

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток підтримки організації змагань зі стрільби з лука»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проекткування»

Виконавець роботи: студент групи ІТ-61-8 Стегній Денис Костянтинівич

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК**

з оцінкою _____ «__» _____ 2020 р.

Науковий керівник

(підпис)

доц. к.т.н. Алексенко О. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2020

Сумський державний університет
 Факультет електроніки та інформаційних технологій
 Кафедра комп'ютерних наук
 Секція інформаційних технологій проектування
 Спеціальність 122 «Комп'ютерні науки»
 Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
 «__» _____ 2020 р.

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

Стегній Денис Костянтинович

1 Тема роботи Web-додаток підтримки організації змагань зі стрільби з лука

керівник роботи Алексенко Ольга Василівна,

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи досвід участі у змаганнях та процесі їх організації

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) «Аналіз процесу організації спортивних змагань та інформаційних систем для їх забезпечення», «Постановка задачі», «Розробка».

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) слайди презентації: Актуальність, Мета та задачі, Огляд аналогів, Модель процесу організації змагань зі стрільби з лука, Основний функціонал ІС, Засоби реалізації, Варіанти використання ІС, Потік даних в процесі організації змагань, Діаграма бази даних, Демонстрація додатку на платформі Windows, Демонстрація Web-додатку, Висновки.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Аналіз процесу організації спортивних змагань та інформаційних систем для їх забезпечення	Алексенко О. В.		
Постановка задачі	Алексенко О. В.		
Розробка	Алексенко О. В.		

7. Дата видачі завдання _____ 01.10.2019 _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	Формування ТЗ	07.12.19	19.12.19	
2	Розробка макетів	17.12.19	03.02.19	
3	Розробка БД	06.02.20	08.03.20	
4	Розробка ПЗ для платформи Windows	11.03.20	12.04.20	
5	Розробка Web-додатку	15.04.20	03.05.20	
6	Тестування ПП	17.01.20	10.05.20	
6.1	Підготовка тестів	17.01.20	10.02.20	
6.2	Тестування методом білого ящика	18.03.20	10.05.20	
6.3	Тестування методом чорного ящика	06.05.20	09.05.20	
7	Підготовка документації	06.05.20	13.05.20	
8	Впровадження ПП	06.05.20	15.05.20	

Студент

(підпис)

Стегній Д. К.

Керівник роботи

(підпис)

Алексенко О. В.

РЕФЕРАТ

Тема роботи «Web-додаток підтримки організації змагань зі стрільби з лука».

Пояснювальна записка складається зі вступу, трьох основних розділів, висновків, списку використаних джерел з 17 найменувань та чотирьох додатків. Загальний обсяг пояснювальної записки складає 127 сторінок, в тому числі 29 сторінок основного тексту, 2 сторінки списку використаних джерел, 90 сторінок додатків.

В першому розділі було наведено результати аналізу процесу організації спортивних змагань та інформаційних систем. Виділені проблеми об'єкту автоматизації, проаналізована система-аналог та визначений перелік вимог до інформаційної системи.

В другому розділі сформульовано мету та задачі. Обрано засоби для реалізації on-premises додатку, Web-додатку та бази даних. Виконано структурно-функціональне моделювання процесу організації змагань у нотації IDEF0, створено модель потоків даних у формі DFD-діаграм та спроектовано систему у формі діаграми варіантів використання із використанням нотації UML.

В третьому розділі описано розробку інформаційної системи, що складається з on-premises додатку, Web-додатку та бази даних. Також наведені скріншоти роботи продукту.

Ключові слова: WEB-ДОДАТОК, БАЗА ДАНИХ, ON-PREMISES ДОДАТОК, ІНФОРМАЦІЙНА СИСТЕМА.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРОЦЕСУ ОРГАНІЗАЦІЇ СПОРТИВНИХ ЗМАГАНЬ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ЇХ ЗАБЕЗПЕЧЕННЯ	8
1.1 Загальна характеристика предметної області	8
1.2 Проблеми об'єкту автоматизації предметної області	9
1.3 Аналіз аналогу	10
1.4 Перелік вимог до інформаційної системи	12
1.5 Узагальнена постановка задачі створення програмних засобів для організації змагань зі стрільби із луку	13
2 ПОСТАНОВКА ЗАДАЧІ	15
2.1 Мета та задачі	15
2.2 Вибір засобів реалізації on-premises додатку та Web-додатку	16
2.3 Вибір системи управління базою даних	16
2.4 Проектування ІС	18
3 РОЗРОБКА	24
3.1 Розробка бази даних	24
3.2 Розробка додатку на платформі Windows	25
3.3 Розробка Web-додатку	32
ВИСНОВКИ	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	37
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ	39
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ	51
ДОДАТОК В. ЛІСТИНГ ДОДАТКІВ ТА СЦЕНАРІЮ СТВОРЕННЯ БД	63
ДОДАТОК Г. ІНСТРУКЦІЯ КОРИСТУВАЧА	111

ВСТУП

Швидкість зміни інформації постійно збільшується майже у всіх галузях діяльності людини. Така проблема дуже нагальна для користувачів інформації, які в ході виконання робочих процесів, керують одними і тими ж самими даними. Все частіше доступ до одних й тих же даних повинні мати кілька користувачів. Данна ситуація унеможлиблює варіант зберігання даних на кожній клієнтській машині, бо інформацію відкориговану одним користувачем, не буде видно іншим. Використання локальних копій даних можуть створювати документи з різним наповненням, але одного призначення, що може призвести до плутанини або дезінформованості. Для вирішення проблем такого типу на цей час використовуються Web-технології[1].

Web-засоби дозволяють зберігати данні на сервері, зокрема використовують бази даних (БД) для централізованого зберігання та обробки інформації. БД надає змогу доступу до інформації великій кількості людей, а для розмежування доступу та для покращення взаємодії з даними, створюються інтерфейси у вигляді Web-додатків або програм, що підключені до бази даних.

Усе зазначене актуальне і для інформації щодо спортивних заходів, з якою повинні працювати як спортсмени, так і організатори змагань, у т.ч. змагань зі стрільби з лука.

Виходячи із визначеної вище актуальності було сформульовано мету проекту – розробка програмного продукту (ПП) для підтримки процесу організації змагань зі стрільби з лука, який буде складатись із Web-додатку (сайту) та додатку на платформі Windows. Сайт надасть змогу розповсюджувати актуальну інформацію серед користувачів. А додаток на платформі Windows призначений для автоматизації процесу керування даними в БД.

Щоб виконати поставлену у роботі мету, потрібно розв'язати такі задачі:

- вивчити процеси організації змагань зі стрільби із лука та провести аналіз систем-аналогів для розроблення технічного завдання;
- обрати та налаштувати інструменти реалізації;
- визначити структуру даних та спроектувати веб-додаток і додаток на платформі Windows;
- реалізувати додатки та розробити інструкції користувача.

1 АНАЛІЗ ПРОЦЕСУ ОРГАНІЗАЦІЇ СПОРТИВНИХ ЗМАГАНЬ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ЇХ ЗАБЕЗПЕЧЕННЯ

1.1 Загальна характеристика предметної області

Розвиток спортивної стрільби з лука в Україні почався з 1956 року. За цей час українські лучники завоювали багато медалей, здобули перемоги та встановили рекорди чемпіонатів Європи, світу і Олімпіад. А тренери розробили новітні техніки та методики. Для розповсюдження інформації щодо змагань та іншої інформації стосовно стрільби із лука на цей час використовується сайт Львівської обласної федерації стрільби із лука (<http://www.archery.lviv.net/>) (рис. 1.1) [2].



Рисунок 1.1 – Головна сторінка сайту Львівської федерації

Процес організації змагань зі стрільби з лука складається з таких етапів:

- Створення календаря змагань (Назва змагання, Тип, Початок змагання, Кінець змагання);
- Прийом заяв на участь;
- Допущення до змагань (формування списку учасників).

Зараз даний процес зовсім не автоматизований, він займає багато часу та потребує багато людських ресурсів.

1.2 Проблеми об'єкту автоматизації предметної області

Проведений аналіз процесів організації змагань зі стрільби з лука показав, що в даній області не має відомих засобів автоматизації процесу організації. А сайт (рис 1.1) не відповідає всім вимогам замовника, не в повній мірі вирішує поставлені цілі та не містить потрібних можливостей.

Виявлений в ході аналізу перелік проблем області автоматизації наводиться у таблиці 1.1.

Таблиця 1.1 – Перелік проблем

№	Назва /Проблема	Зміст	Проблема, яка уточнюється	Примітка: роль/ чий інтерес
1	Інформованість	Важко дізнаватись інформацію, щодо змагань	Немає ПЗ для інформування	Спортсмени, тренери, організатори,
2	Друкарські помилки	Секретарі допускають помилки при заповненні звітів	ПЗ не поперджає виникнення помилок	Секретарі
3	Відсутність єдиного реєстру	Інформація про спортсменів зберігаються на різних носіях	Файли Excel не оновлюються одночасно	Організатори
4	Анкети	Паперові анкети відправляються в різному форматі і дуже важко вибрати потрібну інформацію	Немає ПЗ для електронного анкетування	Секретарі Тренери спортсмени

1.3 Аналіз аналогу

В рамках роботи над проектом було проведено аналіз подібної інформаційної системи. Для аналізу обрано систему Ianseο – це система розроблена для структурування результатів змагань зі стрільби з лука. Система складається з сайту, програм (на платформах Windows, Linux, MacOS) та бази даних. Було проаналізовано тільки сайт (<https://www.ianseo.net/>), оскільки лише він доступний для перегляду[3].

На головній сторінці (рис. 1.2) продемонстровані змагання, що проходять зараз та найближчі або недавно закінчені.

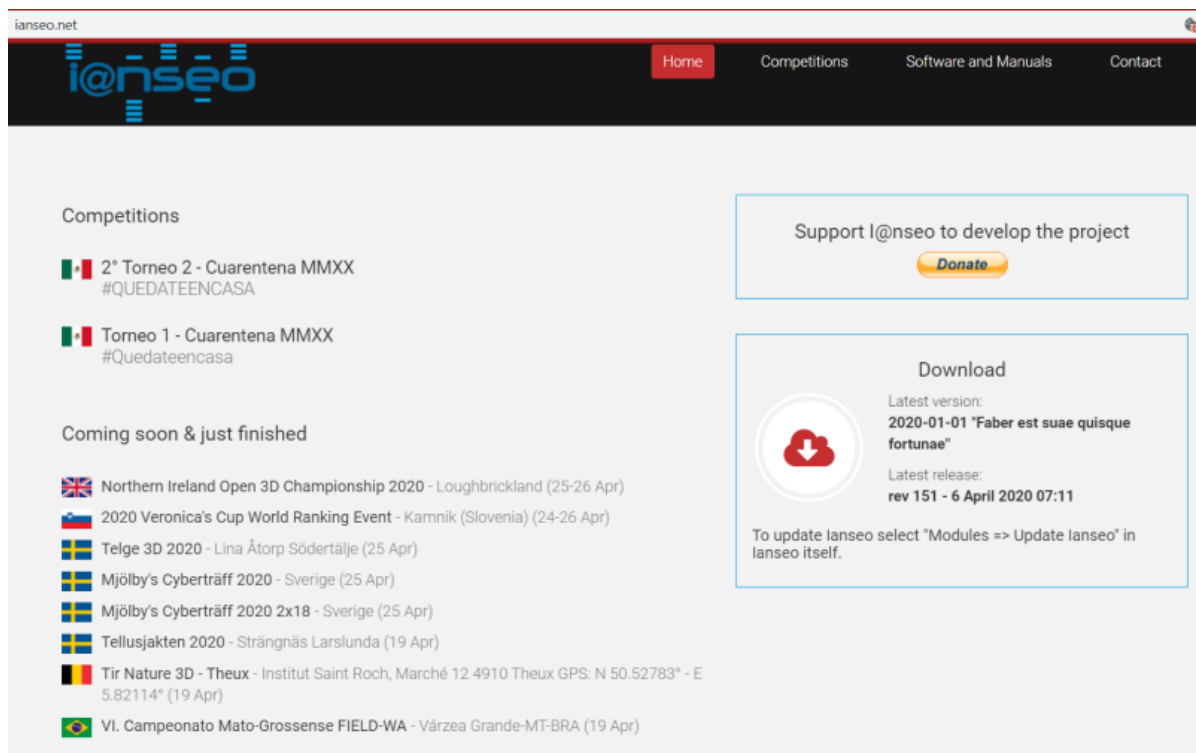


Рисунок 1.2 – Головна сторінка Ianseο

Після натискання на змагання відкривається сторінка з дисциплінами змагання (рис. 1.3).

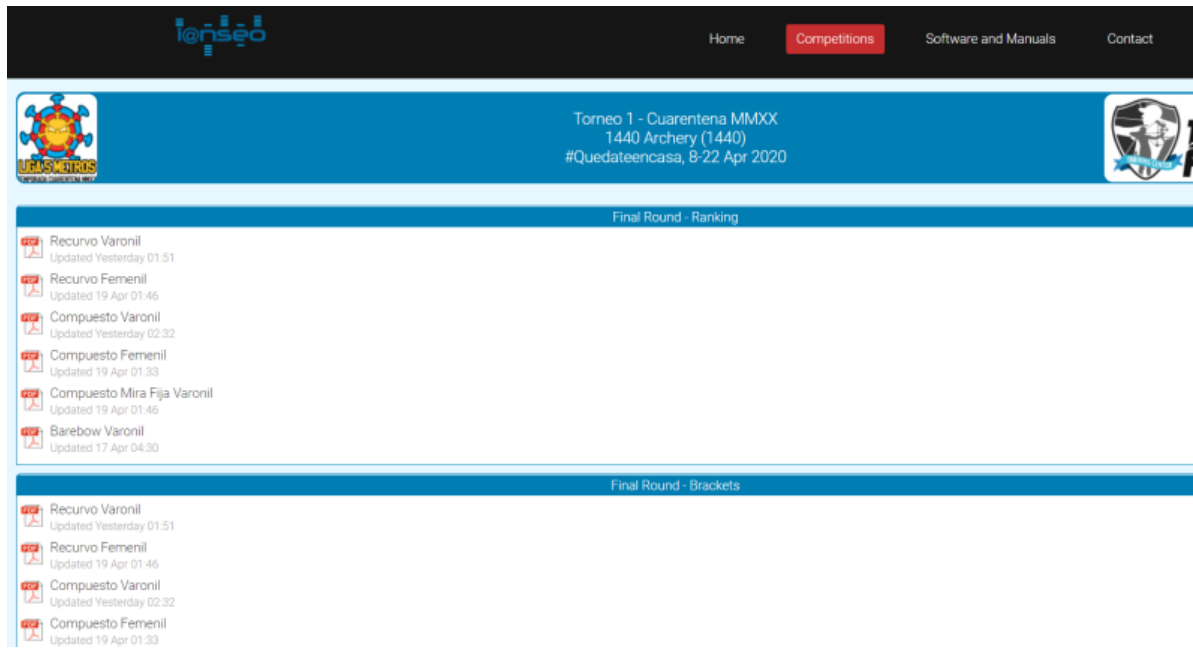


Рисунок 1.3 – Сторінка з дисциплінами

Натиснувши на дисципліну можна переглянути результати (рис. 1.4).

The screenshot shows the results page for the 'Compuesto Varonil' discipline. The table lists the following data:

Pos.	Atleta	P/A/C	Calif.	1/16	1/8	1/4	1/2	Oro
1	FAVELA Sinhue	SIN Sinaloa	1197-1 Pase directo	147	150	150	146	
2	ESPINOSA Adrian	SIN Sinaloa	1167-6 Pase directo	147	122	150	132	
3	SAN Carlos	SIN Sinaloa	1168-5 Pase directo	144	149	149	145	
4	CARRANZA David	SIN Sinaloa	1183-2 Pase directo	146	147	144	143	
5	ZAMORA Oldair	HGO Hidalgo	1147-9 Pase directo	146	147			
6	ZEPEDA Daniel	NL Nuevo Leon	1161-7 Pase directo	149	146			
7	OROZCO Rodrigo	NL Nuevo Leon	1099-13 Pase directo	1	139			
8	LOPEZ LEGY Jose	SIN Sinaloa	1181-3 Pase directo	147	119			
9	CERVANTES Diego	PUE Puebla	910-15 Pase directo	145				
9	CONTRERAS Julio	NL Nuevo Leon	1095-12 Pase directo	133				
9	GARCÍA Camilo	NL Nuevo Leon	1146-10 Pase directo	143				
9	GRANADOS Edgar	CDMX Cdmx	1160-8 Pase directo	140				
9	LEAL Jesus	COAH Coahuila	1170-4 Pase directo	0				
9	OLIVAS Alger	SIN Sinaloa	982-14 Pase directo	126				
9	PEREZ Sergio Arturo	NL Nuevo Leon	333-17	1	0			
9	PRIETO Arturo	PUE Puebla	1124-11 Pase directo	142				
17	PIMENTEL Carlos	NL Nuevo Leon	471-16	0				

Рисунок 1.4 – Сторінка з результатами

На сторінці «Competitions» (рис. 1.5) представлений календар всіх змагань. Для зручного пошуку на сторінці розміщені фільтри.

Competition	Organizer	Location	Date	Updated
Today's Competitions				
T202020	2º Torneo 2 - Cuarentena MMXX	1440 Archery	QUEDATEENCASA	17-27 Apr Today 07:21
T102020	Torneo 1 - Cuarentena MMXX	1440 Archery	Quedateencasa	8-22 Apr Yesterday 02:33
Other Competitions				
TSM2020	Tavel-SM 2020 Inne	Skurups Bågkytteklubb	Skurups Sperbank Arena	31 Oct - 1 Nov 5 Apr 05:16
25.10.20	Cumbria Alf Davis memorial WRS WA18 + Head to Head 2020	Cumbria County Archery Association	Penrith, Cumbria CA11 5JH	25 Oct 10 Feb 19:16
AFOSTM20	Österreichische Staatsmeisterschaften und Österreichische Meisterschaften Feld 2020	Österreichischer Bogenverband Ausschländer Verein, Waldviertler Bogenzentrum Zwettl	Waldviertler Bogenzentrum Zwettl, A-3910 Zwettl- Oberhof	3-6 Sep 19 Apr 20:22
DWC204	DWAA Champs American	DWAA	GWAC, Downton, Wilts	31 Aug 17 Apr 19:21
DWC203	DWAA Champs York/Hereford	DWAA	GWAC, Downton, Wilts	30 Aug 17 Apr 19:15
DWC203N	DWAA Champs National	DWAA	GWAC, Downton, Wilts	30 Aug 17 Apr 19:21
DWC201	DWAA Champs 2-Way Longbow Western	DWAA	GWAC, Downton, Wilts	29 Aug 17 Apr 19:19
DWC202	DWAA Champs Clout	DWAA	GWAC, Downton, Wilts	29 Aug 17 Apr 19:20

Рисунок 1.5 – Сторінка «Competitions»

Проаналізувавши сайт варто відмітити такі переваги:

- Інтуїтивно зрозумілий інтерфейс;
- Зручна навігація;
- Вичерпна інформативність.

1.4 Перелік вимог до інформаційної системи

З метою вирішення низки проблем, що були ідентифіковані під час аналізу предметної області (табл. 1.1.), розроблено вимоги до програмного забезпечення для організації змагань зі стрільби з луку, які наведені у таблиці 1.2.

Таблиця 1.2 – Перелік вимог до інформаційної системи

№	Назва	Зміст	Тип	Вимога, яка уточнюється	№	Зміст тесту
1	Інформованість	Доступ до інформації в програмному продукті повинен бути відкритим	Системна вимога	Вивід інформації на сторінку сайту	1	Безперебійне оновлення з бази даних

Продовження таблиці 1.2 – Перелік вимог до інформаційної системи

№	Назва	Зміст	Тип	Вимога, яка уточнюється	№	Зміст тесту
2	Друкарські помилки	ПЗ повинне попереджувати помилки	Вимога користувача	Данні повинні заповнюватись з бази	2	Перевірка попередження помилок
3	Відсутність єдиного реєстру	Інформація про спортсменів повинна зберігатись в одному місці	Вимога до підсистеми	Данні повинні зберігатись в базу даних	3	Безперебійний доступ до даних
4	Анкети	Формат анкет повинен бути стандартизований	Вимога до компонента	Додаток повинен мати форму анкети	4	Анкети заповнюються і відправляються

1.5 Узагальнена постановка задачі створення програмних засобів для організації змагань зі стрільби із луку

Виходячи із визначених вище проблем було визначено, що необхідно побудувати інформаційну систему «Змагання зі стрільби з луку», яка буде складатись із веб-додатку та додатку для ОС Windows і використовувати єдину базу даних (БД), яка повинна вирішувати наступні задачі:

- Ведення архіву спортсменів;
- Ведення календарю змагань;
- Ведення списків учасників;
- Видача інформації про спортсменів;
- Видача інформації про змагання;
- Ведення анкет на участь.

Обидва додатки повинні:

- Виводити інформацію про змагання та учасників;
- При заповненні списку учасників інформація про спортсменів повинна автоматично додаватись з архіву спортсменів(для попередження друкарських помилок);
- Містити форму для анкет.

Запропонована інформаційна система не має відомих аналогів в своїй сфері застосування. При необхідності додатки можуть бути перекваліфіковані для іншого виду спорту. Продукт проекту буде корисним та актуальним.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Мета та задачі

Виходячи із проведеного аналізу предметної області була сформульована мета даної роботи, яка полягає у створенні інформаційної системи (ІС) для забезпечення організації змагань зі стрільби з лука. ІС буде складатись із програмного on-premises додатку для платформи Windows, який буде містити засоби для ведення реєстру спортсменів, ведення календаря змагань, формування списку учасників, та Web-додатку, який повинен виводити актуальну інформацію про змагання та їх учасників, також повинен містити форму анкети учасника.

Вивчення поточного стану організації змагань дозволило визначити функціонал, який повинна забезпечити розроблювана ІС:

- Додавання, видалення та оновлення інформації про спортсменів виконується лише адміністратором через програмний додаток;
- Додавання, видалення та оновлення інформації про змагання виконуються лише адміністратором через програмний додаток;
- Додавання та видалення учасників виконується адміністратором через програмний додаток;
- Інформація з анкети співвідноситься з даними в базі автоматично, а додавання до списку учасників виконується адміністратором;
- Вивід актуальної інформації автоматично в додатки.

Повна та детальна інформація про вимоги до розробки програмного продукту представлена в технічному завданні (додаток А).

2.2 Вибір засобів реалізації on-premises додатку та Web-додатку

Для реалізації додатку на платформі Windows, було обрано систему програмування Visual Studio 2017, що забезпечує високоефективну роботу програміста, бо має широкий спектр засобів для проектування, запуску і тестування програм. Мовою програмування обрана мова C#. Visual Studio 2017 є системою швидкої розробки додатків що, містить повний набір драйверів для забезпечення доступу до всіх основних форматів БД, обладнана необхідними засобами для керування даними, що знаходяться як на локальному диску, так і на віддаленому сервері, а також великим набором візуальних компонентів для конструювання вікон, що необхідно для створення комфортної взаємодії між користувачем і виконуваним кодом.

Для реалізації Web-додатку було обрано текстовий редактор Sublime Text, а в якості мов програмування було обрано CSS, HTML, PHP, SQL. Мова PHP обрана для виконання програмних операцій у тому числі і зв'язування з базою даних. SQL необхідна для здійснення запитів до БД. Вибір Sublime Text зумовлений тим, що даний додаток володіє всіма необхідними інструментами та плагінами для роботи з обраними мовами програмування.

2.3 Вибір системи управління базою даних

В ході вибору інструменту роботи із даними було порівняно 3 найбільш популярних СУБД для вибору підходящої для реалізації завдання, а саме:

- SQLite;
- MySQL;
- PostgreSQL.

SQLite – це файлова база даних, що вбудовується в додаток, який її використовує[4]. Надає широкий набір засобів та інструментів для більш простого (на відміну від серверних СУБД) опрацювання даних.

Для підвищення швидкості виконання операцій, зв'язок між додатком і СУБД відбувається з виконанням безпосереднього виклику файлів, які містять данні, а не інтерфейсу.

Переваги:

- Для більш простого переміщення, БД зберігається в одному файлі;
- СУБД використовує мову SQL.

Недоліки:

- СУБД допускає одночасне виконання не більше однієї операції запису;
- Використовується лише одним додатком.

MySQL – найбільш популярна серверна база даних. Не зважаючи на те, що СУБД не намагається цілком слідувати стандартам SQL, MySQL надає широкий функціонал. Для роботи з БД, додатки використовують процес-демон.

Переваги:

- Підтримує основний функціонал SQL;
- СУБД містить вбудовані функції безпеки;
- СУБД може працювати з великими об'ємами даних;
- Підтримується більшою частиною веб-ресурсів.

Недоліки:

- Недостатня сумісність з SQL (можуть бути проблеми при інтеграції);
- Операції читання-запису виконані одночасно можуть викликати збій.

PostgreSQL – СУБД, направлена на цілковиту відповідність стандартам SQL та масштабованість. PostgreSQL містить об'єктно-орієнтований функціонал

та повністю підтримує концепт ACID (Atomicity, Consistency, Isolation, Durability).

Система управління основана на технології Postgres і тому відмінно виконує одночасну обробку декількох задач. Підтримка конкурентності розроблена з застосуванням MVCC (Multiversion Concurrency Control), що так само підтримує сумісність з концептом ACID.

Переваги:

- Повністю сумісна з SQL;
- Орієнтована на складні операції;
- Об'єктно-орієнтована.

Недоліки:

- В простих операціях уступає в швидкості конкурентам;
- Складна в освоєнні;
- Важко знайти провайдера.

Зважаючи на всі переваги та недоліки було обрано СУБД MySQL, бо це – серверна система управління (SQLite локальна) та на відміну від PostgreSQL більш розповсюджена та краще справляється з простими операціями.

2.4 Проектування ІС

Перед проектуванням розроблюваної системи було виконано моделювання бізнес-процесу організації змагань зі стрільби з лука у нотації IDEF0 (рис.2.1)[5,6,7]. Показана модель процесу AS-IS.

На контекстній діаграмі (рис.2.1) процесу визначені такі вхідні потоки:

- Календар змагань – данні щодо змагань;
- Зміни в розкладі – данні щодо можливих змін в розкладі;
- Анкети на участь – данні щодо анкет що відправляють;

- Списки учасників – данні щодо списку учасників;
- Данні про спортсменів – данні про спортсмена, що відомі організаторам.

Результатом процесу визначено звіти про результати змагань. Потоки управління та використані механізми вказані на рис.2.1.



Рисунок 2.1 – Контекстна діаграма бізнес-процесу організація змагань зі стрільби з лука

На рис. 2.2 представлена декомпозиція бізнес-процесу організації змагань зі стрільби з лука, у якій показані виділені підпроцеси: «Створення списків учасників», «Проведення змагання», «Упорядкування результатів».

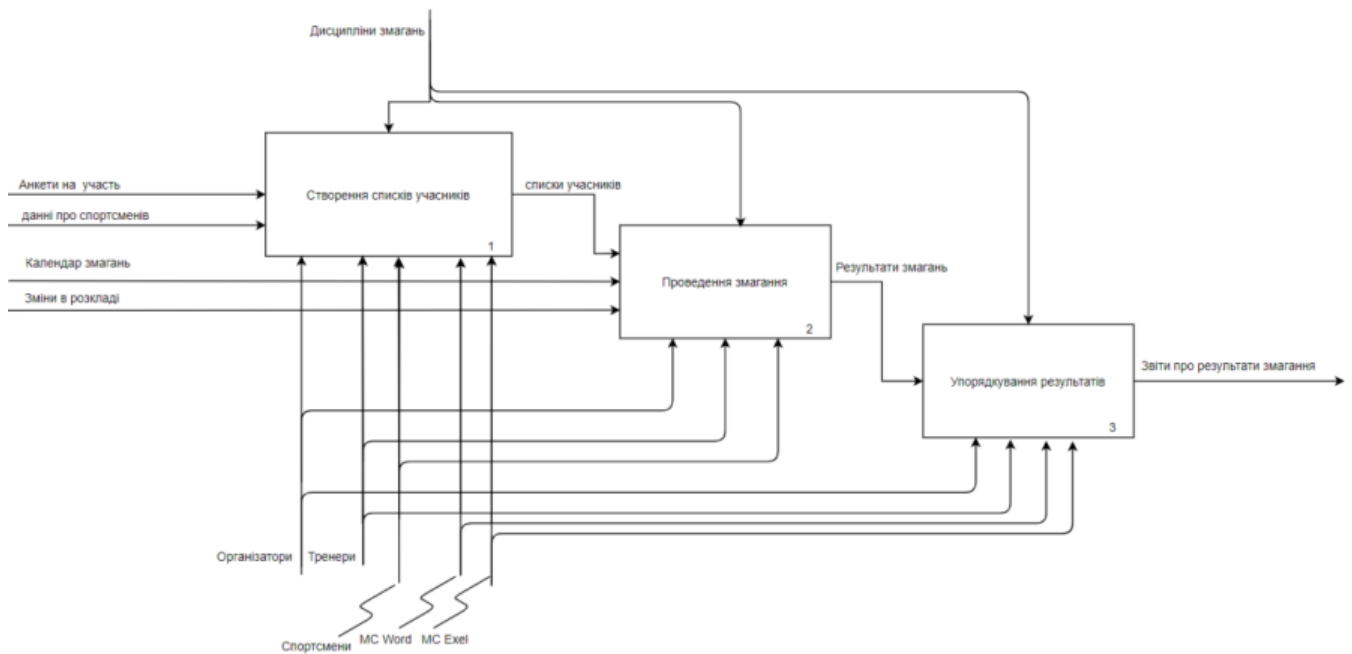


Рисунок 2.2 – Декомпозиція 1-го рівня контекстної діаграми бізнес-процесу організація змагань зі стрільби з лука

Для вивчення потоків даних, які використовуються в процесі організації змагань зі стрільби з лука, було виконано їх моделювання у формі діаграм потоків даних 0-го та 1-го рівнів, результати чого наведені на рис. 2.3 та 2.4.

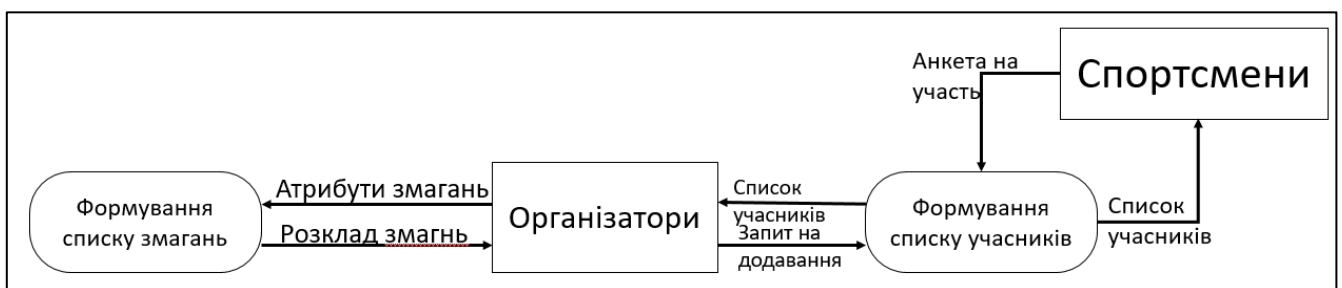


Рисунок 2.3 – DFD 0-го рівня

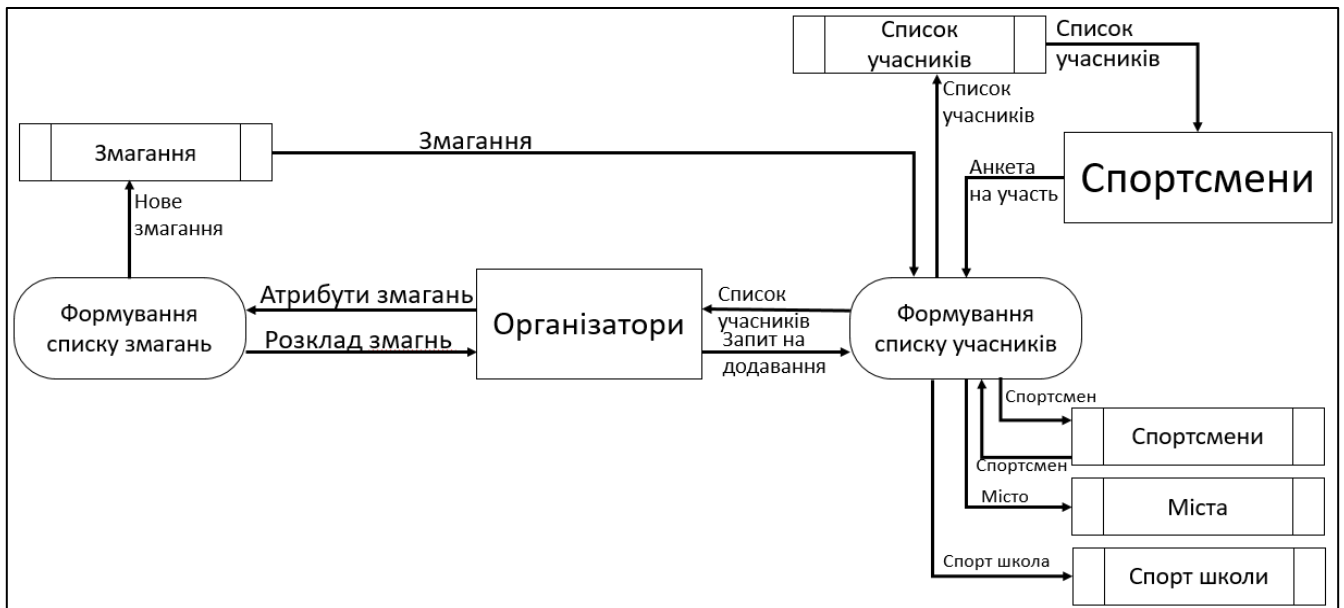


Рисунок 2.4 – DFD 1-го рівня

За визначеними у технічному завданні вимогами до розробки та моделю процесу організації змагань, поданою вище, було встановлено акторів та варіанти використання системи[8,9,10].

Актори:

- Адміністратор – користувач, що має повний доступ в системі;
- Гість – користувач, що має обмежений доступ в системі;
- Авторизований користувач – користувач що має обмежений доступ в системі;
- БД – база, що містить данні системи.

Варіанти використання:

- ВВ Додавання нової інформації про спортсменів – Програмний додаток додає данні про спортсменів до бази.
- ВВ Додавання нової інформації про змагання – Програмний додаток додає данні про змагання до бази.

- ВВ Додавання спортсменів до списку учасників – Програмний додаток додає новий запис до списку учасників в базі даних.
- ВВ Видалити інформацію про спортсменів – Програмний додаток видалляє данні про спортсменів з бази.
- ВВ Видалити інформацію про змагання – Програмний додаток видалляє данні про змагання з бази.
- ВВ Оновити інформацію про змагання – Програмний додаток оновлює данні про змагання в базі.
- ВВ Оновити інформацію про спортсменів – Програмний додаток оновлює данні про спортсменів в базі.
- ВВ Переглянути данні про змагання – Програмний додаток або Web-додаток виводить данні про змагання з бази.
- ВВ Переглянути данні про спортсменів – Програмний додаток або Web-додаток виводить данні про змагання з бази.
- ВВ Анкетування – Програмний додаток або Web-додаток додає анкетні данні до бази.

Діаграма, варіантів використання інформаційної системи, представлена на рис. 2.5. У системі окремо виділені ВВ, які реалізуються у windows-додатку та веб-додатку.

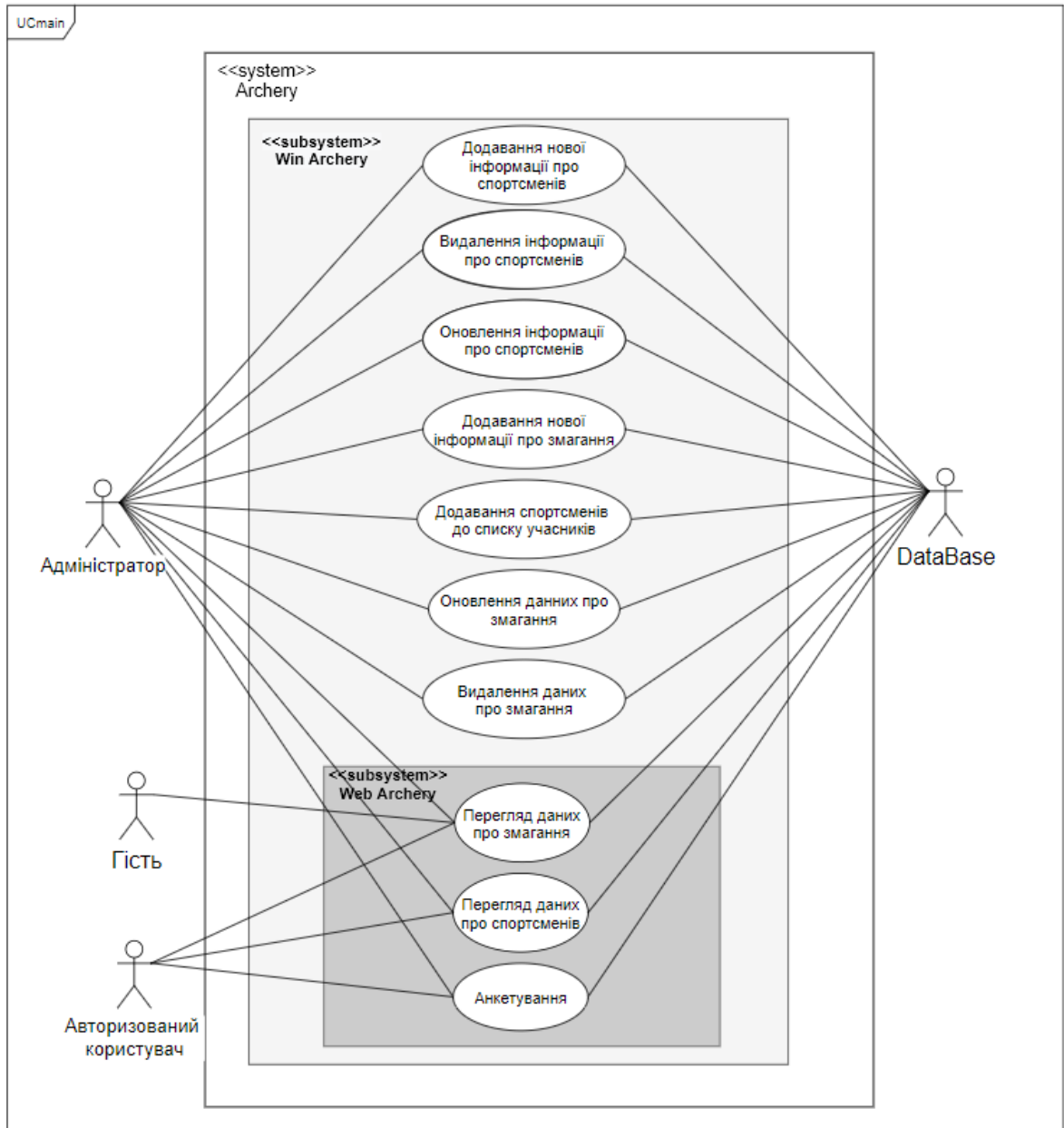


Рисунок 2.5 – Діаграма варіантів використання ІС

3 РОЗРОБКА

3.1 Розробка бази даних

На основі DFD діаграм (рис. 2.3-2.4) було спроектовано базу даних (рис.3.1)[11,12,13]. Як видно БД складається із 5 таблиць:

- Sity – призначена для зберігання унікальних назв міст;
- School – призначена для зберігання унікальних назв спортивних шкіл;
- Sportsman – призначена для зберігання даних про спортсменів;
- Comp – призначена для зберігання даних про змагання;
- List – призначена для формування списку змагань;
- Anketa – призначена для зберігання анкет на участь.

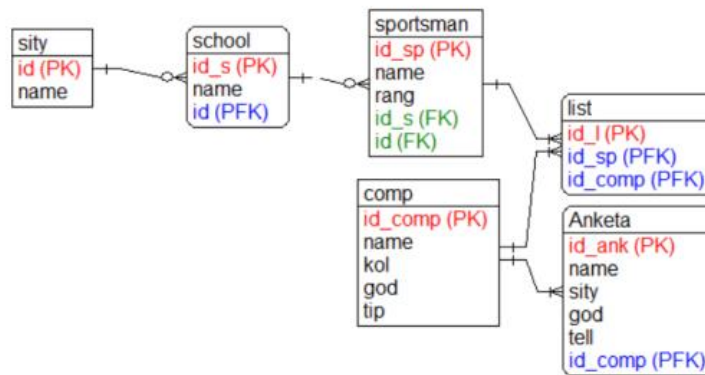


Рисунок 3.1 - ERD-діаграма

В таблиці 3.1 представлена структура бази даних[14,15].

Таблиця 3.1 – Структура БД

Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
Sity	Id_sity	Id	integer	PK	not null
	name	Назва міста	char(30)		not null

Продовження таблиці 3.1 – Структура БД

Таблиця	Таблиця	Таблиця	Таблиця	Таблиця	Таблиця
School	Id_s	Id	integer		
	Name	Назва дисципліни	char(30)		not null
	Id_sity	Id міста	integer	PFK	not null
sportsman	Id_sp	id	integer	PK	not null
	Name	ПІБ	varchar(50)		not null
	God	Рік народження	integer		not null
	Rang	Ранг	integer		not null
	Id_s	id школи	integer	PFK	not null
	Id_sity	Id міста	integer	PFK	not null
comp	Id_comp	id студента	integer	PK	not null
	Name	Назва змагання	char(30)		not null
	God	Мін. рік народження	char(30)		not null
	Tip	Тип	char(30)		not null
	Start	Початок	date		not null
	Final	Кінець	date		not null
list	Id_l	Id	integer	PK	not null
	Id_comp	Id змагання	integer	PFK	not null
	Id_sp	Id спортсмена	integer	PFK	not null

Сценарій створення структури бази даних наведено в «Додатку В».

3.2 Розробка додатку на платформі Windows

Для розробки додатку було обрано середовище Visual Studio 2017 та мову програмування C# (П.2.2). Першим кроком було створено проект «WindowsForms» (рис. 3.2).

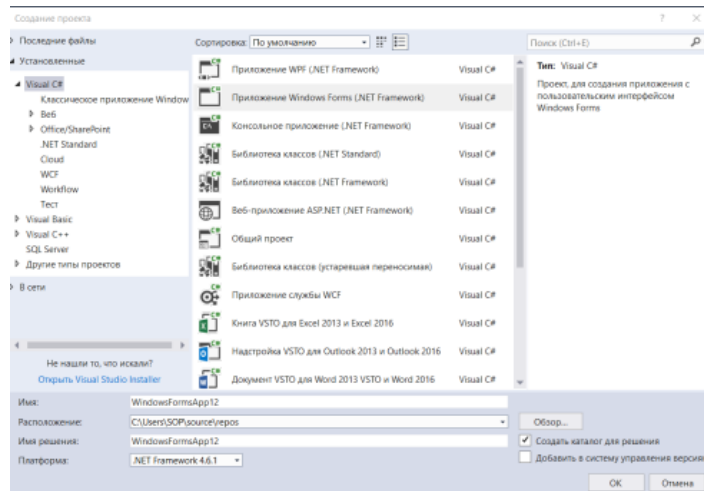


Рисунок 3.2 – Створення нового проекту

Відповідно до проектних рішень, поданих у Технічному завданні (додаток А), щодо користувацького інтерфейсу додатку було розділено на три розділи:

- Адміністративний розділ;
- Гостьовий розділ;
- Авторизація[16,17].

Першим було створено «Адміністративний розділ». Цей розділ розділений на шість вкладок (рис.3.3-3.8).

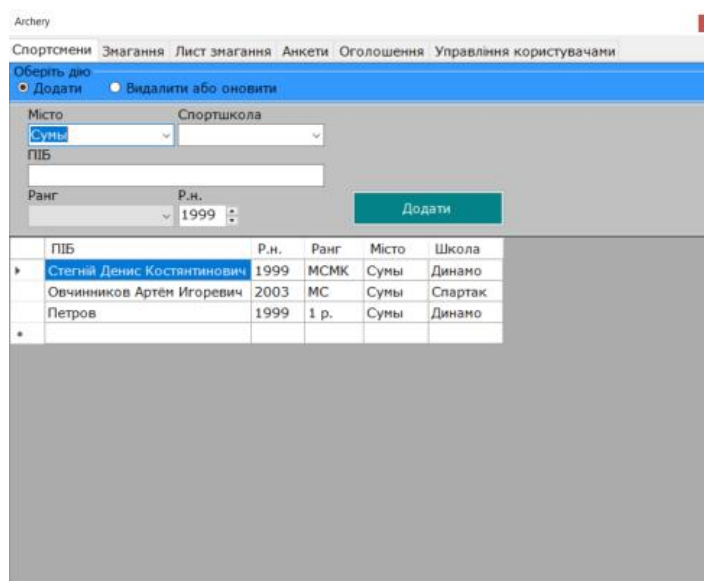


Рисунок 3.3 – Вкладка «Спортсмени»

Archery

Спортсмени Змагання Лист змагання Анкети Оголошення Управління користувачами

Оберть дію

Додати Видалити Оновити

Назва змагання

Тип Місто

Початок Кінець Р.н.

10.05.2020 10.05.2020 1999

Додати

№	Назва	Р.н.	Початок	Кінець	Місто	Тип
6	Київські Вогні	2002	01.12.2020	04.12.2020	Київ	Першість
5	Золота Осінь	1974	17.09.2020	30.09.2020	Львів	Міжнародн
7	Чемпіонат міста Суми серед юніорів	2000	25.06.2020	30.06.2020	Суми	Першість м
3	Кубок України серед дорослих	1998	03.05.2020	16.05.2020	Чернівці	Кубок
4	Кубок України серед юніорів	1999	03.05.2020	03.05.2020	Одеса	Кубок
1	Чемпіонат України серед кадетів	2002	22.06.2019	25.06.2019	Львів	Чемпіо
2	Чемпіонат України серед юніорів	1999	18.06.2019	21.06.2019	Нова Каховка	Чемпіо

Рисунок 3.4 – Вкладка «Змагання»

Archery

Спортсмени Змагання Лист змагання Анкети Оголошення Управління користувачами

Оберть змагання

id	Назва	Р.н.	Початок	Кінець	Місто	Тип
6	Київські Вогні	2002	01.12.2020	04.12.2020	Київ	Першість к
5	Золота Осінь	1974	17.09.2020	30.09.2020	Львів	Міжнародн
7	Чемпіонат міста Суми серед юніорів	2000	25.06.2020	30.06.2020	Суми	Першість м

Обрати змагання

Створити дос.

Список учасників змагання

Обрати спортсмена

Обраний спортсмен

ПІБ:

Р.н.: Ранг:

Операції

Допустити до змагань

Видалити з списку

Список спортсменів

Обрати спортсмена Місто Спортшкола

Рисунок 3.5 – Вкладка «Лист змагання»

Archery

Спортсмени Змагання Лист змагання Анкети Оголошення Управління користувачами

Оберіть змагання

id	Назва	Р.н.	Початок	Кінець	Місто	Тип
6	Київські Вогні	2002	01.12.2020	04.12.2020	Київ	Першість к
5	Золота Осінь	1974	17.09.2020	30.09.2020	Львів	Міжнародн
7	Червона Осінь	2000	25.05.2020	30.05.2020	Суми	Першість

Обрати змагання

Оберіть анкету

Обрати анкету

Анкета №
 ПІБ
 Р.н.
 Спорт школа

Н. тел.
 Місто

Операції
 Видалити анкету

Рисунок 3.6 – Вкладка «Анкети»

Archery

Спортсмени Змагання Лист змагання Анкети Оголошення Управління користувачами

Оберіть дію
 Додати Видалити Оновити

Оголошення

Обрати оголошення

Видалити

№	Оголошення	Автор
1	Через Covid19, всі назначені змагання переносяться ...	Адміністратор
2	Через Covid19, всі назначені змагання переносяться ...	Стегній Денис Костянтинович
3	Shoot CancellationDue to Covid19 the Ballyvally Com...	Адміністратор

Рисунок 3.7 – Вкладка «Оголошення»

Рисунок 3.8 – Вкладка «Управління користувачами»

Наступним було створено «Гостьовий розділ», що розділений на дві вкладки (рис. 3.11-3.12).

Рік	Тип	id	Назва	Р.н.	Початок	Кінець	Місто	Тип
2		2	Чемпіонат України серед юніорів	1999	18.06.2019	21.06.2019	Нова Каховка	Чемпі
1		1	Чемпіонат України серед кадетів	2002	22.06.2019	25.06.2019	Львів	Чемпі
3		3	Кубок України серед дорослих	1998	03.05.2020	16.05.2020	Чернівці	Кубок
4		4	Кубок України серед юніорів	1999	03.05.2020	03.05.2020	Одеса	Кубок
7		7	Чемпіонат міста Суми серед юніорів	2000	25.06.2020	30.06.2020	Суми	Перш

ПІБ	Р.н.	Ранг	Місто	Школа
Овчинников Артем Игоревич	2003	МС	Сумы	Спартак
Петров	1999	1 р.	Сумы	Динамо

Рисунок 3.9 – Вкладка «Змагання»

Місто	Ранг	ПІБ	Р.н.	Ранг	Місто	Школа
Сумы		Стегній Денис Костянтинович	1999	МСМК	Сумы	Динамо
		Овчинников Артем Игоревич	2003	МС	Сумы	Спартак
		Петров	1999	1 р.	Сумы	Динамо

[Авторизуватись](#)

Рисунок 3.10 – Вкладка «Спортсмени»

На рисунку 3.11 представлено розділ «Авторизація», який забезпечує розподіл доступу до функціоналу додатку.

Авторизація

Логін

Пароль

Рисунок 3.11 – Розділ «Авторизації»

В таблиці 3.2 описано елементи використані в додатку.

Таблиця 3.2 – Елементи користувацького інтерфейсу, використані в додатку

№	Елемент	Описання
1	Button	При клацанні на елемент відбувається подія
2	CheckBox	Дозволяє користувачу вибрати або видалити певний параметр.
3	ComboBox	Текстове поле з списком допустимих значень, що випадає.
4	DateTimePicker	Дозволяє користувачу обрати дату та час та відобразити їх в обраному форматі.
5	Label	Надає опис елемента управління чи інформацію під час виконання.
6	NumericUpDown	Відображує числове значення, яке користувач може збільшити або зменшити за допомогою кнопок ввєрх/вниз елемента управління.
7	RadioButton	Дозволяє користувачу обрати один з варіант з групи в комбінації з іншими RadioButton-ами
8	TextBox	Дозволяє користувачу вводити текст та забезпечує редагування одразу декількох рядків та маскування символів пароля.
9	GroupBox	Відображує рамку навколо групи елементів з можливістю включення заголовка.

Продовження таблиці 3.2 – Елементи використані в додатку

№	Елемент	Описання
10	TabControl	Відображує та оброблює для користувача відповідну колекцію вкладок, містять елементи управління та компоненти.
11	DataGridView	Відображує строки та стовбці даних в сітці, які користувач може змінити

Лістинг коду додатку представлено в «Додатку В». Інструкція користувача в «Додатку Г».

3.3 Розробка Web-додатку

За допомогою мов програмування html, php та css, вибір яких обгрунтовано у п.2.2, розроблено Web-додаток, що складається з шести сторінок (рис.3.12-3.17). Лістинг коду сайтів в «Додатку В». Інструкція користувача в «Додатку Г».



Рисунок 3.12 – Головна сторінка

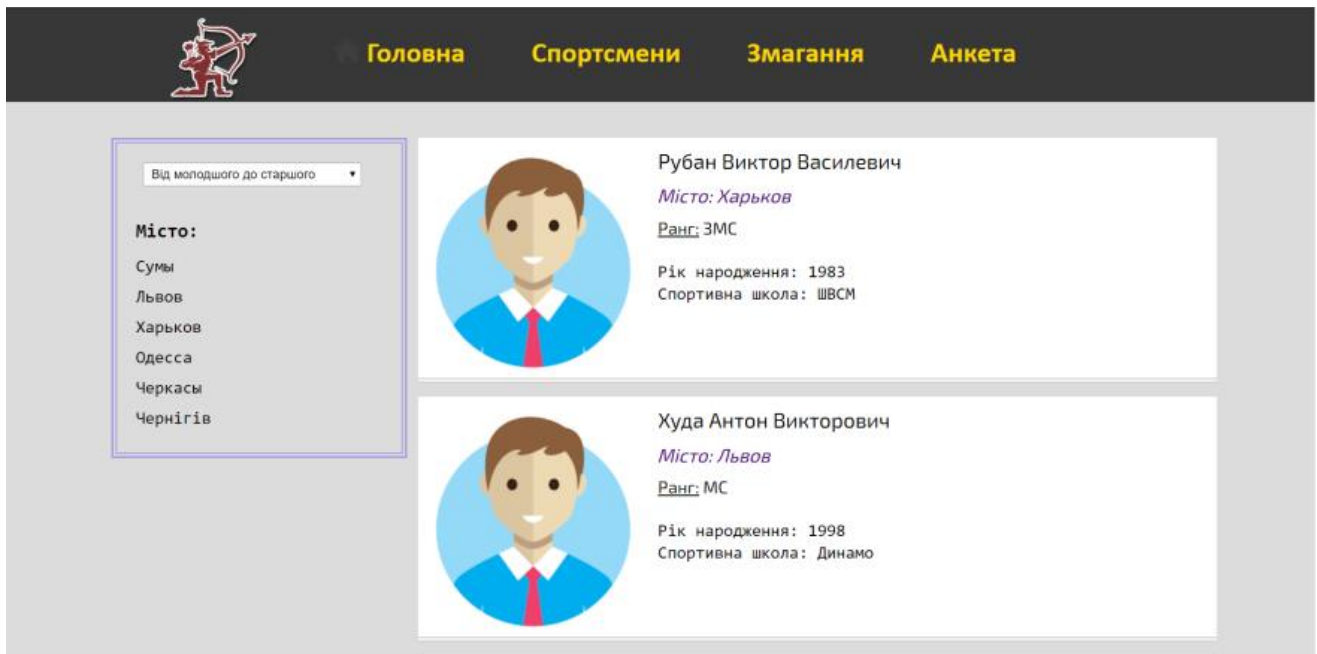


Рисунок 3.13 – Сторінка «Спортсмени»

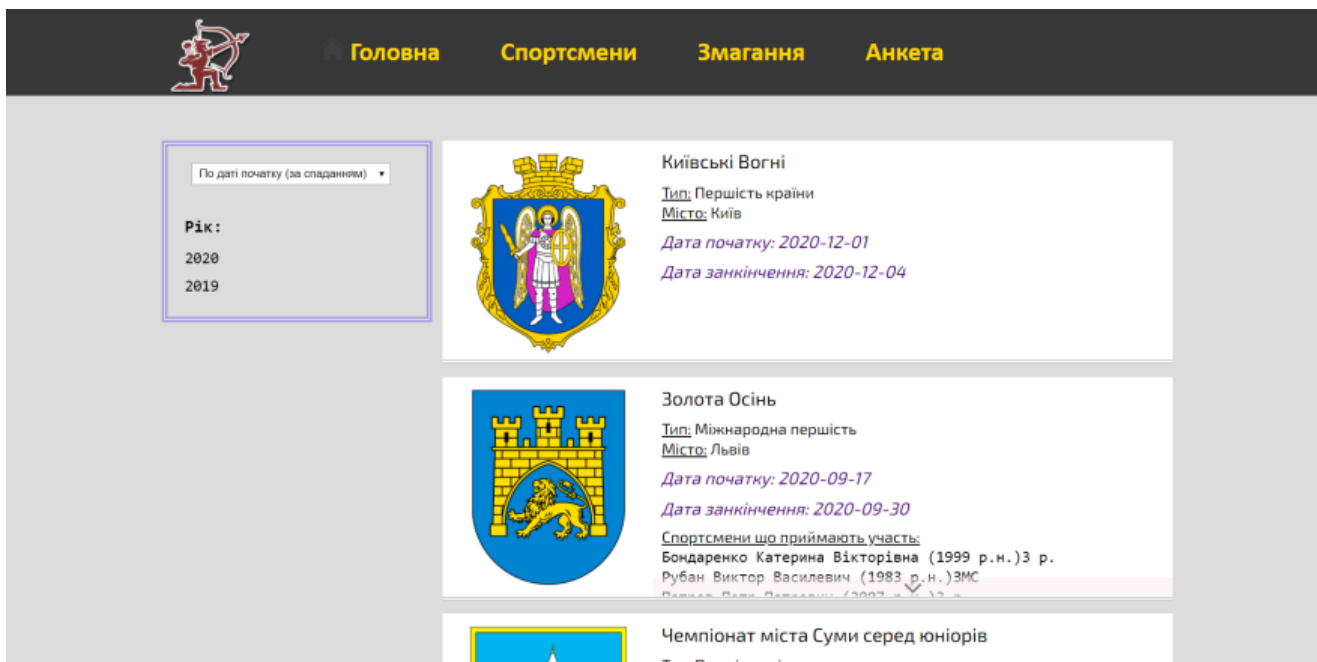


Рисунок 3.14 – Сторінка «Змагання»

АНКЕТА НА УЧАСТЬ

Змагання
Чемпіонат міста Суми серед юніорів

ПІБ

Рік народження
1999

Номер тел.

Спортивна школа

Місто

ВІДПРАВИТИ ЗАПИТ

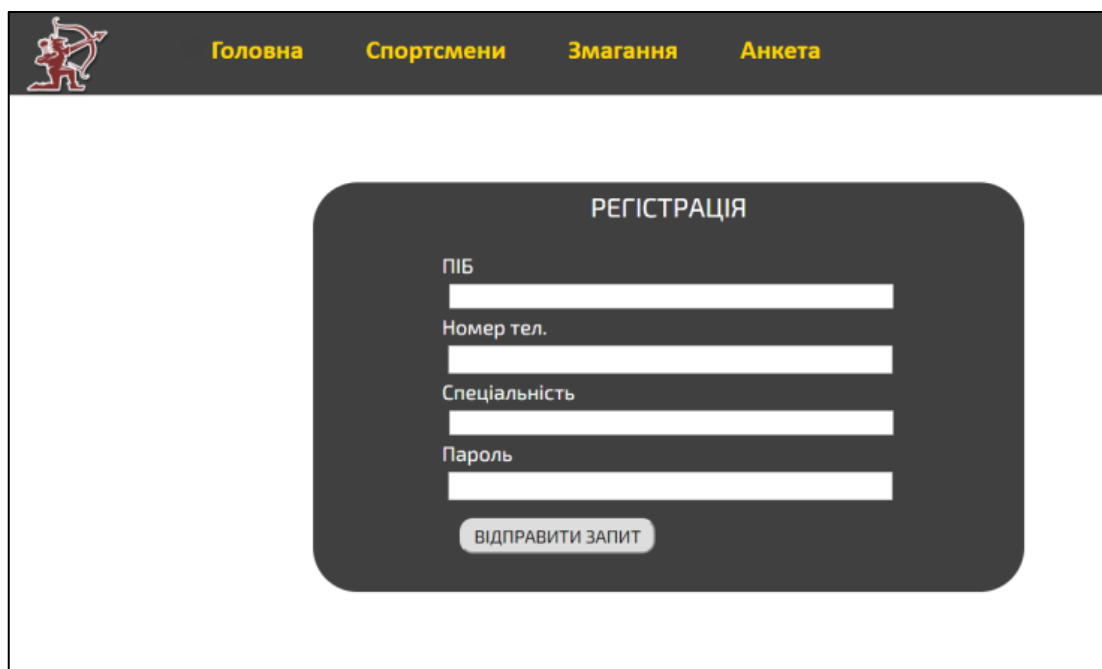
Рисунок 3.15 – Сторінка «Анкета»

Логін

Пароль

Ввійти

Рисунок 3.16 – Сторінка «Авторизації»



РЕГІСТРАЦІЯ

ПІБ

Номер тел.

Спеціальність

Пароль

ВІДПРАВИТИ ЗАПИТ

Рисунок 3.17 – Сторінка «Регістрації»

ВИСНОВКИ

В рамках дипломної роботи було вирішено актуальну проблему забезпечення федерації стрільби з лука засобом автоматизації процесів організації змагань. В результаті був розроблений ПП для підтримки процесу організації змагань зі стрільби з лука, який складається із Web-додатку та додатку на платформі Windows.

В ході роботи був проведений аналіз процесу організації змагань із стрільби з лука та визначено проблеми такі як:

- Важко дізнаватись інформацію, щодо змагань
- Секретарі допускають помилки при заповненні звітів
- Інформація про спортсменів зберігаються на різних носіях
- Паперові анкети відправляються в різному форматі і дуже важко вибирати потрібну інформацію.

За результатами моделювання процесу організації змагань із стрільби з лука були визначені вимоги до розроблюваної системи, які детально описані у розробленому технічному завданні.

Також за результатами аналізу предметної області було обрано інструменти для реалізації проекту (VisualStudio 2017, MySQL, Sublime Text).

Для забезпечення якості реалізації системи було виконане її проектування. Сформований проект складається із схеми даних, діаграми варіантів використання та прототипу інтерфейсу системи.

За результатами проведеної роботи було створено систему підтримки організації змагань зі стрільби із лука як сукупність on-premises додатку для платформи Windows та Web-додатку, які мають єдину БД.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Карпенко М. Ю. Технології створення програмних продуктів та інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 93 с.
2. Archery.lviv.net [Електронний ресурс] – Режим доступу до ресурсу: <http://www.archery.lviv.net/> (дата звернення: 07.04.2020).
3. Ianseo [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ianseo.net/> (дата звернення: 08.04.2020).
4. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems [Електронний ресурс] – Режим доступу до ресурсу: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems> (дата звернення: 05.04.2020).
5. Методология IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: <https://itteach.ru/bpwin/metodologiya-idef0> (дата звернення: 11.04.2020).
6. Нетепчук В. В. Управління бізнес-процесами / В. В. Нетепчук. – Рівне: НУВГП, 2014. – 158 с.
7. Allison L. Step-to-Step Guide - How to Create IDEF0 Diagram [Електронний ресурс] / Lynch Allison // edraw. – 2019. – Режим доступу до ресурсу: <https://www.edrawsoft.com/how-to-create-idef0-diagram.html>;
8. What is the use case? [Електронний ресурс] – Режим доступу до ресурсу: <https://searchsoftwarequality.techtarget.com/definition/use-case> (дата звернення: 12.04.2020).
9. UML Tutorial [Електронний ресурс] // tutorialspoint. – 2016. – Режим доступу до ресурсу: <https://www.tutorialspoint.com/uml/index.htm>;

10. Use Case Diagram [Электронный ресурс] // visual paradigm. – 2018. – Режим доступа до ресурсу: <https://online.visual-paradigm.com/diagrams/tutorials/use-case-diagram-tutorial/>;
11. ER Diagram [Электронный ресурс] // Creately. – 2019. – Режим доступа до ресурсу: <https://creately.com/blog/diagrams/er-diagrams-tutorial/>;
12. What is an Entity Diagram (ERD)? [Электронный ресурс] – Режим доступа до ресурсу: [<https://medium.com/@soni.dumitru/what-is-an-entity-relationship-diagram-erd-13daee5b2a>]
13. MySQL Database Diagram [Электронный ресурс] // devart. – 2018. – Режим доступа до ресурсу: <https://www.devart.com/dbforge/mysql/studio/database-designer.html>;
14. MySQL Primary Key [Электронный ресурс] // MySQL. – 2018. – Режим доступа до ресурсу: <https://www.mysqltutorial.org/mysql-primary-key/>;
15. Using Foreign Keys [Электронный ресурс] // MySQL. – 2018. – Режим доступа до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/example-foreign-keys.html>;
16. MVC Definition [Электронный ресурс] – Режим доступа до ресурсу: [<https://techterms.com/definition/mvc>].
17. Model view controller [Электронный ресурс] – Режим доступа до ресурсу: [<https://whatis.techtarget.com/definition/model-view-controller-MVC>].

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

1. Призначення й мета створення web-додатку

1.1. Призначення web-додатку

Додаток повинен забезпечувати діяльність організаторів змагань зі стрільби з луку щодо збереження та представлення інформації про змагання та результати учасників. Додаток повинен містити всі необхідні данні про змагання зі стрільби з луку та спортсменів(учасників).

1.2. Мета створення web-додатку –

Організація зберігання даних (п.2.2.1.3) про змагання та їх учасників в web-просторі, для підтримки та поширення тільки актуальної інформації.

1.3. Цільова аудиторія

У цільовій аудиторії web-додатку можна виділити наступні групи:

- Організатори змагань.
- Федерація стрільби з луку України.
- Тренери.
- Спортсмени.

2. Вимоги до web-додатку

2.1. Вимоги до web-додатку в цілому

2.1.1. Вимоги до структури й функціонування web-додатку

Система повинна бути реалізована у вигляді сайту та десктопного додатку, що пов'язані з базою даних. Сайт та додаток повинні складатися із взаємозалежних розділів (п.2.2.1.1) із чітко розділеними функціями.

2.1.2. Вимоги до персоналу

Для підтримки й експлуатації сайту та додатку від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним веб-браузером.

2.1.3. Вимоги до збереженні інформації

У системі керування сайтом повинен бути передбачений механізм резервного копіювання структури й вмісту бази даних. Процедура резервного копіювання повинна проводитися співробітником, відповідальним за підтримку сайту, не рідше 1 рази в тиждень. Резервне копіювання вмісту повинне здійснюватися вручну.

2.1.4. Вимоги до розмежування доступу

Користувачів сайту можна розділити на 3 групи відповідно до прав доступу:

- Відвідувачі
- Авторизований користувач
- Редактор (Адміністратор)

Відвідувачі мають доступ тільки до загальнодоступної частини сайту та додатку. А саме до календаря змагнь та оголошень.

Авторизований користувач може відправляти анкети на участь та передивлятися списки учасників.

Редактор може редагувати данні бази даних через десктопний додаток. Також додавати й видаляти розділи сайту.

Доступ до адміністративної частини повинен здійснюватися з використанням унікального логіна й пароля.

2.2. Вимоги до функцій, виконуваних сайтом та додатком

2.2.1. Основні вимоги

2.2.1.1. Структура сайту

Сайт повинен складатися з наступних розділів:

- Головна – виводяться данні про 3 найближчі змагання та важливі повідомлення.
- Змагання – виводяться данні про всі змагання та їх результати.
- Спортсмени – виводяться данні про спортсменів (структура описана у п.2.2.1.1.4).

- Анкета (структура описана у п.2.2.1.1.4) – авторизований користувач може відправити анкету на участь у змаганні.

2.2.1.1.1. Навігація

Користувацький інтерфейс сайту повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до розділів і сторінок. Навігаційні елементи повинні забезпечувати однозначне розуміння користувачем їх змісту: посилання на сторінки повинні бути мати заголовок, умовні позначки відповідати загальноприйнятим. Графічні елементи навігації повинні мати альтернативний підпис.

Для навігації повинна використовуватися система контент-меню, шапка сайту, що містить посилання на всі сторінки сайту. Макет продемонстровано на рисунках 2 - 5.

2.2.1.1.2. Наповнення сайту (контент)

Сторінки всіх розділів сайту повинні формуватися програмним шляхом на підставі інформації з бази даних на сервері.

2.2.1.1.3. Система навігації (карта сайту)

Взаємозв'язок між розділами й підрозділами сайту (карта сайту) представлено на рисунку 1.

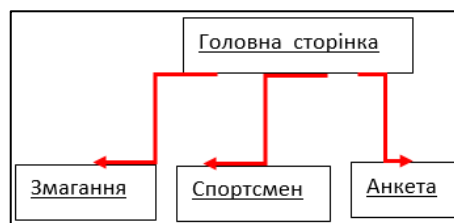


Рисунок 1– Карта сайту

2.2.1.1.4. Структура розділів

На головній сторінці будуть представлені наступні елементи:

- Данні про 3 найближчі змагання ;
- Оголошення;

На сторінці «Спортсмени» будуть представлені наступні елементи:

- Данні про спортсменів (описані у п.2.2.1.3);
- Списки учасників змагань;

На сторінці «Анкета» будуть представлені наступні елементи:

- Поля для заповнення анкети на участь у змаганнях (описані у п.2.2.1.3).

На сторінці «Змагання» будуть представлені наступні елементи:

- Данні про змагання та їх результати (описані у п.2.2.1.3).

2.2.1.1.5. Загальні вимоги

Оформлення не повинне защемляти інформативність: хоча на сайті не повинно бути досить багато графіки, він повинен бути зручний користувачам у плані навігації й цікавий для багаторазового відвідування.

Макети сторінок сайту наведені на рис. 3-5.

2.2.1.2. Структура додатку

Додаток повинен складатися з наступних розділів:

- Адміністративний розділ – керування даними бази даних .
- Гостьовий розділ – не авторизований користувач може передивитися інформацію про змагання.

2.2.1.2.1. Інтерфейс

Розташування елементів на головній сторінці сайту схематично показано на рис. 3.

Типові навігаційні й інформаційні елементи:

- Шапка сайту
- Основне поле контенту

Шапка сайту повинна містити логотип, назву сайту та посилання на всі розділи першого рівня (які представляють собою випадаючі списки з вкладеними в них підрозділами).

Основне поле контенту повинне розташовуватися в центрі сторінки. У цьому полі відображається основний зміст обраного розділу. Стильове оформлення матеріалів і їх елементів (посилань, заголовків, основного тексту, зображень, форм, таблиць і т.п.) повинне бути єдиним для всього веб-сайту.

<div style="display: flex; justify-content: space-between; padding: 5px;"> головна спортсмени анкета Змагання </div>			
Оголошення			
Оголошення			
Оголошення			
Назва змагання	Рік народження спортсмена	Місто проведення	Дата початку та завершення
Чемпіонат	2000	Нова Каховка	21.01.20-25.02.20

Рисунок 2 – Макет головної сторінки

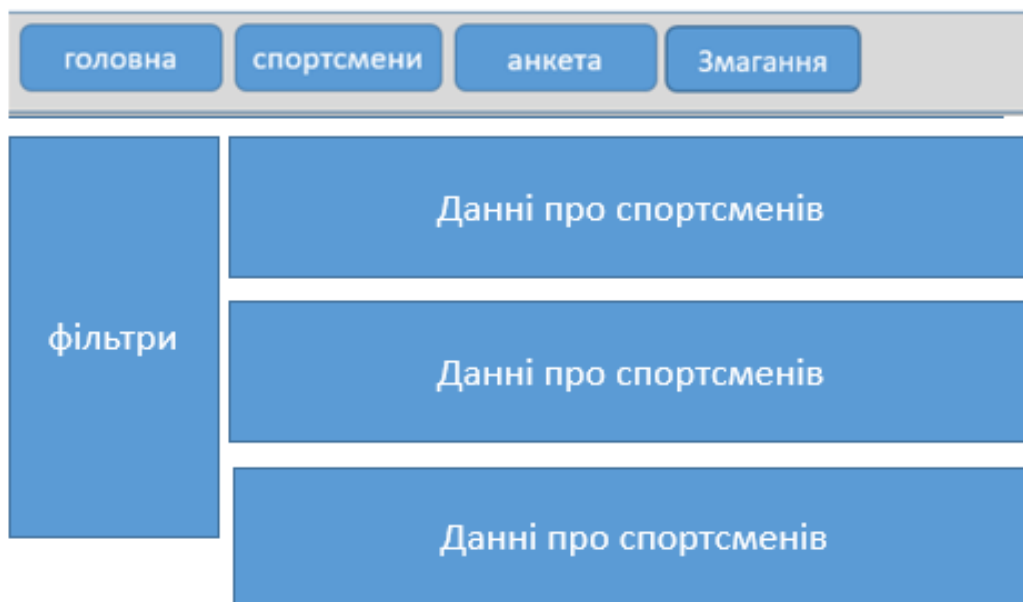


Рисунок 3 – Макет сторінки «Спортсмени»



Рисунок 4 – Макет сторінки «Змагання»

Рисунок 5 – Макет сторінки «Анкета»

2.2.1.2.2. Вимоги до функціональних можливостей

Система керування контентом (додаток) повинна надавати можливість додавання, редагування й видалення вмісту БД.

На рисунках 6 – 8 продемонстровано макети вкладок додатка.

Рисунок 6 – Макет додатка «Авторизація»

Рисунок 7 – Макети додатка «Гостьовий режим»

Рисунок 8 – Макети додатка «Адміністративна панель»

2.2.1.3. Вимоги до бази даних

Дані що будуть зберігатися в базі даних:

- Відомості про спортсменів (ПІБ, Рік народження, Ранг, Місто, Спортивна школа);
- Відомості про змагання (Назва змагання, Тип, Початок змагання, Кінець змагання);
- Список учасників;
- Логіни та паролі користувачів;
- Анкети(ПІБ, Рік народження, Ранг, Місто, Спортивна школа, Назва змагання).

2.3. Вимоги до видів забезпечення

2.3.1. Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням мов програмування:

- PHP
- MySQL
- HTML
- CSS

Реалізація додатку відбувається з використанням мови програмування C# та програми VisualStudio 2017

Реалізація бази даних відбувається з використанням MySql Workbench 8.0

2.3.2. Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;
- Включена підтримка javascript, Flash і cookies.

2.3.3. Вимоги до апаратного забезпечення

Апаратне забезпечення серверної частини повинне задовольняти наступним вимогам:

- Веб-сервер Apache з модулем MOD_REWRITE;
- Не менш 500 МБ вільного місця на диску.

3. Склад і зміст робіт зі створення проекту

Докладний опис етапів роботи зі створення проекту наведено в табл. 1.

Таблиця 1 – Етапи створення проекту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Проектування бази даних	1
2	Реалізація бази даних	2
3	Розробка каркасу: проектування розмітки та наповнення додатку	2
4	Авторизація: розробка та реалізація блоку Авторизації в додатку	1
5	Розробка та реалізація блоку Адміністративної панелі додатку: Оперування даними про спортсменів	2
6	Розробка та реалізація блоку Адміністративної панелі додатку: Оперування даними про змагання	2
7	Розробка та реалізація блоку Адміністративної панелі додатку: Оперування даними про учасників змагань	2
8	Розробка та реалізація блоку Гостьового режиму додатку	2
9	Розробка каркасу: проектування розмітки та наповнення сайту	1
10	Авторизація: розробка та реалізація блоку Авторизації на сайті	1
11	Розробка та реалізація блоку головної сторінки сайту: Блок оголошень	1

Продовження таблиці 1 – Етапи створення проекту

№	Склад і зміст робіт	Строк розробки (у робочих днях)
12	Розробка та реалізація блоку головної сторінки сайту: Календар найближчих змагань	1
13	Розробка та реалізація блоку сторінки сайту «Спортсмени»: Блок фільтрів для пошуку	1
14	Розробка та реалізація блоку сторінки сайту «Спортсмени»: Блок виводу результатів пошуку	1
15	Розробка та реалізація блоку сторінки сайту «Змагання»: Блок фільтрів для пошуку	1
16	Розробка та реалізація блоку сторінки сайту «Змагання»: Блок виводу результатів	1
17	Розробка та реалізація сторінки сайту «Анкета»	1
18	Завершення роботи: Проведення стилістичних виправлень веб-сайту, перевірка (тестування) реалізованого функціоналу	1
	Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту	25

4. Вимоги до складу й змісту робіт із введення сайту та додатку в експлуатацію

Для встановлення додатку та стабільної роботи, ПК повинен відповідати таким умовам:

- Операційна система Windows XP і вище;
- Процесор : 1.6 GHz (не менше);
- Оперативна пам'ять : 32 bit - 1 GB, 64 bit - 2 GB;
- Вільного місця на жорсткому диску : 3 GB;

- Підключення до мережі інтернет.

Додаток встановлюється за допомогою установочного файлу.

Для переносу сайту на хостинг необхідно, щоб параметри хостинга відповідали вимогам зазначеним в ТЗ (п.2.3.3). На хостинг переноситься програма (сайт), зверстаний шаблон дизайну й структура й наповнення бази даних з подальшою їх доробкою.

ДОДАТОК Б. Планування робіт

1 Планування змісту структури робіт IT-проекту (WBS)

WBS є засобом для поділу всіх робіт по проекту на керовані, що визначаються пакети робіт, що дозволяють досягти рівень деталізації інформації, що надається, який відповідає потребам керівництва проекту в контролі. WBS дозволяє визначити роботу по проекту з точки зору життєвого циклу проекту.

WBS дозволяє звести цілі проекту до ієрархії засобів їх досягнення, або отримання результатів, передбачених проектом. WBS є також інструментом, що дозволяє керівнику проекту отримати опис кінцевого результату (продукту, послуги) проекту і всіх підпроектів, в результаті яких буде досягнуто запланований результат. Далі WBS може розділятися (і результати поділятися) на частини для спеціалізації видів і обсягів робіт учасників проекту, координації їх дій і закріплення відповідальності за обсягами робіт, аж до рівня, що забезпечує керованість і належне адміністрування проекту.

На рисунку Б.1 представлено WBS структуру даного проекту, яка містить перелік операцій необхідний для реалізації поставленої мети.

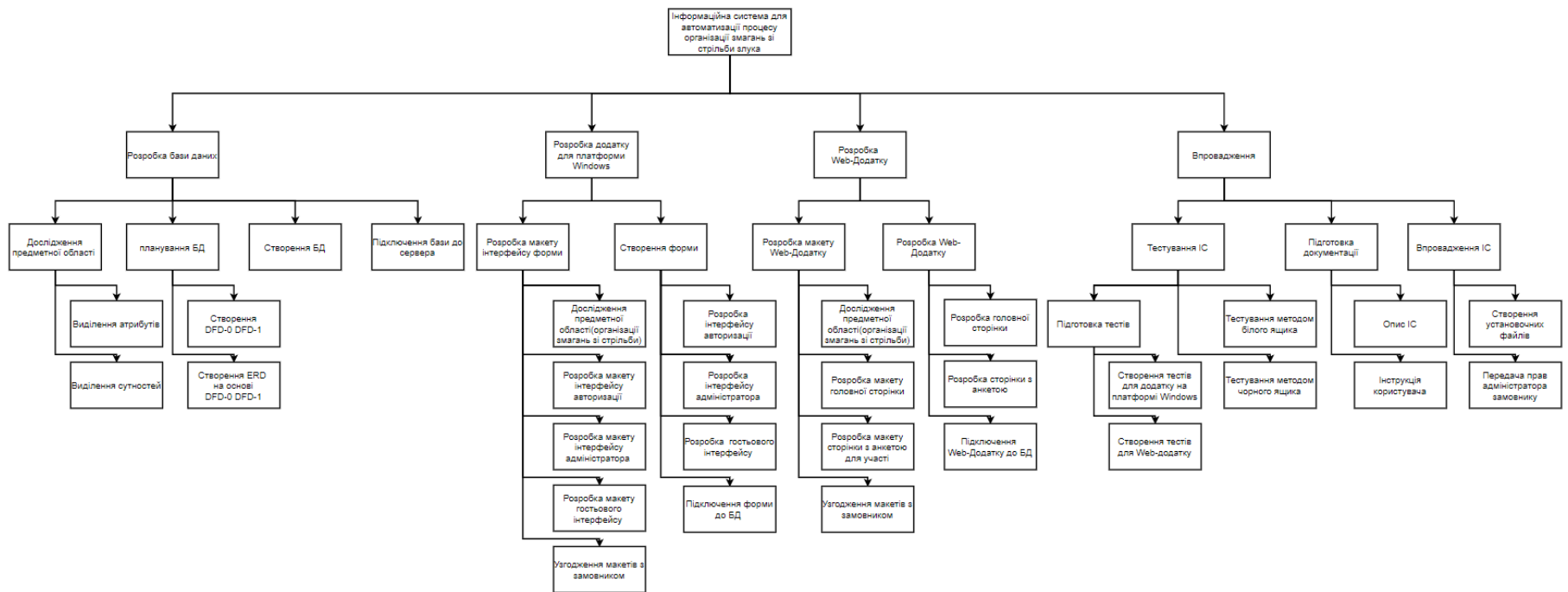


Рисунок Б.1 – WBS проекту «Організація змагань зі стрільби з лука»

2 Планування структури організації, для впровадження готового проекту (OBS)

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – є графічним відображенням учасників проекту (фізичних та юридичних осіб) та їхніх відповідальних осіб, залучених до реалізації проекту.

На рисунку Б.2 представлено OBS структуру даного проекту, яка включає перелік учасників та виконавців проекту відповідно до їх обов'язків.

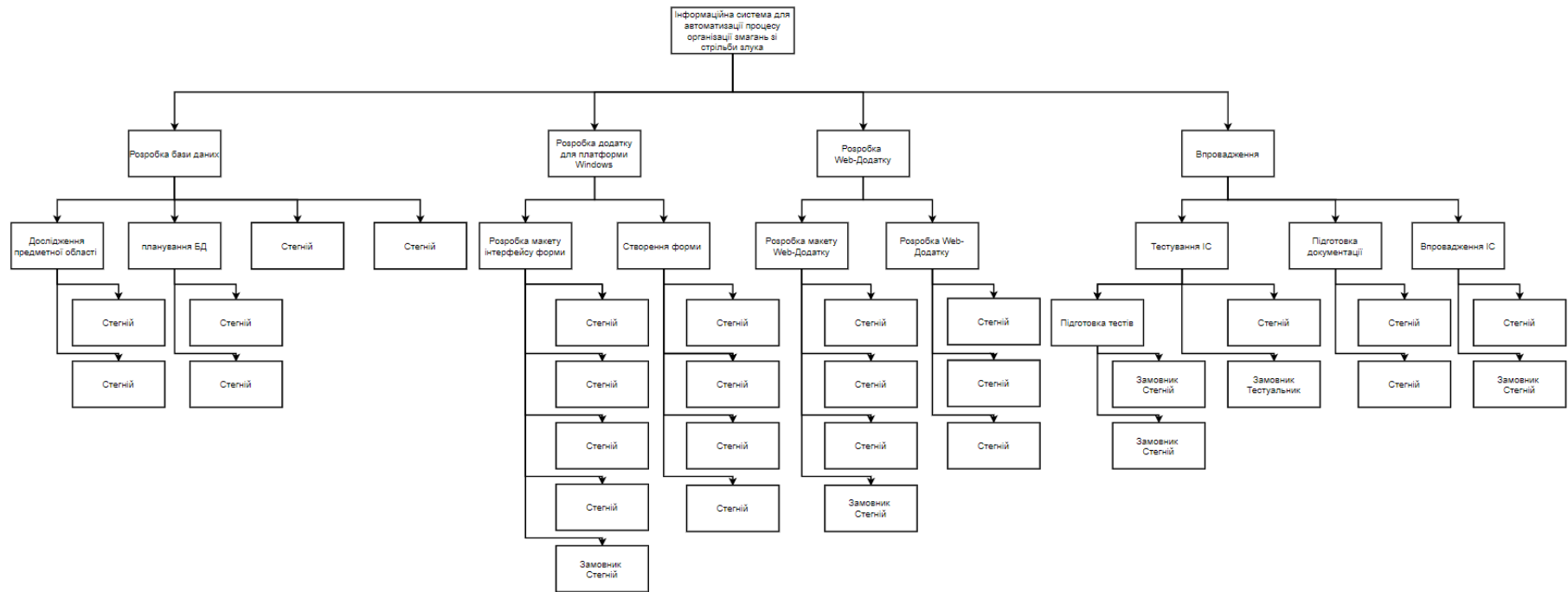


Рисунок Б.2 – OBS проекту «Організація змагань зі стрільби з лука»

3 Побудова матриці відповідальності

Матриця відповідальності забезпечує опис і узгодження структури відповідальності за виконання пакетів робіт. Вона являє собою форму опису розподілу відповідальності за реалізацію робіт по проекту, із зазначенням ролі кожного з підрозділів в їх виконанні.

На підставі OBS та WBS структур побудовано матрицю відповідальності проекту (табл. Б.1). Матриця відповідальності закріплює за кожною елементарною роботою виконавця.

Таблиця Б.1 - Матриця відповідальності

WBS/OBS		Тестувальник	Стегній	Замовник
Розробка БД	Дослідження предметної області	Виділення сутностей		
		Виділення атрибутів		
	Планування БД	Створення DFD-0 DFD-1		
		Створення ERD на основі DFD-0 DFD-1		
	Створення БД			
Підключення бази до сервера				
Розробка ПЗ для платформи Windows	Розробка макету інтерфейсу форми	Дослідження предметної області		
		Розробка макету інтерфейсу авторизації		
		Розробка макету інтерфейсу адміністратора		
		Розробка макету гостевого інтерфейсу		
		Узгодження макетів з замовником		
	Створення форми	Розробка інтерфейсу авторизації		
		Розробка інтерфейсу адміністратора		
		Розробка гостевого інтерфейсу		
		Підключення до БД		
Розробка Web-додатку	Розробка макету Web-додатку	Дослідження предметної області		
		Розробка макету інтерфейсу авторизації		
		Розробка макету інтерфейсу адміністратора		
		Розробка макету гостевого інтерфейсу		
		Узгодження макетів з замовником		
	Розробка Web-додатку	Розробка головної сторінки		
		Розробка сторінки з анкетною		
		Підключення Web-додатку до БД		

Продовження таблиці Б.1 - Матриця відповідальності

WBS/OBS			Тестувальник	Стегній	Замовник
Тестування ПП	Підготовка тестів	Створення тестів для ПЗ на платформі Windows			
		Створення тестів для Web-додатку			
	Тестування методом білого ящика				
	Тестування методом чорного ящика				
Підготовка документів	Опис ПП				
	Підготовка інструкції користувача				
Впровадження ПП	Передача прав адміністратора замовнику				
	Створення установочних файлів				

4 Розробка PDM-мережі

Мережевий графік — граф, який відображає роботи проекту, зв'язки між ними, стан проекту.

Може будуватися в 2-х варіантах:

а) вершини графа відображають стан деякого об'єкта (наприклад, будівництва), а дуги — роботи, що ведуться на цьому об'єкті.

б) вершини графа відбивають роботи, а зв'язки між ними — залежності між роботами.

PDM-мережа проекту дає уявлення про його критичний шлях і тривалість у робочих днях за умови, що нема обмежень на ресурси для виконання робіт проекту

На рисунку Б.3 представлена PDM-мережа виконання робіт.

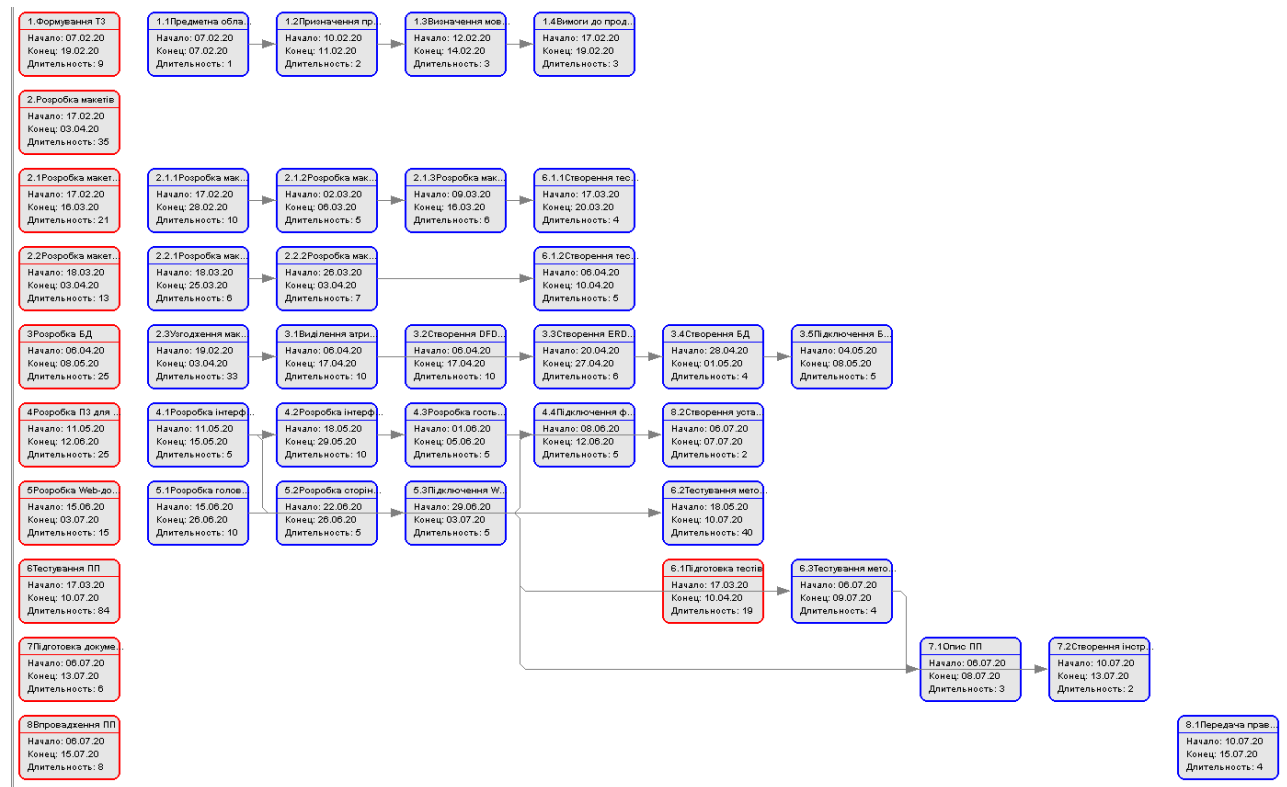


Рисунок Б.3 – PDM-мережа виконання робіт

5 Побудова календарного графіку виконання ІТ-проекту

Для того щоб мати реальне уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі, а також, проекту в цілому з урахуванням вихідних та святкових днів, будують календарний графік робіт. Він є реальним розподілом робіт з пакету за календарними датами, тобто своєрідним розкладом виконання робіт.

На діаграмі Гантта всі роботи за проектом представлені у вигляді горизонтальних відрізків, паралельних осі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Гантта також показується залежність між завданнями.

Використання моделі проекту, побудованої в вигляді діаграми Гантта, дозволяє контролювати і оптимізувати план виконання робіт, наочно відстежувати хід його виконання.

На наступному рисунку Б.4 представлено діаграму Гантта розроблюваного проекту.

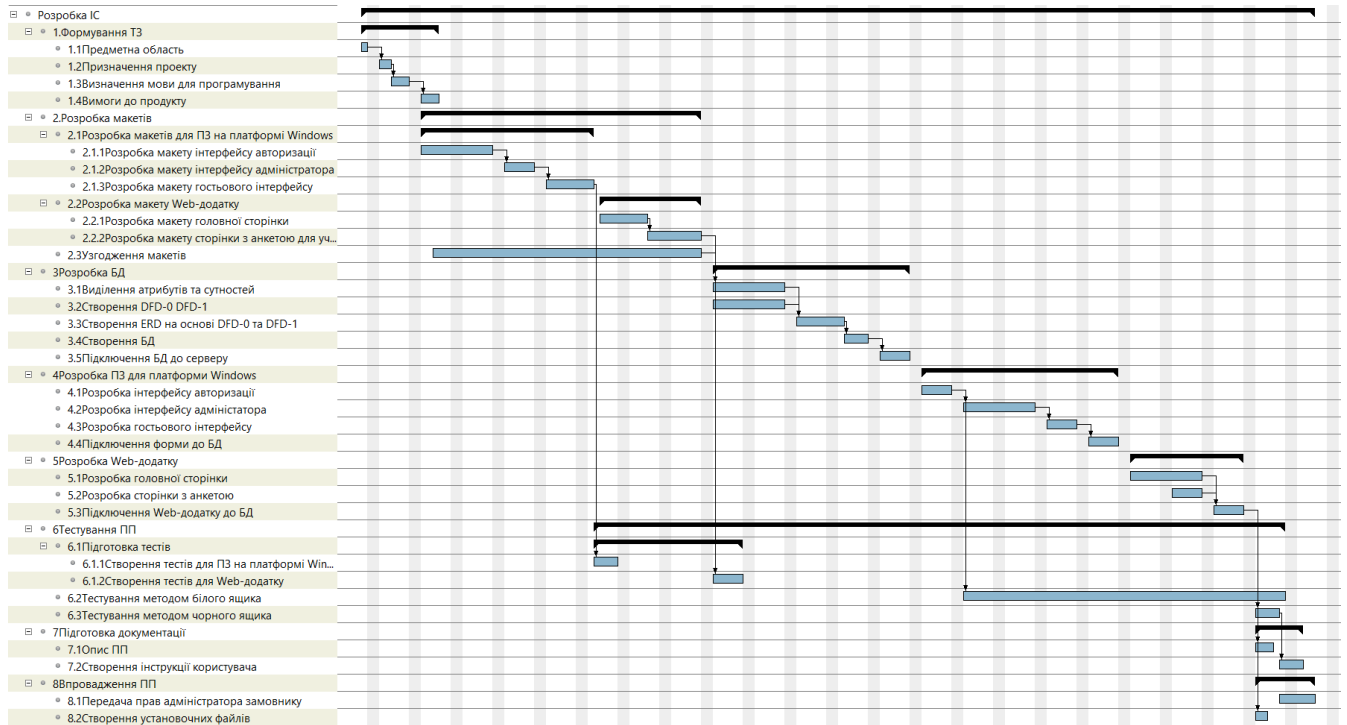


Рисунок Б.4 – Діаграма Гантта

6 Управління ризиками

Ризик — поєднання ймовірності та наслідків настання несприятливих подій. Також ризиком часто називають безпосередньо певну подію, здатну принести кому-небудь збиток.

Ідентифікація ризиків – це виявлення ризиків, здатних вплинути на проект, і документальне оформлення їх характеристик. Це ітеративний процес, який періодично повторюється на всьому протязі проекту, оскільки в рамках його життєвого циклу можуть виявлятися нові ризики.

Якісний аналіз передує кількісному. Він передбачає визначення факторів ризику, ідентифікацію потенційних областей виникнення ризику, виявлення напрямків діяльності та етапів, на яких може реалізуватися ризик. Протягом якісного аналізу також встановлюється можливість кількісної оцінки ризиків, реалізація яких може вплинути на діяльність підприємства.

Кількісна оцінка ризиків часто супроводжує якісну оцінку і також вимагає процес ідентифікації ризиків. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків.

Планування реагування на ризики – це процес розробки шляхів і визначення дій із збільшення можливостей і зниження погроз для цілей проекту. Даний процес зачинається після проведення якісного і кількісного аналізу ризиків.

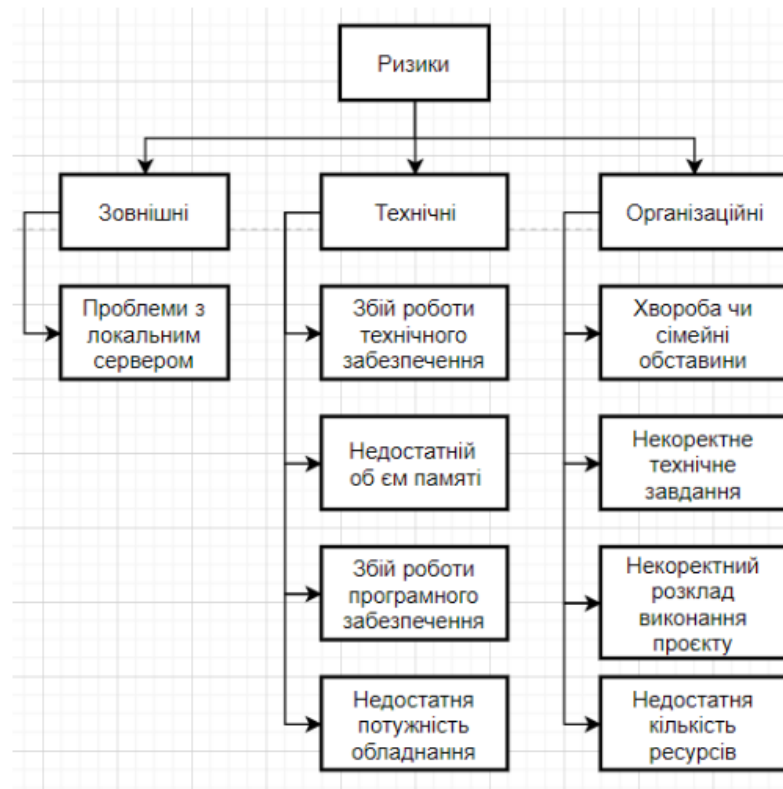


Рисунок Б.5 – RBS

Ймовірність виникнення:

- 1 Слабкоймовірно
- 2 Малоюмовірно
- 3 Ймовірно
- 4 Вельми ймовірно
- 5 Майже можливо

Величина втрат:

- 1 Мінімальна
- 2 Низька
- 3 Середня
- 4 Висока
- 5 Максимальна

Таблиця Б.3 – Ймовірність втрат

Ризик		ймовірність виникнення	величина втрат
Проблеми з локальним сервером	R1	3	5
Збій роботи технічного забезпечення	R2	3	2
Збій роботи програмного забезпечення	R3	2	5
Недостатня потужність обладнання	R4	2	2
Недостатній об'єм пам'яті	R5	2	2
Хвороба чи сімейні обставини розробника	R6	1	3
Некоректне технічне завдання	R7	3	3
Некоректний розклад виконання проекту	R8	3	3
Недостатня кількість ресурсів	R9	2	4

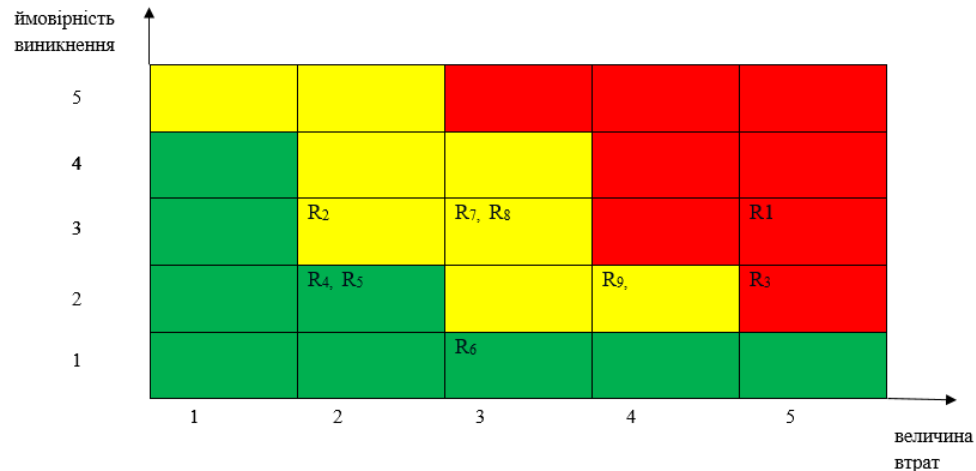


Рисунок Б.6 - Матриця ймовірності втрат

Класифікація за ступенем впливу:

- ігноровані ($1 \leq R \leq 4$);
- незначні ($5 \leq R \leq 8$);
- помірні ($9 \leq R \leq 11$);
- вагомі ($12 \leq R \leq 19$);
- критичні ($20 \leq R \leq 25$).

Класифікація за рівнем ризику:

- прийнятні ризики;
- виправданні ризики;
- недопустимі ризики;
-

Таблиця Б.4 – Класифікація за ступенем впливу та за рівнем ризику

Ризик		Ступінь впливу	Рівень ризику
Проблеми з локальним сервером	R1	15	недопустимі ризики
Збій роботи технічного забезпечення	R2	6	виправданні ризики
Збій роботи програмного забезпечення	R3	10	недопустимі ризики
Недостатня потужність обладнання	R4	4	прийнятні ризики
Недостатній об'єм пам'яті	R5	4	прийнятні ризики
Хвороба чи сімейні обставини розробника	R6	3	прийнятні ризики
Некоректне технічне завдання	R7	9	виправданні ризики
Некоректний розклад виконання проекту	R8	9	виправданні ризики
Недостатня кількість ресурсів	R9	8	виправданні ризики

План по усуненню ризиків:

- Вибір потужного обладнання для виконання проекту
- Періодичні поставки тестових версій ПП замовнику
- Безперервна взаємодія з замовником
- Резервувати час на випадок помилок планування та виникнення непередбачених обставин.
- Використовувати програми страхування технічних ризиків.
- Ретельний вибір інструментів виконання проекту.

ДОДАТОК В. ЛІСТИНГИ ДОДАТКІВ ТА СЦЕНАРІЮ СТВОРЕННЯ БД

Сценарій створення структури бази даних

```
CREATE TABLE `anketa` (  
  `id_ank` int(11) NOT NULL,  
  `id_comp` int(11) NOT NULL,  
  `name` varchar(30) NOT NULL,  
  `tell` varchar(10) NOT NULL,  
  `god_r` varchar(4) NOT NULL,  
  `school` varchar(10) NOT NULL,  
  `sity` varchar(30) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `att` (  
  `id_att` int(11) NOT NULL,  
  `txt` longtext NOT NULL,  
  `id_kor` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `comp` (  
  `id_comp` int(11) NOT NULL,  
  `name` char(50) DEFAULT NULL,  
  `god` char(4) DEFAULT NULL,  
  `tip` char(50) DEFAULT NULL,  
  `start` date NOT NULL,  
  `final` date NOT NULL,  
  `sity_comp` varchar(30) NOT NULL,  
  `god_comp` varchar(4) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `cook` (  
  `id` int(11) NOT NULL,  
  `cook` varchar(4) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `list` (  
  `id_l` int(11) NOT NULL,  
  `id_sp` int(11) NOT NULL,  
  `id_comp` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `login` (  
  `id` int(11) NOT NULL,  
  `login` varchar(30) NOT NULL,  
  `password` varchar(30) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

`id_kor` varchar(11) NOT NULL,
`spec` varchar(30) NOT NULL,
`pass` varchar(30) NOT NULL,
`name` varchar(30) NOT NULL,
`cook` varchar(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `reg` (
  `id_reg` int(11) NOT NULL,
  `spec` varchar(30) NOT NULL,
  `pass` varchar(30) NOT NULL,
  `name` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `school` (
  `id_s` int(11) NOT NULL,
  `name` char(10) DEFAULT NULL,
  `id_sity` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `sity` (
  `id_sity` int(11) NOT NULL,
  `name` char(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `sportsman` (
  `id_sp` int(11) NOT NULL,
  `name` char(50) DEFAULT NULL,
  `god` varchar(4) NOT NULL,
  `rang` char(10) DEFAULT NULL,
  `id_s` int(11) NOT NULL,
  `id_sity` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Лістинг коду додатку на платформі Windows

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MySql.Data.MySqlClient;
using Word = Microsoft.Office.Interop.Word;
using System.Windows.Forms;

namespace Praktika

```



```

{
public partial class Form1 : Form
{
    string conn = "server=localhost;user=root;database=sport;password=";
    public string ank_nom="";
    public string ank_comp;
    public string ank_name;
    public string ank_school;
    public string ank_tel;
    public string ank_sity;
    public string ank_god;
    public string list_comp="";
    public string list_sport= "";
    public string up_comp = "";
    public string del_comp = "";
    public string comp_comp = "";
    public string kor_id = "0996159924";
    public string kor_name = "Адміністратор";
    public string att_id = "";
    public string log_id = "";
    public string reg_id = "";
    public int kor = 0;
    // string conn = "server=serverdenisa2.mysql.database.azure.com; user=stehni;
database=dbdenis;password=Denis1234; ";
    //string conn = "server=serverdenisa.database.windows.net;
user=azure_superuser;database=dbdenis;password=Denis1234;";
    public Form1()
    {
        InitializeComponent();
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();

        string s1 = "SELECT Name FROM sity";

        MySqlCommand comm1 = new MySqlCommand(s1, conn2);
        MySqlDataReader reader1 = comm1.ExecuteReader();
        while (reader1.Read())
        {
            comboBox1.Items.Add(reader1[0].ToString());
            comboBox8.Items.Add(reader1[0].ToString());
            comboBox12.Items.Add(reader1[0].ToString());
        }
        reader1.Close();

        string s2 = "SELECT distinct Name FROM comp";

        MySqlCommand comm2= new MySqlCommand(s2, conn2);
        MySqlDataReader reader2 = comm2.ExecuteReader();
        while (reader2.Read())
        {
            comboBox6.Items.Add(reader2[0].ToString());
        }
        reader2.Close();

        string s21 = "SELECT distinct sity_comp FROM comp";

        MySqlCommand comm21 = new MySqlCommand(s21, conn2);
        MySqlDataReader reader21 = comm21.ExecuteReader();
        while (reader21.Read())
        {
            comboBox7.Items.Add(reader21[0].ToString());
        }
        reader21.Close();
    }
}

```

```

string s3 = "SELECT distinct tip FROM comp";

MySQLCommand comm3 = new MySQLCommand(s3, conn2);
MySQLDataReader reader3 = comm3.ExecuteReader();
while (reader3.Read())
{
    comboBox2.Items.Add(reader3[0].ToString());
}
reader3.Close();
textBox2.Items.Add("1 p.");
textBox2.Items.Add("2 p.");
textBox2.Items.Add("3 p.");
textBox2.Items.Add("KMC");
textBox2.Items.Add("MC");
textBox2.Items.Add("MCMK");
textBox2.Items.Add("3MC");
}
private void Form1_Load(object sender, EventArgs e)
{
    groupBox1.Visible = false;
    dataGridView1.Visible = false;
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox4.Items.Clear();
    comboBox4.Text = "";
    comboBox3.Items.Clear();
    comboBox3.Text = "";
    if (radioButton2.Checked)
    {
        textBox2.Text = "";
    }
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string s1 = "SELECT school.Name FROM school where id_sity= (select id_sity
from sity where name ='" + comboBox1.Text+"'");
    MySQLCommand comm1 = new MySQLCommand(s1, conn2);
    MySQLDataReader reader1 = comm1.ExecuteReader();
    while (reader1.Read())
    {
        comboBox4.Items.Add(reader1[0].ToString());
    }
    reader1.Close();
    string s2 = "SELECT Name FROM sportsman where id_sity= (select id_sity from
sity where name ='" + comboBox1.Text + "')";
    MySQLCommand comm2 = new MySQLCommand(s2, conn2);
    MySQLDataReader reader2 = comm2.ExecuteReader();
    while (reader2.Read())
    {
        comboBox3.Items.Add(reader2[0].ToString());
    }
    reader2.Close();
    dataGridView1.Visible = true;
    DataSet ds = new DataSet();
    string sql4 = "SELECT sportsman.name as 'ИИБ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Метро',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox1.Text + "'";
    MySQLDataAdapter dataAdapter = new MySQLDataAdapter(sql4, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView1.DataSource = ds.Tables["Users"];
}
private void button1_Click(object sender, EventArgs e)

```

```

    {
        if ( textBox1.Text == "" || textBox2.Text == "" || comboBox1.Text == "" ||
comboBox4.Text == "" )
        {
            MessageBox.Show("Необхідно заповнити всі поля");
        }
        else
        {
            try
            {
                string name = textBox1.Text;
                string rank = textBox2.Text;
                string god = Convert.ToString(numericUpDown1.Value);
                string sity = comboBox1.Text;
                string sch = comboBox4.Text;
                int id = 0;
                int id2 = 0;
                int m = 0;
                int n = 0;
                dataGridView1.Visible = true;
                MySqlConnection conn2 = new MySqlConnection(conn);
                conn2.Open();
                string s = "select sity.name, school.name from sity inner join school on
school.id_sity=sity.id_sity ";
                MySqlCommand comm1 = new MySqlCommand(s, conn2);
                MySqlDataReader reader3 = comm1.ExecuteReader();
                while (reader3.Read())
                {
                    if (reader3[0].ToString() == sity)
                    {
                        m = 1;
                    }
                    if (reader3[1].ToString() == sch && reader3[0].ToString() == sity)
                    {
                        n = 1;
                    }
                }
                reader3.Close();

                if (m != 1)
                {
                    string s4 = "select max(id_sity) from sity";
                    MySqlCommand comm4 = new MySqlCommand(s4, conn2);
                    MySqlDataReader reader4 = comm4.ExecuteReader();
                    while (reader4.Read())
                    {
                        string id1 = reader4[0].ToString();
                        id2 = Convert.ToInt32(id1) + 1;
                    }
                    reader4.Close();
                    string sql4 = ("INSERT INTO sity (`id_sity`, `name`) VALUES (" + id2
+ "," + sity + "));");
                    MySqlCommand comm5 = new MySqlCommand(sql4, conn2);
                    comm5.ExecuteNonQuery();
                    comboBox1.Items.Clear();
                    comboBox8.Items.Clear();
                    string s18 = "SELECT Name FROM sity";
                    MySqlCommand comm18 = new MySqlCommand(s18, conn2);
                    MySqlDataReader reader18 = comm18.ExecuteReader();
                    while (reader18.Read())
                    {
                        comboBox1.Items.Add(reader18[0].ToString());
                        comboBox8.Items.Add(reader18[0].ToString());
                    }
                    reader18.Close();
                }
            }
            catch { }
        }
    }

```

```

}
if (n != 1)
{
    string s4 = "select max(id_s) from school";
    MySqlCommand comm4 = new MySqlCommand(s4, conn2);
    MySqlDataReader reader4 = comm4.ExecuteReader();
    while (reader4.Read())
    {
        string id1 = reader4[0].ToString();
        id2 = Convert.ToInt32(id1) + 1;
    }
    reader4.Close();
    string sql4 = ("INSERT INTO school (`id_s`, `name`, `id_sity`) VALUES
(" + id2 + ", '" + sch + "', (select id_sity from sity where name='" + sity + "'))");
    MySqlCommand comm5 = new MySqlCommand(sql4, conn2);
    comm5.ExecuteNonQuery();
}
string s2 = "select max(id_sp) from sportsman";
MySqlCommand comm2 = new MySqlCommand(s2, conn2);
MySqlDataReader reader1 = comm2.ExecuteReader();
while (reader1.Read())
{
    string id1 = reader1[0].ToString();
    id = Convert.ToInt32(id1) + 1;
}
reader1.Close();
int k = 0;
string s6 = "SELECT sportsman.name as 'ПИБ',sportsman.god as 'Год
пожд.',sportsman.rang as 'Ранг',sity.name as 'Мисто',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox1.Text + "'";
MySqlCommand comm6 = new MySqlCommand(s6, conn2);
MySqlDataReader reader6 = comm6.ExecuteReader();
while (reader6.Read())
{
    if (reader6[0].ToString() == name && reader6[1].ToString() == god)
    {
        k = 1;
    }
}
reader6.Close();
if (k == 1)
{
    MessageBox.Show("Спортсмен " + name + " " + god + " Р.Н. из Мистоа "
+ sity + " уже есть в базе данных");
}
else
{
    string sql = ("INSERT INTO `sportsman` (`id_sp`, `name`, `god`,
`rang`, `id_s`, `id_sity`) VALUES ('" + id + "', '" + name + "', '" + god + "', '" + rank +
"', (select DISTINCT id_s from school where name='" + sch + "' and id_sity =(select
DISTINCT id_sity from sity where name='" + sity + "')), (select DISTINCT id_sity from sity
where name='" + sity + "'))");
    MySqlCommand comm = new MySqlCommand(sql, conn2);
    comm.ExecuteNonQuery();
}
DataSet ds = new DataSet();
string sql5 = "SELECT sportsman.name as 'ПИБ',sportsman.god as
'Р.Н.',sportsman.rang as 'Ранг',sity.name as 'Мисто',school.name as 'Спортшкола' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox1.Text + "'";
MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
dataAdapter.Fill(ds, "Users");
dataGridView1.DataSource = ds.Tables["Users"];
conn2.Close();

```

```

    }
    catch { MessageBox.Show("Нет соединения с базой"); }
}
}
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    comboBox4.Text = "";
    comboBox3.Text = "";
    comboBox1.Text = "";
    textBox2.Text = "";
    textBox1.Text = "";
    groupBox1.Visible = true;
    if(radioButton1.Checked)
    {
        button1.Visible = true;
        textBox1.Visible = true;
        comboBox3.Visible = false;
        button2.Visible = false;
        button3.Visible = false;
    }
}
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    comboBox4.Text = "";
    comboBox3.Text = "";
    comboBox1.Text = "";
    textBox2.Text = "";
    textBox1.Text = "";
    groupBox1.Visible = true;
    if (radioButton2.Checked)
    {
        button1.Visible = false;
        textBox1.Visible = false;
        comboBox3.Visible=true;
        button2.Visible = true;
        button3.Visible = true;
    }
}
private void button3_Click(object sender, EventArgs e)
{
    if (comboBox3.Text == "" || comboBox1.Text == "" )
    {
        MessageBox.Show("Необходимо заполнить поля (Місто, ПІВ, Год рожд.)");
    }
    else
    {
        string name = comboBox3.Text;
        string god = Convert.ToString(numericUpDown1.Value);
        string sity = comboBox1.Text;
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        string s1 = " DELETE FROM sportsman WHERE name='" + name + "' and god
="+god+ "' and id_sity = (select DISTINCT id_sity from sity where name='" + sity + "')";
        MySqlCommand comm1 = new MySqlCommand(s1, conn2);
        comm1.ExecuteNonQuery();
        DataSet ds = new DataSet();
        string sql4 = "SELECT sportsman.name as 'ПІВ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox1.Text + "'";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql4, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView1.DataSource = ds.Tables["Users"];
        MessageBox.Show("Удалено");
        conn2.Close();
    }
}

```

```

    }
}
private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string name = comboBox3.Text;
    textBox5.Text = comboBox3.Text;
    string s6 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as
'Школа',sportsman.id_sp FROM `sportsman` INNER JOIN school on school.id_s=sportsman.id_s
INNER JOIN sity on sity.id_sity=sportsman.id_sity WHERE sportsman.name = '" + name + "'";
    MySqlCommand comm6 = new MySqlCommand(s6, conn2);
    MySqlDataReader reader6 = comm6.ExecuteReader();
    while (reader6.Read())
    {
        comboBox4.Text = reader6[4].ToString();
        numericUpDown1.Value = Convert.ToDecimal( reader6[1].ToString());
        textBox2.Text = reader6[2].ToString();
        textBox4.Text = reader6[5].ToString(); ;
    }
    reader6.Close();
}
private void button2_Click(object sender, EventArgs e)
{
    if ( comboBox1.Text == "" || textBox2.Text == "" || comboBox1.Text == "" ||
comboBox4.Text == "")
    {
        MessageBox.Show("Необхідно заповнити всі поля");
    }
    else
    {
        try
        {
            string name = comboBox3.Text;
            string rank = textBox2.Text;
            string god = Convert.ToString(numericUpDown1.Value);
            string sity = comboBox1.Text;
            string sch = comboBox4.Text;
            int id = 0;
            int id2 = 0;
            int m = 0;
            int n = 0;
            dataGridView1.Visible = true;
            MySqlConnection conn2 = new MySqlConnection(conn);
            conn2.Open();
            string s = "select sity.name, school.name from sity inner join school on
school.id_sity=sity.id_sity ";
            MySqlCommand comm1 = new MySqlCommand(s, conn2);
            MySqlDataReader reader3 = comm1.ExecuteReader();
            while (reader3.Read())
            {
                if (reader3[0].ToString() == sity)
                {
                    m = 1;
                }
                if (reader3[1].ToString() == sch && reader3[0].ToString() == sity)
                {
                    n = 1;
                }
            }
            reader3.Close();
            if (m != 1)
            {
                string s4 = "select max(id_sity) from sity";

```

```

MySQLCommand comm4 = new MySQLCommand(s4, conn2);
MySQLDataReader reader4 = comm4.ExecuteReader();
while (reader4.Read())
{
    string id1 = reader4[0].ToString();
    id2 = Convert.ToInt32(id1) + 1;
}
reader4.Close();
string sql4 = ("INSERT INTO sity (`id_sity`, `name`) VALUES (" + id2
+ "," + sity + "));");
MySQLCommand comm5 = new MySQLCommand(sql4, conn2);
comm5.ExecuteNonQuery();
}
if (n != 1)
{
    string s4 = "select max(id_s) from school";
    MySQLCommand comm4 = new MySQLCommand(s4, conn2);
    MySQLDataReader reader4 = comm4.ExecuteReader();
    while (reader4.Read())
    {
        string id1 = reader4[0].ToString();
        id2 = Convert.ToInt32(id1) + 1;
    }
    reader4.Close();
    string sql4 = ("INSERT INTO school (`id_s`, `name`, `id_sity`) VALUES
(" + id2 + "," + sch + ",(select id_sity from sity where name='" + sity + "));");
    MySQLCommand comm5 = new MySQLCommand(sql4, conn2);
    comm5.ExecuteNonQuery();
}
string s2 = "select max(id_sp) from sportsman";
MySQLCommand comm2 = new MySQLCommand(s2, conn2);
MySQLDataReader reader1 = comm2.ExecuteReader();
while (reader1.Read())
{
    string id1 = reader1[0].ToString();
    id = Convert.ToInt32(id1) + 1;
}
reader1.Close();
int k = 0;
string s6 = "SELECT sportsman.name as 'ПИБ',sportsman.god as 'Год
рожд.',sportsman.rang as 'Ранг',sity.name as 'Мисто',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox1.Text + "'";
MySQLCommand comm6 = new MySQLCommand(s6, conn2);
MySQLDataReader reader6 = comm6.ExecuteReader();
while (reader6.Read())
{
    if (reader6[0].ToString() == name && reader6[1].ToString() == god)
    {
        k = 1;
    }
}
reader6.Close();
if (k == 1 && comboBox3.Text!=textBox5.Text)
{
    MessageBox.Show("Спортсмен " + name + " " + god + " Р.н. из Мистоа "
+ sity + " уже есть в базе данных");
}
else
{
    string sql = ("UPDATE `sportsman` SET `name` = '"+name+"', `god` =
 '"+god+"', `rang` = '"+rank+"', `id_s` = (select id_s from school where name='"+sch+"'
 and id_sity=(select id_sity from sity where name='"+sity+"')) WHERE `sportsman`.`id_sp`
 = '"+textBox4.Text+"'");
    MySQLCommand comm = new MySQLCommand(sql, conn2);
}

```

```

        comm.ExecuteNonQuery();
        textBox5.Text = comboBox3.Text;
    }
    DataSet ds = new DataSet();
    string sql5 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox1.Text + "'";
    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView1.DataSource = ds.Tables["Users"];
    conn2.Close();
    }
    catch { MessageBox.Show("Нет соединения с базой"); }
}
}
private void button6_Click(object sender, EventArgs e)
{
    if ( comboBox6.Text == "" || comboBox2.Text == "" || comboBox7.Text == "" )
    {
        MessageBox.Show("Необхідно заповнити всі поля");
    }
    else
    {
        try
        {
            dataGridView2.Visible = true;
            string name = comboBox6.Text;
            string rank = comboBox2.Text;
            string sity_comp = comboBox7.Text;
            string god = Convert.ToString(numericUpDown2.Value);
            string q = dateTimePicker3.Text;
            string den = Convert.ToString(q[6]) + q[7] + q[8] + q[9] + "-" + q[3] +
q[4] + "-" + q[0] + q[1];
            string god_comp = Convert.ToString(q[6]) + q[7] + q[8] + q[9];
            string b = dateTimePicker1.Text;
            string den2 = Convert.ToString(b[6]) + b[7] + b[8] + b[9] + "-" + b[3] +
b[4] + "-" + b[0] + b[1];
            int id = 0;
            MySqlConnection conn2 = new MySqlConnection(conn);
            conn2.Open();
            string s2 = "select max(id_comp) from comp";
            MySqlCommand comm2 = new MySqlCommand(s2, conn2);
            MySqlDataReader reader1 = comm2.ExecuteReader();
            while (reader1.Read())
            {
                string id1= reader1[0].ToString();
                id = Convert.ToInt32(id1) + 1;
            }
            reader1.Close();
            string sql = ("INSERT INTO `comp` (`id_comp`, `name`, `god`, `tip`,
`start`, `final`, `sity_comp`, `god_comp`) VALUES ('" + id+"", '"+name+"', '"+god+"',
 '"+rank+"', '"+den+"', '"+den2+"', '" + sity_comp + "', '" + god_comp + "'");
            MySqlCommand comm = new MySqlCommand(sql, conn2);
            comm.ExecuteNonQuery();
            DataSet ds = new DataSet();
            string sql5 = "SELECT comp.id_comp as '№',comp.name as 'Назва',comp.god
as 'П.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` ORDER BY `comp`.`start` desc";
            MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
            dataAdapter.Fill(ds, "Users");
            dataGridView2.DataSource = ds.Tables["Users"];
            conn2.Close();
        }
        catch { MessageBox.Show("Нет соединения с базой"); }
    }
}

```



```

    }
}
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    comboBox6.Text = "";
    comboBox2.Text = "";
    dataGridView2.Visible = true;
    groupBox4.Visible = true;
    if (radioButton4.Checked)
    {
        button6.Visible = true;
        button4.Visible = false;
        button5.Visible = false;
        comboBox2.Visible = true;
        numericUpDown2.Visible = true;
        label7.Visible = true;
        label3.Visible = true;
        dateTimePicker1.Visible = true;
        label11.Visible = true;
        label9.Visible = true;
        comboBox6.Visible = true;
        label10.Visible = true;
        dateTimePicker3.Visible = true;
        label8.Visible = true;
        comboBox7.Visible = true;
        button20.Visible = false;
        label16.Visible = false;
        label17.Visible = false;
    }
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    DataSet ds = new DataSet();
    string sql5 = "SELECT comp.id_comp as '№',comp.name as 'Назва',comp.god as
'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` ORDER BY `comp`.`start` desc";
    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView2.DataSource = ds.Tables["Users"];
}
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    comboBox6.Text = "";
    comboBox2.Text = "";
    dataGridView2.Visible = true;
    groupBox4.Visible = true;
    if (radioButton3.Checked)
    {
        button6.Visible = false;
        button4.Visible = true;
        button5.Visible = false;
        comboBox2.Visible = false;
        numericUpDown2.Visible = false;
        label7.Visible = false;
        label3.Visible = false;
        dateTimePicker1.Visible = false;
        label11.Visible = false;
        label9.Visible = false;
        comboBox6.Visible = false;
        label10.Visible = false;
        dateTimePicker3.Visible = false;
        label8.Visible = false;
        comboBox7.Visible = false;
        button20.Visible = true;
        label16.Visible = true;
        label17.Visible = true;
    }
}

```

```

    }
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    DataSet ds = new DataSet();
    string sql5 = "SELECT comp.id_comp as '№',comp.name as 'Назва',comp.god as
'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` ORDER BY `comp`.`start` desc";
    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView2.DataSource = ds.Tables["Users"];
}

private void button4_Click(object sender, EventArgs e)
{
    if (comboBox6.Text == "")
    {
        MessageBox.Show("Необходимо заполнить поле (Назва соревнования)");
    }
    else
    {
        string q = dateTimePicker3.Text;
        string den = Convert.ToString(q[6]) + q[7] + q[8] + q[9] + "-" + q[3] +
q[4] + "-" + q[0] + q[1];

        string name = comboBox6.Text;
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        string s1 = " DELETE FROM comp WHERE name='" + name + "' and start ='" +
den + "' ";
        MySqlCommand comm1 = new MySqlCommand(s1, conn2);
        comm1.ExecuteNonQuery();
        DataSet ds = new DataSet();
        string sql5 = "SELECT comp.name as 'Назва',comp.god as 'Р.н.', comp.start
as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто', comp.tip 'Тип' FROM
`comp` ORDER BY `comp`.`start` desc";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView2.DataSource = ds.Tables["Users"];
        conn2.Close();
    }
}

private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    comboBox6.Text = "";
    comboBox2.Text = "";
    groupBox4.Visible = true;
    dataGridView2.Visible = true;
    if (radioButton5.Checked)
    {
        button6.Visible = false;
        button4.Visible = false;
        button5.Visible = true;
        comboBox2.Visible = true;
        numericUpDown2.Visible = true;
        label7.Visible = true;
        label3.Visible = true;
        dateTimePicker1.Visible = true;
        label11.Visible = true;
        label9.Visible = true;
        comboBox6.Visible = true;
        label10.Visible = true;
        dateTimePicker3.Visible = true;
        label8.Visible = true;
        comboBox7.Visible = true;
        button20.Visible = true;
    }
}

```

```

        label16.Visible = false;
        label17.Visible = false;
    }
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    DataSet ds = new DataSet();
    string sql5 = "SELECT comp.id_comp as '№',comp.name as 'Назва',comp.god as
'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` ORDER BY `comp`.`start` desc";
    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView2.DataSource = ds.Tables["Users"];
}
private void button5_Click(object sender, EventArgs e)
{
    if ( comboBox6.Text == "" || comboBox2.Text == "" || comboBox7.Text == "" )
    {
        MessageBox.Show("Необхідно заповнити всі поля");
    }
    else
    {
        try
        {
            dataGridView2.Visible = true;
            string name = comboBox6.Text;
            string rank = comboBox2.Text;
            string sity_comp = comboBox7.Text;
            string god = Convert.ToString(numericUpDown2.Value);
            string q = dateTimePicker3.Text;
            string den = Convert.ToString(q[6]) + q[7] + q[8] + q[9] + "-" + q[3] +
q[4] + "-" + q[0] + q[1];
            string b = dateTimePicker1.Text;
            string den2 = Convert.ToString(b[6]) + b[7] + b[8] + b[9] + "-" + b[3] +
b[4] + "-" + b[0] + b[1];
            string den3 = Convert.ToString(q[6]) + q[7] + q[8] + q[9];
            MySqlConnection conn2 = new MySqlConnection(conn);
            conn2.Open();
            string sql = ("UPDATE `comp` SET `name` = '"+name+"', `god` =
 '"+god+"', `tip` = '"+rank+"', `start` = '"+den+"', `final` = '"+den2+ "' , `sity_comp`=
 '"+sity_comp+"', `god_comp`= '"+den3+" WHERE `comp`.`id_comp` = ' " + up_comp+"";");
            MySqlCommand comm = new MySqlCommand(sql, conn2);
            comm.ExecuteNonQuery();
            textBox5.Text = comboBox3.Text;
            DataSet ds = new DataSet();
            string sql5 = "SELECT comp.id_comp as '№',comp.name as 'Назва',comp.god
as 'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` ORDER BY `comp`.`start` desc";
            MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
            dataAdapter.Fill(ds, "Users");
            dataGridView2.DataSource = ds.Tables["Users"];
            conn2.Close();
        }
        catch { MessageBox.Show("Нет соединения с базой"); }
    }
}
private void comboBox6_SelectedIndexChanged(object sender, EventArgs e)
{
    textBox6.Text = comboBox6.Text;
}

private void comboBox8_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox5.Items.Clear();
    comboBox5.Text = "";
    MySqlConnection conn2 = new MySqlConnection(conn);

```

```

        conn2.Open();
        string s1 = "SELECT school.Name FROM school where id_sity= (select id_sity
from sity where name =' " + comboBox8.Text + "')";

        MySqlCommand comm1 = new MySqlCommand(s1, conn2);
        MySqlDataReader reader1 = comm1.ExecuteReader();
        while (reader1.Read())
        {
            comboBox5.Items.Add(reader1[0].ToString());
        }
        reader1.Close();

        string s2 = "SELECT sportsman.name as 'ПИБ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' " +
            "FROM `sportsman` INNER JOIN school on school.id_s=sportsman.id_s
INNER JOIN sity on sity.id_sity=sportsman.id_sity" +
            " where sportsman.id_sity= (select id_sity from sity where name =' " +
comboBox8.Text + "')";

        /* MySqlCommand comm2 = new MySqlCommand(s2, conn2);
        MySqlDataReader reader2 = comm2.ExecuteReader();
        while (reader2.Read())
        {
            comboBox9.Items.Add(reader2[0].ToString());
        }
        reader2.Close();*/

        DataSet ds = new DataSet();
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(s2, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView11.DataSource = ds.Tables["Users"];
    }
    private void comboBox5_SelectedIndexChanged(object sender, EventArgs e)
    {
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        string s2 = "SELECT sportsman.name as
'ПИБ',sportsman.god as 'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name
as 'Школа' " +
            "FROM `sportsman` INNER JOIN school on school.id_s=sportsman.id_s
INNER JOIN sity on sity.id_sity=sportsman.id_sity" +
            " where sportsman.id_s= (select id_s from school where name =' " +
comboBox5.Text + "' and id_sity= (select id_sity from sity where name =' " +
comboBox8.Text + "')";
        DataSet ds = new DataSet();
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(s2, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView11.DataSource = ds.Tables["Users"];
    }
    private void button8_Click(object sender, EventArgs e)
    {
        string l = textBox9.Text;
        string p = textBox8.Text;
        if (l != "" && p != "")
        {
            if (l == "1")
            {
                if (p == "1")
                {
                    groupBox8.Visible = false;
                    tabControl1.Visible = true;
                }
                else { MessageBox.Show("Неправильный пароль"); }
            }
            else { MessageBox.Show("Неправильный логин"); }
        }
    }
}

```

```

else
{
    MessageBox.Show("Заполните логин или пароль");
}
//-----
dataGridView7.Visible = true;
MySQLConnection conn2 = new MySqlConnection(conn);
conn2.Open();
//-----
DataSet ds = new DataSet();
string sql5 = "SELECT att.id_att as '№',att.txt as 'Оголошення', login.name
as 'Автор' FROM `att` INNER JOIN login on att.id_kor= login.id_kor";
MySQLDataAdapter dataAdapter = new MySQLDataAdapter(sql5, conn2);
dataAdapter.Fill(ds, "Users");
dataGridView13.DataSource = ds.Tables["Users"];
dataGridView13.Columns[0].Width = 40;
dataGridView13.Columns[1].Width = 440;
dataGridView13.Columns[2].Width = 300;
//-----
DataSet ds1 = new DataSet();
string sql1 = "SELECT comp.id_comp as 'id', comp.name as 'Назва',comp.god as
'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` " +
"where comp.start > '" + DateTime.Now.ToString("yyyy-MM-dd") + "' ORDER
BY `comp`.`start` desc";
//textBox10.Text = sql1;
MySQLDataAdapter dataAdapter1 = new MySQLDataAdapter(sql1, conn2);
dataAdapter1.Fill(ds1, "Users");
dataGridView8.DataSource = ds1.Tables["Users"];
DataSet ds2 = new DataSet();
string sql2 = "SELECT comp.id_comp as 'id', comp.name as 'Назва',comp.god as
'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` " +
" ORDER BY `comp`.`start` desc";
MySQLDataAdapter dataAdapter2 = new MySQLDataAdapter(sql2, conn2);
dataAdapter2.Fill(ds2, "Users");
dataGridView10.DataSource = ds2.Tables["Users"];
DataSet ds3 = new DataSet();
string sql3 = "SELECT id_reg as 'Телефон', name as 'Ім`я', spec as
'Спеціальність', pass 'пароль' FROM `reg`";
MySQLDataAdapter dataAdapter3 = new MySQLDataAdapter(sql3, conn2);
dataAdapter3.Fill(ds3, "Users");
dataGridView12.DataSource = ds3.Tables["Users"];
DataSet ds4 = new DataSet();
string sql4 = "SELECT id_kor as 'Телефон', name as 'Ім`я', spec as
'Спеціальність', pass 'пароль' FROM `login`";
MySQLDataAdapter dataAdapter4 = new MySQLDataAdapter(sql4, conn2);
dataAdapter4.Fill(ds4, "Users");
dataGridView14.DataSource = ds4.Tables["Users"];
conn2.Close();
}
private void button7_Click(object sender, EventArgs e)
{
    groupBox8.Visible = false;
    tabControl2.Visible = true;
    MySQLConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    DataSet ds = new DataSet();
    string sql5 = "SELECT comp.id_comp as 'id',comp.name as 'Назва',comp.god as
'Р.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` ORDER BY `comp`.`start` ASC";
    MySQLDataAdapter dataAdapter = new MySQLDataAdapter(sql5, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView4.DataSource = ds.Tables["Users"];
}

```

```

string s2 = "SELECT distinct god_comp FROM comp";
comboBox9.Items.Clear();
MySQLCommand comm2 = new MySQLCommand(s2, conn2);
MySQLDataReader reader2 = comm2.ExecuteReader();
while (reader2.Read())
{
    comboBox9.Items.Add(reader2[0].ToString());
}
reader2.Close();
conn2.Close();
}

private void button11_Click(object sender, EventArgs e)
{
    groupBox8.Visible = true;
    tabControl2.Visible = false;
}
private void comboBox12_SelectedIndexChanged(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    dataGridView1.Visible = true;
    DataSet ds = new DataSet();
    string sql4 = "SELECT sportsman.name as 'ПИБ',sportsman.god as
'П.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox12.Text + "'";
    MySQLDataAdapter dataAdapter = new MySQLDataAdapter(sql4, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView6.DataSource = ds.Tables["Users"];
}
private void button12_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string s2 = "select count(id_l) from list where id_comp = '" + list_comp + "'
";
    MySQLCommand comm2 = new MySQLCommand(s2, conn2);
    MySQLDataReader reader1 = comm2.ExecuteReader();
    int id = 0;
    while (reader1.Read())
    {
        string id1 = reader1[0].ToString();
        id = Convert.ToInt32(id1);
    }
    reader1.Close();
    if (id == 0)
    {
        MessageBox.Show("Список пуст");
    }
    else
    {
        Word.Application app = new Word.Application();
        Word.Document doc = app.Documents.Add(Visible: false);
        Word.Range r = doc.Range();
        r.Text = "doc1";
        string[,] sp = new string[id, 6];
        string s3 = "SELECT sportsman.name as 'ПИБ',sportsman.god as
'П.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =

```

```

list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where comp.id_comp ='" + list_comp + "'";
    MySqlCommand comm3 = new MySqlCommand(s3, Conn2);
    MySqlDataReader reader3 = comm3.ExecuteReader();
    int i = 0;
    while (reader3.Read())
    {
        for (int j = 0; j < 6; j++)
        {
            if (j == 0)
            {
                sp[i, j] = Convert.ToString(i + 1);
            }
            else
            {
                sp[i, j] = reader3[j - 1].ToString();
            }
        }
        i++;
    }
    reader3.Close();
    Word.Table t = doc.Tables.Add(r, id + 1, 6);
    t.Borders.Enable = 1;
    t.Columns.AutoFit();
    // string[,] mas = new string[id, 10];
    string[] as2 = new string[6];
    as2[0] = "№";
    as2[1] = "ПИБ";
    as2[2] = "П.Н.";
    as2[3] = "Рагг";
    as2[4] = "Місто";
    as2[5] = "Школа";
    foreach (Word.Row row in t.Rows)
    {
        foreach (Word.Cell cell in row.Cells)
        {
            if (cell.RowIndex == 1)
            {
                cell.Range.Text = as2[cell.ColumnIndex - 1];
            }
            else
            {
                cell.Range.Text = sp[cell.RowIndex - 2, cell.ColumnIndex -
1].ToString();
            }
        }
    }
    doc.Save();
    doc.Close();
    conn2.Close();
}
}
private void button9_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string s2 = "select count(id_1) from list where id_comp = '" + comp_comp + "'
";
    MySqlCommand comm2 = new MySqlCommand(s2, conn2);
    MySqlDataReader reader1 = comm2.ExecuteReader();
    int id = 0;
    while (reader1.Read())
    {
        string id1 = reader1[0].ToString();
        id = Convert.ToInt32(id1);
    }
}

```

```

}
reader1.Close();
if (id == 0)
{
    MessageBox.Show("Список пуст");
}
else
{
    Word.Application app = new Word.Application();
    Word.Document doc = app.Documents.Add(Visible: false);
    Word.Range r = doc.Range();
    r.Text = "doc1";
    string[,] sp = new string[id, 6];
    string s3 = "SELECT sportsman.name as 'ИИБ',sportsman.god as
'П.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where comp.id_comp ='" + comp_comp + "' ";
    MySqlCommand comm3 = new MySqlCommand(s3, conn2);
    MySqlDataReader reader3 = comm3.ExecuteReader();
    int i = 0;
    while (reader3.Read())
    {
        for (int j = 0; j < 6; j++)
        {
            if (j == 0)
            {
                sp[i, j] = Convert.ToString(i + 1);
            }
            else
            {
                sp[i, j] = reader3[j - 1].ToString();
            }
        }
        i++;
    }
    reader3.Close();
    Word.Table t = doc.Tables.Add(r, id + 1, 6);
    t.Borders.Enable = 1;
    t.Columns.AutoFit();
    string[] as2 = new string[6];
    as2[0] = "№";
    as2[1] = "ИИБ";
    as2[2] = "П.Н.";
    as2[3] = "Ранг";
    as2[4] = "Місто";
    as2[5] = "Школа";
    foreach (Word.Row row in t.Rows)
    {
        foreach (Word.Cell cell in row.Cells)
        {
            if (cell.RowIndex == 1)
            {
                cell.Range.Text = as2[cell.ColumnIndex - 1];
            }
            else
            {
                cell.Range.Text = sp[cell.RowIndex - 2, cell.ColumnIndex -
1].ToString();// "колонка " + cell.ColumnIndex.ToString();
            }
        }
    }
    doc.Save();
    doc.Close();
    conn2.Close();
}

```



```

    }
    }
private void button13_Click(object sender, EventArgs e)
{
    groupBox8.Visible = true;
    tabControl2.Visible = false;
}
private void button14_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    try
    {
        double x;
        x = Convert.ToDouble(dataGridView7.SelectedCells[0].Value.ToString());
        double id_ank = Convert.ToDouble(x);

        string s3 = "SELECT anketa.`id_ank` as '№', anketa.`id_comp` as '',
anketa.`name` as 'ПИБ', anketa.`god_r` as 'П.Н.', anketa.`school` as 'Спортшкола',
anketa.`tell` as 'Тел.', anketa.`sity` as 'Місто'" +
        "FROM anketa inner join `comp` on comp.id_comp=anketa.id_comp " +
        "where anketa.id_ank='" + id_ank + "'";
        MySqlCommand comm3 = new MySqlCommand(s3, conn2);
        MySqlDataReader reader3 = comm3.ExecuteReader();
        int i = 0;
        while (reader3.Read())
        {
            groupBox14.Text = "Анкета №" + reader3[0].ToString();
            label20.Text = "ПИБ: " + reader3[2].ToString();
            label25.Text = "П.Н.:" + reader3[3].ToString();
            label26.Text = "Спорт школа: " + reader3[4].ToString();
            label27.Text = "Н. тел.: " + reader3[5].ToString();
            label28.Text = "Місто: " + reader3[6].ToString();

            ank_nom = reader3[0].ToString(); ;
            ank_comp = reader3[1].ToString();
            ank_name = reader3[2].ToString();
            ank_school = reader3[4].ToString();
            ank_tel = reader3[5].ToString();
            ank_sity = reader3[6].ToString();
            ank_god = reader3[3].ToString();

        }
        reader3.Close();
    }
    catch { }
}

private void dataGridView7_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    double x;
    x = Convert.ToDouble(dataGridView7.SelectedCells[0].Value.ToString());
    //textBox7.Text = Convert.ToString(x);
}
private void button15_Click(object sender, EventArgs e)
{
    try {
        double x;
        x = Convert.ToDouble(dataGridView8.SelectedCells[0].Value.ToString());
        double id_comp = Convert.ToDouble(x);
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        DateTime thisDay = DateTime.Today;
    }
}

```

```

string s2 = "select name, start, final from comp where id_comp ='" + id_comp
+ "'";

MySQLCommand comm2 = new MySQLCommand(s2, conn2);
MySQLDataReader reader1 = comm2.ExecuteReader();
while (reader1.Read())
{
    label18.Text = " Обрано: " + reader1[0].ToString() ;
}
reader1.Close();
//-----
DataSet ds = new DataSet();
string sql5 = "SELECT anketa.`id_ank` as '№', anketa.`name` as 'ПІБ',
anketa.`god_r` as 'Р.н.', anketa.`school` as 'Спортшкола', anketa.`tell` as 'Тел.',
anketa.`sity` as 'Місто' " +
"FROM anketa inner join `comp` on comp.id_comp=anketa.id_comp " +
"where anketa.id_comp='" + id_comp + "'";
MySQLDataAdapter dataAdapter = new MySQLDataAdapter(sql5, conn2);
dataAdapter.Fill(ds, "Users");
dataGridView7.DataSource = ds.Tables["Users"];

//-----
DataSet ds2 = new DataSet();
string sql2 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'Р.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp ='" + id_comp + "' ";
MySQLDataAdapter dataAdapter2 = new MySQLDataAdapter(sql2, conn2);
dataAdapter2.Fill(ds2, "Users");
dataGridView9.DataSource = ds2.Tables["Users"];
}
catch { }
}
private void button16_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    if (ank_nom == "")
    {
        MessageBox.Show("Оберіть анкету");
    }
    else
    {
        int id = 0;
        string s2 = "select max(id_1) from list";
        MySQLCommand comm2 = new MySQLCommand(s2, conn2);
        MySQLDataReader reader2 = comm2.ExecuteReader();
        while (reader2.Read())
        {
            string id1 = reader2[0].ToString();
            id = Convert.ToInt32(id1) + 1;
        }
        reader2.Close();
        string s1 = "SELECT id_sp FROM sportsman " +
            "where name='" + ank_name + "' and god='" + ank_god + "'";
        MySQLCommand comm1 = new MySQLCommand(s1, conn2);
        MySQLDataReader reader1 = comm1.ExecuteReader();
        string id_sp = "";
        while (reader1.Read())
        {
            id_sp = reader1[0].ToString();
        }
        reader1.Close();

        string s12 = "SELECT id_1 FROM list " +

```

```

        "where id_sp='" + id_sp + "' and id_comp='" + ank_comp + "'";
        MySqlCommand comm12 = new MySqlCommand(s12, conn2);
        MySqlDataReader reader12 = comm12.ExecuteReader();
        string id_list = "";
        while (reader12.Read())
        {
            id_list = reader12[0].ToString();
        }
        reader12.Close();
        if (id_list != "")
        {
            MessageBox.Show("Спортсмен вже додан у список");
        }
        else
        {
            if (id_sp != "")
            {
                string sql = ("INSERT INTO `list` (`id_l`, `id_sp`, `id_comp`)
VALUES ('" + id + "','" + id_sp + "','" + ank_comp + "')");
                MySqlCommand comm = new MySqlCommand(sql, conn2);
                comm.ExecuteNonQuery();
            }
            else
            {
                string s3 = "SELECT id_sity FROM sity" +
                " where name= '" + ank_sity + "'";
                MySqlCommand comm3 = new MySqlCommand(s3, conn2);
                MySqlDataReader reader3 = comm3.ExecuteReader();
                string id_sity = "";
                while (reader3.Read())
                {
                    id_sity = reader3[0].ToString();
                }
                reader3.Close();
                string s5 = "SELECT max(id_s) FROM school";
                MySqlCommand comm5 = new MySqlCommand(s5, conn2);
                MySqlDataReader reader5 = comm5.ExecuteReader();
                int maxid_school = 0;
                while (reader5.Read())
                {
                    string id3 = reader5[0].ToString();
                    maxid_school = Convert.ToInt32(id3) + 1;
                }
                reader5.Close();
                if (id_sity == "")
                {
                    string s4 = "SELECT max(id_sity) FROM sity";
                    MySqlCommand comm4 = new MySqlCommand(s4, conn2);
                    MySqlDataReader reader4 = comm4.ExecuteReader();
                    int maxid_sity = 0;
                    while (reader4.Read())
                    {
                        string id2 = reader4[0].ToString();
                        maxid_sity = Convert.ToInt32(id2) + 1;
                    }
                    reader4.Close();
                    string sql1 = ("INSERT INTO `sity`(`id_sity`, `name`) VALUES
(" + maxid_sity + "','" + ank_sity + "')");
                    string sql2 = ("INSERT INTO `school`(`id_s`, `name`, `id_sity`) VALUES
(" + maxid_school + "','" + ank_school + "','" + maxid_sity + "')");
                    MySqlCommand comm11 = new MySqlCommand(sql1, conn2);
                    comm11.ExecuteNonQuery();
                }
                string s6 = "SELECT id_s FROM school" +
                " where name= '" + ank_school + "'";
            }
        }
    }
}

```

```

        MySqlCommand comm6 = new MySqlCommand(s6, conn2);
        MySqlDataReader reader6 = comm6.ExecuteReader();
        string id_school = "";
        while (reader6.Read())
        {
            id_school = reader6[0].ToString();
        }
        reader6.Close();
        if (id_school == "")
        {
            string sql7 = ("INSERT INTO `school`(`id_s`, `name`,
`id_sity`) VALUES (" + maxid_school + ",'" + ank_school + "'," + id_sity + ")");

            MySqlCommand comm7 = new MySqlCommand(sql7, conn2);
            comm7.ExecuteNonQuery();
        }
        string s8 = "SELECT school.id_s, sity.id_sity FROM school" +
" inner join sity on school.id_sity=sity.id_sity " +
"where school.name= '" + ank_school + "'";
        MySqlCommand comm8 = new MySqlCommand(s8, conn2);
        MySqlDataReader reader8 = comm8.ExecuteReader();
        id_school = "";
        id_sity = "";
        while (reader8.Read())
        {
            id_school = reader8[0].ToString();
            id_sity = reader8[1].ToString(); ;
        }
        reader8.Close();
        string s10 = "SELECT max(id_sp) FROM sportsman";
        MySqlCommand comm10 = new MySqlCommand(s10, conn2);
        MySqlDataReader reader10 = comm10.ExecuteReader();
        int maxid_sp = 0;
        while (reader10.Read())
        {
            string id2 = reader10[0].ToString();
            maxid_sp = Convert.ToInt32(id2) + 1;
        }
        reader10.Close();
        string sql9 = ("INSERT INTO `sportsman`(`id_sp`, `name`, `god`,
`rang`, `id_s`, `id_sity`) " +
" VALUES ('" + maxid_sp + "'," + ank_name + "'," + ank_god
+ "','-'," + id_school + "'," + id_sity + "')");
        MySqlCommand comm9 = new MySqlCommand(sql9, conn2);
        comm9.ExecuteNonQuery();
        string sql = ("INSERT INTO `list`(`id_l`, `id_sp`, `id_comp`)
VALUES('" + id + "'," + maxid_sp + "'," + ank_comp + "' )");
        MySqlCommand comm = new MySqlCommand(sql, conn2);
        comm.ExecuteNonQuery();
    }
    DataSet ds2 = new DataSet();
    string sql2 = "SELECT sportsman.name as 'ПИБ',sportsman.god as
'П.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp ='" + ank_comp + "' ";
    MySqlDataAdapter dataAdapter2 = new MySqlDataAdapter(sql2, conn2);
    dataAdapter2.Fill(ds2, "Users");
    dataGridView9.DataSource = ds2.Tables["Users"];
}
}
}
private void button17_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);

```

```

conn2.Open();
if (checkBox1.Checked != true)
{
    MessageBox.Show("Для видалення необхідно поставити відмітку");
}
else
{
    string sql = ("DELETE FROM `anketa` WHERE id_ank='"+ank_nom+ "'");

    MySqlCommand comm = new MySqlCommand(sql, conn2);
    comm.ExecuteNonQuery();
    MessageBox.Show("Анкета №"+ank_nom+" видалена");

    DataSet ds = new DataSet();
    string sql5 = "SELECT anketa.`id_ank` as '№', anketa.`name` as 'ПІБ',
anketa.`god_r` as 'Р.н.', anketa.`school` as 'Спортшкола', anketa.`tell` as 'Тел.',
anketa.`sity` as 'Місто' +
        "FROM anketa inner join `comp` on comp.id_comp=anketa.id_comp " +
        "where anketa.id_comp='" + ank_comp + "'";
    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView7.DataSource = ds.Tables["Users"];
    checkBox1.Checked = false;
}
}

private void button18_Click(object sender, EventArgs e)
{
    try
    {
        double x;
        x = Convert.ToDouble(dataGridView10.SelectedCells[0].Value.ToString());
        double id_comp = Convert.ToDouble(x);
        list_comp = Convert.ToString(x);
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        DateTime thisDay = DateTime.Today;
        string s2 = "select name, start, final from comp where id_comp =' " + id_comp +
        """;

        MySqlCommand comm2 = new MySqlCommand(s2, conn2);
        MySqlDataReader reader1 = comm2.ExecuteReader();
        while (reader1.Read())
        {
            groupBox15.Text = "Список учасників змагання (" + reader1[0].ToString() +
            ")";
        }
        reader1.Close();
        //-----
        DataSet ds2 = new DataSet();
        string sql2 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'Р.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp =' " + id_comp + " " ";
        MySqlDataAdapter dataAdapter2 = new MySqlDataAdapter(sql2, conn2);
        dataAdapter2.Fill(ds2, "Users");
        dataGridView3.DataSource = ds2.Tables["Users"];
    }
    catch { }
}

private void button19_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    try

```

```

    {
        string x;
        x = Convert.ToString(dataGridView3.SelectedCells[0].Value.ToString());
        string sport_pib = Convert.ToString(x);
        string s3 = "SELECT sportsman.id_sp ,sportsman.name as
'ПІБ',sportsman.god as 'Р.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name
as 'Школа' " +
                "FROM `sportsman` INNER JOIN school on school.id_s=sportsman.id_s
INNER JOIN sity on sity.id_sity=sportsman.id_sity WHERE sportsman.name = '" + sport_pib +
                "'";

        MySqlCommand comm3 = new MySqlCommand(s3, conn2);
        MySqlDataReader reader3 = comm3.ExecuteReader();
        int i = 0;
        while (reader3.Read())
        {
            list_sport = reader3[0].ToString();
            label36.Text = "ПІБ: " + reader3[1].ToString();
            label32.Text = "Р.Н.:" + reader3[2].ToString();
            label33.Text = "Ранг: " + reader3[3].ToString();

        }
        reader3.Close();
    }
    catch {
    }
}
private void button21_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    try
    {
        string x;
        x = Convert.ToString(dataGridView11.SelectedCells[0].Value.ToString());
        string sport_pib = Convert.ToString(x);
        string s3 = "SELECT sportsman.id_sp ,sportsman.name as
'ПІБ',sportsman.god as 'Р.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name
as 'Школа' " +
                "FROM `sportsman` INNER JOIN school on school.id_s=sportsman.id_s
INNER JOIN sity on sity.id_sity=sportsman.id_sity WHERE sportsman.name = '" + sport_pib +
                "'";

        MySqlCommand comm3 = new MySqlCommand(s3, conn2);
        MySqlDataReader reader3 = comm3.ExecuteReader();
        int i = 0;
        while (reader3.Read())
        {
            list_sport = reader3[0].ToString();
            label36.Text = "ПІБ: " + reader3[1].ToString();
            label32.Text = "Р.Н.:" + reader3[2].ToString();
            label33.Text = "Ранг: " + reader3[3].ToString();

        }
        reader3.Close();
    }
    catch
    {
    }
}
private void button23_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    if (list_sport == "")
    {
        MessageBox.Show("Оберіть спортсмена");
    }
}

```

```

else
{
    int id = 0;
    string s2 = "select max(id_l) from list";
    MySqlCommand comm2 = new MySqlCommand(s2, conn2);
    MySqlDataReader reader2 = comm2.ExecuteReader();

    while (reader2.Read())
    {
        string id1 = reader2[0].ToString();
        id = Convert.ToInt32(id1) + 1;
    }
    reader2.Close();
    string s12 = "SELECT id_l FROM list " +
        "where id_sp='" + list_sport + "' and id_comp='" + list_comp + "'";
    MySqlCommand comm12 = new MySqlCommand(s12, conn2);
    MySqlDataReader reader12 = comm12.ExecuteReader();
    string id_list = "";
    while (reader12.Read())
    {
        id_list = reader12[0].ToString();
    }
    reader12.Close();
    if (id_list != "")
    {
        MessageBox.Show("Спортсмен вже додан у список");
    }
    else
    {
        string sql = ("INSERT INTO `list` (`id_l`, `id_sp`, `id_comp`)
VALUES('" + id + "', '" + list_sport + "', '" + list_comp + "')");
        MySqlCommand comm = new MySqlCommand(sql, conn2);
        comm.ExecuteNonQuery();
    }
}
DataSet ds2 = new DataSet();
string sql2 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'Р.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp ='" + list_comp + "' ";
MySqlDataAdapter dataAdapter2 = new MySqlDataAdapter(sql2, conn2);
dataAdapter2.Fill(ds2, "Users");
dataGridView3.DataSource = ds2.Tables["Users"];
}
private void button22_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    if (checkBox2.Checked != true)
    {
        MessageBox.Show("Для видалення необхідно поставити відмітку");
    }
    else
    {
        string sql = ("DELETE FROM list WHERE id_sp='" + list_sport + "'");

        MySqlCommand comm = new MySqlCommand(sql, conn2);
        comm.ExecuteNonQuery();
        MessageBox.Show("Спортсмен видален з списку");
        DataSet ds2 = new DataSet();
        string sql2 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'Р.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =

```

```

list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp ='" + list_comp + "' ";
    MySqlDataAdapter dataAdapter2 = new MySqlDataAdapter(sql2, conn2);
    dataAdapter2.Fill(ds2, "Users");
    dataGridView3.DataSource = ds2.Tables["Users"];
    checkBox2.Checked = false;
    }
}
private void button20_Click_1(object sender, EventArgs e)
{
    try
    {
        double x;
        x = Convert.ToDouble(dataGridView2.SelectedCells[0].Value.ToString());
        double id_comp = Convert.ToDouble(x);
        up_comp = Convert.ToString(x);
        del_comp = Convert.ToString(x);
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        DateTime thisDay = DateTime.Today;
        string s2 = "select name,start, final,god,tip,sity_comp from comp where
id_comp ='" + id_comp + "'";
        MySqlCommand comm2 = new MySqlCommand(s2, conn2);
        MySqlDataReader reader1 = comm2.ExecuteReader();
        while (reader1.Read())
        {
            if (radioButton5.Checked)
            {
                comboBox6.Text = reader1[0].ToString();
                dateTimePicker3.Text = reader1[1].ToString();
                dateTimePicker1.Text = reader1[2].ToString();
                string g = reader1[3].ToString();
                numericUpDown2.Value = Convert.ToUInt32(g);
                comboBox2.Text = reader1[4].ToString();
                comboBox7.Text = reader1[5].ToString();
            }
            string q = reader1[1].ToString();
            string den = Convert.ToString( q[0]) + q[1]+ "." + q[3] + q[4] + "."
+ q[6] + q[7] + q[8] + q[9];
            q = reader1[2].ToString();
            string den2 = Convert.ToString(q[0]) + q[1] + "." + q[3] + q[4] + "."
+ q[6] + q[7] + q[8] + q[9];
            if (radioButton3.Checked)
            {
                label16.Text = " Обрано: " + reader1[0].ToString() + " (м." +
reader1[5].ToString()+")";
                label17.Text = " (з " + den+ " по "+ den2 +")";
            }
        }
        reader1.Close();
        DataSet ds2 = new DataSet();
        string sql2 = "SELECT sportsman.name as 'ПИБ',sportsman.god as
'П.Н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp ='" + id_comp + "' ";
        MySqlDataAdapter dataAdapter2 = new MySqlDataAdapter(sql2, conn2);
        dataAdapter2.Fill(ds2, "Users");
        dataGridView3.DataSource = ds2.Tables["Users"];
    }
    catch { }
}
private void comboBox9_SelectedIndexChanged(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);

```



```

        conn2.Open();
        string s2 = "SELECT distinct tip FROM comp where god_comp=
"+comboBox9.Text+"";
        comboBox11.Items.Clear();
        MySqlCommand comm2 = new MySqlCommand(s2, conn2);
        MySqlDataReader reader2 = comm2.ExecuteReader();
        while (reader2.Read())
        {
            comboBox11.Items.Add(reader2[0].ToString());
        }
        reader2.Close();
        DataSet ds = new DataSet();
        string sql5 = "SELECT comp.id_comp as 'id',comp.name as 'Назва',comp.god as
'П.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` where god_comp = '"+comboBox9.Text+"' ORDER BY `comp`.`start`
ASC";

        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView4.DataSource = ds.Tables["Users"];
        conn2.Close();
    }
    private void comboBox11_SelectedIndexChanged(object sender, EventArgs e)
    {
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        DataSet ds = new DataSet();
        string sql5 = "SELECT comp.id_comp as 'id',comp.name as 'Назва',comp.god as
'П.н.', comp.start as 'Початок',comp.final as 'Кінець', comp.sity_comp as 'Місто',
comp.tip 'Тип' FROM `comp` where god_comp = '" + comboBox9.Text + "' and tip = '" +
comboBox11.Text + "' ORDER BY `comp`.`start` ASC";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView4.DataSource = ds.Tables["Users"];
        conn2.Close();
    }
    private void button24_Click(object sender, EventArgs e)
    {
        try
        {
            double x;
            x = Convert.ToDouble(dataGridView4.SelectedCells[0].Value.ToString());
            double id_comp = Convert.ToDouble(x);
            comp_comp = Convert.ToString(x);
            MySqlConnection conn2 = new MySqlConnection(conn);
            conn2.Open();
            DateTime thisDay = DateTime.Today;
            string s2 = "select name, start, final from comp where id_comp ='" +
id_comp + "'";
            MySqlCommand comm2 = new MySqlCommand(s2, conn2);
            MySqlDataReader reader1 = comm2.ExecuteReader();
            while (reader1.Read())
            {
                groupBox7.Text = "Список учасників змагання (" +
reader1[0].ToString() + ")";
            }
            reader1.Close();
            //-----
            DataSet ds2 = new DataSet();
            string sql2 = "SELECT sportsman.name as 'ПІБ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM `list`
INNER JOIN comp ON comp.id_comp = list.id_comp INNER JOIN sportsman on sportsman.id_sp =
list.id_sp INNER JOIN sity on sportsman.id_sity = sity.id_sity INNER JOIN school ON
sportsman.id_s = school.id_s where list.id_comp ='" + id_comp + "' ";
            MySqlDataAdapter dataAdapter2 = new MySqlDataAdapter(sql2, conn2);
            dataAdapter2.Fill(ds2, "Users");

```

```

        dataGridView5.DataSource = ds2.Tables["Users"];
    }
    catch { }
}
private void comboBox10_SelectedIndexChanged(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string s2 = "SELECT distinct(sportsman.rang) as 'Ранг' FROM `sportsman` INNER
JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity " +
    "WHERE sity.name = '" + comboBox12.Text + "'";
    comboBox10.Items.Clear();
    MySqlCommand comm2 = new MySqlCommand(s2, conn2);
    MySqlDataReader reader2 = comm2.ExecuteReader();
    while (reader2.Read())
    {
        comboBox10.Items.Add(reader2[0].ToString());
    }
    reader2.Close();
    dataGridView1.Visible = true;
    DataSet ds = new DataSet();
    string sql4 = "SELECT sportsman.name as 'ИИВ',sportsman.god as
'П.н.',sportsman.rang as 'Ранг',sity.name as 'Місто',school.name as 'Школа' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity WHERE sity.name = '" + comboBox12.Text + "'";
    MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql4, conn2);
    dataAdapter.Fill(ds, "Users");
    dataGridView6.DataSource = ds.Tables["Users"];
}
private void button27_Click(object sender, EventArgs e)
{
    if (textBox3.Text == "")
    {
        MessageBox.Show("Поле з оголошенням порожнє");
    }
    else
    {
        try
        {
            dataGridView13.Visible = true;
            string txt = textBox3.Text;
            int id = 0;
            MySqlConnection conn2 = new MySqlConnection(conn);
            conn2.Open();
            string s2 = "select max(id_att) from att";
            MySqlCommand comm2 = new MySqlCommand(s2, conn2);
            MySqlDataReader reader1 = comm2.ExecuteReader();
            while (reader1.Read())
            {
                string id1 = reader1[0].ToString();
                id = Convert.ToInt32(id1) + 1;
            }
            reader1.Close();
            string sql = ("INSERT INTO `att`(`id_att`, `txt`, `id_kor`) VALUES ('" +
id + "', '" + txt + "', '" + kor_id + "')");
            MySqlCommand comm = new MySqlCommand(sql, conn2);
            comm.ExecuteNonQuery();
            DataSet ds = new DataSet();
            string sql5 = "SELECT att.id_att as '№',att.txt as 'Оголошення',
login.name as 'Автор' FROM `att` INNER JOIN login on att.id_kor= login.id_kor";
            MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
            dataAdapter.Fill(ds, "Users");
            dataGridView13.DataSource = ds.Tables["Users"];
            dataGridView13.Columns[0].Width = 40;

```

```

        dataGridView13.Columns[1].Width = 440;
        dataGridView13.Columns[2].Width = 300;
        conn2.Close();
    }
    catch { MessageBox.Show("Нет соединения с базой"); }
}
}
private void radioButton8_CheckedChanged(object sender, EventArgs e)
{
    groupBox19.Visible = true;
    button26.Visible = false;
    button27.Visible = true;
    button28.Visible = false;
    button10.Visible = false;
}
private void radioButton7_CheckedChanged(object sender, EventArgs e)
{
    groupBox19.Visible = true;
    button26.Visible = false;
    button27.Visible = false;
    button28.Visible = true;
    button10.Visible = true;
}
private void button26_Click(object sender, EventArgs e)
{
    try
    {
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        string sql = "UPDATE `att` SET `txt`=' " + textBox3.Text + " ',`id_kor`="
        + kor_id + "' WHERE id_att='" + att_id + "'";
        MySqlCommand comm = new MySqlCommand(sql, conn2);
        comm.ExecuteNonQuery();
        DataSet ds = new DataSet();
        string sql5 = "SELECT att.id_att as '№',att.txt as 'Оголошения',
login.name as 'Авrop' FROM `att` INNER JOIN login on att.id_kor= login.id_kor";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView13.DataSource = ds.Tables["Users"];
        dataGridView13.Columns[0].Width = 40;
        dataGridView13.Columns[1].Width = 440;
        dataGridView13.Columns[2].Width = 300;
    }
    catch { }
}
private void radioButton6_CheckedChanged(object sender, EventArgs e)
{
    groupBox19.Visible = true;
    button26.Visible = true;
    button27.Visible = false;
    button28.Visible = false;
    button10.Visible = true;
}
private void button10_Click(object sender, EventArgs e)
{
    try
    {
        double x;
        x = Convert.ToDouble(dataGridView13.SelectedCells[0].Value.ToString());
        double id_att = Convert.ToDouble(x);
        att_id= Convert.ToString(x);
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        DateTime thisDay = DateTime.Today;
        string s2 = "select txt from att where id_att =' " + id_att + "'";
    }
}

```

```

        MySqlCommand comm2 = new MySqlCommand(s2, conn2);
        MySqlDataReader reader1 = comm2.ExecuteReader();
        while (reader1.Read())
        {
            textBox3.Text = reader1[0].ToString();
        }
        reader1.Close();
    }
    catch { }
}
private void button28_Click(object sender, EventArgs e)
{
    try
    {
        MySqlConnection conn2 = new MySqlConnection(conn);
        conn2.Open();
        string sql = "DELETE FROM `att` WHERE id_att="+att_id ;
        MySqlCommand comm = new MySqlCommand(sql, conn2);
        comm.ExecuteNonQuery();
        DataSet ds = new DataSet();
        string sql5 = "SELECT att.id_att as '№',att.txt as 'Оголошення',
login.name as 'Автор' FROM `att` INNER JOIN login on att.id_kor= login.id_kor";
        MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sql5, conn2);
        dataAdapter.Fill(ds, "Users");
        dataGridView13.DataSource = ds.Tables["Users"];
        dataGridView13.Columns[0].Width = 40;
        dataGridView13.Columns[1].Width = 440;
        dataGridView13.Columns[2].Width = 300;
        //-----
    }
    catch { }
}
private void button25_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string sql = "";
    if (checkBox3.Checked)
    {
        if (kor == 0)
        {
            MessageBox.Show("Оберіть заяву або користувача!");
        }
        else
        {
            if (kor == 1)
            {
                sql = " DELETE FROM `reg` WHERE id_reg = '" + reg_id + "' ";
                MessageBox.Show("Заява видалена");
            }
            if (kor == 2)
            {
                sql = " DELETE FROM `login` WHERE id_kor = '" + log_id + "' ";
                MessageBox.Show("Користувач видален");
            }
            MySqlCommand comm = new MySqlCommand(sql, conn2);
            comm.ExecuteNonQuery();
            DataSet ds3 = new DataSet();
            string sql3 = "SELECT id_reg as 'Телефон', name as 'Ім`я', spec as
'Спеціальність', pass 'пароль' FROM `reg`";
            MySqlDataAdapter dataAdapter3 = new MySqlDataAdapter(sql3, conn2);
            dataAdapter3.Fill(ds3, "Users");
            dataGridView12.DataSource = ds3.Tables["Users"];
            DataSet ds4 = new DataSet();

```

```

        string sql4 = "SELECT id_kor as 'Телефон', name as 'Ім`я', spec as
'Sпеціальність', pass 'пароль' FROM `login`";
        MySqlDataAdapter dataAdapter4 = new MySqlDataAdapter(sql4, conn2);
        dataAdapter4.Fill(ds4, "Users");
        dataGridView14.DataSource = ds4.Tables["Users"];
    }
    checkBox3.Checked = false;
}
else
{
    MessageBox.Show("Поставте відмітку для видалення");
}
}
private void button30_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    kor = 1;
    try
    {
        string x;
        x = Convert.ToString(dataGridView12.SelectedCells[0].Value.ToString());
        reg_id = Convert.ToString(x);
        string s3 = "SELECT * " +
            "FROM reg WHERE id_reg = '" + reg_id + "'";
        MySqlCommand comm3 = new MySqlCommand(s3, conn2);
        MySqlDataReader reader3 = comm3.ExecuteReader();
        int i = 0;
        while (reader3.Read())
        {
            comboBox13.Text = reader3[1].ToString();
            textBox7.Text = reader3[3].ToString();
            textBox10.Text = reader3[0].ToString();
            textBox11.Text = reader3[2].ToString();
        }
        reader3.Close();
    }
    catch
    {
    }
}
private void button31_Click(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    kor = 2;
    try
    {
        string x;
        x = Convert.ToString(dataGridView14.SelectedCells[0].Value.ToString());
        log_id = Convert.ToString(x);

        string s3 = "SELECT * " +
            "FROM login WHERE id_kor = '" + log_id + "'";
        MySqlCommand comm3 = new MySqlCommand(s3, conn2);
        MySqlDataReader reader3 = comm3.ExecuteReader();
        int i = 0;
        while (reader3.Read())
        {
            comboBox13.Text = reader3[1].ToString();
            textBox7.Text = reader3[3].ToString();
            textBox10.Text = reader3[0].ToString();
            textBox11.Text = reader3[2].ToString();
        }
    }
}

```

```

        reader3.Close();
    }
    catch { }
}
private void button29_Click_1(object sender, EventArgs e)
{
    MySqlConnection conn2 = new MySqlConnection(conn);
    conn2.Open();
    string sql = "";
    if (comboBox13.Text == "" || textBox7.Text == "" || textBox10.Text == "" ||
textBox11.Text == "" )
    {
        MessageBox.Show("Заповніть всі поля!");
    }
    else
    {
        string spec = comboBox13.Text;
        string name = textBox7.Text;
        string tel = textBox10.Text;
        string pass = textBox11.Text;
        sql = "INSERT INTO `login`(`id_kor`, `spec`, `pass`, `name`) VALUES
('"+tel+"', '"+spec+"', '"+pass+"', '"+name+"')";
        MySqlCommand comm = new MySqlCommand(sql, conn2);
        comm.ExecuteNonQuery();
        DataSet ds3 = new DataSet();
        string sql3 = "SELECT id_reg as 'Телефон', name as 'Ім'я', spec as
'Спеціальність', pass 'пароль' FROM `reg`";
        MySqlDataAdapter dataAdapter3 = new MySqlDataAdapter(sql3, conn2);
        dataAdapter3.Fill(ds3, "Users");
        dataGridView12.DataSource = ds3.Tables["Users"];
        DataSet ds4 = new DataSet();
        string sql4 = "SELECT id_kor as 'Телефон', name as 'Ім'я', spec as
'Спеціальність', pass 'пароль' FROM `login`";
        MySqlDataAdapter dataAdapter4 = new MySqlDataAdapter(sql4, conn2);
        dataAdapter4.Fill(ds4, "Users");
        dataGridView14.DataSource = ds4.Tables["Users"];
    }
}
}
}

```

Лістинг коду Web-додатку

Головна сторінка

```

<html>
<head>
<title>Archery: Competition</title>
<link rel="stylesheet" href="sport/home4.css">
<link rel="stylesheet" href="sport/golovna4.css">
<link rel="stylesheet" href="o_nas/o_nas.css">
<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">
<link rel='icon' href='sport\Archery.png'>
<meta charset='utf-8'>
<body class="body">
<div class="shap">
<ul id="nav">
    <div class="logo"></div>
    <li><a href="home1.php"> <div class="glav"></div> Головна</a> </li>
    <li><a href="sportsmans.php"> Спортсмени</a>
    <ul id="second">
</ul>
</li>
</ul>
</div>
<div class="php">
<pre>
$db = mysql_connect("localhost","root","")
or die("Could not connect: " . mysql_error());
mysql_select_db("sport", $db)
or die("Could not select DB: " . mysql_error());
$s= mysql_query("SELECT * from `sity` ", $db);
while($sity = mysql_fetch_array($s)){

```

```

        echo"<li><a href='sportsmans.php?id_sity=" . $sity['id_sity'] . "'>
". $sity['name'] . "</a>      ";
    }
    ?>
        </ul>
        </li>
        <li><a href="comp.php"> Змагання</a>
        <ul id="second">
<?php
$v= mysql_query("SELECT distinct god_comp from `comp` order by god_comp DESC
", $db);
while($god_comp = mysql_fetch_array($v)){
    echo"<li><a href='comp.php?god_comp=" . $god_comp['god_comp'] . "'>
". $god_comp['god_comp'] . "</a>      ";
}
?>
        </ul>
        </li>
        <li><a href="anketa.php">Анкета</a> </li>
        <li style='float:right'><a href="admin.php"> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
</a></li>
</ul>
</div>
<div class='ots'></div>
<div align="center">
<div class='Competition' align="center">
    <?php
$today = date("Y-m-d");
$g= mysql_query("SELECT * from `comp` where final > '$today' order by start ", $db);
while($gr = mysql_fetch_array($g)){
    //echo $gr[name];
    if($c1==''){
        $c1=$gr[name] . ' (м. ' . $gr[sity_comp] . ')';
        $c11='(з ' . $gr[start] . ' по ' . $gr['final'] . ') ' ;
        $c111=$gr[id_comp];
    }
    else{
        if($c2==''){
            $c2=$gr[name] . ' (м. ' . $gr[sity_comp] . ')';
            $c22='(з ' . $gr[start] . ' по ' . $gr['final'] . ') ' ;
            $c222=$gr[id_comp];
        }
        else{
            $c3=$gr[name] . ' (м. ' . $gr[sity_comp] . ')';
            $c33='(з ' . $gr[start] . ' по ' . $gr['final'] . ') ' ;
            $c333=$gr[id_comp];
        }
    }
}}
if($c1==''){
    $c1="ЗМАГАННЯ НЕ НАЗНАЧЕНО";
}
if($c2==''){
    $c2="ЗМАГАННЯ НЕ НАЗНАЧЕНО";
}
if($c3==''){
    $c3="ЗМАГАННЯ НЕ НАЗНАЧЕНО";
}
?>
<?php
echo '<p class=nadp>' . " НАГАЛЬНІ ТА НАСТУПНІ ЗМАГАННЯ
". '</p>';
echo '<a href="comp.php#'. $c111 . "'><div class=Competition1 align="center">
';
        echo $c1 ;
        echo '<pre>

```

```

        </pre>';
        echo $c11;
        echo ' </div></a>';
echo '<a href="comp.php#'. $c222. '"><div class=Competition1 align="center"> ' ;
        echo $c2 ;
        echo '<pre>
        </pre>';
        echo $c22;
        echo '
</div></a>';
echo '<a href="comp.php#'. $c333. '"><div class=Competition1 align="center">
    ' ;
        echo $c3 ;
        echo '<pre>
        </pre>';
        echo $c33;
        echo ' </div></a>';

?>
</div>
<div class='att' align="center">
<?php
    $att= mysql_query("SELECT login.name as name, att.txt as txt from `att`
inner join login on att.id_kor=login.id_kor order by att.id_att desc",$db);
while($attr = mysql_fetch_array($att)){
    //echo $gr[name];
    if($a1==''){
        $a1=$attr[name];
        $a11=$attr[txt];
        $a111=$attr[name];
    }
    else{
        if($a2==''){
            $a2=$attr[name];
            $a22=$attr[txt];
            $a222=$attr[name];
        }
        else{
            $a3=$attr[name];
            $a33=$attr[txt];
            $a333=$attr[name];
        }
    }
}}
if($a1==''){
    $a1="ОГОЛОШЕННЯ ВІДСУТНЄ";
}
if($a2==''){
    $a2="ОГОЛОШЕННЯ ВІДСУТНЄ";
}
if($a3==''){
    $a3="ОГОЛОШЕННЯ ВІДСУТНЄ";
}
echo '<p class=nadp>'. "ОГОЛОШЕННЯ
    ". '</p>';
echo '<pre><div class=att2 align="center">';
        echo $a11;
        echo '<div align="right"><div class=att1 align="center">';
        echo $a1 ;
        echo '</div></div>';
        echo '</div>';
echo '<div class=att2 align="center">';
        echo $a22;
        echo '<div align="right"><div class=att1 align="center">';
        echo $a2 ;
        echo '</div></div></a>';
        echo '</div></a>';

```



```

        echo '<div class=att2 align="center">';
            echo $a33;
            echo '<div align="right"><div class=att1 align="center">';
                echo $a3 ;
            echo '</div></div></a>';
            echo '</div></a></pre>';
        echo '<div class=att3> </div>';
        ?>
    </div>
</div>
</body>
</html>

```

Сторінка «Спортсмени»

```

<?php
session_start();
if($_COOKIE['sort']!=''){
$_SESSION['sort']=$_COOKIE['sort'];
}
SetCookie('sort','');
$_COOKIE['sort']='';
?>
<html>
<head>
<title>Archery: Sportsman</title>
<link rel="stylesheet" href="sport/home7.css">
<link rel="stylesheet" href="o_nas/o_nas.css">
<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">
<link rel='icon' href='sport\Archery.png'>
<style>
.opisanie{
    display:inline-block;
    vertical-align:top;
    font-family: 'exo 2';
}
</style>
<meta charset='utf-8'>
<body class="body">

<div class="shap">
<ul id="nav">
    <div class="logo"></div>
    <li><a href="home1.php"> <div class="glav"></div> Головна</a> </li>
    <li><a href="sportsmans.php"> Спортсмени</a>
    <ul id="second">

<?php
$db = mysql_connect("localhost","root","")
or die("Could not connect: " . mysql_error());
mysql_select_db("sport", $db)
or die("Could not select DB: " . mysql_error());
$s= mysql_query("SELECT * from `sity` ",$db);
while($sity = mysql_fetch_array($s)){
    echo"<li><a href='sportsmans.php?id_sity=".$sity['id_sity']."'>
".$sity['name']."</a> ";
}
?>
        </ul>
        </li>
        <li><a href="comp.php"> Змагання</a>
    <ul id="second">

<?php

```

```

$vv= mysql_query("SELECT distinct god_comp from `comp` order by god_comp DESC",$db);
while($god_comp = mysql_fetch_array($vv)){
    echo"<li><a href='comp.php?god_comp=".$god_comp['god_comp']."'>
".$god_comp['god_comp']."</a>";
}
?>
        </ul>
        </li>
        <li><a href="anketa.php">Анкета</a> </li>
        <li style='float:right'><a href="admin.php"> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
</a></li>
</ul>
</div>
<div class='ots'></div>
<div align="center">
<div id='sortdiv'>
    <?php

$cook= mysql_query("SELECT * from cook ",$db);
$cook1 = mysql_fetch_array($cook);
if ($cook1[cook]!='')
{
}
else
{
    echo"<script>>window.location.href='vhod1.php'</script>";
}

$sportsman=$_GET['rang'];
$_SESSION['sportsman']=$sportsman;

if($_SESSION['sort']==' || $_SESSION['sort']=='Від молодшого до старшого'){
$order=' order by `sportsman`.`god`';}
if($_SESSION['sort']=='Від старшого до молодшого'){
$order=' order by `sportsman`.`god` DESC';}
if($_SESSION['sort']=='Нові записи спочатку'){
$order=' order by `sportsman`.`id_sp` DESC';}
if($_SESSION['sort']!='' || $_GET['rang']!=''){
echo"<form class='unset' action=' ' method='post'>";
echo"<button rang='submit' name='subunset'><img src='home/unset.png'> Очистити
фільтри</button></form>";
}

if(isset($_POST['subunset'])){
$_SESSION['sort']='';
$id_sity=$_GET['id_sity'];
echo"<script>>window.loesityon.href='sportsmans.php?id_sity='</script>";
}
echo"<form method='post' multiple>";
echo"<select onchange='document.cookie =
`sort=`+this.options[this.selectedIndex].value;window.loesityon.href=window.loesityon.href'>";
$sort=$_SESSION['sort'];

if($sort=='Від молодшого до старшого' || $sort==''){
echo"<option name='op1' selected value='Від молодшого до старшого'>Від молодшого до
старшого</option>";}
else{
echo"<option name='op1' value='Від молодшого до старшого'>Від молодшого до
старшого</option>";}
if($sort=='Від старшого до молодшого'){
echo"<option name='op2' selected value='Від старшого до молодшого'>Від старшого до
молодшого</option>";}

```

```

else{
echo"<option name='op2' value='Від старшого до молодшого'>Від старшого до
молодшого</option>";}
if($sort=='Нові записи спочатку'){
echo"<option name='op3' selected value='Нові записи спочатку'>Нові записи
спочатку</option></select></fieldset>";}
else{
echo"<option name='op3' value='Нові записи спочатку'>Нові записи
спочатку</option></select>";}

$str="SELECT sportsman.id_sp as 'id', sportsman.name as 'name',sportsman.god as
'god',sportsman.rang as 'rang',sity.name as 'sity',school.name as 'school' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity ";
if($_GET['id_sity']!=''){
$id_sity=$_GET['id_sity'];
$str.="where sity.id_sity='$id_sity' ";}
if($_GET['rang']!=''){
$rangs=$_GET['rang'];
$str.=" and sportsman.rang='$rangs'";}
$str.= $order;
$qr_result = mysql_query($str,$db);
if (!$qr_result)
{echo $str; }
if($_GET['id_sity']!=''){
echo"<br><h1 style='font-size:20px;margin:30px auto 3px'>Ранг:</h1>";
$sort1=mysql_query("SELECT distinct rang from `sportsman` where
id_sity='$id_sity'", $db);
while($data1 = mysql_fetch_array($sort1)){
if($_GET['rang']==$data1['rang']){ echo"<font style='font-
weight:bold;background:gold;font-size:20px'>";}
echo"<a
href='sportsmans.php?id_sity=$id_sity&rang=".$data1['rang']."'>".$data1['rang']."</
a><br>";
if($_GET['rang']==$data1['rang']){ echo"</font>";}
}}
else{
echo"<br><h1 style='font-size:20px;margin:30px auto
3px'>Micro:</h1>";
$sort1=mysql_query("SELECT * from `sity`", $db);
while($data1 = mysql_fetch_array($sort1)){
echo"<a
href='sportsmans.php?id_sity=".$data1['id_sity']."'>".$data1['name']."</a><br>";
}}

echo"</form>";

?>
</div>

<?php

echo"<div class='opisanie'>";

while($data = mysql_fetch_array($qr_result)){
/*echo '<a>
<div class="product" border-color=silver id="p'.$data['id']."'>';*/

echo '<a>
<div class="product" border-color=silver id="p'.$data['id']."'>';

```

```

        echo '<div class="text" id="'. $data['id']. '"><p
class="name_product">'. $data['name']. '</p>';
        echo ' <p class="price">Micro: ' . $data['sity'] . ' </p>';
        echo '<p style="margin-left:10px;"><strike>Ранг:</strike>
'. $data['rang']. '</p>';

        $list="SELECT * from `list`";
        $list_result = mysql_query($list,$db);
        $list2='';
        while($data2 = mysql_fetch_array($list_result)){

                $list2.= $data2['name']. " (". $data2['god']. " р.н.)" . $data2['rang']. "
                ";

        }

if($data['school']!=''){
        echo '<p class="komentar"><br><pre>Рік народження: '. $data['god']. '
        Спортивна школа: '. $data['school']. '</pre></p>';}
        echo"<div class='mores' id='more". $data['id']. "' style='display:none;'
        OnClick='var t=document.getElementById(\". $data['id']. "\");
if(this.childNodes[0].src.indexOf(`home/s1.png`)>0){
                t.style.height=`auto`;
                this.style.bottom=0;

                document.getElementById(`p". $data['id']. "`).style.overflow=`visible`;

                this.childNodes[0].src=`home/s2.png`;
        }
        else{
                var i=t.clientHeight;
                t.height=`250px`;
                t.style.height=`250px`;
                this.style.bottom=i-250+2;

                document.getElementById(`p". $data['id']. "`).style.overflow=`hidden`;
                this.childNodes[0].src=`home/s1.png`;
        }
        '><img src='home/s1.png'></div>";
echo"</div>";
echo"</div></a>";
        echo"<script>
        var t=document.getElementById(\". $data['id']. "\");
        var i=t.clientHeight;
        if(i>250){
                t.height='250px';
                t.style.height='250px';
                var m=document.getElementById('more". $data['id']. "'");
                m.style.display='';
                m.style.bottom=i-250+30;
        }
        </script>";
}
mysql_close($db);

?>
</div>
</body>

</html>

```

Сторінка «Змагання»

<?php

```

session_start();
if($_COOKIE['sort']!=''){
$_SESSION['sort']=$_COOKIE['sort'];
}
SetCookie('sort','');
$_COOKIE['sort']='';
?>
<html>
<head>
<title>Archery: Sportsman</title>
<link rel="stylesheet" href="sport/home7.css">
<link rel="stylesheet" href="o_nas/o_nas.css">
<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">
<link rel='icon' href='sport\Archery.png'>
<style>
.opisanie{
    display:inline-block;
    vertical-align:top;

    font-family: 'exo 2';
}
</style>
<meta charset='utf-8'>
<body class="body">

<div class="shap">
<ul id="nav">
    <div class="logo"></div>
    <li><a href="homel.php"> <div class="glav"></div> Головна</a> </li>
    <li><a href="sportsmans.php"> Спортсмени</a>
    <ul id="second">

<?php
$db = mysql_connect("localhost","root","")
or die("Could not connect: " . mysql_error());
mysql_select_db("sport", $db)
or die("Could not select DB: " . mysql_error());
$s= mysql_query("SELECT * from `sity` ",$db);
while($sity = mysql_fetch_array($s)){
    echo"<li><a href='sportsmans.php?id_sity=".$sity['id_sity']."'>
".$sity['name']."</a>      ";
}
?>

        </ul>
    </li>
    <li><a href="comp.php"> Змагання</a>
    <ul id="second">

<?php
$v= mysql_query("SELECT distinct god_comp from `comp` order by god_comp DESC", $db);
while($god_comp = mysql_fetch_array($v)){
    echo"<li><a href='comp.php?god_comp=".$god_comp['god_comp']."'>
".$god_comp['god_comp']."</a>      ";
}
?>

        </ul>
    </li>
    <li><a href="anketa.php">Анкета</a> </li>
    <li style='float:right'><a href="admin.php"> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
</a></li>
</ul>
</div>
<div class='ots'></div>
<div align="center">
<div id='sortdiv'>

```

```

<?php

$cook= mysql_query("SELECT * from cook ",$db);
$cook1 = mysql_fetch_array($cook);
if ($cook1[cook]!='')
{
}
else
{
    echo"<script>>window.location.href='vhod1.php'</script>";
}

$sportsman=$_GET['rang'];
$_SESSION['sportsman']=$sportsman;

if($_SESSION['sort']==' || $_SESSION['sort']=='Від молодшого до старшого'){
$order=' order by `sportsman`.`god`;
}
if($_SESSION['sort']=='Від старшого до молодшого'){
$order=' order by `sportsman`.`god` DESC';
}
if($_SESSION['sort']=='Нові записи спочатку'){
$order=' order by `sportsman`.`id_sp` DESC';
}
if($_SESSION['sort']!=' || $_GET['rang']!='){
echo"<form class='unset' action='' method='post'>";
echo"<button rang='submit' name='subunset'><img src='home/unset.png'> Очистити
фільтри</button></form>";
}

if(isset($_POST['subunset'])){
$_SESSION['sort']='';
$id_sity=$_GET['id_sity'];
echo"<script>>window.losityion.href='sportsmans.php?id_sity='</script>";
}
echo"<form method='post' multiple>";
echo"<select onchange='document.cookie =
`sort`+this.options[this.selectedIndex].value;window.losityion.href=window.losityi
on.href'>";
$sort=$_SESSION['sort'];

if($sort=='Від молодшого до старшого' || $sort==''){
echo"<option name='op1' selected value='Від молодшого до старшого'>Від молодшого до
старшого</option>";}
else{
echo"<option name='op1' value='Від молодшого до старшого'>Від молодшого до
старшого</option>";}
if($sort=='Від старшого до молодшого'){
echo"<option name='op2' selected value='Від старшого до молодшого'>Від старшого до
молодшого</option>";}
else{
echo"<option name='op2' value='Від старшого до молодшого'>Від старшого до
молодшого</option>";}
if($sort=='Нові записи спочатку'){
echo"<option name='op3' selected value='Нові записи спочатку'>Нові записи
спочатку</option></select></fieldset>";}
else{
echo"<option name='op3' value='Нові записи спочатку'>Нові записи
спочатку</option></select>";}

$str="SELECT sportsman.id_sp as 'id', sportsman.name as 'name',sportsman.god as
'god',sportsman.rang as 'rang',sity.name as 'sity',school.name as 'school' FROM
`sportsman` INNER JOIN school on school.id_s=sportsman.id_s INNER JOIN sity on
sity.id_sity=sportsman.id_sity ";
if($_GET['id_sity']!='){
    $id_sity=$_GET['id_sity'];
}

```

```

$str.="where sity.id_sity='$id_sity' ";}
if($_GET['rang']!=''){
$rangs=$_GET['rang'];
$str.=" and sportsman.rang='$rangs'";}
$str.=$order;
$qr_result = mysql_query($str,$db);
if (!$qr_result)
{echo $str; }
if($_GET['id_sity']!=''){
echo"<br><h1 style='font-size:20px;margin:30px auto 3px'>Ранг:</h1>";
$sort1=mysql_query("SELECT distinct rang from `sportsman` where
id_sity='$id_sity'", $db);
while($data1 = mysql_fetch_array($sort1)){
if($_GET['rang']==$data1['rang']){ echo"<font style='font-
weight:bold;background:gold;font-size:20px'>";}
echo"<a
href='sportsmans.php?id_sity=$id_sity&rang=".$data1['rang']."'>".$data1['rang']."</
a><br>";
if($_GET['rang']==$data1['rang']){ echo"</font>";}
}}
else{
echo"<br><h1 style='font-size:20px;margin:30px auto
3px'>Micro:</h1>";
$sort1=mysql_query("SELECT * from `sity`", $db);
while($data1 = mysql_fetch_array($sort1)){
echo"<a
href='sportsmans.php?id_sity=".$data1['id_sity']."'>".$data1['name']."</a><br>";
}}

echo"</form>";

?>
</div>

<?php

echo"<div class='opisanie'>";

while($data = mysql_fetch_array($qr_result)){
/*echo '<a>
<div class="product" border-color=silver id="p'.$data['id']."'>*/

echo '<a>
<div class="product" border-color=silver id="p'.$data['id']."'>';

echo '<div class="text" id="'.$data['id']."'><p
class="name_product">'.$data['name'].'</p>';
echo '<p class="price">Micro: ' . $data['sity'] . ' </p>';
echo '<p style="margin-left:10px;"><strike>Ранг:</strike>
'.$data['rang'].'</p>';

$list="SELECT * from `list`";
$list_result = mysql_query($list,$db);
$list2='';
while($data2 = mysql_fetch_array($list_result)){

$list2.= $data2['name']." (".$data2['god']." p.н.)" . $data2['rang']."
";

}
}

```

```

if($data['school']!=''){
    echo '<p class="koment"><br><pre>Рік народження: '.$data['god'].'
    Спортивна школа: '.$data['school'].'</pre></p>';}
    echo"<div class='mores' id='more".$data['id']."' style='display:none;'
    OnClick='var t=document.getElementById(\".$data['id']."');
if(this.childNodes[0].src.indexOf(`home/s1.png`)>0){
    t.style.height=`auto`;
    this.style.bottom=0;

    document.getElementById(`p".$data['id']."`).style.overflow=`visible`;

        this.childNodes[0].src=`home/s2.png`;
    }
    else{
        var i=t.clientHeight;
        t.height=`250px`;
        t.style.height=`250px`;
        this.style.bottom=i-250+2;

        document.getElementById(`p".$data['id']."`).style.overflow=`hidden`;
        this.childNodes[0].src=`home/s1.png`;
    }
    '><img src='home/s1.png'></div>";
echo"</div>";
echo"</div></a>";
echo"<script>
var t=document.getElementById('".$data['id']."');
var i=t.clientHeight;
if(i>250){
    t.height='250px';
    t.style.height='250px';
    var m=document.getElementById('more".$data['id']."');
    m.style.display='';
    m.style.bottom=i-250+30;
}
</script>";

}
mysql_close($db);

?>
</div>
</body>

</html>

```

Сторінка «Анкета»

```

<html>
<head>
<title>Archery: Anketa</title>
<link rel="stylesheet" href="sport/home4.css">
<link rel="stylesheet" href="sport/golovna9.css">
<link rel="stylesheet" href="o_nas/o_nas.css">
<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">
<link rel='icon' href='sport\Archery.png'>
<meta charset='utf-8'>
<body class="body">

<div class="shap">
<ul id="nav">
    <div class="logo"></div>
    <li><a href="home1.php"> <div class="glav"></div> Головна</a> </li>
    <li><a href="sportsmans.php"> Спортсмени</a>

```



```

        $sql_ank = "INSERT INTO `anketa`(`id_ank`, `id_comp`, `name`, `tell`,
`god_r`, `school`, `sity`) VALUES
('$max_id2', '$id_comp2', '$pib', '$tel', '$god', '$school', '$sity');"
        //echo $sql_ank;
        mysql_query($sql_ank, $db);
        //session_unset();
        //exit();
        $p_id= mysql_query("SELECT * from anketa where id_ank=$max_id2", $db);
        $p_id1 = mysql_fetch_array($p_id);
        if($p_id1[id_ank]!=''){
            $otv= 'Запит на участь у змаганні відправлено';
        }
    }
}
else
{
    echo"<script>window.location.href='vhod.php'</script>";
}
?>

```

```

<div class='anketa1' >
    <p class=nadp1>АНКЕТА НА УЧАСТЬ</p>
<div class='anketa2' align="center">
    <form method="POST">

```

```

<fieldset class='field1' style='width:700px; font-size: 23; text-align:left;margin:0px auto;'>

```

```

Змагання<br>
<select class= txt name="comp" method="POST">
<?PHP
$today = date("Y-m-d");
$toyear= date("Y");
$result=mysql_query("SELECT * FROM comp where start > '$today' order by
start", $db);
$i=0;
while ($doc = mysql_fetch_array($result))
{
echo "<option value='".$doc['name']."'>".$doc['name']."</option>";
$i++;
}
    echo '</select><br>

    ПІВ<br><input class= txt name="pib" type="text" required><br>
    Рік народження <br><input class= txt2 name="god" type="number" max="$toyear"
value="1999" required><br>
    Номер тел. <br><input class= txt3 name="tel" type="number" required><br>
    Спортивна школа <br><input class= txt name="school" type="text"
required><br>
    Місто<br><input class= txt name="sity" type="text" required><br>';
    ?>
<input class="center1" name="submit" type="submit" value="ВІДПРАВИТИ ЗАПИТ">
</fieldset>
</form>
<div class='otv' align="center">
<?php
echo $otv;
?>
</div>
</div>

```

```

</div>
</body>

</html>

```

Сторінка «Авторизації»

```

<html>
<head>
<title>Archery: Competition</title>
<link rel="stylesheet" href="sport/home4.css">
<link rel="stylesheet" href="sport/golovna7.css">
<link rel="stylesheet" href="o_nas/o_nas.css">
<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">
<link rel='icon' href='sport\Archery.png'>
<meta charset='utf-8'>
<body class="body">

<div class="shap">
<ul id="nav">
<div class="logo"></div>
<li><a href="homel.php"> <div class="glav"></div> Головна</a> </li>
<li><a href="sportsmans.php"> Спортсмени</a>
<ul id="second">
<?php
$db = mysql_connect("localhost","root","")
or die("Could not connect: " . mysql_error());
mysql_select_db("sport", $db)
or die("Could not select DB: " . mysql_error());
$s = mysql_query("SELECT * from `sity` ", $db);
while($sity = mysql_fetch_array($s)){
echo"<li><a href='sportsmans.php?id_sity=".$sity['id_sity']."'>
".$sity['name']."</a>      ";
}
?>
</ul>
</li>
<li><a href="comp.php"> Змагання</a>
<ul id="second">
<?php
$v = mysql_query("SELECT distinct god_comp from `comp` order by god_comp DESC
", $db);
while($god_comp = mysql_fetch_array($v)){
echo"<li><a href='comp.php?god_comp=".$god_comp['god_comp']."'>
".$god_comp['god_comp']."</a>      ";
}
?>
</ul>
</li>
<li><a href="anketa.php">Анкета</a> </li>
<li style='float:right'><a href="admin.php"> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
</a></li>
</ul>
</div>
<div class='ots'></div>
<div align="center">
<div class='anketa' align="center">

<div>

<?php
$login=$_POST['login'];

```

```

    $parol=$_POST['parol'];

    $t= mysql_query("SELECT * from login where pass like '$parol' and id_kor
like '$login' ", $db);
    $rez = mysql_fetch_array($t);
    if($rez['id_kor']!=''){

        $UP= mysql_query("UPDATE `cook` SET `cook`='ank' WHERE id=1", $db);
        echo"<script>window.location.href='anketa.php'</script>";

    }

if($_SESSION['login']==''){
    echo"<form action='' method='post'><fieldset class='field'
style='width:600px;text-align:right;margin:0px auto;line-height:3;'>";

        echo"Логін <input class= txt3 type='text' name='login' required><br>";
        echo"Пароль <input class= txt3 type='password' name='parol'
required></fieldset>";

        echo"<fieldset class='center1' style='margin:0px auto;'><input type='submit'
name='subenter' value='Ввійти'></fieldset>";
        echo"</form>";
    }
    ?>

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

</html>

```

Сторінка «Регістрації»

```

<html>
<head>
<title>Archery: Anketa</title>
<link rel="stylesheet" href="sport/home4.css">
<link rel="stylesheet" href="sport/golovna10.css">
<link rel="stylesheet" href="o_nas/o_nas.css">
<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">
<link rel='icon' href='sport\Archery.png'>
<meta charset='utf-8'>
<body class="body">

<div class="shap">
<ul id="nav">
    <div class="logo"></div>
    <li><a href="home1.php"> <div class="glav"></div> Головна</a> </li>
    <li><a href="sportsmans.php"> Спортсмени</a>
    <ul id="second">

<?php
$db = mysql_connect("localhost","root","")
or die("Could not connect: " . mysql_error());
mysql_select_db("sport", $db)
or die("Could not select DB: " . mysql_error());
$s= mysql_query("SELECT * from `sity` ", $db);
while($sity = mysql_fetch_array($s)){
    echo"<li><a href='sportsmans.php?id_sity=".$sity['id_sity']."'>
".$sity['name']."</a> ";

```

```

}
?>
    </ul>
    </li>
    <li><a href="comp.php"> Змагання</a>
    <ul id="second">
<?php
$v= mysql_query("SELECT distinct god_comp from `comp` order by god_comp DESC
",$db);
while($god_comp = mysql_fetch_array($v)){
    echo"<li><a href='comp.php?god_comp=".$god_comp['god_comp']."'>
".$god_comp['god_comp']."</a>      ";
}
?>
    </ul>
    </li>
    <li><a href="anketa.php">Анкета</a> </li>
    <li style='float:right'><a href="admin.php"> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
</a></li>
</ul>
</div>
<div class='ots'></div>
<div align="center">

<?PHP
$cook= mysql_query("SELECT * from cook ",$db);
$cookl = mysql_fetch_array($cook);
/*if ($cookl[cook]!='')
{
*/
if(isset($_POST['submit']))
{
    $pib = $_POST['pib'];
    $tel = $_POST['tel'];
    $spec = $_POST['spec'];
    $pass = $_POST['pass'];

    $sql_ank = "INSERT INTO `reg` (`id_reg`, `spec`, `pass`, `name`) VALUES
('$tel','$spec','$pass','$pib)";
    //echo $sql_ank;
    mysql_query($sql_ank,$db);
    //session_unset();
    //exit();
    $p_id= mysql_query("SELECT * from reg where id_reg=$tel",$db);
    $p_idl = mysql_fetch_array($p_id);
    if($p_idl[id_reg]!=''){
        $otv= 'Запит відправлено';
    }
}
/*}

else
{
    //echo"<script>window.location.href='vhod.php'</script>";
}*/
?>

<div class='reg' >

```

```

        <p class=nadp1>РЕГІСТРАЦІЯ</p>
        <div class='anketa2' align="center">
            <form method="POST">

                <fieldset class='field1' style='width:700px; font-size: 23; text-
                align:left;margin:0px auto;'>

                    ПІВ<br>                                <input class= txt name="pib" type="text"
                    required><br>
                    Номер тел. <br>                        <input class= txt3 name="tel" type="number"
                    required><br>
                    Спеціальність <br> <input class= txt name="spec" type="text" required><br>
                    Пароль <br>                            <input class= txt3 name="pass" type="password"
                    required><br>
                    <input class="center1" name="submit" type="submit" value="ВІДПРАВИТИ ЗАПИТ">
                </fieldset>
                </form>
                <div class='otv' align="center">
                    <?php
                    echo $otv;
                    ?>
                </div>
            </div>
        </div>
    </body>

</html>

```

ДОДАТОК Г. ІНСТРУКЦІЯ КОРИСТУВАЧА

Загальні відомості про користувальницький інтерфейс додатків

Додаток на платформі Windows призначений для керування даними, що зберігаються в базі даних. Web-додаток слугує для розповсюдження інформації щодо змагань та прийняття анкет на участь у змаганнях.

Запуск додатку на платформі Windows

Після запуску програми на екрані з'явиться вікно «Авторизації». (рис Г.1):

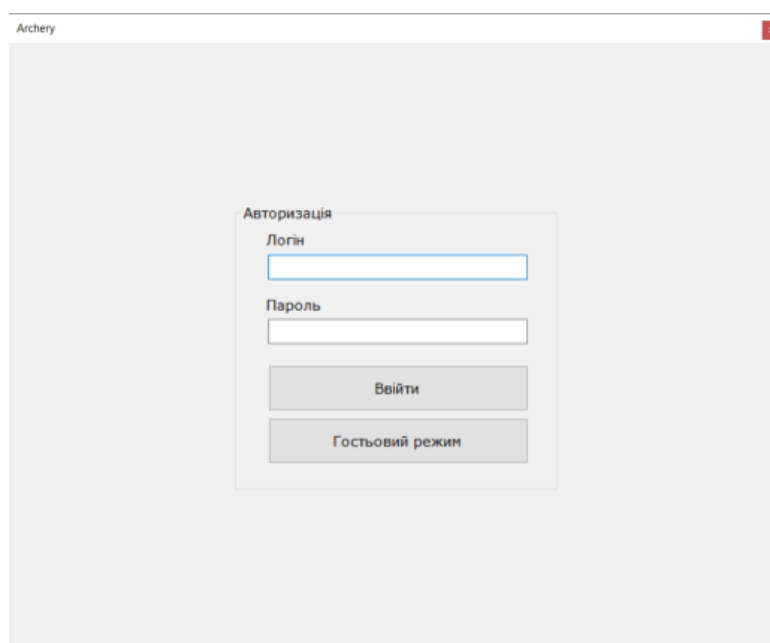


Рисунок Г.1 – Вікно «Авторизації»

Авторизація в додатку

Для авторизації Вам необхідно зробити наступні дії:

- 1) Ввести логін;
- 2) Ввести пароль.

З метою забезпечення безпеки під час введення пароля замість реальних символів в поле «Пароль» вводяться символи «*», в тій же кількості, що і кількість символів у паролі.

3) Натисніть кнопку «Вхід». Після успішної авторизації Ви зможете продовжити роботу з програмою. У випадку помилки на екрані з'явиться відповідне повідомлення про помилку.

4) Користувач який не знає логін і пароль, може продивлятися данні в гостьовому режимі, якщо натисне кнопку «Гостьовий режим».

Керування даними о спортсменах

Щоб додати дані про нового спортсмена до бази даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Спортсмени» (рис. Г.2);
- 3) Обрати дію «Додати» (рис. Г.2);
- 4) Заповнити відповідні поля (рис. Г.2);
- 5) Натиснути кнопку «Додати» (рис. Г.2).

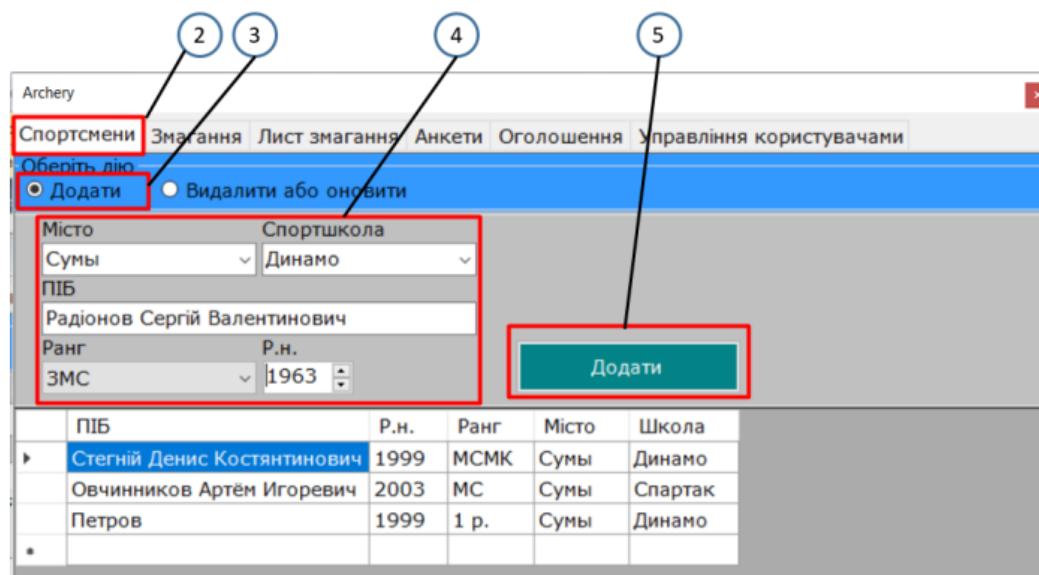


Рисунок Г.2 – Додавання даних про нового спортсмена

Щоб видалити дані про спортсмена з бази даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);

- 2) Натиснути на вкладку «Спортсмени» (рис. Г.3);
- 3) Обрати дію «Видалити або оновити» (рис. Г.3);
- 4) Обрати необхідне місто з списку варіантів (рис. Г.3);
- 5) Обрати необхідного спортсмена з списку варіантів (рис. Г.4);
- 6) Натиснути кнопку «Видалити» (рис. Г.4).

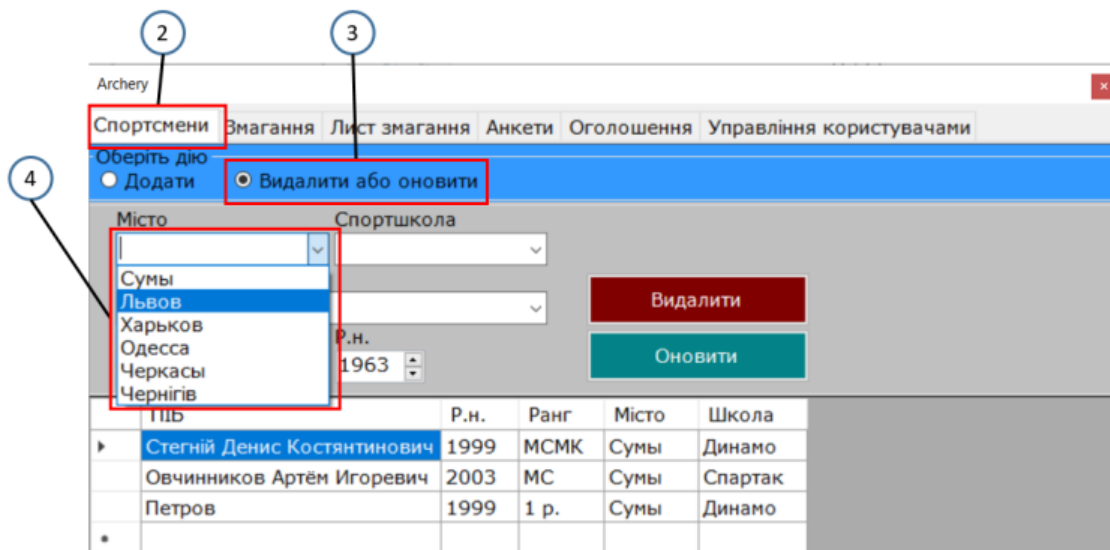


Рисунок Г.3 – Видалення даних про спортсмена

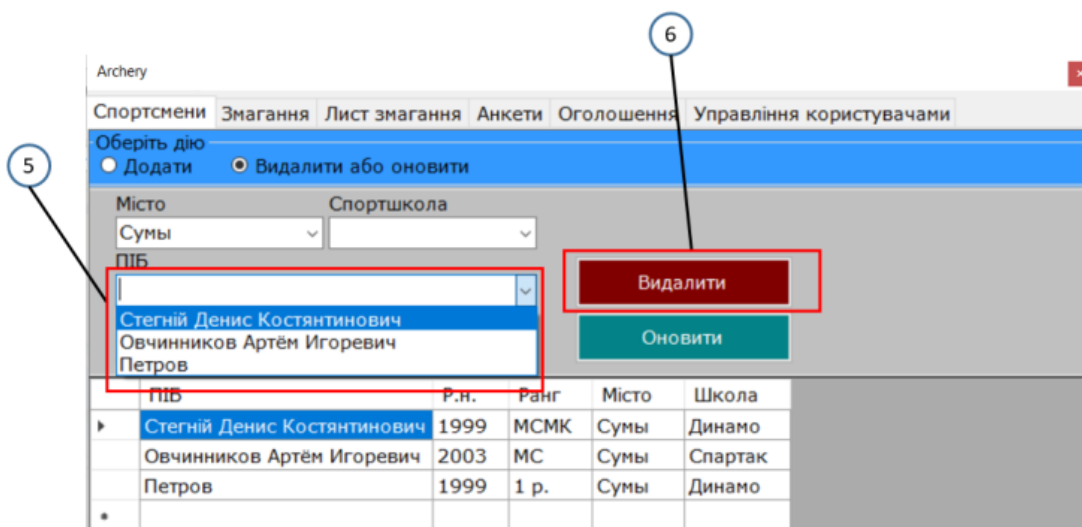


Рисунок Г.4 – Видалення даних про спортсмена

Щоб оновити дані про спортсмена в базі даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Спортсмени» (рис. Г.5);
- 3) Обрати дію «Видалити або оновити» (рис. Г.5);
- 4) Обрати необхідне місто з списку варіантів (рис. Г.5);
- 5) Обрати необхідного спортсмена з списку варіантів (рис. Г.6);
- 6) Зробити необхідні правки в полях для вводу (рис. Г.6);
- 7) Натиснути кнопку «Оновити» (рис. Г.6).

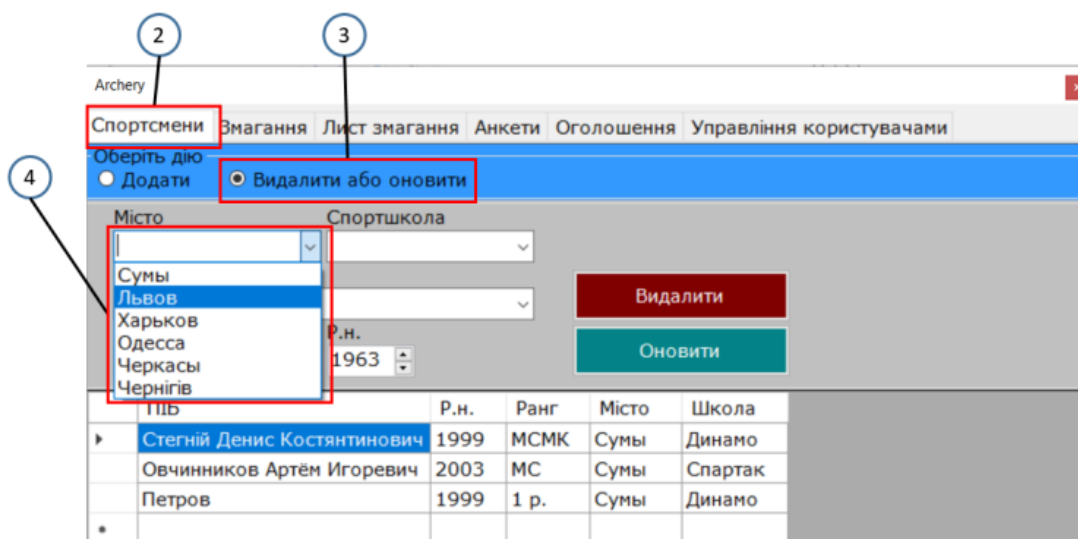


Рисунок Г.5 – Оновлення даних про спортсмена

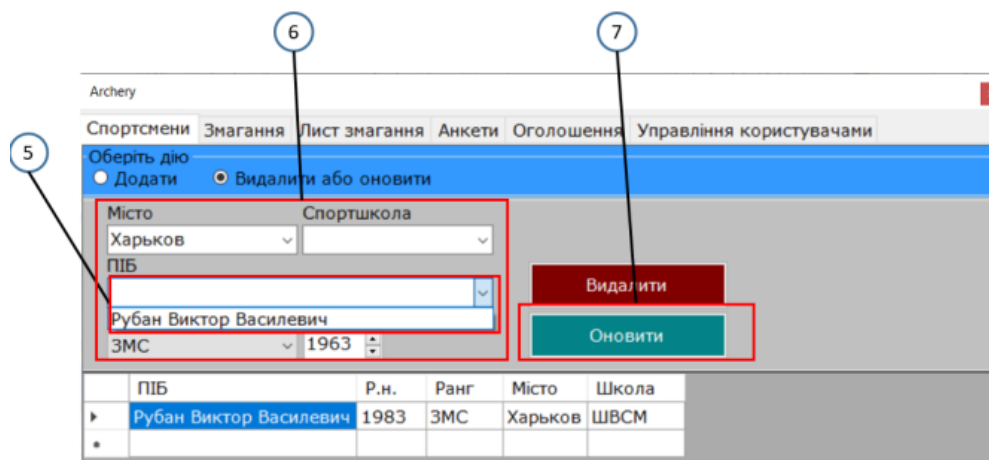


Рисунок Г.6 – Оновлення даних про спортсмена

Керування календарем змагань

Щоб додати дані про змагання до бази даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Змагання» (рис. Г.7);
- 3) Обрати дію «Додати» (рис. Г.7);
- 4) Заповнити відповідні поля (рис. Г.7);
- 5) Натиснути кнопку «Додати» (рис. Г.7).

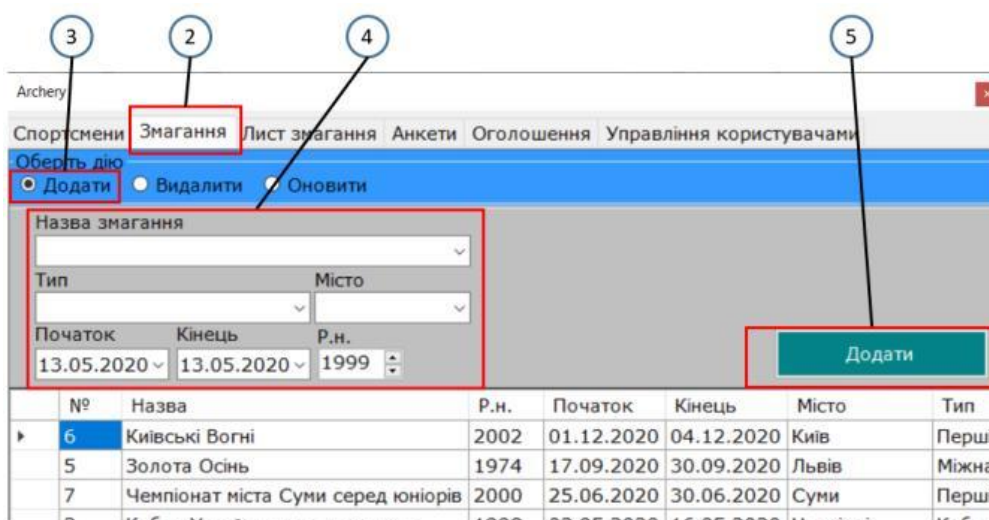


Рисунок Г.7 – Додавання даних про змагання

Щоб видалити дані про змагання з бази даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Змагання» (рис. Г.8);
- 3) Обрати дію «Видалити» (рис. Г.8);
- 4) Обрати необхідне змагання в таблиці (рис. Г.8);
- 5) Натиснути кнопку «Обрати змагання» (рис. Г.8);
- 6) Натиснути кнопку «Видалити» (рис. Г.8).

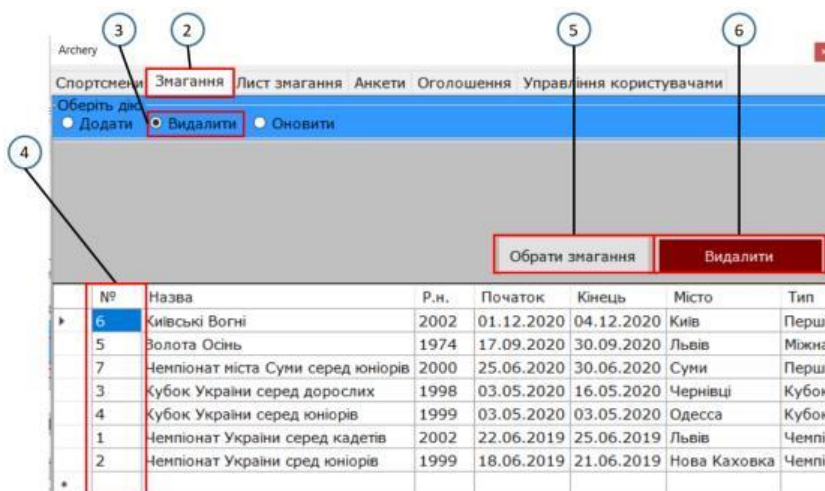


Рисунок Г.8 – Видалення даних про змагання

Щоб оновити дані про спортсмена в базі даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Змагання» (рис. Г.9);
- 3) Обрати дію «Оновити» (рис. Г.9);
- 4) Обрати необхідне змагання в таблиці (рис. Г.9);
- 5) Натиснути кнопку «Обрати змагання» (рис. Г.9);
- 6) Зробити необхідні правки в полях для вводу (рис. Г.9);
- 7) Натиснути кнопку «Оновити» (рис. Г.9).

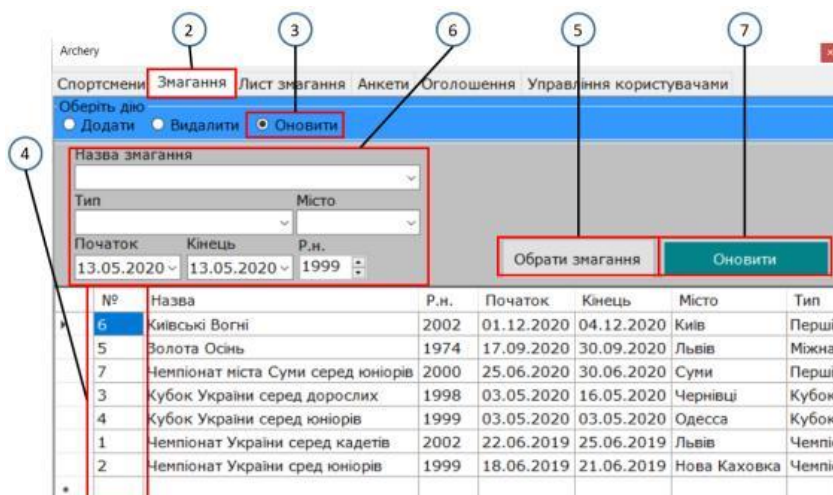


Рисунок Г.9 – Оновлення даних про змагання

Керування списком учасників змагань

Щоб додати спортсмена до списку учасників, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Лист змагання» (рис. Г.10);
- 3) Обрати змагання з списку змагань (рис. Г.10);
- 4) Натиснути кнопку «Обрати змагання» (рис. Г.10);
- 5) Обрати спортсмена з списку спортсменів (рис. Г.10);
- 6) Натиснути кнопку «Обрати спортсмена» (рис. Г.10);
- 7) Натиснути кнопку «Допустити до змагань» (рис. Г.10).

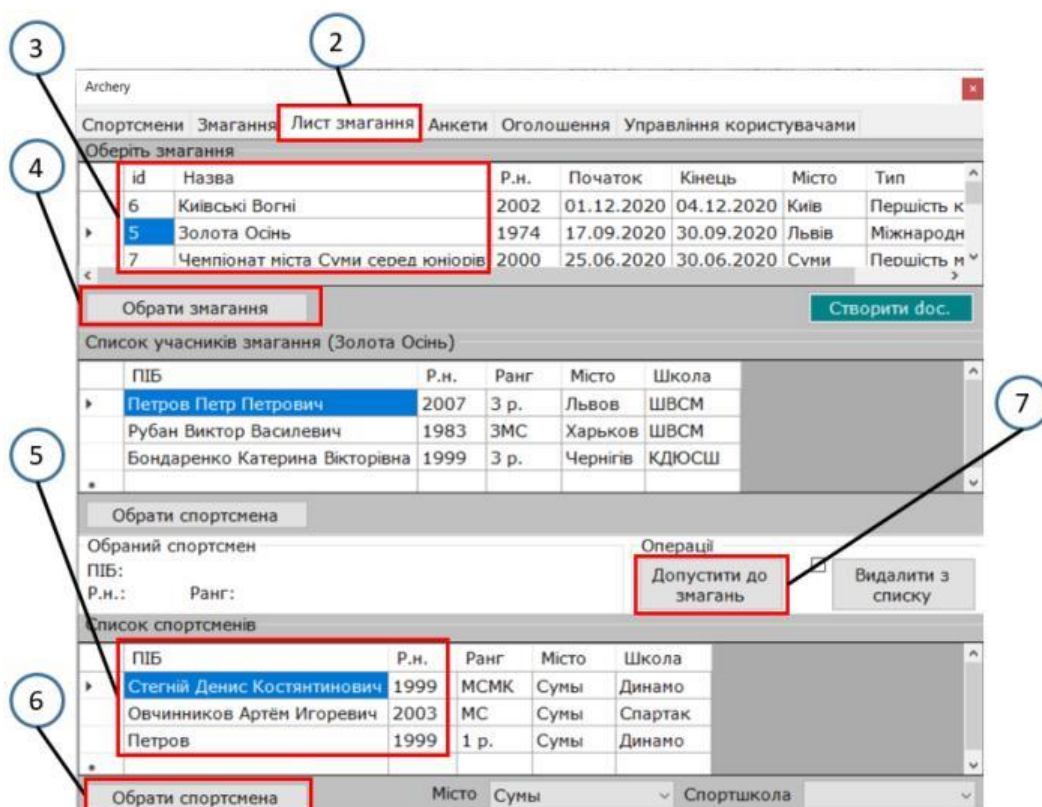


Рисунок Г.10 – Додавання спортсменів до списку учасників

Щоб видалити спортсмена з списку учасників, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Лист змагання» (рис. Г.11);
- 3) Обрати змагання з списку змагань (рис. Г.11);

- 4) Натиснути кнопку «Обрати змагання» (рис. Г.11);
- 5) Обрати спортсмена з списку учасників (рис. Г.11);
- 6) Натиснути кнопку «Обрати спортсмена» (рис. Г.11);
- 7) Поставити відмітку (рис. Г.11);
- 8) Натиснути кнопку «Видалити з списку» (рис. Г.11).

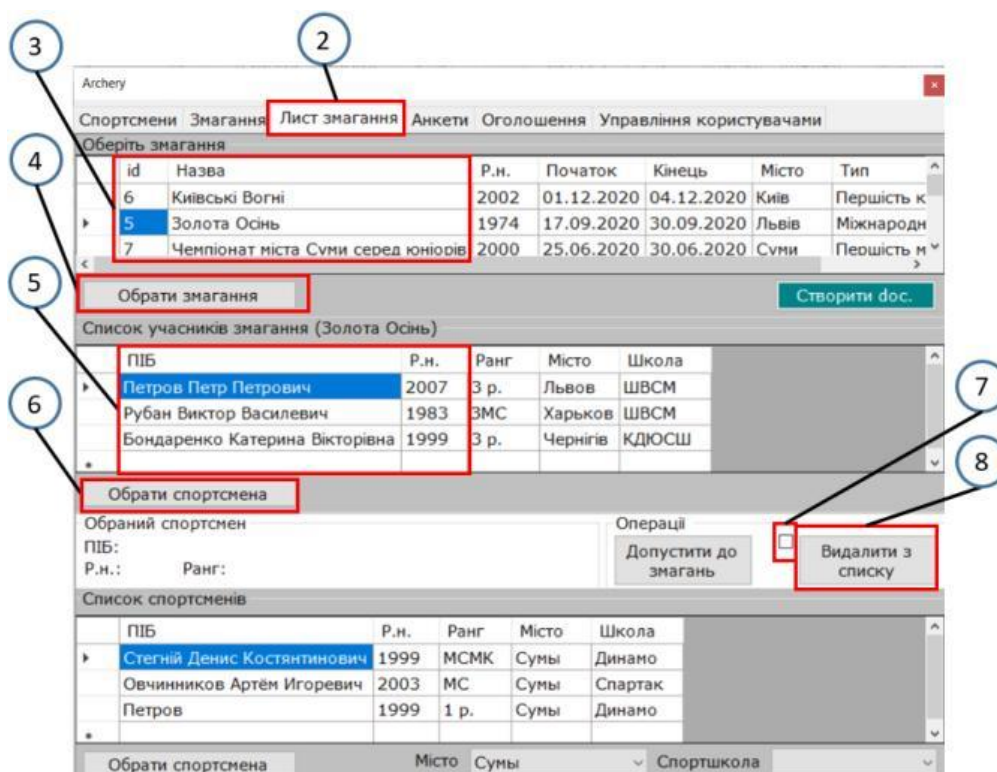


Рисунок Г.11 – Видалення спортсменів з списку учасників

Для додавання спортсмену до списку учасників за анкетною, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Анкети» (рис. Г.12);
- 3) Обрати змагання з списку змагань (рис. Г.12);
- 4) Натиснути кнопку «Обрати змагання» (рис. Г.12);
- 5) Обрати анкету з списку анкет (рис. Г.12);
- 6) Натиснути кнопку «Обрати анкету» (рис. Г.12);
- 7) Натиснути кнопку «Допустити до змагань» (рис. Г.12).

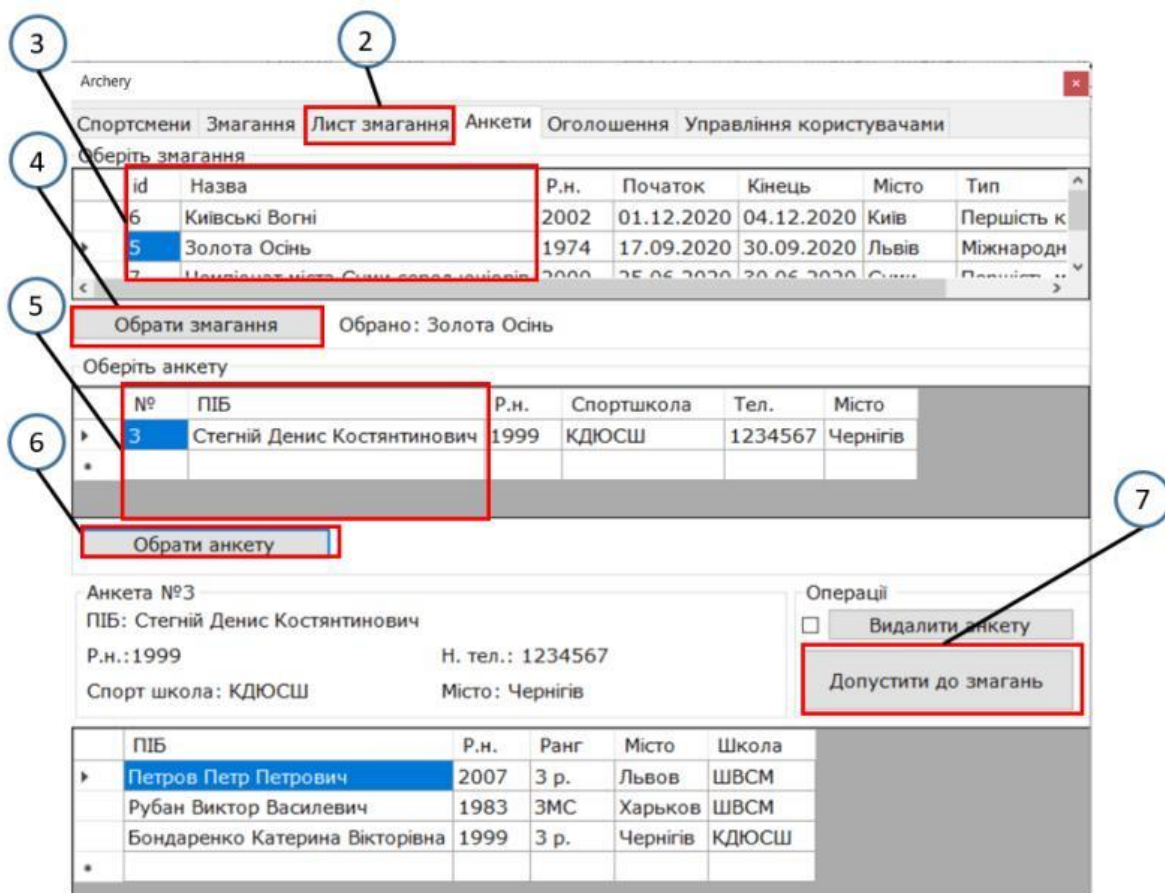


Рисунок Г.12 – Додавання спортсмена до списку учасників за анкетною

Для видалення анкети з бази даних, необхідно:

- 1) Авторизуватись (див. Авторизація в додатку);
- 2) Натиснути на вкладку «Анкети» (рис. Г.13);
- 3) Обрати змагання з списку змагань (рис. Г.13);
- 4) Натиснути кнопку «Обрати змагання» (рис. Г.13);
- 5) Обрати анкету з списку анкет (рис. Г.13);
- 6) Натиснути кнопку «Обрати анкету» (рис. Г.13);
- 7) Поставити відмітку (рис. Г.13);
- 8) Натиснути кнопку «Видалити анкету» (рис. Г.13).

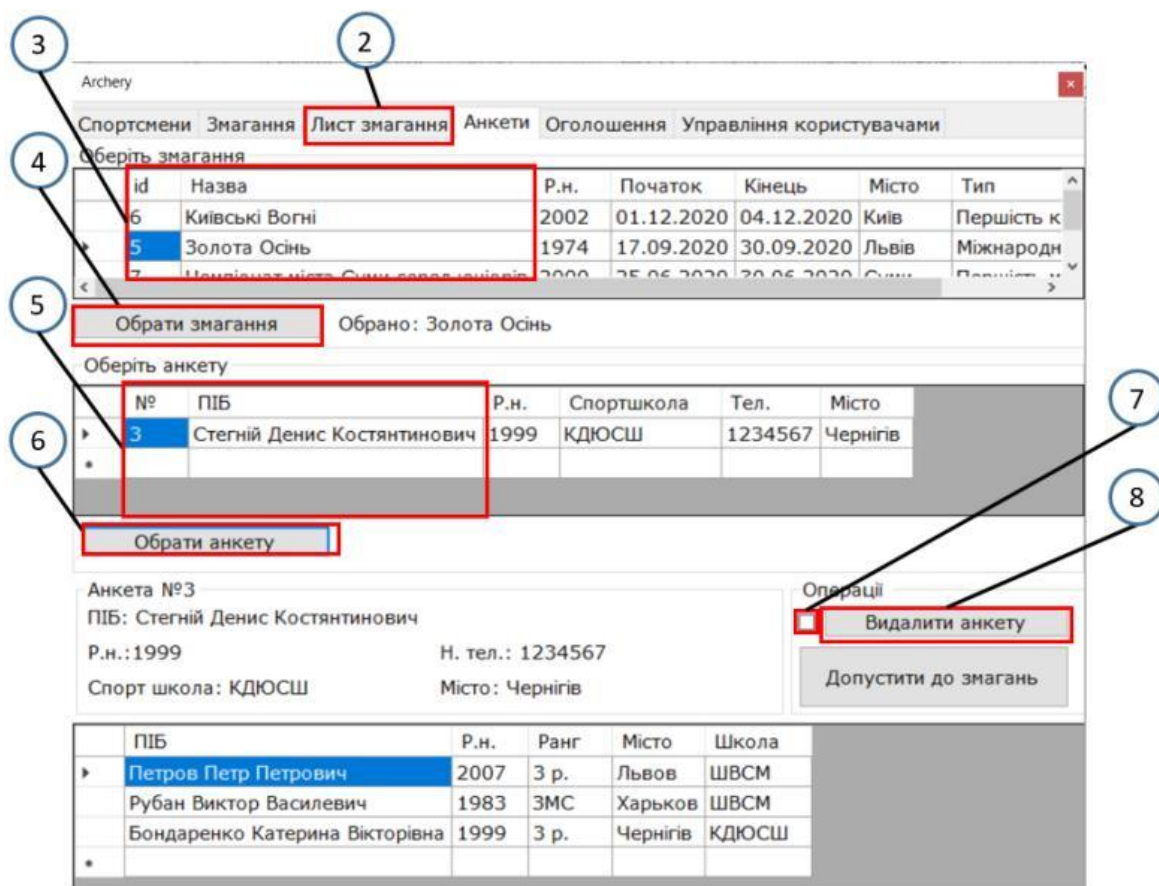


Рисунок Г.13 – Видалення анкети

Управління оголошеннями, що виводяться на головній сторінці Web-додатку

Для додавання нового оголошення, необхідно:

- 1) Авторизуватись (див. Авторизація додатку);
- 2) Натиснути на вкладку «Оголошення» (рис. Г.14);
- 3) Обрати дію «Додати» (рис. Г.14);
- 4) Ввести необхідну інформацію (рис. Г.14);
- 5) Натиснути кнопку «Додати» (рис. 1.14).

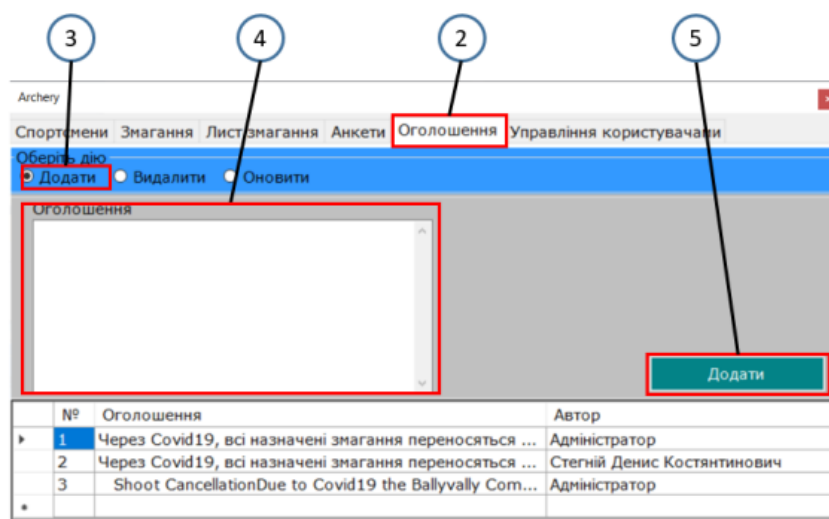


Рисунок Г.14 – Додавання нового оголошення

Для додавання нового оголошення, необхідно:

- 1) Авторизуватись (див. Авторизація додатку);
- 2) Натиснути на вкладку «Оголошення» (рис. Г.15);
- 3) Обрати дію «Видалити» (рис. Г.15);
- 4) Обрати оголошення з списку оголошень (рис. Г.15);
- 5) Натиснути кнопку «Обрати оголошення» (рис. Г.15);
- 6) Натиснути кнопку «Видалити» (рис. 1.15).

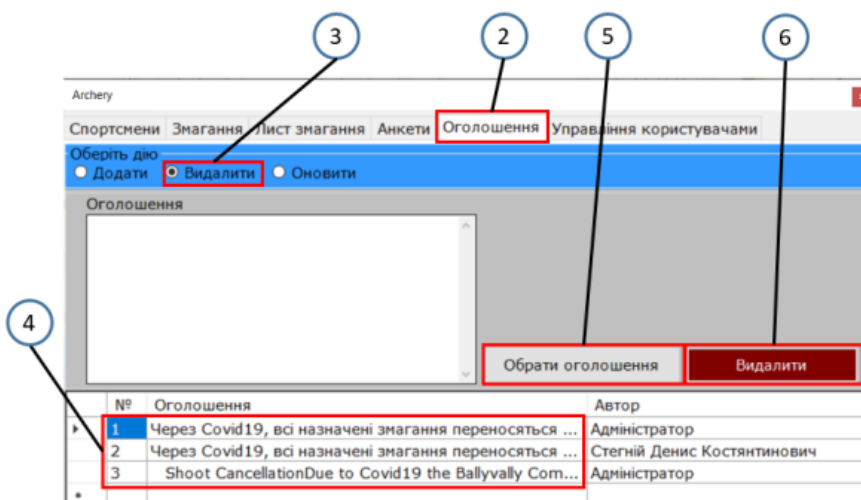


Рисунок Г.15 – Видалення оголошення

Для додавання нового оголошення, необхідно:

- 1) Авторизуватись (див. Авторизація додатку);
- 2) Натиснути на вкладку «Оголошення» (рис. Г.16);
- 3) Обрати дію «Оновити» (рис. Г.16);
- 4) Обрати оголошення з списку оголошень (рис. Г.16);
- 5) Натиснути кнопку «Обрати оголошення» (рис. Г.16);
- 6) Зробити необхідні правки (рис. Г.16);
- 7) Натиснути кнопку «Оновити» (рис. 1.16).

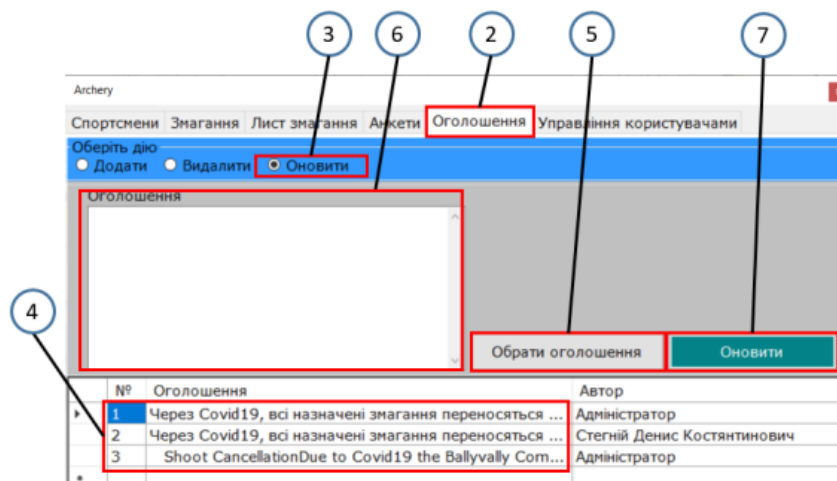


Рисунок Г.16 – Оновлення оголошення

Управління користувачами

Для додавання користувача за заяво, необхідно:

- 1) Авторизуватись (див. Авторизація додатку);
- 2) Натиснути на вкладку «Управління користувачами» (рис. Г.17);
- 3) Обрати заяву з списку заяв (рис. Г.17);
- 4) Натиснути кнопку «Обрати заяву» (рис. Г.17);
- 5) Зробити необхідні правки (рис. Г.17);
- 6) Натиснути кнопку «Додати користувача» (рис. 1.17).

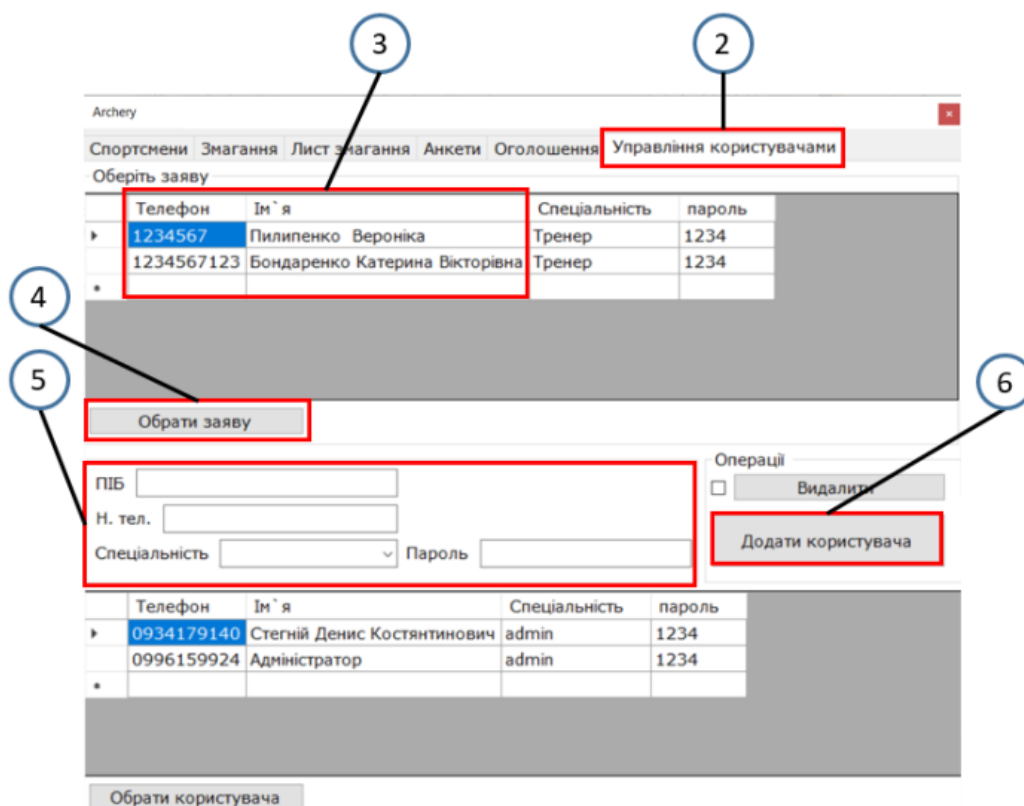


Рисунок Г.17 – Додавання користувача за заявою

Для видалення заявки, необхідно:

- 1) Авторизуватись (див. Авторизація додатку);
- 2) Натиснути на вкладку «Управління користувачами» (рис. Г.18);
- 3) Обрати заяву з списку заяв (рис. Г.18);
- 4) Натиснути кнопку «Обрати заяву» (рис. Г.18);
- 5) Поставити відмітку (рис. Г.18);
- 6) Натиснути кнопку «Видалити» (рис. 1.18).

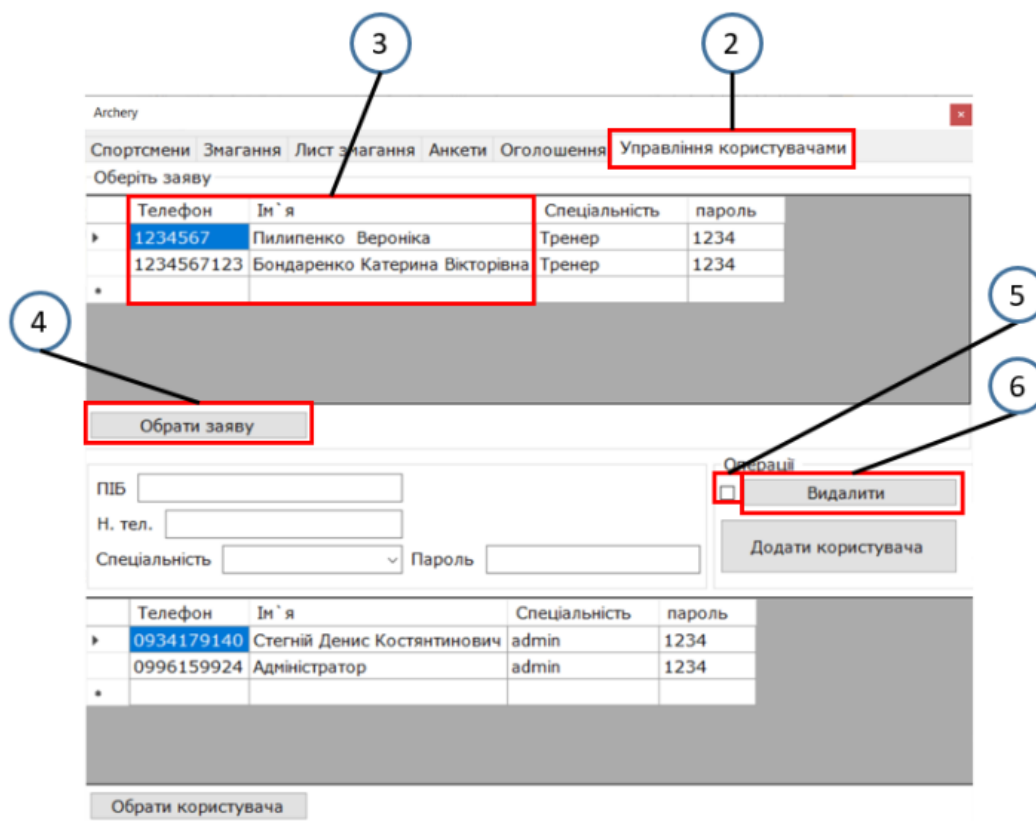


Рисунок Г.18 – Видалення заявки

Для видалення заявки, необхідно:

- 1) Авторизуватись (див. Авторизація додатку);
- 2) Натиснути на вкладку «Управління користувачами» (рис. Г.19);
- 3) Обрати користувача з списку користувачів (рис. Г.19);
- 4) Натиснути кнопку «Обрати користувача» (рис. Г.19);
- 5) Поставити відмітку (рис. Г.19);
- 6) Натиснути кнопку «Видалити» (рис. 1.19).

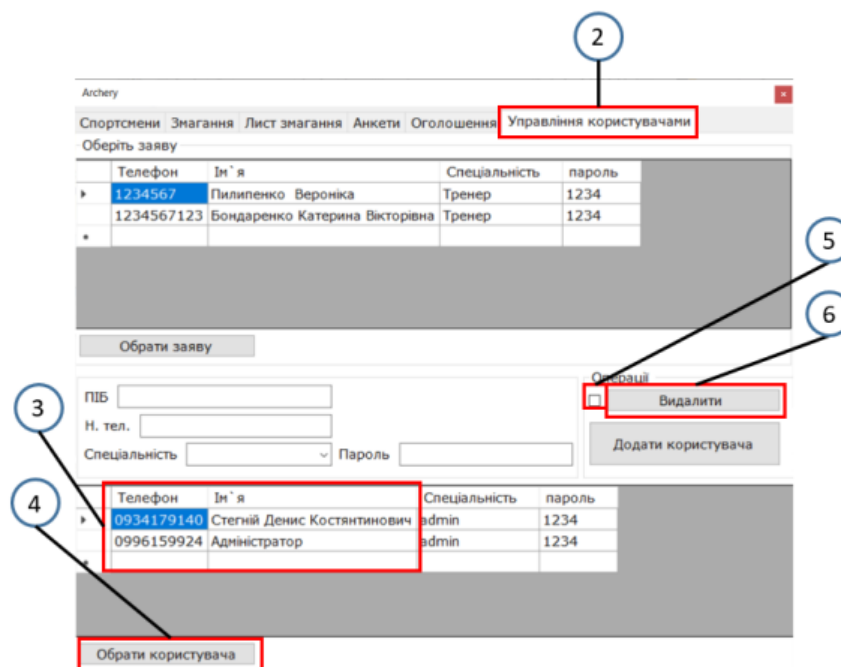


Рисунок Г.19 – Видалення користувача

Авторизація у Web-додатку

Необхідність в авторизації на сайті виникає при спробі перейти на сторінку «Спортсмени» або «Анкета» (рис. Г.20). Не авторизований користувач при переході на ці сторінки перенаправляється на сторінку авторизації (рис. Г.21).

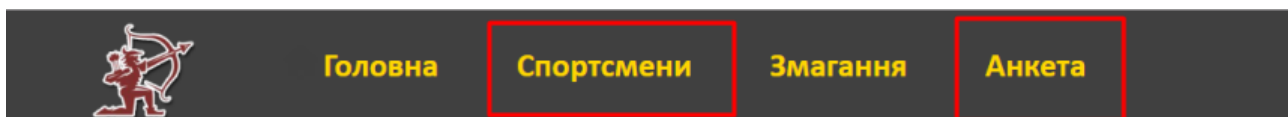


Рисунок Г.20 – Шапка сайту

Рисунок Г.21 – Сторінка авторизації

Щоб авторизуватись необхідно заповнити поля «Логін» та «Пароль», потім натиснути кнопку «Ввійти» (рис. Г.21).

Регістрація в Web-додатку

Щоб зареєструватись на сайті необхідно:

- 1) На сторінці авторизації натиснути на посилання «Регістрація» (рис. Г.21);
- 2) На сторінці реєстрації заповнити всі необхідні поля (рис. Г.22);
- 3) Натиснути кнопку «Відправити запит» (рис. Г.22).

Рисунок Г.22 – Сторінка реєстрації

Відправлення анкети для участі у змаганні

Щоб відправити анкету, необхідно:

- 1) Авторизуватись (див. Авторизація у Web-додатку);
- 2) Відкрити сторінку «Анкета» (рис. Г.23);
- 3) Заповнити всі необхідні поля (рис. Г.23);
- 4) Натиснути кнопку «Відправити запит» (рис. Г.23).

The image shows a web interface for submitting an application. At the top, a dark navigation bar contains a logo on the left and four menu items: 'Головна', 'Спортсмени', 'Змагання', and 'Анкета'. The 'Анкета' item is highlighted with a red box and labeled with a circled '2'. Below the navigation bar is a form titled 'АНКЕТА НА УЧАСТЬ'. The form contains several input fields: a dropdown menu for 'Змагання' (currently showing 'Чемпіонат міста Суми серед юніорів'), a text field for 'ПІБ', a text field for 'Рік народження' (with '1999' entered), a text field for 'Номер тел.', a text field for 'Спортивна школа', and a text field for 'Місто'. A red box highlights the entire form area, with a circled '3' pointing to it. At the bottom of the form is a button labeled 'ВІДПРАВИТИ ЗАПИТ', which is also highlighted with a red box and labeled with a circled '4'.

Рисунок Г.23 – Відправлення анкети