_score'].'" name="f-
participant-score'.$row['id'].'" min="0" placeholder="0" disabled/>';

                echo '<b class="text-center" style="padding: 5px
0; width: 5%">-</b>';

                echo '<input class="text-center" style="width:
50px; padding-left: 15px; font-size: 16px; border: none;"
type="number" value="'.$row['s_participant_score'].'" name="s-
participant-score'.$row['id'].'" min="0" placeholder="0" disabled/>';

                echo '<div style="width: 250px; background: #fff;
display: flex; align-items: center; justify-content: center; font-
size: 16px;">'.$row['s_participant_id'].'</div>

                </div>';

        } else {

            echo '<div style="display: flex; justify-content:
center; margin: 0; border-bottom: 1px solid #e7e7e7;">

                <div style="width: 250px; background: #fff;
display: flex; align-items: center; justify-content: center; font-
size: 16px;">'.$row['f_participant_id'].'</div>';

```

```

        echo '<input class="text-center" style="width:
50px; padding-left: 15px; font-size: 16px; border: none;"
type="number" value="'. $row['f_participant_score']. '" name="f-
participant-score'. $row['id']. '" min="0" placeholder="0" required/>';

        echo ' <b class="text-center" style="padding: 5px
0; width: 5%">-</b>';

        echo '<input class="text-center" style="width:
50px; padding-left: 15px; font-size: 16px; border: none;"
type="number" value="'. $row['s_participant_score']. '" name="s-
participant-score'. $row['id']. '" min="0" placeholder="0" required/>';

        echo ' <div style="width: 250px; background: #fff;
display: flex; align-items: center; justify-content: center; font-
size: 16px;">'. $row['s_participant_id']. '</div>

        </div>';

    }

    if ( ( $row['f_participant_id'] != NULL ||
$row['s_participant_id'] != NULL ) && $row['s_participant_score'] ) {

        $m = true;

    }

    $i++;

    if($i % $matchesInTour == 0){

        if(!$m){

            echo ' <div class="text-center" style="padding:
20px; margin: 5px 0 50px 0;"><button class="btn send-results"
type="submit" name="send-results">Відправити
результат</button></div></form>';

            }else {

                echo ' <div class="text-center" style="padding:
20px; margin: 5px 0 50px 0;"><button class="btn send-results"
type="submit" name="send-results" disabled>Відправити
результат</button></div></form>';

            }

        }

    }

?>

</div>

</div>

</div>

```

**ДОДАТОК Б**

Файл /round-robin/style/admin.css

```
* {  
    margin: 0;  
    padding: 0;  
}  
  
body {  
    background-color: #f3f3f3;  
}  
  
a {  
    text-decoration: none !important;  
}  
  
button:focus {  
    outline: none;  
}  
  
.btn.disabled, .btn:disabled {  
    background: #9c9c9c;  
}  
  
.btn.disabled:hover, .btn:disabled:hover {  
    background: #9c9c9c;  
}  
  
.admin-form {  
    height: 100vh;
```



```
display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

background: #38AECC;

}

.admin-form > h2 {

font-weight: 600;

font-size: 40px;

font-weight: bold;

color: #ffffff;

}

.admin-form > form {

padding: 35px 60px 5px 60px;

width: 500px;

box-shadow: 0 1px 8px 0 rgba(0, 0, 0, .2), 0 3px 4px 0 rgba(0, 0, 0, .14), 0 3px 3px -2px rgba(0, 0, 0, .12);

border-radius: 10px;

background: #ffffff;

}

.admin-form > form .form-group {

padding: 20px 0;

}

.admin-form > form input {

border: none;

border-bottom: 2px solid #38AECC;
```

```
    transition: 1s;

    border-radius: 0;

    font-size: 20px;

    color: #38AECC;
}

.admin-form > form input::placeholder {

    color: #c8c8c8;
}

.admin-form > form input:focus {

    border: none;

    border-bottom: 2px solid #EE6352;

    transform: scale(1.15);

    transition: 1s;

    color: #EE6352;
}

.admin-form > form button:focus,
.admin-form > form button:active,
.btn-primary:not(:disabled):not(.disabled).active,
.btn-primary:not(:disabled):not(.disabled):active,
.show > .btn-primary.dropdown-toggle,
.btn-primary:not(:disabled):not(.disabled).active:focus,
.btn-primary:not(:disabled):not(.disabled):active:focus,
.show > .btn-primary.dropdown-toggle:focus {

    box-shadow: 0 1px 2px 0 rgba(60, 64, 67, 0.302), 0 1px 3px 1px
    rgba(60, 64, 67, 0.149);

    outline: none;
}
```

```
    background-color: #0090C1;

    border-color: #0090C1;
}

.form-error {

    color: #f44336;

    padding-top: 10px;

    font-weight: 600;
}

.form-control {

    border-radius: 5px;

    transition: .3s;
}

.form-control:focus {

    border-color: #0090C1;

    border-width: 2px;

    box-shadow: none;
}

.btn-primary {

    border-radius: 3px;

    background: #EE6352;

    box-shadow: 0 1px 8px 0 rgba(0, 0, 0, .2), 0 3px 4px 0 rgba(0, 0, 0, .14), 0 3px 3px -2px rgba(0, 0, 0, .12);

    color: #f3f3f3;

    width: 120px;

    font-weight: 600;

    text-transform: uppercase;
```

```
        border: none;
    }

    .btn-primary:hover {
        background-color: #0090C1;
    }

    .form-error {
        margin: 10px 0;
        color: #EE6352;
    }

    /*Profile Styles*/

    .burger {
        position: absolute;
        left: 260px;
        top: 10px;
        display: none;
        align-items: center;
        justify-content: center;
        flex-direction: column;
        cursor: pointer;
    }

    .burger > span {
        width: 30px;
        margin: 2px 0;
        height: 5px;
        border-radius: 2px;
```

```
    background: #183446;

    transition: all 1s ease;
}

.burger-opened > span:nth-child(1) {

    transform: rotate(45deg) translate(6px, 6px);
}

.burger-opened > span:nth-child(2) {

    opacity: 0;
}

.burger-opened > span:nth-child(3) {

    transform: rotate(-45deg) translate(6px, -6px);
}

.profile-menu {

    position: fixed;

    width: 250px;

    height: 100vh;

    background: #183446;

    color: #eee;

    transition: .3s;

    box-shadow: 0 1px 8px 0 rgba(0, 0, 0, .2), 0 3px 4px 0 rgba(0, 0,
0, .14), 0 3px 3px -2px rgba(0, 0, 0, .12);
}

.profile-menu-header {

    padding: 10px;
}

.welcome-text {
```

```
    text-align: left;

    cursor: default;

    margin-bottom: 0;
}

.welcome-text > i {

    color: #EE6352;
}

.visit-website {

    color: #38AECC;
}

.visit-website:hover {

    color: #0090C1;
}

.menu-option {

    position: relative;

    display: flex;

    justify-content: flex-start;

    align-items: center;

    padding: 10px;

    color: #eee !important;

    transition: .3s;
}
```

## Файл /round-robin/style/main.css

```
* {  
    margin: 0;  
    padding: 0;  
}  
  
a:hover, a:focus {  
    text-decoration: none;  
    box-shadow: none;  
}  
  
.header {  
    height: 70px;  
    background: #f3f3f3;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}  
  
.header .home, .header img {  
    height: 100%;  
    padding-left: 5px;  
}  
  
.header > .login {  
    margin-right: 5px;  
    border-radius: 3px;  
    background: #EE6352;
```

```
    font-weight: 600;

    color: #eee;

    box-shadow: 0 1px 8px 0 rgba(0, 0, 0, .2), 0 3px 4px 0 rgba(0, 0,
0, .14), 0 3px 3px -2px rgba(0, 0, 0, .12);
}

.header > .login:hover {

    background: #0090C1;

}

.tournaments-participants-matches {

    display: flex;

}

.welcome-user {

    height: calc(100vh - 70px);

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

}

.welcome-user img {

    width: 50%;

}

.welcome-user p {

    font-size: 20px;

    font-weight: 300;

}

.participants-matches {
```



```
width: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: flex-start;
}

.tournament-table {

margin: 10px 0;

width: 70%;
}

caption {

caption-side: top !important;

font-size: 20px;

color: #183446;

font-weight: 700;
}

.tournament-table tr:hover {

background: #FAFAFA;
}

.tournament-table tr th {

border-right: 2px solid #e7e7e7;

border-left: 2px solid #e7e7e7;

padding: 10px 0px;

background: #e7e7e7;

text-align: center;
```

```
}  
  
.tournament-table tr td {  
    border-right: 2px solid #e7e7e7;  
    border-left: 2px solid #e7e7e7;  
    border-bottom: 2px solid #e7e7e7;  
    text-align: center;  
    padding: 10px 20px;  
}  
  
.tournament-table tr td.option {  
    padding: 0;  
    border-right: 2px solid #e7e7e7;  
    border-left: 2px solid #e7e7e7;  
}  
  
.tournamentList {  
    min-width: 300px;  
    min-height: calc(100vh - 70px);  
    background: #38AECC;  
    color: #eee;  
    transition: .3s;  
}  
  
.tournamentList h5 {  
    padding: 18px 0;  
}  
  
.tournamentChoice {  
    position: relative;
```

```
display: flex;

justify-content: flex-start;

align-items: center;

padding: 10px;

color: #eee !important;

transition: .3s;

}

.tournamentChoice:after {

    content: '';

    position: absolute;

    right: 30px;

    border-top: 5px solid #eee;

    border-right: 5px solid #eee;

    border-left: 5px solid transparent;

    border-bottom: 5px solid transparent;

    transform: rotate(45deg);

    transition: .3s;

}

.tournamentChoice:hover {

    background: rgba(0, 0, 0, .2);

    transition: .3s;

}

.tournamentChoice:hover:after {

    right: 15px;

    transition: .3s;

}
```

```
}  
  
.tournamentChoice > i {  
    padding-right: 5px;  
}  
  
.active-tournament {  
    background: rgba(0, 0, 0, .2);  
}  
  
.active-tournament:hover {  
    background: rgba(0, 0, 0, .2);  
}  
  
.active-tournament:after {  
    right: 15px;  
}  
  
caption {  
    caption-side: top !important;  
    text-align: center;  
}
```

