

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

**на тему:**

**«WEB-орієнтована CRM-система управління змаганнями клубів спортивно-бальних танців»**

**Завідувач  
випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Проценко О.Б.**

**Студента групи ІН – 71**

**Бєлим І.К.**

**СУМИ 2021**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедри Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ  
до випускної роботи**

Студента четвертого курсу, групи ІН-71 спеціальності “Комп'ютерні науки”  
денної форми навчання Белім Ілони Костянтинівни.

**Тема: “WEB-орієнтована CRM-система управління змаганнями клубів  
спортивно-бальних танців”**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ від \_\_\_\_\_ 20 \_\_\_\_\_ р.

**Зміст пояснювальної записки:** 1) аналітичний огляд методів реалізації веб-орієнтованих систем; 2) порівняння з існуючими аналогами; 3) формулювання та постановка завдання; 5) проектування та моделювання інформаційної системи; 6) тестування та аналіз отриманого інформаційної системи та результатів.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 2021 р.

Керівник випускної роботи \_\_\_\_\_ Проценко О.Б.

Завдання прийняв до виконання \_\_\_\_\_ Белім І.К.

## РЕФЕРАТ

**Записка:** 71 стор., 46 рис., 2 табл., 2 додатки, 17 джерел.

**Об'єкт дослідження** – CRM-система управління змаганнями клубів спортивно-бальних танців.

**Мета роботи** – розробка web-орієнтованої системи, яка дає можливість доступу з будь-якої точки світу та з будь-якого гаджету з виходом до Інтернету зареєструватися на змагання зі спортивно-бальних танців, переглянути та занести нові результати минулих змагань та подати заявку на суддівство.

**Методи дослідження** – емпіричні, комплексні та теоретичні методи.

**Результати** – розроблено web-орієнтовану CRM систему для реєстрації на спортивно бальних танцях, також реалізовано можливість створення «Танцювальної книги» для внесення результатів минулих змагань та можливість подання заявки на суддівство на основі клієнт-серверних технологій з використанням новітніх технологій, таких як php, JavaScript, HTML, CSS, jQuery та MySQL.

CRM-СИСТЕМА, WEB-ДОДАТОК, СПОРТИВНО-БАЛЬНІ ТАНЦІ,  
ТАНЦЮВАЛЬНА КНИГА, РЕЄСТРАЦІЯ, ЗМАГАННЯ, СУДДІВСТВО.

# ЗМІСТ

ВСТУП.....	5
1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
1.1 Огляд останніх досліджень та публікацій.....	6
1.2 Огляд подібних рішень .....	7
1.3 Постановка задачі .....	9
2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ.....	11
2.1 Обрані технології для реалізації бекенду.....	13
2.2 Обрані технології для реалізації фронтенду .....	13
2.3 Діаграмне та схематичне представлення .....	14
3. ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ.....	16
3.1 Моделювання .....	16
3.2 Проектування моделі бази даних .....	18
3.3 Програмна реалізація.....	23
3.4 Програмна реалізація клієнтської частини .....	34
ВИСНОВКИ.....	55
СПИСОК ЛІТЕРАТУРИ.....	56
ДОДАТКИ.....	58

## ВСТУП

В сучасному світі жодна людина не зможе уявити свого життя без використання інтернету та новітніх технологій. Повсякденні та рутинні задачі переходять до автоматизованого вигляду, щоб можна було зробити в два кліки.

Кожного дня люди стикаються з великою кількістю задач, які необхідно виконати. Але мабуть не залишилось жодної людини, яка б не володіла телефоном й змогла б уявити себе без нього. Використання сучасних гаджетів нерéalно полегшує життя кожного, завдяки йому можемо вільно спілкуватися з людьми з різних куточків світу, обмінюватися інформацією та зберігати важливі моменти у фотокартках та відео, які можна через багато років буде переглядати.

Кожен спортсмен, який активно приймає участь в змаганнях, для того щоб перевірити свою професійність та продемонструвати свою майстерність, витрачає велику кількість часу на підготовку до вже відомого заходу. Необхідно зібрати всі документи та тільки під час прибуття на місце проведення заходу матимете можливість повністю зареєструватися на змаганнях. З цієї причини, розробка web-орієнтованої системи для реєстрації на змаганнях є дуже актуальною, тому що за допомогою Інтернету та будь-якого новітнього гаджету, чи то телефону, планшету або ж комп'ютеру, у з'являється можливість в лічені хвилини пройти реєстрацію.

Використання web-додатку полегшить життя як учасників змагання, так і організаторів, об'єднає все в одній системі. Організатори зможуть створювати змагання, а танцівники зможуть обирати та реєструватися на відповідні змагання, а також журі зможуть надсилати свої кандидатури. Для учасників буде зберігатися інформація щодо минулих змагань та їх результатів.

# 1. ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Огляд останніх досліджень та публікацій

Спортивно-бальні танці ніколи не покидали арени слави та з кожним роком набувають все більшої популярності. Звичайно, що кількість клубів на території України та за її межами зростає, щорічно склад цих клубів поповнюється перспективними танцівниками, які мають неймовірний запал та хочуть продемонструвати все, на що спроможні. А саме, мають бажання взяти участь у змаганнях, але для цього необхідно пройти довгий шлях реєстрації, яку можна зробити тільки очно по прибуттю на саме місце проведення заходу.

У кожного професійного танцівника є «Танцювальна книга» - паспорт, в якому зберігається інформація про всі змагання та результати. До цієї книги заносяться дані по прибуттю про саме змагання, організаторів та класи, в яких будуть брати участь, а по закінченню – результати та кількість набраних балів для переходу в наступний танцювальний клас.

В еру стрімкого розвитку інформаційних технологій, всі книги, документи, паперові матеріали перетворюються на електронні ресурси, великою перевагою є створення системи, де будуть зберігатися «Танцювальна книга» танцівника, буде можливість зареєструватися на змагання заздалегідь і всі результати будуть зберігатися в одному місці.

Отже, проект, в результаті якого буде створено програмний продукт – WEB-орієнтована CRM-система управління змаганнями клубів спортивно-бальних танців.

Говорячи про різні аспекти, а саме про 5 найголовніших, то дана система зробить чималий внесок в кожен із них, а саме:

- Щодо технічного аспекту система дає можливість доступу у будь-який час та з будь-якого куточку світу, використовуючи будь-яку платформу.
- Щодо соціально аспекту даний програмний продукт збільшить кількість учасників змагань, спростить процес реєстрації та об'єднає все в одній системі.

- Цінність – програмний продукт буде цінною знахідкою для танцівників, які хочуть прийняти участь у змаганнях та клубів-організаторів цих змагань, а також для суддів, які будуть оцінювати майстерність учасників.
- Щодо комерційного аспекту дана система об'єднає дані та можливість одразу учасникам змагань та організаторам бути в курсі всіх актуальних подій.
- Проміжок затребуваності – залежить в першу чергу від кількості змагань, популярності спортивно-бальних танців. Не менш важливу роль будуть відігравати використані в системі технології, новітні чи вже застарілі та алгоритми актуальні чи вже доживають свої останні дні під час написання додатку, доки їх актуальність буде високою та буде підтримуватися різними операційними системами, доти буде затребуваний на ринку інформаційних технологій.

## 1.2 Огляд подібних рішень

Після проведення ретельного дослідження, було виявлено декілька продуктів-аналогів до майбутньої системи, про які на сьогоднішній день майже ніхто не знає – web-сайти [audsf.com.ua](http://audsf.com.ua) (рис.1.1) та [flymark.com.ua](http://flymark.com.ua) (рис.1.2).

Web-сайт [audsf.com.ua](http://audsf.com.ua) – це об'єднання Всеукраїнської Федерації Танцювального Спорту, де зібрані минулі та майбутні змагання на території України за певний проміжок часу. Основний функціонал – це інформація для головних суддів та різні новини в спорті, список зі змаганнями, на які можна зареєструватися будь-якому користувачу. Дизайн досить застарілий та не дуже зручний зі сторони UX. Також даний продукт не дає можливість переглянути у власному кабінеті користувача-танцівника його результати за пройдени змагання, тобто немає «Танцювальної книги». Також на сайті наявна влаштована реклама.

Web-сайт [flymark.com.ua](http://flymark.com.ua) – це також сервіс, який дозволяє будь-яким користувачам, але не суддям, зареєструватися на змаганнях зі спортивно-бальних танців. Він має гарний та зручний дизайн, але деякі блоки можуть бути невірно розташовані на сторінці, можна переглянути майбутні змагання, але на рівні з цим на

сайті присутня велика кількість іншого непотрібного функціоналу, який розсіює увагу користувачів. Влаштована реклама також з’являється час від часу.



Рисунок 1.1 – Головна сторінка сайту audsf.com.ua

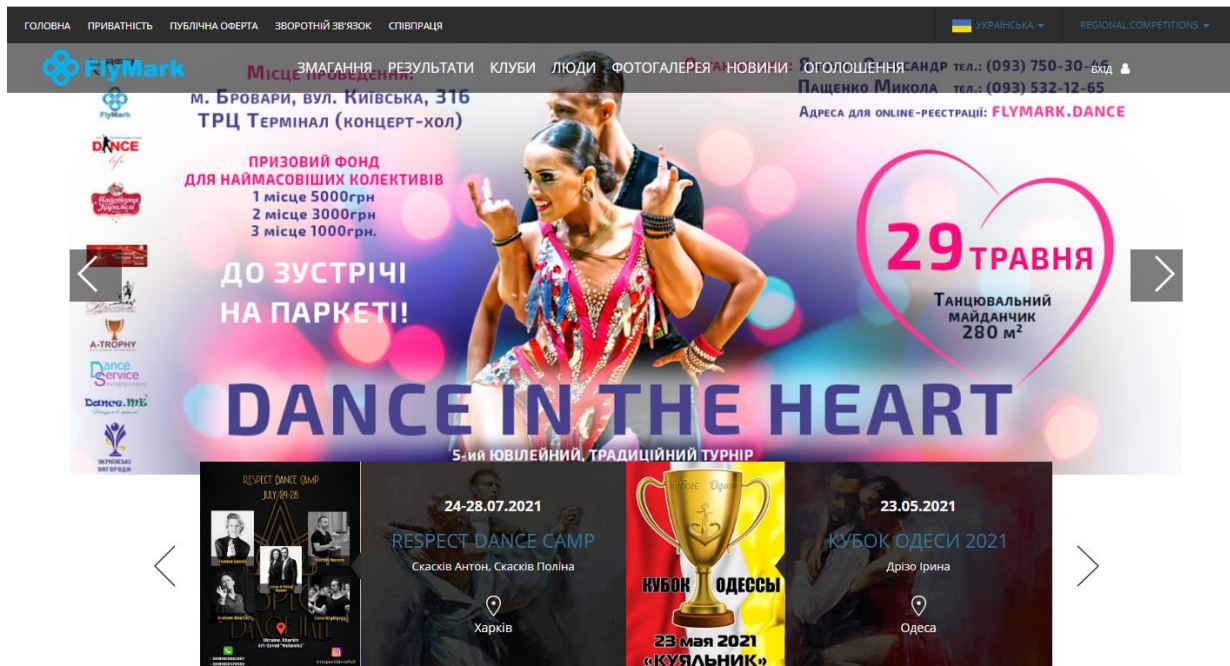


Рисунок 1.2 – Головна сторінка сайту flymark.com.ua



Для більш наглядного та зручного розуміння всіх переваг та недоліків всю інформацію було зібрано в порівняльну характеристику додатків-аналогів та майбутньої системи. Дана інформація подана в порівняльній таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця основних характеристик додатків-аналогів

<b>Характеристики</b>	<b>audsf.com.ua</b>	<b>flymark.com.ua</b>	<b>Створювана система</b>
Сучасний та зручний UX/UI	-	+	+
Можливість реєстрації для суддів	-	-	+
Наявність власних результатів танцівника	-	-	+
Наявність влаштованої реклами	+	+	-
Швидке завантаження сторінок та робота сервісу	-	+	+
Наявність «Танцювальної книги»	-	-	+

### **1.3 Постановка задачі**

Головною метою написання роботи є розробка WEB-орієнтованої CRM-система управління змаганнями клубів спортивно-бальних танців. Основними функціями інформаційної системи є:

- реєстрація на змаганнях з будь-якої точки світу до моменту прибуття на самі змагання;
- можливість перегляду власних результатів змагання у «Танцювальній книзі»;
- можливість доступу до системи з будь-якого гаджету, незважаючи на тип ОС;

- немає необхідності встановлювати окремий додаток, для того щоб скористуватися необхідним функціоналом.

Для створення та розробки даного завдання виокремлено такі необхідні частини дослідження, а саме:

- проведення аналізу додатків-аналогів;
- створення дизайну та майбутнього інтерфейсу;
- проектування та розробка додатку;
- розробка модулів реєстрації та входу користувачів;
- розробка модуля редагування власної інформації;
- розробка модуля реєстрації на змагання;
- розробка модуля зі всіма змаганнями;
- розробка модуля «Танцювальної книги»;
- тестування програмного продукту;
- написання документації.

## 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

Перед початком реалізації інформаційної системи було проаналізовано та досліджено велику кількість методів та відібрано найбільш доцільні та релеванті для даного виду роботи.

Так як, розроблювана система є веб орієнтованою, то це означає, що дана система базується на архітектурі «клієнт-серверних» технологій.

Це говорить, що всі запити, які надходять від клієнта, а в свою чергу від конкретного користувача, оброблюються на стороні сервера, а вже далі він надсилає відповідь, що інтерпретується браузером та клієнт відображає результат у зрозумілому для звичайних користувачів вигляді. Весь цей процес неможливий без такої геніальної мережі як Інтернет. На рисунку 2.1 схематично відображено це спілкування між клієнтом та сервером.

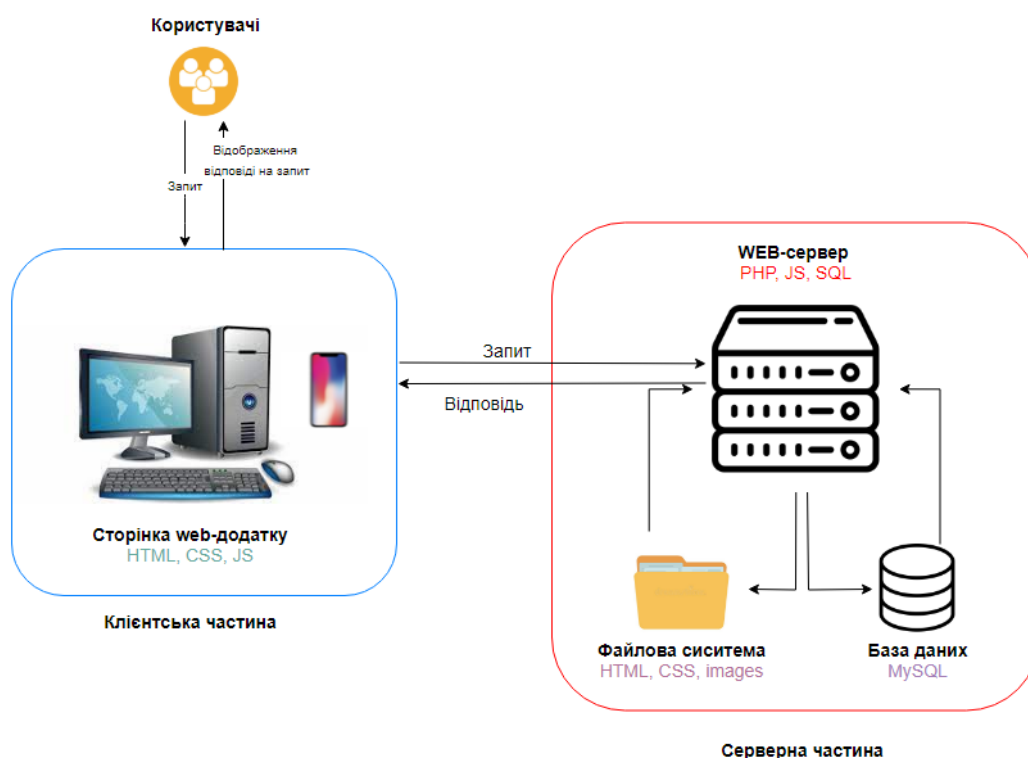


Рисунок 2.1 – Схематична архітектура системи

Так як розроблювана система – це web-додаток, який орієнтований на користувачів, а саме на роботу з ними, тому дуже важливим є скрупульозне опрацювання кожного запиту від користувача та коректне відображення відповіді на нього. У випадках коли у відповідь на запит немає інформації або процес закінчився помилкою, яку необхідно проінформувати користувача зрозумілим для нього способом – сторінкою з помилкою.

Відповідно до схематичного зображення видно, що дана система складається з двох великих частин – сервера та клієнта, які водночас окремими елементами, але взаємозалежними. А вже всередині цих двох складових, велика кількість менших деталей, які взаємодіють між собою.

Клієнт або клієнтська частина виконує такі функції, як правильне відображення веб-сторінок за допомогою браузерів комп'ютерів чи мобільних девайсів, тобто адаптивне відображення незалежне від типу пристрою, за допомогою таких технологій, як HTML, CSS, JS [9]. Це безпосередньо те, що дає відповіді на конкретні запити користувача.

Сервер або серверна частина складається з великої кількості компонентів, її функціонал дуже широкий, але головні функції, які виконує веб-сервер – оброблює запити, які надходять від клієнта, за допомогою PHP, JS, SQL [4, 6, 17]. До серверної частини також відноситься й файлова система, її задачею є робота з файлами каскадних стилів, що додають сторінці стилізований та структурований вигляд.

Й не менш важливою складовою серверної частини є база даних, в якій зберігається інформація стосовно користувачів, змагань, і всі введені дані й не тільки, бо ще кожному елементу який заноситься до неї присвоюється унікальний номер, який далі ідифікує ці дані. Виконання такого процесу відбувається наступним чином: аналіз запиту, який надійшов від клієнта, перевірка, якщо вона наявна, а далі вже надсилання відповіді клієнту, який інтерпритує у вигляді зрозумілому користувачу.

## 2.1 Обрані технології для реалізації бекенду

У якості серверу було обрано Denwer – найбільш відповідний для реалізації даної інформаційної системи в реальних умовах. В першу чергу це безкоштовний сервіс, який може встановити кожний. По друге, це пакет, який містить всередині спеціальні програми, за допомогою яких на комп'ютер встановлюються локальний сервер. Отже, в результаті один фізичний комп'ютер виступає в якості серверу та одразу ж і в якості клієнту.

Відповідно до того, що Denwer має всередині себе вже влаштовану адмін-панель PhpMyAdmin для роботи з базами даних MySQL [18], тому звичайно було обрано її, бо вона є вільна у доступі та зручна для використання реляційна система керування базами даних.

PhpMyAdmin дає можливість з нуля створити БД та далі за допомогою написаних движків підключити її до розробленого сервісу.

Для написання запитів до БД обрано SQL (Structured Query Language) – мову програмування, яка має декларативний вигляд та застосовується для управління даними, тобто додавання нових, редагування занесених та перегляд існуючих в системі даних [4, 6].

І звичайно для написання всієї серверної частини було обрано PHP (Hypertext Preprocessor) – скриптову мову програмування [15, 17, 18].

## 2.2 Обрані технології для реалізації фронтенду

Для реалізації клієнтської частини було обрано чималу кількість різних сервісів та технологій, вільних у доступі, таких як:

- Google Fonts – бібліотека від компанії Google із вільними у доступі шрифтами, які було використано для гармонійного відображення текстові інформації на сторінках [3];
- Font Awesome – частково вільний у доступі сервіс для додавання графічних іконок на сторінки додатку, для того щоб звичайному користувачу було простіше навігуватися на сторінках та звичайно для більш естетичного вигляду;

- Bootstrap – вільний у доступі фреймворк, що дає можливість використовувати готові HTML- та CSS-шаблони для швидкого створення структури сайту та оформлення стилем, звичайного було використано для того, щоб переглянути вже готові UX/UI дизайни, щоб відповідно до них розмістити зручно та красиво блоки на сторінках [8].

Щодо головних технологіями, а саме релевантних для розробки клієнтської частини даної системи стали приведені нижче технології:

- HTML (HyperText Markup Language) – мова тегів, гіпертекстової розмітки, використовується для створення структури сторінки, розбиття її на блоки та послідовність їх розміщення [1];

- CSS (Cascading Style Sheets) – мова, за допомогою якої HTML-сторінки отримують стильове наповнення, змінюються кольори та розмір та розміщення блоків в залежності від обраного гаджету, бо для даної системи важливим аспектом є адаптивність до різних розширень [2, 9];

- JS (JavaScript) – мова з широким функціоналом, яка водночас є об'єктно-орієнтованою, прототипною та динамічною, застосовується в даному додатку для додавання на сторінку інтерактивності та анімації [5, 12, 14];

- jQuery – бібліотека JavaScript, яка є вільною у доступі та дозволяє швидше та зручніше створювати анімацію та інтерактивність на сторінці [11, 13].

У поєднанні всіх вище перерахованих технологій в результаті виходить програмний продукт, який взаємодіє між користувачами, за допомогою комп'ютера або інших гаджетів та сервером, який надає коректні відповіді на вхідні запити.

### **2.3 Діаграмне та схематичне представлення**

Для структурного відображення системи, її основних функцій, взаємодій між модулями було обрано:

- IDEF – техніку моделювання, за допомогою поєднання графічних та структурних правил, призначені для опису процесів системи. IDEF0 – формалізує та

відображає опис бізнес-процесів системи. IDEF1 – відображає інформаційні потоки всередині системи для аналізу їх структури та взаємозв'язку;

- Також для демонстрації можливих способів використання системи було обрано UML-діаграму Варіантів-Використання – відображає відношення в системі між акторами (роль, що виконує сутність) та прецедентами (варіантами використання);

- ER-діаграму – відображення відношення сутностей між собою, в тому числі і кількісно, у якій сутності скільки зв'язків з іншою;

- Діаграма взаємодії модулів – структурне відображення залежності модулів системи між собою та опис головних функціональних можливостей системи.

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ

#### 3.1 Моделювання

Першою задачею було створити IDEF діаграми двох рівнів, щоб розуміти, які основні процеси, вхідні та вихідні дані чи потоки будуть у створюваній системі.

Рисунок 3.1 є формалізує описану систему та є відображенням контекстної діаграми першого рівня.

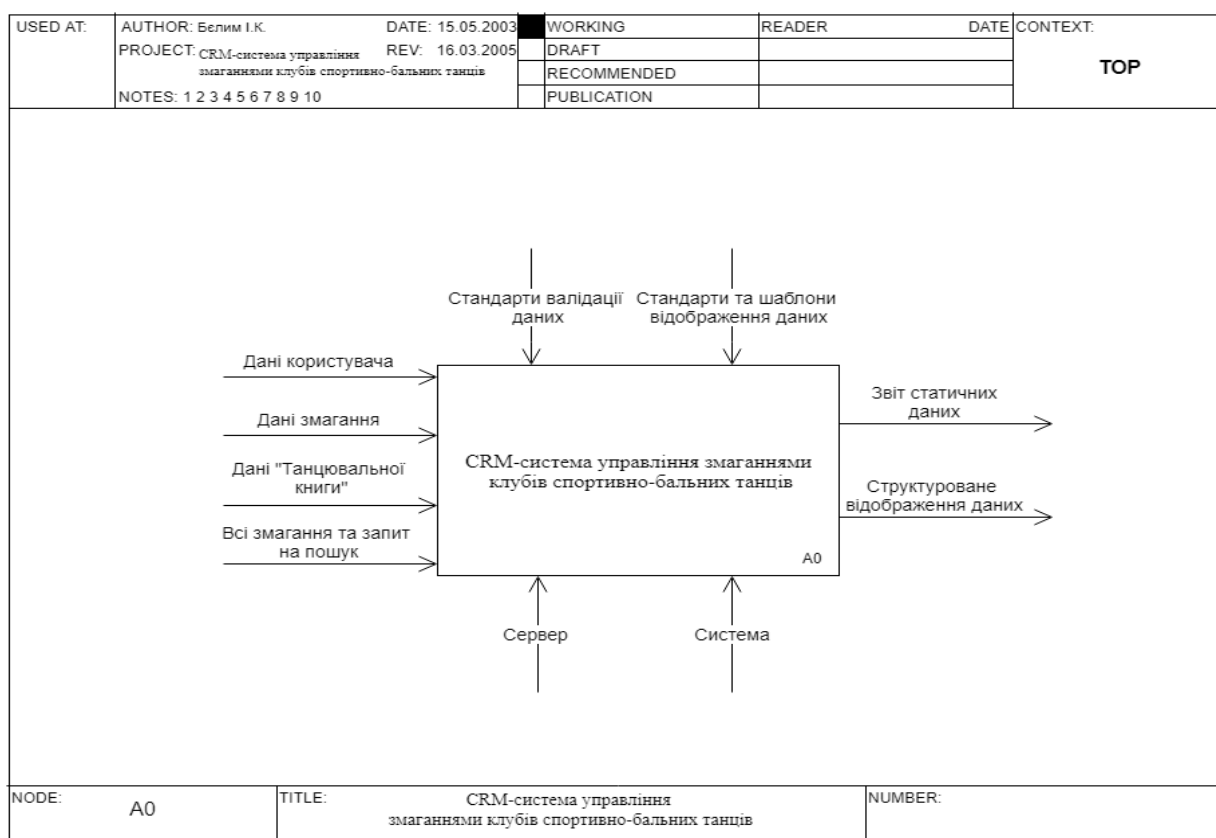


Рисунок 3.1 – IDEF0, контекстна діаграма першого рівня



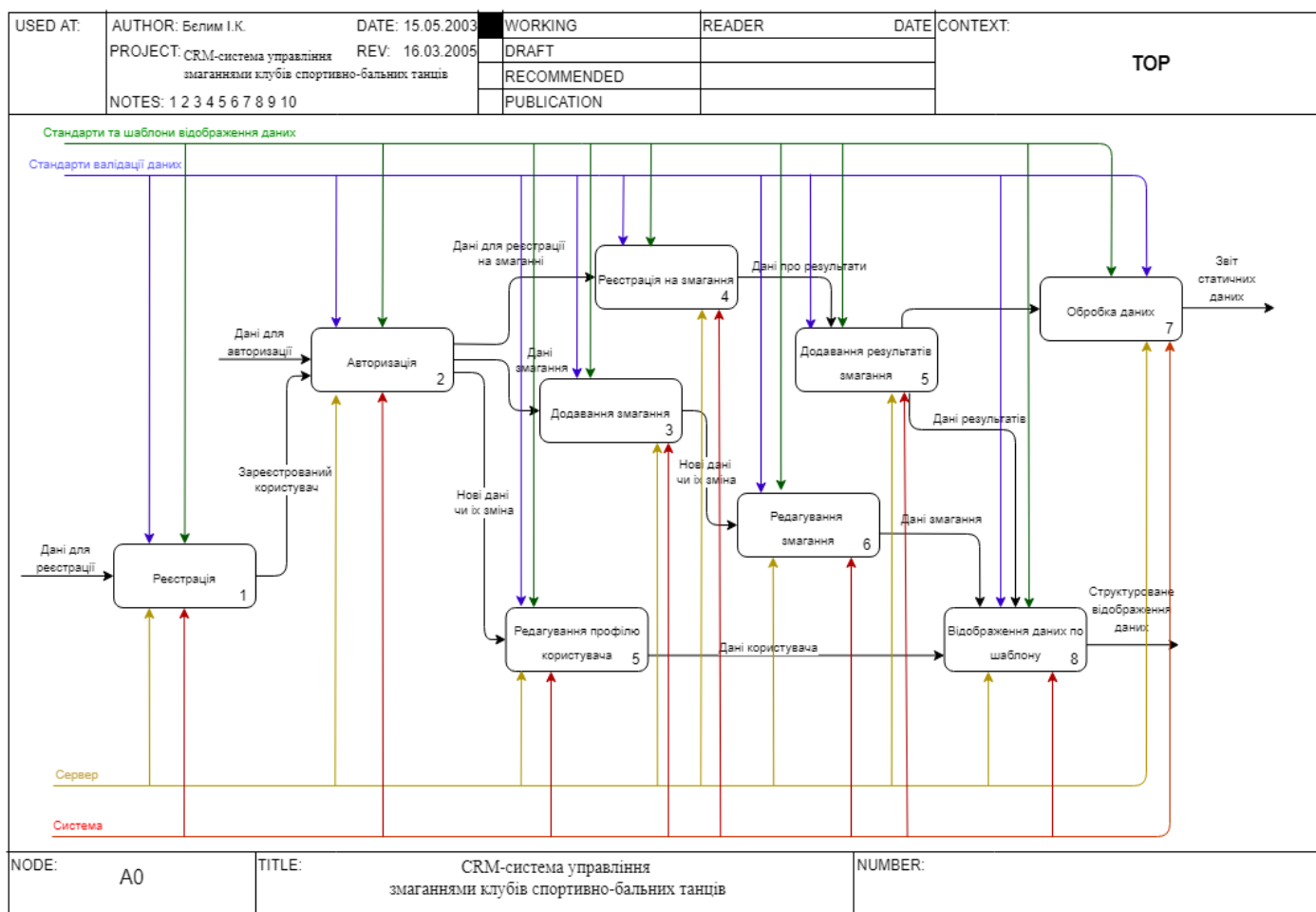


Рисунок 3.2 – IDEF1, декомповована контекстна діаграма першого рівня

Рисунок 3.2 відображає діаграму декомпозиції першого рівня для основного процесу системи – «Управління змаганнями клубів спортивно-бальних танців».

Також на рисунку 3.3 зображена UML-діаграма прецедентів або по іншому діаграма Варіантів-Використання.

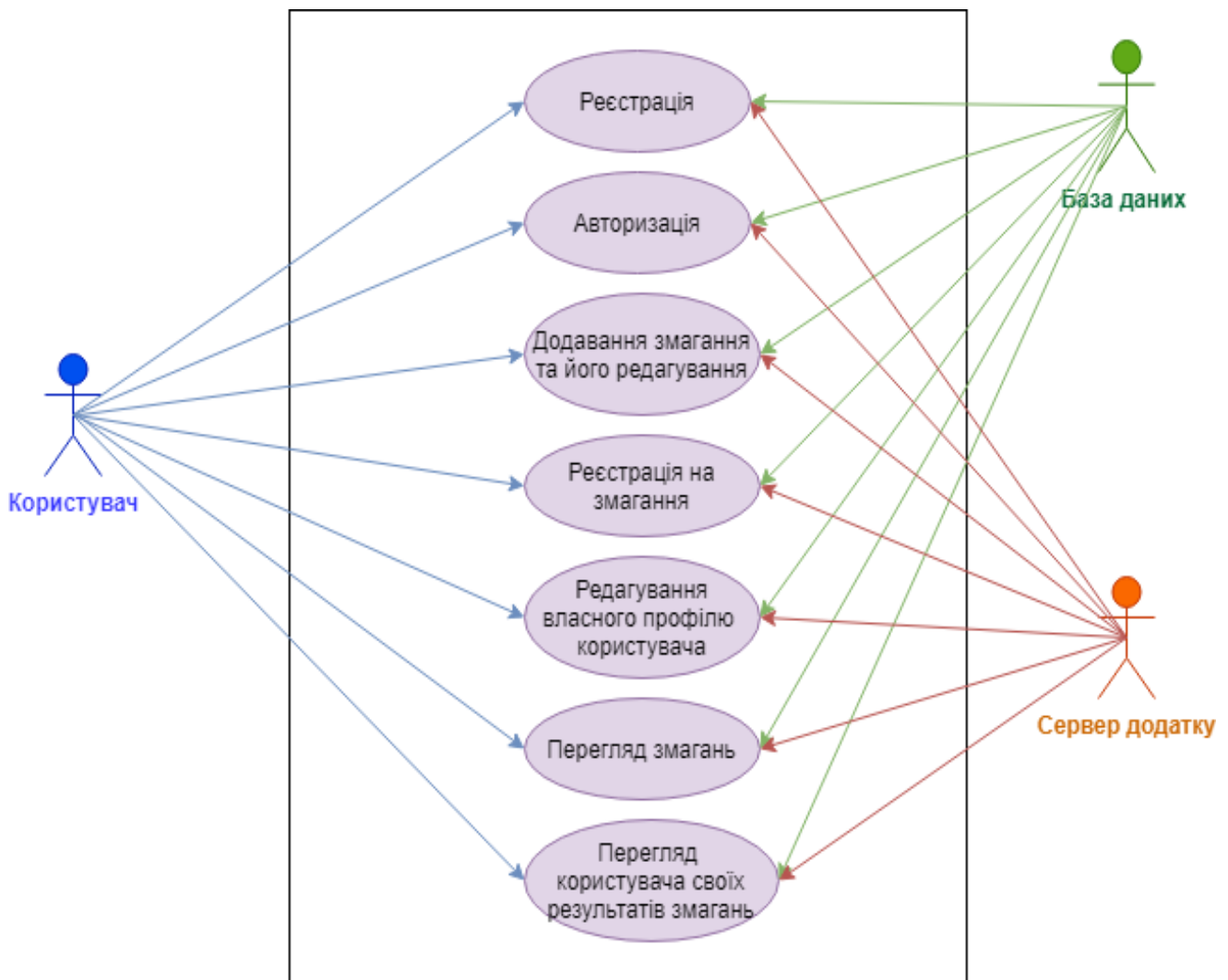


Рисунок 3.3 – Use-case діаграма

### 3.2 Проектування моделі бази даних

Під час проектування бази даних для інформаційної системи головним завданням є визначення сутностей та атрибутів. Далі всі визначені перелічені параметри збирають та формують в ER-діаграму для наглядного відображення зв'язків між сутностями – реальними або уявними об'єктами.

Було виділено основні сутності системи для коректного функціонування web-орієнтованої системи:

- Звичайні користувачі;
- Танцівники;
- Судді;

- Танцювальні книги танцівників;
- Записи танцювальної книги;
- Клуби;
- Змагання;
- Запит на участь в змаганні;
- Запит у суддівстві змагання.

Згідно даних сутностей було сформовано таблицю 3.1, де представлено опис кожної сутності та відповідних до неї атрибутів.

Таблиця 3.1 – Опис сутностей БД та їх атрибутів

Сутність	Опис	Атрибут	Опис
users	Персональна інформація звичайного користувача	user_id	Ідентифікатор зареєстрованого користувача
		name	Ім'я користувача
		surname	Прізвище користувача
		gender	Стать користувача
		email	Електронна адреса користувача
		birthday	Дата народження користувача
		role	Роль користувача(журі, танцівник)
		password	Пароль для входу в систему
		location	Місце проживання
dancers	Персональна інформація користувачів, які є танцівниками	dancer_id	Ідентифікатор танцівника
		user_id	Ідентифікатор користувача системи
		club_id	Ідентифікатор танцювального клубу танцівника
		d_class	Танцювальний клас танцівника
		program	Танцювальна програма танцівника
judgers	Інформація про журі та їх досвід	judge_id	Ідентифікатор судді
		user_id	Ідентифікатор користувача системи
		description	Опис досвіду журі
dancer-book	Інформація про танцювальну книгу танцівника	book_id	Ідентифікатор танцювальної книги
		dancer_id	Ідентифікатор танцівника
		b_date	Дата створення танцювальної книги
notes	Список результатів минулих змагань танцівника	note_id	Ідентифікатор запису (результату змагання)
		book_id	Ідентифікатор танцювальної книги
		competition_name	Назва змагання, де брав участь

Таблиця 3.1 – Продовження таблиці опису сутностей БД та їх атрибутів

Сутність	Опис	Атрибут	Опис
		n_class	Клас, в якому змагався
		pairs_quantity	Кількість пар на змаганні
		program	Програма, в якій змагався
		c_date	Дата змагання
		place	Місце, яке посіли
clubs	Інформація про танцювальні клуби	club_id	Ідентифікатор клубу
		club_name	Назва клубу
		club_address	Повна адреса клубу
		owner	Власник-створювач даного клубу
competitions	Інформація про майбутні змагання	competition_id	Ідентифікатор змагання
		user_id	Ідентифікатор користувача-адміна, який заніс дані про змагання
		comp_name	Назва змагання
		comp_address	Адреса проведення змагання
		start_time	Початок змагання
		comp_image	Фотографія до змагання
		organizer	Клуб-організатор
d_request	Заявка танцівника на реєстрацію в змаганнях	d_request_id	Ідентифікатор заявки від танцівника
		dancer_id	Ідентифікатор танцівника
		competition_id	Ідентифікатор змагання
j_request	Заявка судді на те, щоб судити змагання	d_request_id	Ідентифікатор заявки від судді
		dancer_id	Ідентифікатор судді
		competition_id	Ідентифікатор змагання

Також згідно до вище описаних сутностей та їх атрибутів було реалізовано ER-діаграму (відношення сутностей між собою, в тому числі і кількісно, у якій сутності скільки зв'язків з іншою). Її можна переглянути на рисунку 3.4.

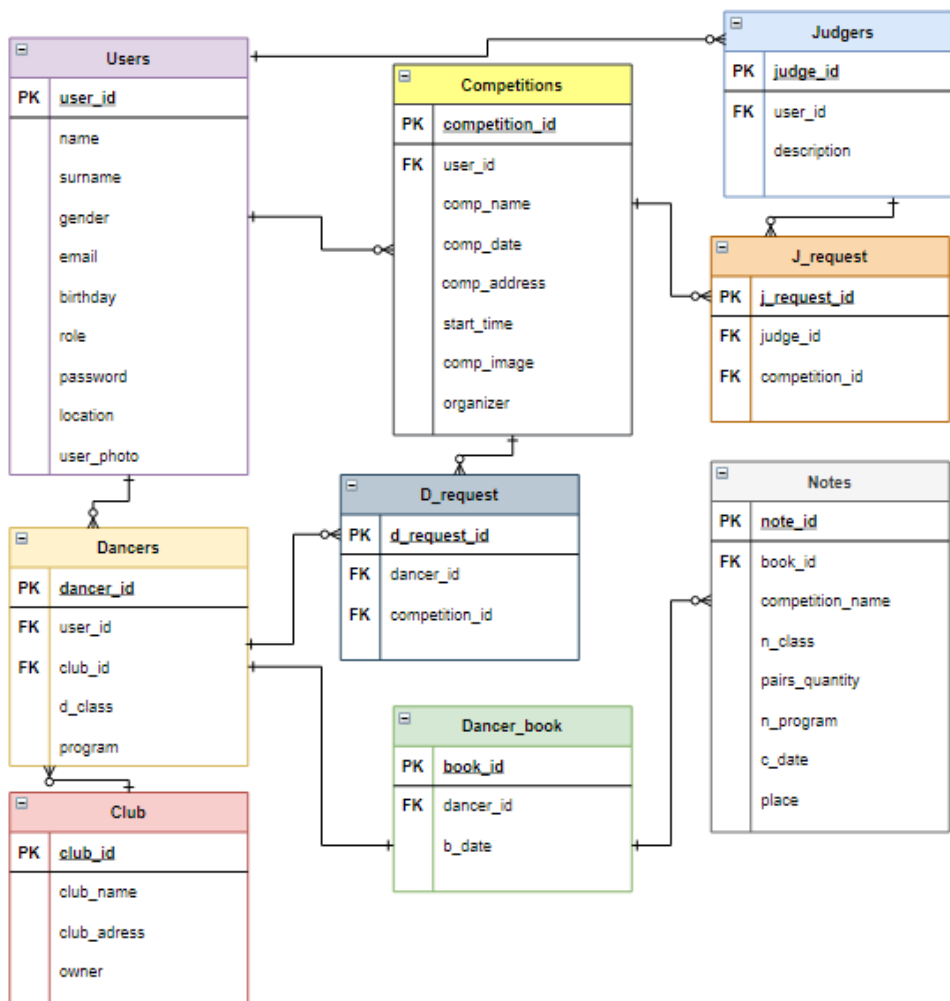


Рисунок 3.4 – ER діаграма

Для сутностей «users» та «dancers», «users» та «judgers» обрано зв'язок один-до-багатьох, який говорить, що користувачі можуть бути танцівниками або ж суддями, але конкретний танцівник чи суддя має одного конкретного користувача. Зв'язок один-до-багатьох також відображає відношення між «users» та «competitions», один користувач (адмін) може мати/створити багато змагань, але у конкретного змагання є тільки один користувач-творець.

Один-до-багатьох зв'язок у відношенні «dancers» та «clubs» показує, що у клуба може бути багато танцівників, але у танцівника тільки один конкретний клуб. Зв'язок один-до-одного гарно описує відношення між «dancers» та «dancers\_book», бо одного танцівника є одна танцювальна книга. А в свою чергу між «dancers\_book» та «notes» зв'язок один-до-багатьох, який говорить, що в танцювальній книзі може бути

багато записів, але у багатьох записів одного користувача є тільки одна танцювальна книга. Відношення між «dancers» та «d\_request», «judgers» та «j\_request» описане зв'язком один-до-багатьох, що означає, що танцівник може один раз зареєструватися на конкретне змагання так само як і суддя може один раз відправити одну заявку на суддівство в конкретному змаганні. Подібні зв'язки між сутностями «competitions» та «d\_request», «competitions» та «j\_request», говорять що одне на змагання може бути відправлено багато заявок на суддівство та зареєстрована велика кількість танцівників.

На рисунку 3.5 продемонстрована фізична реалізація БД в MySQL, також у вигляді ER-діаграми з позначеними зв'язками між сутностями та вказаними типами даних для кожного з атрибутів [6]. А на рисунку 3.6 відображено структуровані залежності модулів між собою та основні функції системи.

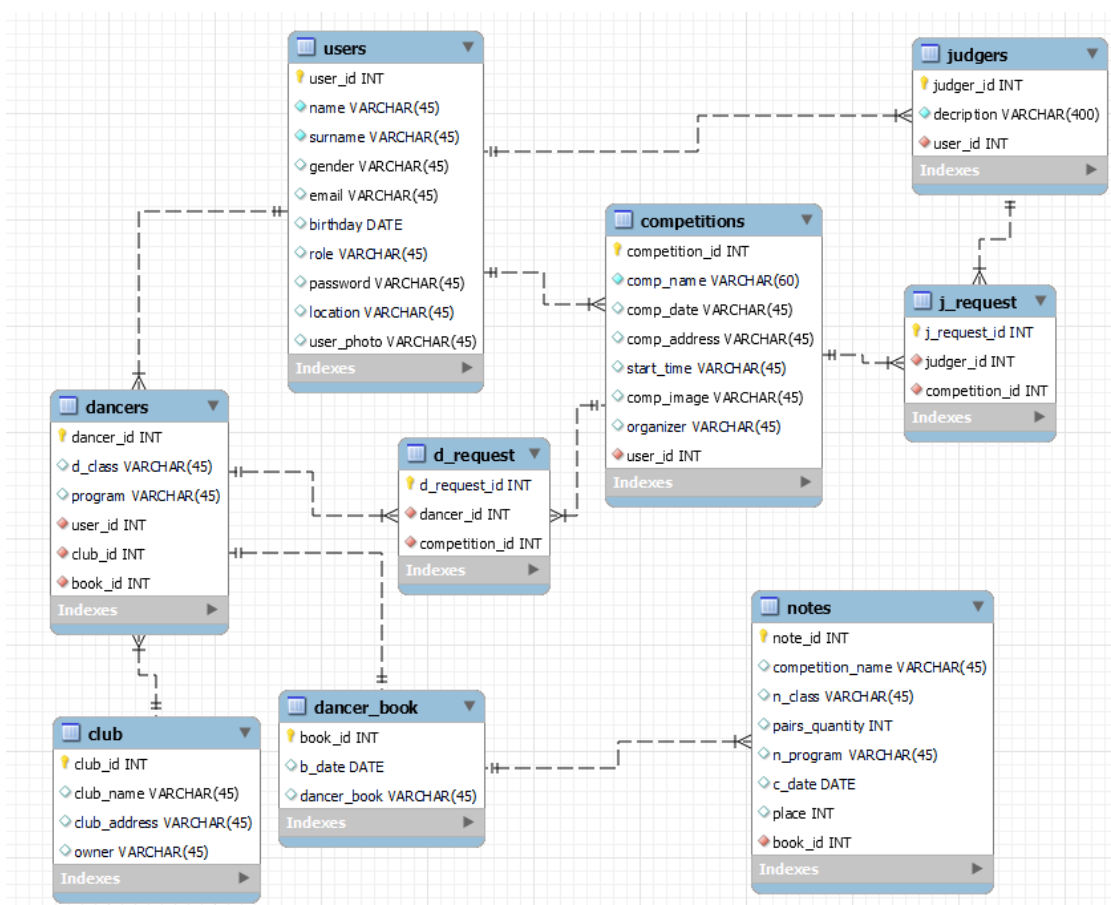


Рисунок 3.5 – Реалізація БД

### 3.3 Програмна реалізація

На рисунку 3.6 зображено структуру системи, згідно до якої видно, що основними модулями є: можливість реєстрації, входу до системи, додавання власної інформації та щодо майбутніх змагань, редагування цих внесених даних, а також можливість зареєструватися на змагання чи податися у якості судді.

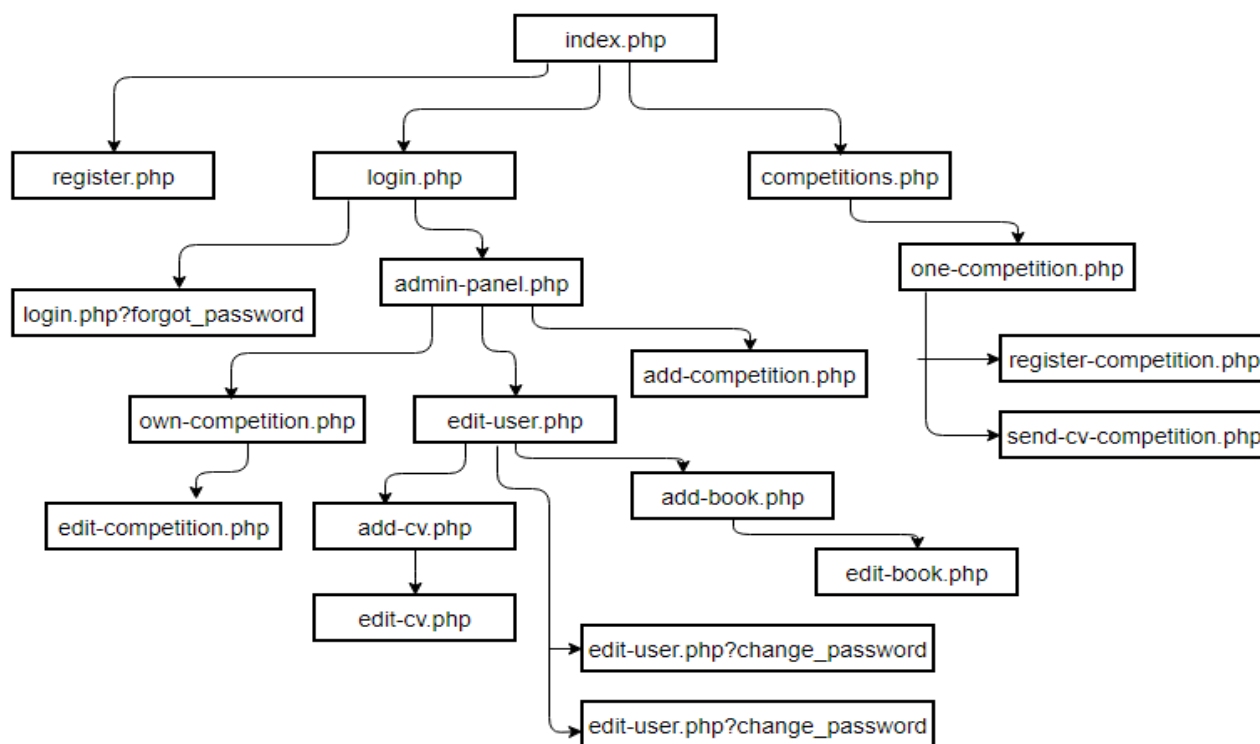


Рисунок 3.6 – Схематичне зображення структури системи

#### 3.3.1 Реєстрація користувача

Для реєстрації нового користувача заповнюється 5 полів, таких як: ім'я та фамілія, імейл, пароль та повторення введеного паролю. Далі для коректної реалізації процесу реєстрації для кожного з полів, яке користувач власноруч заповняє, створено змінну, в яку заносяться та перевіряються дані введені користувачем, далі вони оброблюються та заносяться в БД, як представлено на рисунку 3.7.

Тож, до основних змінних, які зберігають дані введені безпосередньо користувачем і зберігаються з методу POST, належать:

- \$n\_name – створена для зберігання введеного ім'я користувача;
- \$n\_surname – створена для зберігання введеної фамілії користувача;
- \$n\_email – створена для зберігання введеного імейлу користувача;
- \$n\_password - створена для зберігання введеного паролю користувача;
- \$n\_password2 - створена для зберігання повторно введеного паролю користувача.

Після введення даних користувачем відбувається перевірка чи немає вже користувача з таким імейлом в системі (занесеного в базі даних). Такий процес відбувається шляхом порівняння, за допомогою змінної \$check\_email й використання даних змінної \$n\_email. Якщо щойно введена користувачем електронна пошта вже наявна в системі, то користувача повідомлено сповіщенням «Помилка! Користувач з таким імейлом вже зареєстрований!».

Також відбувається перевірка чи збігаються введені значення в полях Пароль та Повторити пароль. Якщо ні, то користувач буду проінформований за допомогою повідомлення «Помилка! Значення в полі пароль та Повторити пароль не збігаються!».

Після пройдених двох перевірок змінна \$query формує запит, щоб додати введену інформацію в базу даних, а змінна \$result безпосередньо надсилає цей запит до бази даних. Далі користувача проінформовано про вдалу реєстрацію повідомленням «Реєстрація відбулася успішно. Пройдіть авторизацію».



```

<?php
if(isset($_POST["n_name"])){$n_name = $_POST[n_name];}
if(isset($_POST["n_surname"])){$n_surname = $_POST[n_surname];}
if(isset($_POST["n_email"])){$n_email = $_POST[n_email];}
if(isset($_POST["n_password"])){$n_password = $_POST[n_password];}
if(isset($_POST["n_password2"])){$n_password2 = $_POST[n_password2];}
if(isset($_POST["n_send"])){
    $check_email = mysql_query("SELECT 'email' FROM 'users' WHERE 'email' = '$n_email'");
    if(mysql_num_rows($check_email) != 0){
        echo '
        <div class="alert alert-danger alert-dismissible fade show" role="alert">
            <strong>Помилка!</strong> Користувач з таким імейлом вже зареєстрований!
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>';
    }else if(mysql_num_rows($check_email) == 0){
        if($n_password != $n_password2){
            echo '
            <div class="alert alert-danger alert-dismissible fade show" role="alert">
                <strong>Помилка!</strong> Значення в полі Пароль та Повторити пароль не збігаються!
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>';
        }else{
            $date = date("m.d.y");
            $query = "INSERT INTO 'users' ('name', 'surname', 'email', 'password', 'role',
            'location', 'user_photo') VALUES ('$n_name', '$n_surname', '$n_email', '$n_password', '2',
            'Відсутня', 'user.png');";
            $result = mysql_query($query) or die ("Error: ".mysql_error() );
            if($result){
                echo '<div class="alert alert-info" role="alert">
                Реєстрація відбулася успішно. Пройдіть <a href="login.php">авторизацію</a></div>';
                exit();
            }
        }
    }
}
?>

```

Рисунок 3.7 – Реєстрації нового користувача

### 3.3.2 Вхід користувача до системи

Для входу в систему зареєстрованому користувачу необхідно заповнити 2 поля форми авторизації – імейл (електронна пошта) та пароль. Також на формі входу/авторизації представлено редіректи на сторінки реєстрації та відновлення паролю для пришвидшення навігації користувача.

Змінні, які виконують головну роль в цьому модулі та приймають дані з методу POST:

- \$enter – використовується для збереження імейлу користувача, також виконується перевірка допустимих символів перед тим, як зберегти ці дані, за допомогою `mysql_real_escape_string` – методу, не дозволяє запускатися скриптам (та

вірусному коду), за допомогою додавання символу «\» перед кожним існуючим символом;

- \$password – використовується для збереження паролю користувача. Для безпеки та цілісності даних користувачів, як це робиться у багатьох системах, які використовують особисті дані, пароль зберігається в зашифрованому вигляді, що досягнене за допомогою використання методу md5.

Для входу в систему ще використовуються 2 змінних \$sql\_log та \$row\_log, одна для створення запиту до бази даних, а друга для відправки цього запиту до БД. Також виконується перевірка на збіг даних, які вже були занесені до системи, тобто якщо користувач з введеними даними не буде знайдений в базі даних, то він буде проінформований повідомленням «Помилка! Користувача з такими даними не знайдено. Перевірте будь-ласка чи введено правильні дані.». Якщо ж дані збіглися з із вже занесеними в базу даних, то користувач успішно ввійде до системи. В цей момент (один з найважливіших, бо виконується запам'ятовування) до сесії заноситься унікальний ідентифікатор та ім'я конкретного користувача, а сам користувач перенаправлений на головну сторінку. Це представлено на рисунку 3.8.

```

if (isset($_POST['send']))
{
    $sender = mysql_real_escape_string($_POST['email']);
    $password = md5($_POST['password']);
    $sql_log = mysql_query("SELECT 'user_id', 'role', 'name' FROM 'users'
WHERE 'email'='$sender' and 'password'='$password'
LIMIT 1", $link) or die(mysql_error());
    $row_log = mysql_fetch_array($sql_log);
    $id_prof = $row_log['user_id'];
    $_SESSION['user_role'] = $row_log['role'];
    if (mysql_num_rows($sql_log) == 1) {
        if($row_log['role']==1){
            $_SESSION['user_name'] = 'Админ';
            $_SESSION['user_id'] = $row_log['user_id'];
            echo '<meta http-equiv="refresh" content="0; url=admin-panel.php?
used_id='.$row_log['user_id'].'>';
        }
    }
    else{
        echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">
<strong>Помилка!</strong> Користувача з такими даними не знайдено. Перевірте будь-ласка
чи введено правильні дані.
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&time;</span>
</button>
</div>';
    }
}

```

Рисунок 3.8 – Вхід до системи зареєстрованого користувача

### 3.3.3 Редагування персональних даних

Так як під час реєстрації користувач заповнює лише 5 полів, для того щоб процес реєстрації був швидким та не вимагав введення великої кількості інформації на перших кроках використання системи, то реалізовано можливість редагування персональних даних, де є можливість заповнити більш детально інформацію про себе. В даному модулі користувач може видалити фотокартку, видалити створений профіль під час реєстрації або ж відредагувати свій профіль: заповнити більше даних про себе та в залежності від даних, які користувач буде вносити буде 2 типи профілів – танцівники та судді.

Модуль видалення фотокартки доступний тільки в тому разі, якщо користувач вже власноруч додавав якесь фото до свого профілю, у іншому випадку, коли встановлене зображення за замовчуванням, у користувача немає можливості виконати цю дію, що й продемонстровано на рисунку 3.9 й дає опис задуманої логіки.

Редагування профілю відбувається за таким процесом: представлена форма з полями, де відображені вже наявні в системі дані та порожні поля, де інформація ще була занесена. Логіка роботи даного модулю така: методом POST збирається введена інформація, далі додається в змінну `$sql_refresh` в sql запит, а вже останнім кроком відправляється запит до БД.

Реалізований код дає можливість системі самостійно розпізнавати, що, наприклад, фотокартка була додана, тому виконає її завантаження на сервер та занесення до БД. Даний код продемонстровано на рисунку 3.10.

```

if (isset($_POST['del_photo'])) {
    $sql_refresh = mysql_query("UPDATE 'users' SET 'user_photo' = 'user.png'
    WHERE 'user_id' = " . $_GET['red_id'] . ", $link);
    echo '<meta http-equiv="Refresh" content="0; URL="admin-panel.php?
    user_id=' . $_GET['red_id'] . "'>';
}

```

Рисунок 3.9 – Редагування (видалення фотокартки) із звичайного профілю користувача

```

if (isset($_POST['refresh_submit'])){
    if($_FILES['red_image']['name'] != ''){
        $path = 'images/';
        $tmp_path = 'temp/';
        //Массив допустимих значений для картинок
        $types = array('image/png', 'image/jpeg');
        //Максимальный объем данного файла
        $size = 4000000;
        $extension_img = strtolower(substr(strchr($_FILES['red_image']['name'], '.'),1));
        $name_img = rand(1, 1000).'-' .md5($_FILES['red_image']['name']).'.'.$extension_img;
        if ($_SERVER['REQUEST_METHOD'] == 'POST')
        {
            // 1) Проверка типа файла
            if (!in_array($_FILES['red_image']['type'], $types))
                die('Тип файла недопустимый. <a href="#">Загрузить другой?</a>'
                .$_FILES['red_image']['type']);
            // 2) Проверка размера файла
            if ($_FILES['red_image']['size'] > $size)
                die('Файл слишком большой. <a href="#">Загрузить другой?</a>');
            // 3) Загрузка подходящего файла и вывод оповещения
            if (!@copy($_FILES['red_image']['tmp_name'], $path.$name_img)){
                echo 'Что-то пошло не так.';
            }
        }
    }else { $name_image = $result['user_photo'];
    $sql_refresh = mysql_query("UPDATE 'users' SET 'name' = '".$_POST['red_name']."',
    'surname' = '".$_POST['red_surname']."', 'gender' = '".$_POST['red_gender']."',
    'birthday' = '".$_POST['red_birthday']."', 'location' = '".$_POST['red_location']."',
    'user_photo' = '".$_name_image"' WHERE 'user_id' = '".$_GET['red_id'];
    if(!$sql_refresh){
        echo '<p style="text-align:center; font-size^18px; color:red;">Помилка! '.
        mysql_error(). '</p></br>';
    }else{
        echo '<meta http-equiv="Refresh" content = "0"; URL="admin-pannel.php?
        user_id='.$_GET['red_id'].'">';
    }
}
}

```

Рисунок 3.10 – Редагування звичайного профілю користувача

### 3.3.4 Додавання Танцювальної книги

Користувачі, під час редагування власного профілю мають можливість обрати чи є вони кваліфіковані судді чи професійні танцівники. Під час вибору варіанта «танцівник» в них з'являється можливість створити власну «Танцювальну книгу». З логікою, яка була закладена ще на етапі проектування бази даних після створення цієї книги користувач зможе заносити записи щодо минулих результатів своїх змагань.

Звичайно перед створенням такої «Танцювальної книги» виконується перевірка чи є користувач танцівником, за допомогою змінної \$check\_level, і у разі відповіді ні, його буде проінформовано повідомленням «Помилка! Ви не є танцівником, тому ви

не можете створювати Танцювальні книги». А в протилежному випадку танцювальну книгу буде успішно додано. Цю логіку продемонстровано на рисунку 3.11.

```

if(isset($_POST["b_date"])){$b_date = $_POST[b_date];}
if(isset($_POST["n_send"])){
    $check_email = mysql_query("SELECT 'email' FROM 'users' WHERE 'email' = '$c_email'");
    $check_level = mysql_query("SELECT 'level' FROM 'users' WHERE 'level' = '$c_level'");
    if(mysql_num_rows($check_email) != 0 && ($check_level) != 2){
        echo'
            <div class="alert alert-danger alert-dismissible fade show" role="alert">
                <strong>Помилка!</strong> Ви не є танцівником, тому ви не можете створювати Танцювальні книги!
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>';
    }else (mysql_num_rows($check_level) == 2){
        $date = date("m.d.y");
        $query = "INSERT INTO 'dancer-book' ('b_date') VALUES ('$b_date')";
        $result = mysql_query($query) or die ("Error: ".mysql_error() );
        if($result){
            echo '<div class="alert alert-info" role="alert">
                Танцювальну книгу успішно додано, тепер ви можете додати результати ваших минух замахань.</div>';
            exit();
        }
    }
}

```

Рисунок 3.11 – Додавання «Танцювальної книги»

### 3.3.5 Додавання запису до Танцювальної книги

Як було описано вище, тільки користувачі, які є танцівниками мають змогу створити танцювальну книгу та додати до неї запис, який складається з 6 полів, для кожного створюється змінна, яка окремо зберігає для цих полів дані. Перед додаванням запису виконується перевірка чи є в цього танцівника взагалі створена книга, за допомогою змінної \$check\_book.

Якщо книги не має, то користувач бачить повідомлення «Помилка! У вас немає танцювальної книги». У протилежному разі запис до книги успішно заноситься та його можна переглянути в цій самій «Танцювальній книзі». Дана логіка продемонстрована на рисунку 3.12.

```

if(isset($_POST["n_name"])){$n_name = $_POST[n_name];}
if(isset($_POST["n_class"])){$n_class = $_POST[n_class];}
if(isset($_POST["n_pairs_quantity"])){$n_pairs_quantity = $_POST[n_pairs_quantity];}
if(isset($_POST["n_program"])){$n_program = $_POST[n_program];}
if(isset($_POST["n_date"])){$n_date = $_POST[n_date];}
if(isset($_POST["n_place"])){$n_place = $_POST[n_place];}
if(isset($_POST["n_send"])){
    $check_book = mysql_query("SELECT 'book_id' FROM 'dancer_book'");
    if(mysql_num_rows($check_book) == 0){
        echo'
        <div class="alert alert-danger alert-dismissible fade show" role="alert">
        <strong>Помилка!</strong> У вас немає танцювальної книги!
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&time;</span>
        </button>
        </div>';
    }else (mysql_num_rows($check_book) != 0 ){
        $date = date("m.d.y");
        $query = "INSERT INTO 'notes' ('competition_name', 'n_class', 'pairs_quantity', 'n_program', 'c_date',
        'place') VALUES ('$n_name', '$n_class', '$n_pairs_quantity', '$n_program', '$n_date', '$n_place')";
        $result = mysql_query($query) or die ("Error: ".mysql_error() );
        if($result){
            echo '<div class="alert alert-info" role="alert">
            Новий запис успішно додано до танцювальної книги.</div>';
            exit();
        }
    }
}
}

```

Рисунок 3.12 – Додавання запису до «Танцювальної книги»

### 3.3.6 Додавання змагання адміном

Для додавання нового змагання в систему, адмін заповнює 6 полів, таких як: назва та дата проведення, адреса місця проведення, час початку, хто є організатором та додає фотокартку змагання. Так само як і під час реєстрації для кожного з полів створено змінну, в яку заносяться та перевіряються дані, потім вони оброблюються та заносяться в БД. Цю реалізацію видно на рисунку 3.13.

Також виконується перевірка на те чи має користувач права додавати змагання чи ні. У випадку, якщо у користувача немає цих, то його буде проінформовано повідомленням «Помилка! У вас немає прав додавати змагання.». У позитивному випадку змагання додано до системи.

```

if(isset($_POST["c_name"])){$c_name = $_POST[c_name];}
if(isset($_POST["c_date"])){$c_date = $_POST[c_date];}
if(isset($_POST["c_address"])){$c_address = $_POST[c_address];}
if(isset($_POST["c_start_time"])){$c_start_time = $_POST[c_start_time];}
if(isset($_POST["c_image"])){$c_image = $_POST[c_image];}
if(isset($_POST["c_organizer"])){$c_organizer = $_POST[c_organizer];}
if(isset($_POST["c_send"])){
    $check_level = mysql_query("SELECT 'level' FROM 'users' WHERE 'level' = '$c_level'");
    if(mysql_num_rows($check_level) != 1){
        echo'
        <div class="alert alert-danger alert-dismissible fade show" role="alert">
            <strong>Помилка!</strong> У вас немає прав додавати змагання!
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&time;</span>
            </button>
        </div>';
    }else (mysql_num_rows($check_level) == 1){
        $date = date("m.d.y");
        $query = "INSERT INTO 'competitions' ('comp_name', 'comp_date', 'comp_address', 'start_time', 'comp_image',
        'organizer') VALUES ('$c_name', '$c_date', '$c_address', '$c_start_time', '$c_image', '$c_organizer')";
        $result = mysql_query($query) or die ("Error: ".mysql_error() );
        if($result){
            echo '<div class="alert alert-info" role="alert">
            Додавання змагання відбулося успішно.</div>';
            exit();
        }
    }
}

```

Рисунок 3.13 – Додавання змагання

### 3.3.7 Редагування змагання адміном

Так само, як і для модуля редагування власного профілю користувача, реалізовано редагування створеного адміном змагання, але його можна редагувати тільки до того моменту, поки на нього ще не встигли зареєструватися танцівники або податися судді, щоб оцінювати дане змагання. Адмін може замінити фотокартку змагання тільки в тому випадку, якщо він її додавав під час створення цього змагання. Дана реалізація продемонстрована на рисунку 3.14.

```

if (isset($_POST['del_image'])) {
    $sql_refresh = mysql_query("UPDATE 'users' SET 'c_image' = 'competition.png'
    WHERE 'user_id' = " . $_GET['red_id'] . ", $link);
    echo '<meta http-equiv="Refresh" content="0; URL="add-competition.php?
    user_id=' . $_GET['red_id'] . '">';
}

```

Рисунок 3.14 – Редагування фотокартки змагання



Також реалізовано редагування даних про змагання, а саме його назви, дати та адреси місця проведення, часу початку та того, хто є організатором. Це представлено формою з полями, які є вже попередньо заповнені, або ж порожні, якщо ці дані не були занесені ще, тобто так як і редагування даних профілю користувача. Логіка роботи цього модулю продемонстровано на рисунку 3.15, вона виконується наступним чином: перший крок – методом POST збирається введена інформація, другий крок – додається в змінну `$sql_refresh` в sql запит, а останнім – відправляється запит до БД.

```

if (isset($_POST['refresh_submit'])){
    if($_FILES['red_image']['name'] != ''){
        $path = 'images/';
        $tmp_path = 'temp/';
        //Массив допустимих значений для картинок
        $types = array('image/png', 'image/jpeg');
        //Максимальный объем данного файла
        $size = 4000000;
        $extension_img = strtolower(substr(strrchr($_FILES['red_image']['name'], '.'),1));
        $name_img = rand(1, 1000).'-' .md5($_FILES['red_image']['name']).'.'.$extension_img;
        if ($_SERVER['REQUEST_METHOD'] == 'POST')
        {
            // 1) Проверка типа файла
            if (!in_array($_FILES['red_image']['type'], $types))
                die('Тип файла недопустимый. <a href="#">Загрузить другой?</a>'
                .$_FILES['red_image']['type']);
            // 2) Проверка размера файла
            if ($_FILES['red_image']['size'] > $size)
                die('Файл слишком большой. <a href="#">Загрузить другой?</a>');
            // 3) Загрузка подходящего файла и вывод оповещения
            if (!@copy($_FILES['red_image']['tmp_name'], $path.$name_img)){
                echo 'Что-то пошло не так.';
            }
        }
        }else {$name_img = $result['c_image'];}
        $sql_refresh = mysql_query("UPDATE 'competitions' SET 'comp_name' = '".$_POST['red_name']."',
        'comp_date' = '".$_POST['red_date']."', 'comp_address' = '".$_POST['red_address']."',
        'start_time' = '".$_POST['red_time']."', 'comp_image' = '".$name_img.",
        'organizer' = '".$_POST['red_organizer']."' WHERE 'user_id' = '".$_GET['red_id'];
        if(!$sql_refresh){
            echo '<p style="text-align:center; font-size^18px; color:red;">Помилка! '.
            mysql_error().'</p><br>';
        }else{
            echo '<meta http-equiv="Refresh" content = "0"; URL="add-competition.php?
            user_id='.$_GET['red_id'].'">';
        }
    }
}

```

Рисунок 3.15 – Редагування даних змагання

### 3.3.8 Відновлення паролю користувачем

У розроблюваній системі реалізовано функціонал відновлення, бо у разі втрати користувачем пароля необхідна бути можливість відновити його, щоб уберегти технічну підтримку від відновлення паролю із їхньої сторони.



У користувача є можливість ввести у форму відновлення свій імейл, далі відбувається перевірка чи наявний такий в базі даних даної системи, якщо ні, то користувач бачить повідомлення, що даного імейла немає в системі. Якщо даний імейл є в системі, то користувач отримує повідомлення, що новий пароль було надіслано йому на пошту та він направлений на сторінку входу до системи. А на пошту йому було відправлено згенерований довільно пароль. Цю логіку можна побачити на рисунку 3.16.

```

if (isset($_POST['send_email'])) {
    $sender = mysql_real_escape_string($_POST['email']);
    $sql_log = mysql_query("SELECT 'name', 'surname' FROM 'users'
WHERE 'email'='{$sender}' LIMIT 1", $link) or die(mysql_error());
    $row_log = mysql_fetch_array($sql_log);
    $name = $row_log['name'];
    $surname = $row_log['surname'];
    if (mysql_num_rows($sql_log) == 1) {
        $array_pass = array ("45", "66", "7", "3", "37", "99", "82", "2", "12", "4",
            "a", "f", "r", "q", "t", "i", "g", "a", "4g", "ro", "71", "k1", "d", "y8", "o",
            "d", "fe", "fh", "oe", "i8", "lq", "p", "f9", "l2", "oq", "0o", "li");
        for ($i = 0; $i < 8; $i++)
        {
            shuffle ($array_pass);
            $string = $string.$array_pass[1];
        }
        $new_password = md5($string);
        $restore_password = mysql_query("UPDATE 'users' SET 'password'='{$new_password}' WHERE
'email'='{$sender}'", $link) or die(mysql_error());
        mail($sender, "Запит на відновлення паролю", "Раді вас вітати, $name $surname.
Вам надіслано новий пароль: $string");
        echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">
            <strong>Відмінно!</strong> Новий пароль надіслано. Можете скористатися
            ним та увійти до системи.<a href="login.php">Авторизація</a>
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&time;</span>
            </button>
        </div>';
    } else {
        echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">
            <strong>Помилка!</strong> Користувача з таким імейлом не знайдено в системі.
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                <span aria-hidden="true">&time;</span>
            </button>
        </div>';
    }
}

```

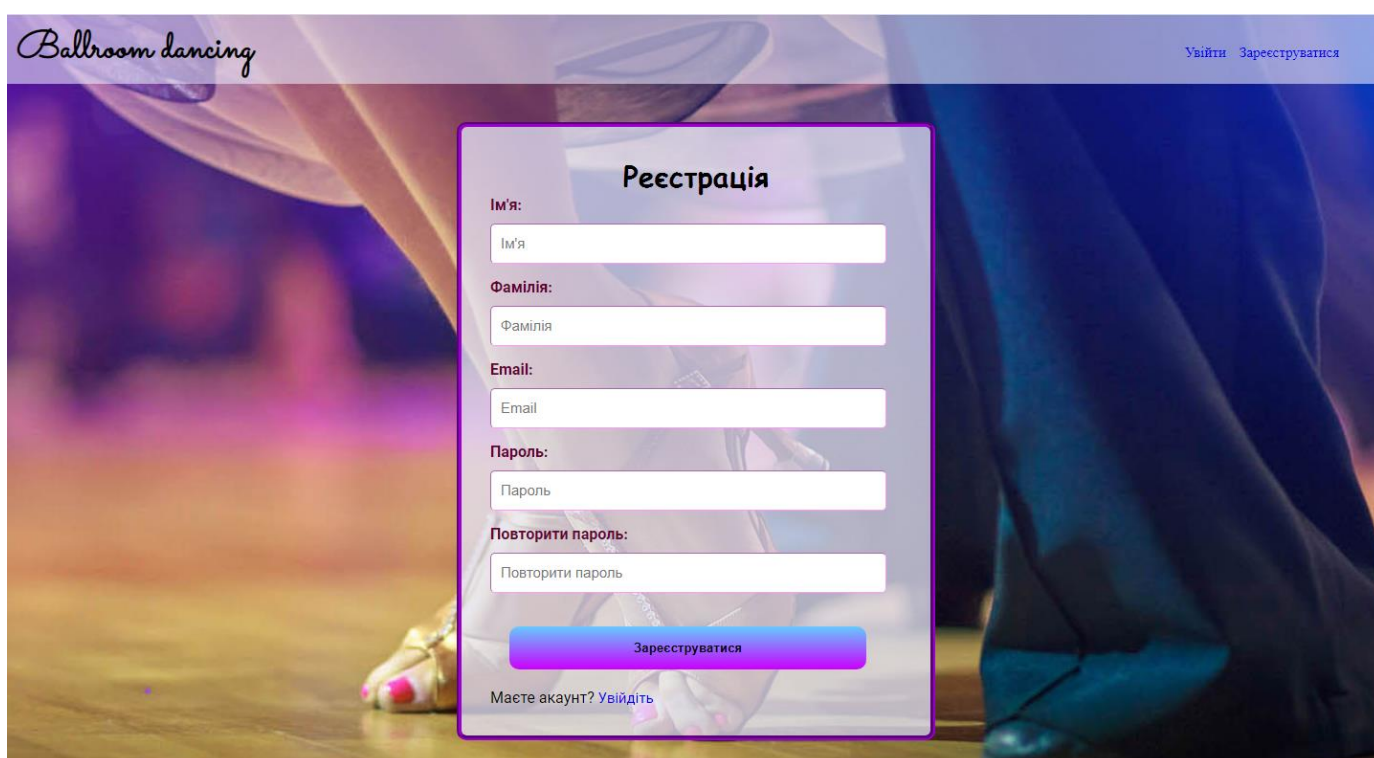
Рисунок 3.16 – Відновлення паролю зареєстрованого користувача

### 3.4 Програмна реалізація клієнтської частини

Першим кроком для користувача, який хоче працювати з даною системою, а не тільки переглядати дані по змаганням та не мати змоги на них зареєструватися, є реєстрація. Дана форма представлена з 5 полів, кожна поле має свої допустимі значення – валідацію:

- Ім'я
- Фамілія
- Імейл
- Пароль
- Повторити пароль

На рисунку 3.17 представлено форму реєстрації, вона містить посилання на сторінку авторизації, тому користувач вільно може перейти на форму входу до системи.



The image shows a registration form titled "Реєстрація" overlaid on a background image of a ballroom dancer. The form includes the following fields and elements:

- Ім'я:
- Фамілія:
- Email:
- Пароль:
- Повторити пароль:
- Зареєструватися (button)
- Маєте акаунт? [Увійдіть](#)

The background image shows a person's legs in a blue dress and high heels, dancing on a wooden floor. The text "Ballroom dancing" is visible in the top left corner, and "Увійти Зареєструватися" is in the top right corner.

Рисунок 3.17 – Форма реєстрації нового користувача

Якщо введені користувачем дані не збігаються з очікуваними, а саме, якщо дані в полі Пароль та Повторити пароль не є однаковими або введений імейл вже занесений до системи, то користувача проінформовано. У випадку успішної реєстрації також буде проінформовано та додано посилання на авторизацію. Дані модальні вікна продемонстровано на рисунку 3.18.

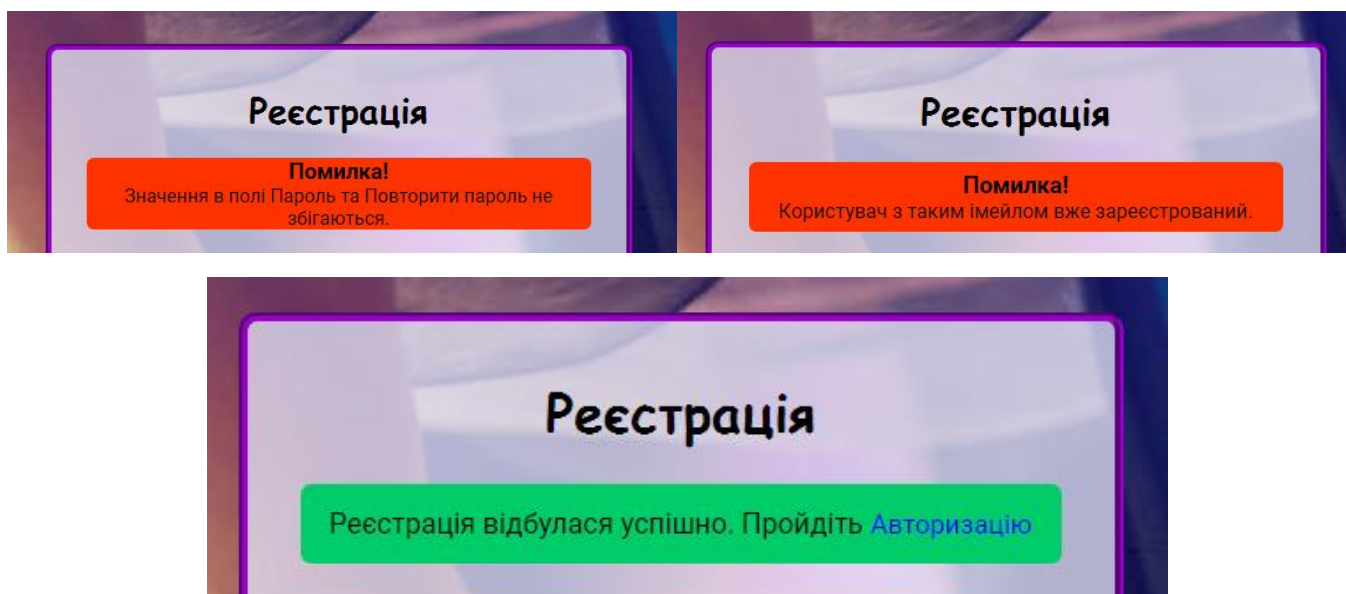


Рисунок 3.18 – Модальні вікна під час реєстрації

Форма авторизації представлена на рисунку 3.19, вона складається з двох полів – імейла та паролю, все подібно до форми реєстрації. Також на цій формі міститься посилання на сторінку відновлення паролю та сторінку проходження реєстрації для швидкої навігації по сайту.

Якщо під входу до системи користувач допустив помилку у введенні своїх даних, то його буде проінформовано, як представлено на рисунку 3.20.

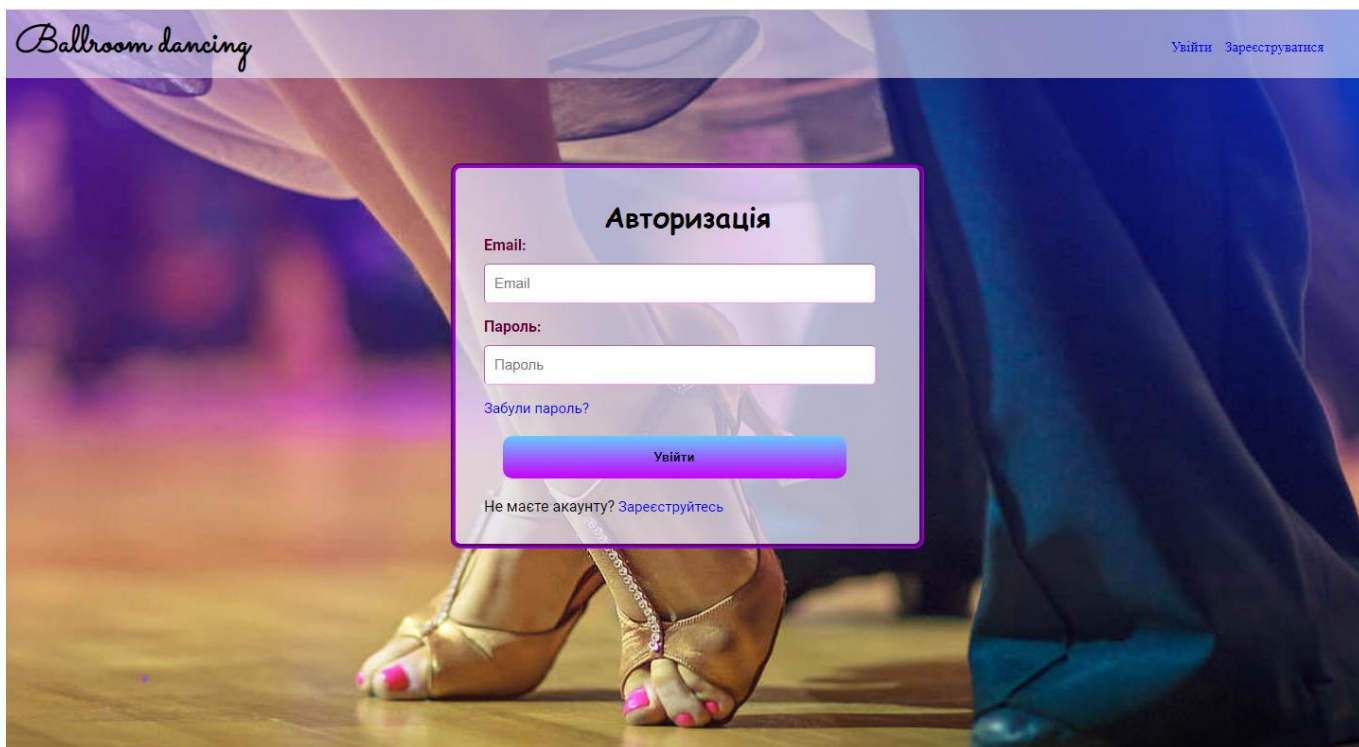


Рисунок 3.19 – Форма авторизації зареєстрованого користувача

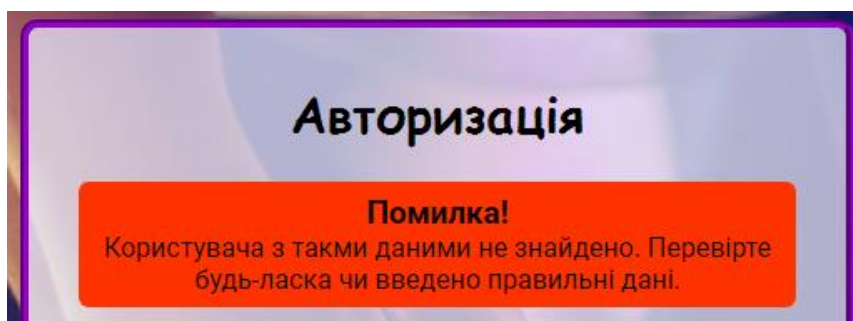


Рисунок 3.20 – Модальне вікно під час входу до системи

Також у випадку, якщо користувач забув свій пароль у нього є можливість відновити його. Він буде направлений на сторінку, де необхідно бути ввести свою електронну адресу, після на пошту буде відправлено автоматично рандомно згенерований пароль. Якщо користувач ввів електронну адресу, якої немає в системі, його буде проінформовано та дана можливість повторно ввести. При успішному заповненні поля, на пошту буде отримано пароль та користувач має можливість

авторизуватися в системі. Сторінку відновлення пароля та модальні вікна продемонстровано на рисунку 3.21.

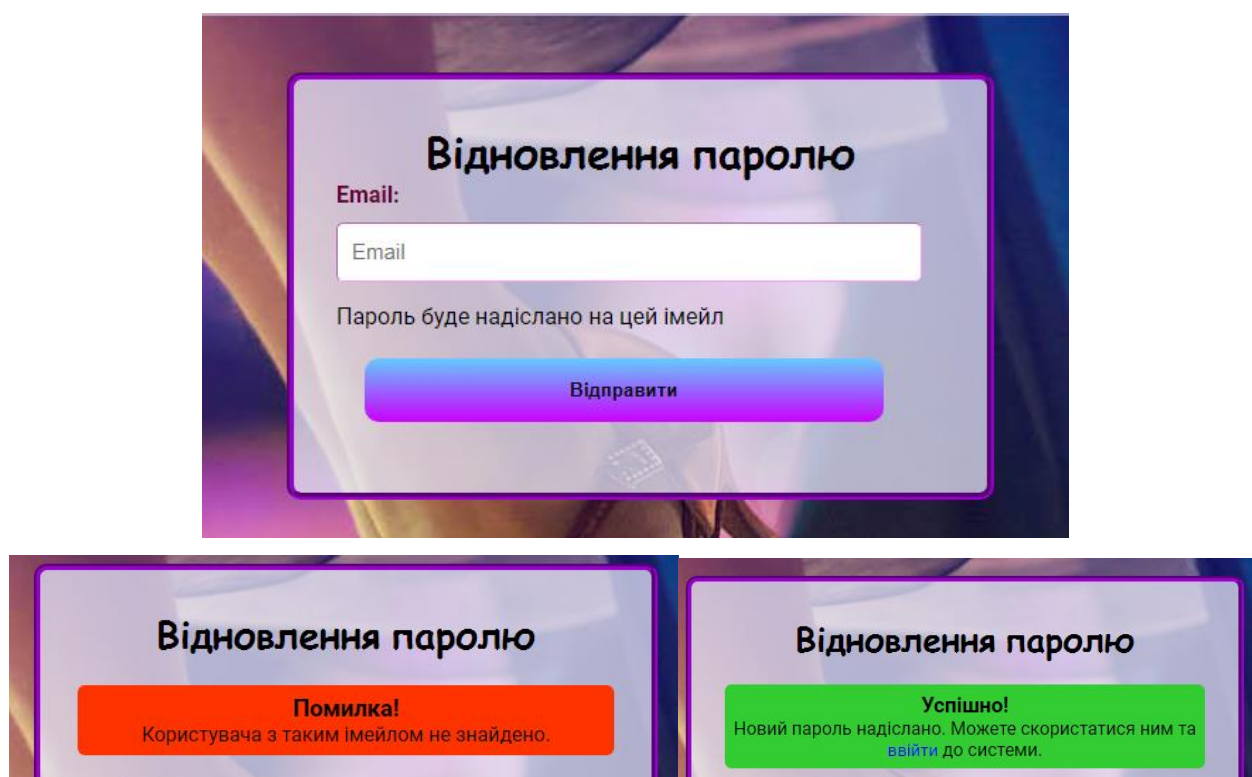


Рисунок 3.21 – Форма відновлення пароля з модальними вікнами

Після того, як користувач успішно зареєструвався в системі, перша сторінка на яку він потрапляє, це власний профіль, де він бачить всю заповнену інформацію під час реєстрації, має змогу відредагувати свій профіль, додати більше інформації або взагалі видалити власний профіль. Дана реалізація представлена на рисунку 3.22.



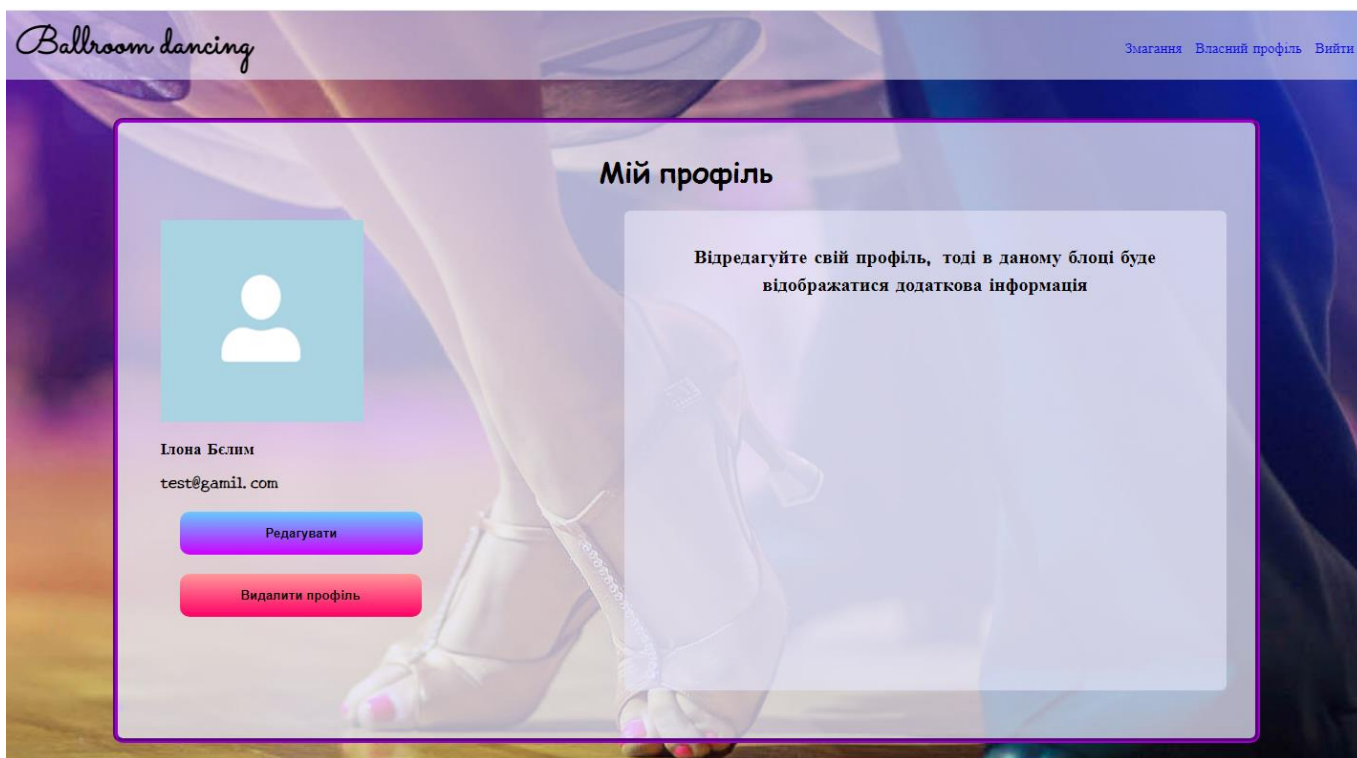


Рисунок 3.22 – Профіль щойно зареєстрованого нового користувача

Після того, як користувач натисне на кнопку «Редагувати» він буде направлений на сторінку редагування власного профілю, де зможе доповнити інформацію про себе, змінити фотокартку з тої, що йде за замовчуванням на власну.

Головним моментом є те, що користувач може обрати чи є він танцівником чи суддею, в залежності від цього він додатково буде доповнювати інформацію. Перший крок редагування представлений на рисунку 3.23.

Ballroom dancing

Змагання Бласний профіль Вийти

### Відредагуйте потрібну інформацію

Ім'я:

Фамілія:

Email:

День народження:

Пароль:

Місто:

Оберіть стать

Чоловіча

Жіноча

Фотокартка:  
 Файл не вибран

Оберіть танцювальну професію:

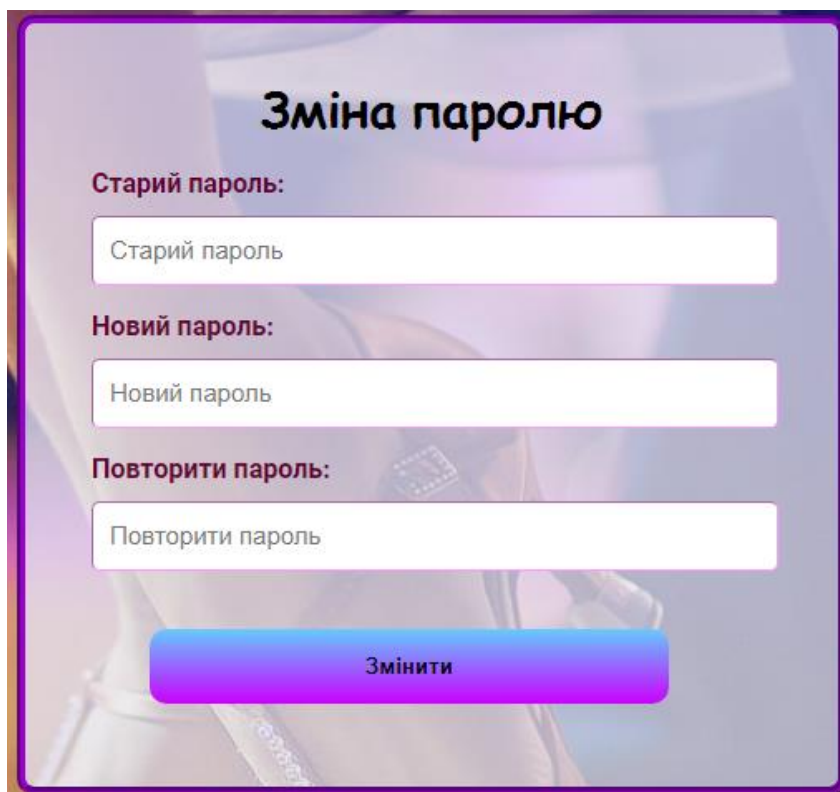
Суддя

Танцівник

Рисунок 3.23 – Форма редагування звичайного профілю

Форма редагування дає можливість змінити всі занесені поля, але для редагування паролю та імейлу реалізовано 2 окремі сторінки, на які перенаправляється користувача після натиснення на кнопку «Змінити».

Для зміни паролю створена форма з 3-ма полями (рис. 3.24) для поточного паролю, нового та повторення його. Також реалізовано 3 модальні вікна (рис. 3.25) для таких подій, як: все вірно і користувач успішно змінив пароль й може перейти на сторінку редагування профілю, помилка, старий пароль не знайдено в системі або ж значення в полях Пароль та Повторити пароль не збігаються.



**Зміна паролю**

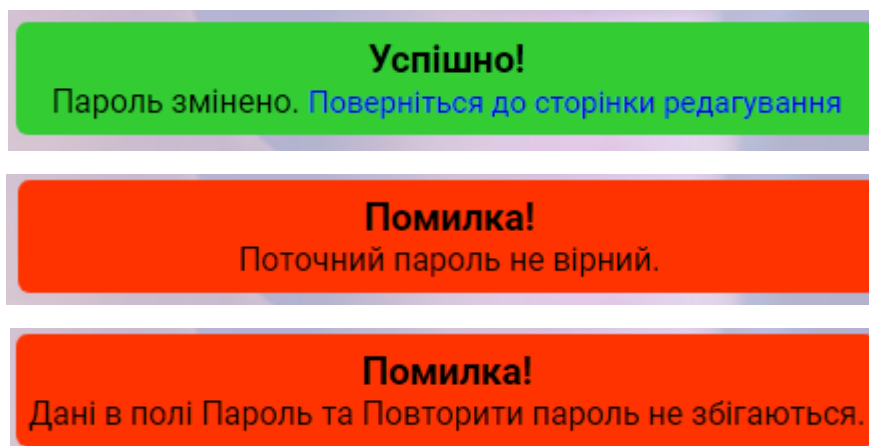
Старий пароль:

Новий пароль:

Повторити пароль:

**Змінити**

Рисунок 3.24 – Форма редагування паролю



**Успішно!**  
Пароль змінено. [Поверніться до сторінки редагування](#)

**Помилка!**  
Поточний пароль не вірний.

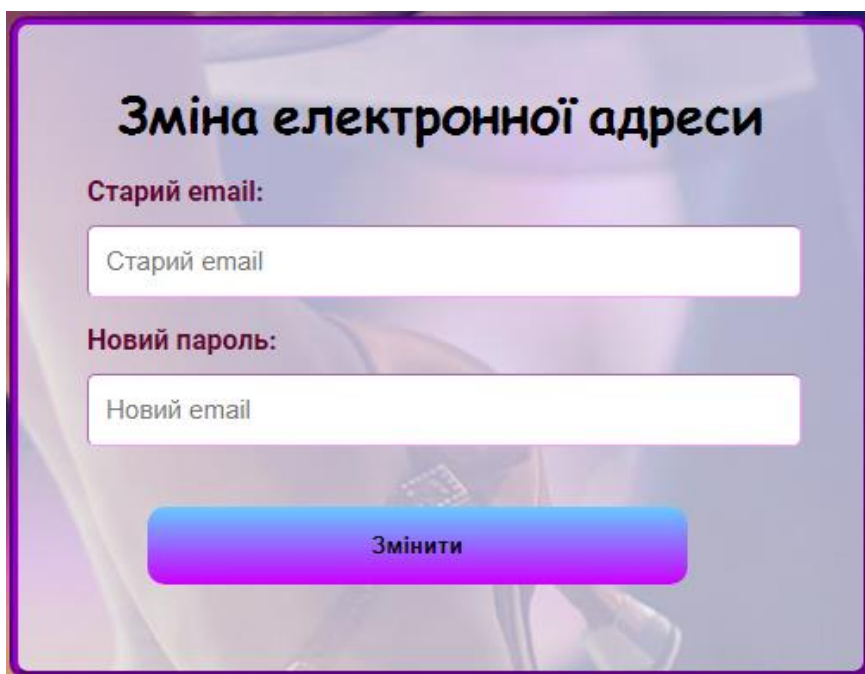
**Помилка!**  
Дані в полі Пароль та Повторити пароль не збігаються.

Рисунок 3.25 – Модальні вікна під час зміни паролю

Редагування електронної адреси (рис. 3.26) подібне до зміни паролю. Після натискання кнопки «Змінити», користувачу відкривається нова сторінка з формою з 2-ма полями для введення поточної адреси та нової.

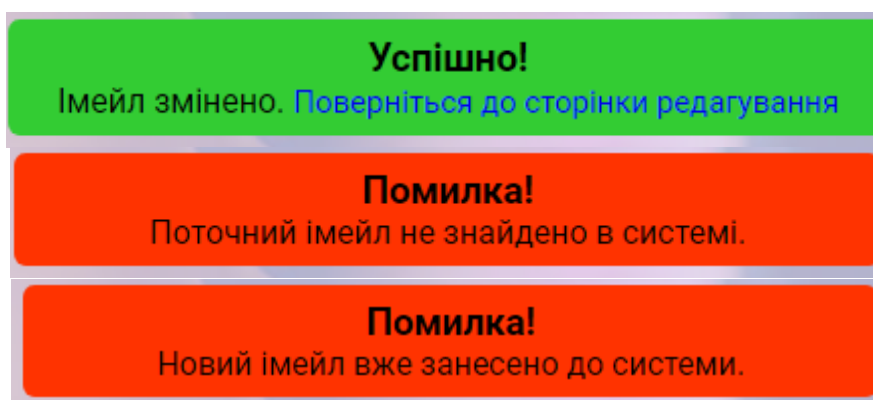


Також реалізовано 3 види модальних вікон (рис. 3.27): коли поточний імейл не знайдено в системі, коли новий імейл вже занесений до системи і успішна зміна імейлу.



The image shows a modal window titled "Зміна електронної адреси" (Change email address). It contains two input fields: "Старий email:" (Old email) and "Новий пароль:" (New password). Below the fields is a blue button labeled "Змінити" (Change).

Рисунок 3.26 – Форма редагування імейлу



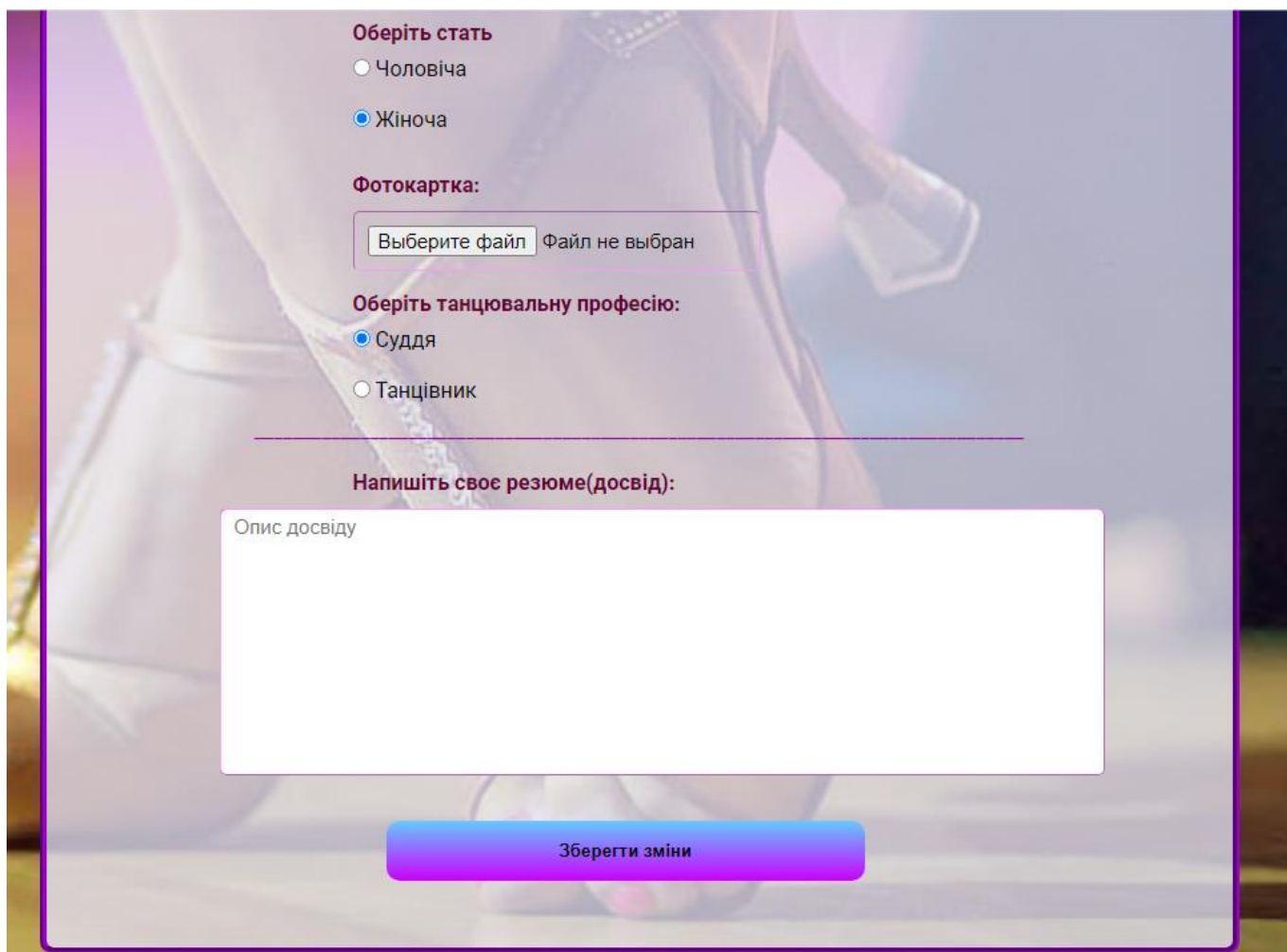
The image shows three stacked modal windows. The top one is green and says "Успішно!" (Success!) and "Імейл змінено. [Поверніться до сторінки редагування](#)" (Email changed. [Return to the edit page](#)). The middle one is red and says "Помилка!" (Error!) and "Поточний імейл не знайдено в системі." (Current email not found in system). The bottom one is red and says "Помилка!" (Error!) and "Новий імейл вже занесено до системи." (New email already entered in system).

Рисунок 3.27 – Модальні вікна під час зміни імейлу

Великою перевагою даної системи є те, що користувач під час редагування може перетворити свій профіль на профіль танцівника або ж на профіль судді. В

залежності від того, що він вибере під час редагування поля «Оберіть танцювальну професію».

Якщо він обирає – «Суддя» (рис. 2.28), то йому відкривається текстове поле для введення інформації про себе, опису свого досвіду у сфері суддівства бальних танців та взагалі досвіду у сфері бальних танців.

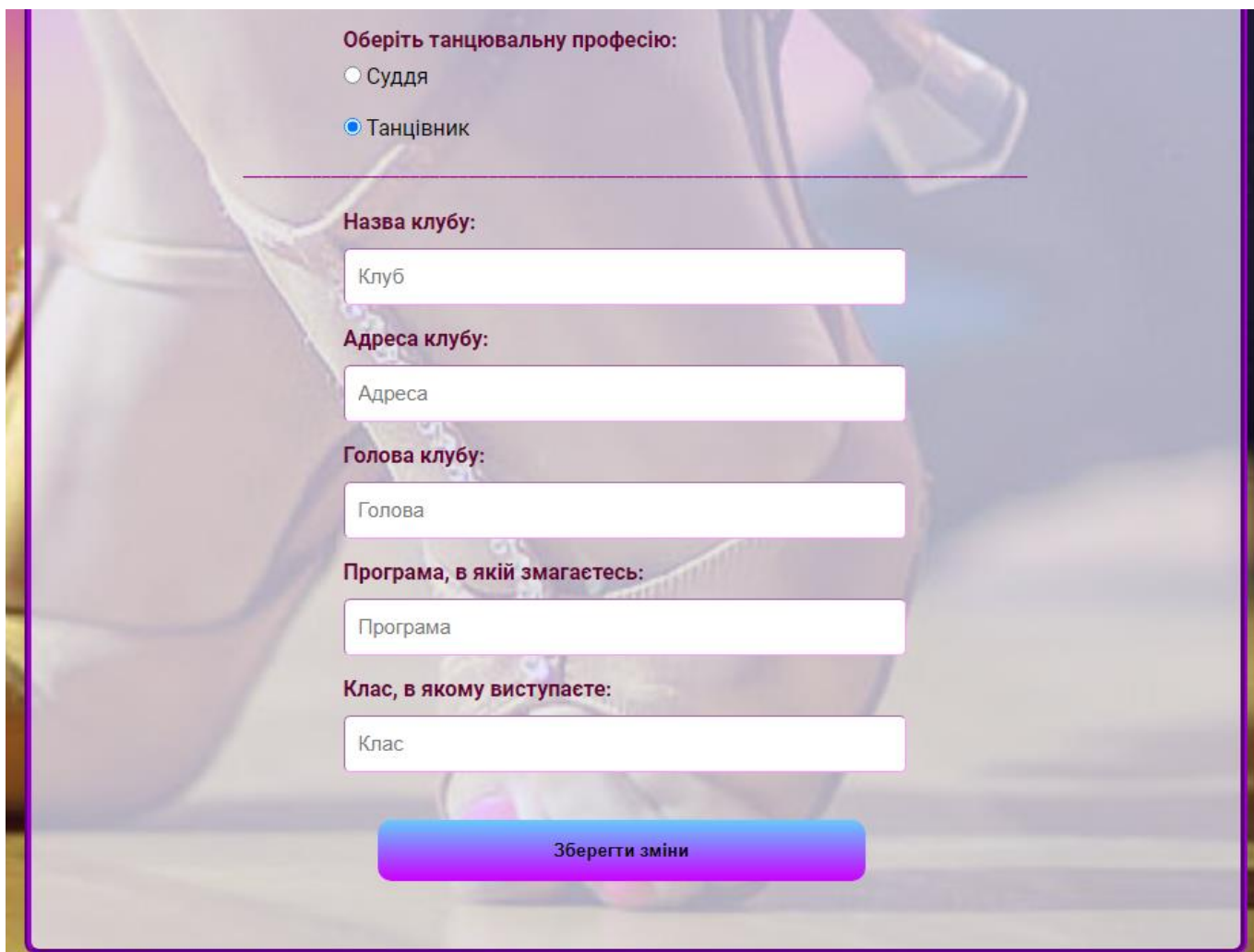


The image shows a web form for editing a profile. The form is overlaid on a background image of a ballroom dancer. The form contains the following elements:

- Оберіть стать** (Select gender):
  - Чоловіча (Male)
  - Жіноча (Female)
- Фотокартка:** (Profile picture)
  - Выберите файл | Файл не выбран (Choose file | File not selected)
- Оберіть танцювальну професію:** (Select dancing profession):
  - Суддя (Judge)
  - Танцівник (Dancer)
- Напишіть своє резюме(досвід):** (Write your resume/experience):
  - Опис досвіду (Description of experience) - a large text input area.
- Зберегти зміни** (Save changes) - a blue button at the bottom.

Рисунок 3.28 – Редагування профілю з вибором професії «Суддя»

Якщо обирається «Танцівник» (рис. 3.29), то користувачу необхідно заповнити додатково п'ять полів стосовно своєї сфери діяльності, а саме: назву, адресу та голову клубу, програму в якій на даний момент змагається та клас, в якому виступає.



Оберіть танцювальну професію:

Суддя

Танцівник

---

Назва клубу:

Адреса клубу:

Голова клубу:

Програма, в якій змагаєтесь:

Клас, в якому виступаєте:

Зберегти зміни


Рисунок 3.29 – Редагування профілю з вибором професії «Танцівник»

В залежності від того, які дані користувач заповнив під час редагування свого профілю є 3 варіанти відображення даних сторінці власного профілю користувача, а саме:

- Профіль, де користувач не обирає чи є він танцівником, чи суддею, тому там відображається коротка загальна інформація;
- Профіль, де користувач заповнює інформацію стосовно суддівства;
- Профіль, де користувач заповнює дані стосовно клубу, програми та свого класу (рис. 3.30).

*Ballroom dancing* Змагання Власний профіль Вийти

### Мій профіль



Глона Бєлим  
test@gmail.com

[Редагувати](#)

[Видалити профіль](#)

**Додаткова інформація**

---

Дата народження: 09. 10. 2000

Місто: Полтава

Стать: Жіноча

Професія: Танцівник

**Про клуб**

---

Назва: "Visavi"

Адреса: вул. Новослвська 103, Полтава, Україна

Голова клубу: Навородський І. Б.

**Кваліфікація**

---

Програма: Стандартна, Латино-американська

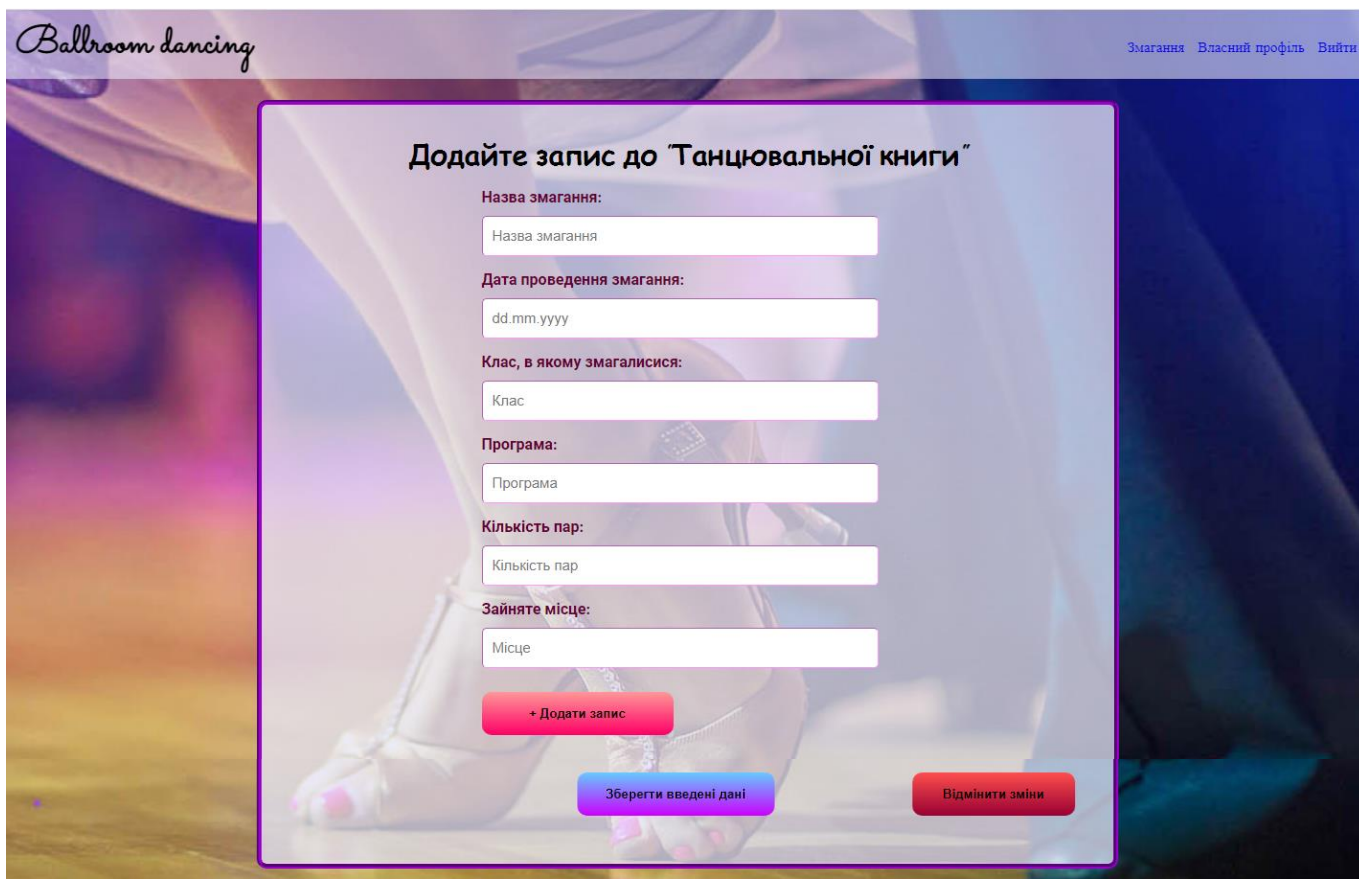
Клас: C-St, D-Lt

[Додати "Танцювальну книгу"](#)

Рисунок 3.30 – Профіль користувача-танцівника після редагування

Звичайно реалізовано можливість редагування вже внесених цих додаткових даних, користувач зі звичайним профілем має можливість відредагуватися та перетворити його на профіль танцівника чи судді. Користувач-суддя чи користувач-танцівник може змінити свої дані вже тільки в залежності від типу свого профілю.

Для користувача-танцівника реалізована можливість створення «Танцювальної книги» після натиснення на кнопку «Додати «Танцювальну книгу»» його буде направлено на сторінку її створення (рис. 3.31), де можна додати результати минулих змагань. За один раз можна додати від одного до кількох результатів (рис. 3.32).



*Ballroom dancing* Змагання [Власний профіль](#) [Вийти](#)

### Додайте запис до «Танцювальної книги»

Назва змагання:

Дата проведення змагання:

Клас, в якому змагалися:

Програма:

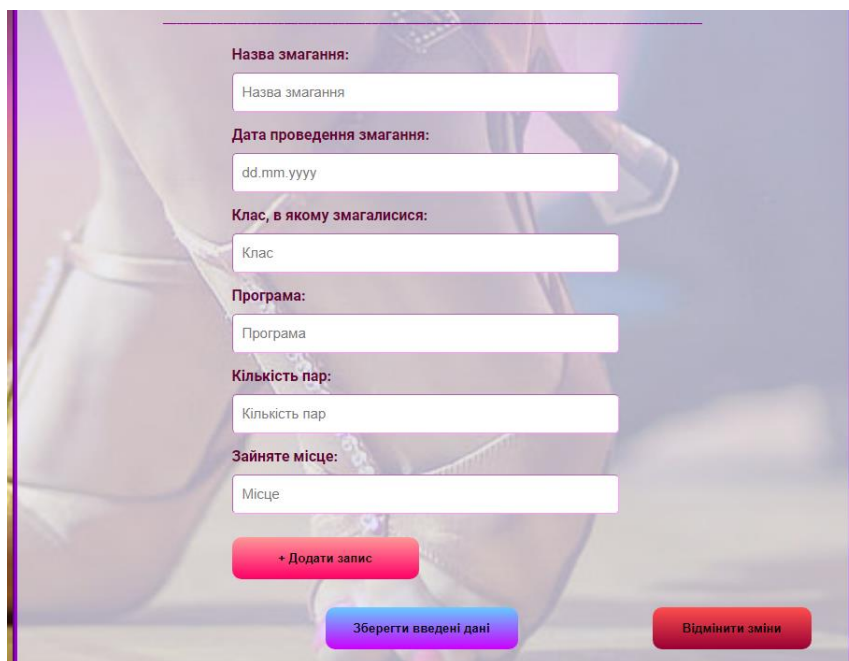
Кількість пар:

Зайняте місце:

[+ Додати запис](#)

[Зберегти введені дані](#) [Відмінити зміни](#)

Рисунок 3.31 – Створення одного запису в «Танцювальній книзі»



Назва змагання:

Дата проведення змагання:

Клас, в якому змагалися:

Програма:

Кількість пар:

Зайняте місце:

[+ Додати запис](#)

[Зберегти введені дані](#) [Відмінити зміни](#)

Рисунок 3.32 – Створення ще одного запису в «Танцювальній книзі»



Після збереження введених даних до «Танцювальної книги» користувач може переглянути їх та відредагувати, або ж ще додати нові дані до танцювальної книги.

Переглянути танцювальну книгу можна перейшовши на сторінку власного профілю та натиснувши на кнопку «Переглянути «Танцювальну книгу»», як зображено на рисунку 3.33.

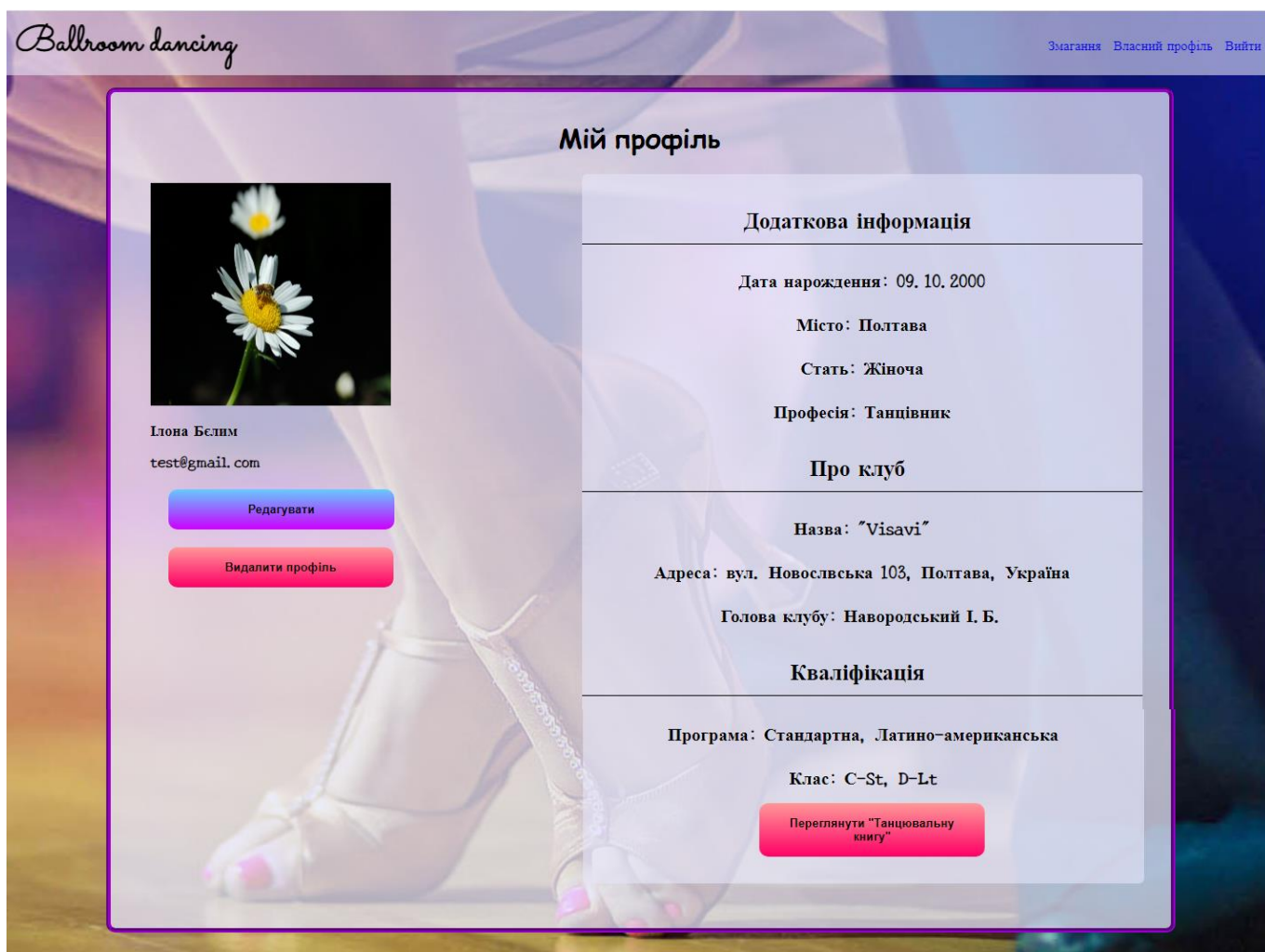


Рисунок 3.33 – Профіль користувача-танцівника, який створив «Танцювальну книгу»

Відкривається сторінка профілю, з відображеними загальними даними з лівої сторони, а з правої – занесені записи до «Танцювальної книги» (рис. 3.34), які можна відредагувати натиснувши на кнопку редагування, після буде направлено на окрему сторінку з редагуванням (рис. 3.35), де водночас можна одразу додати ще нові записи

до книги. А також можна відмінити внесені зміни і користувач повернеться на сторінку профілю із даними танцювальної книги.

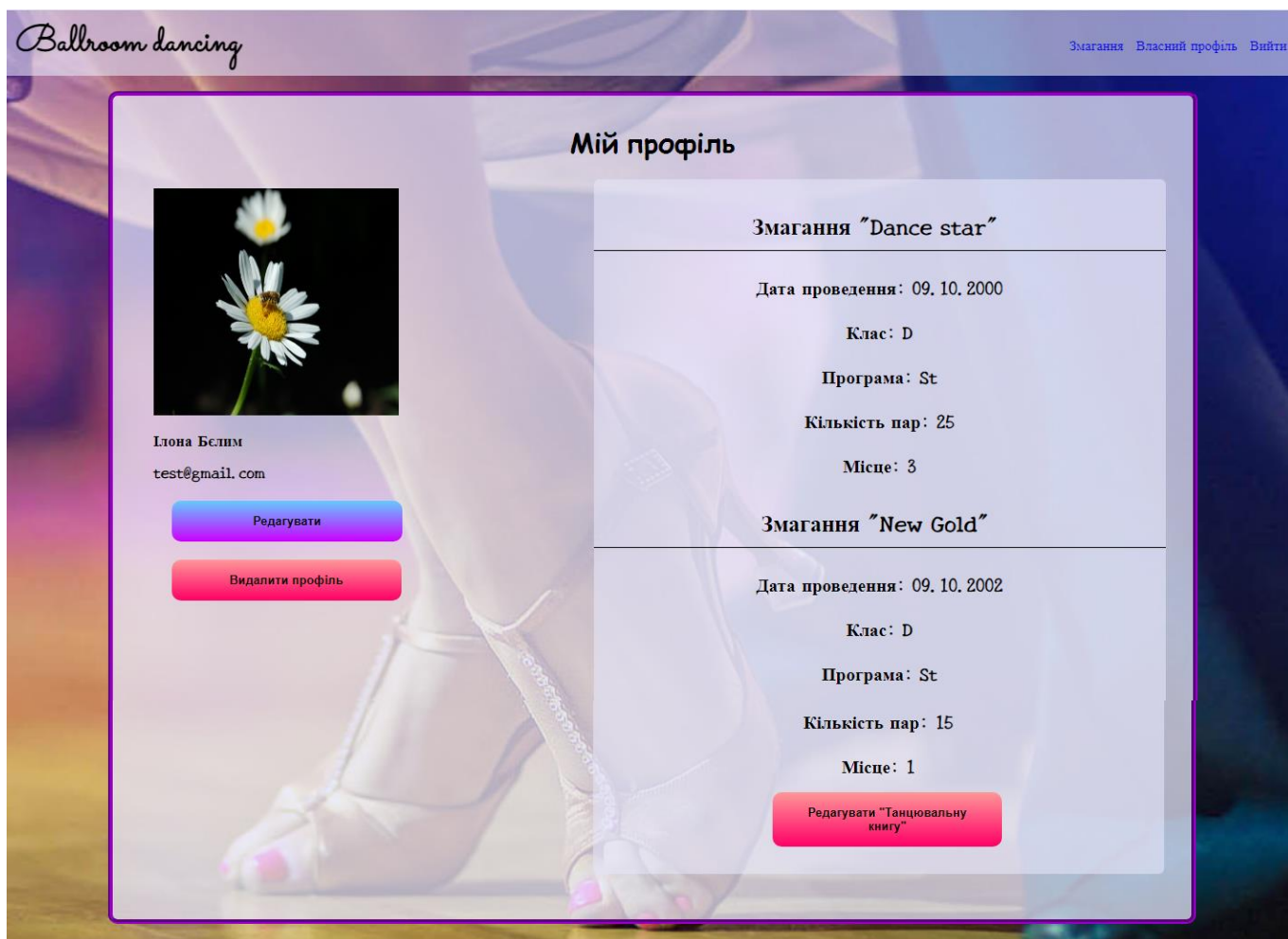
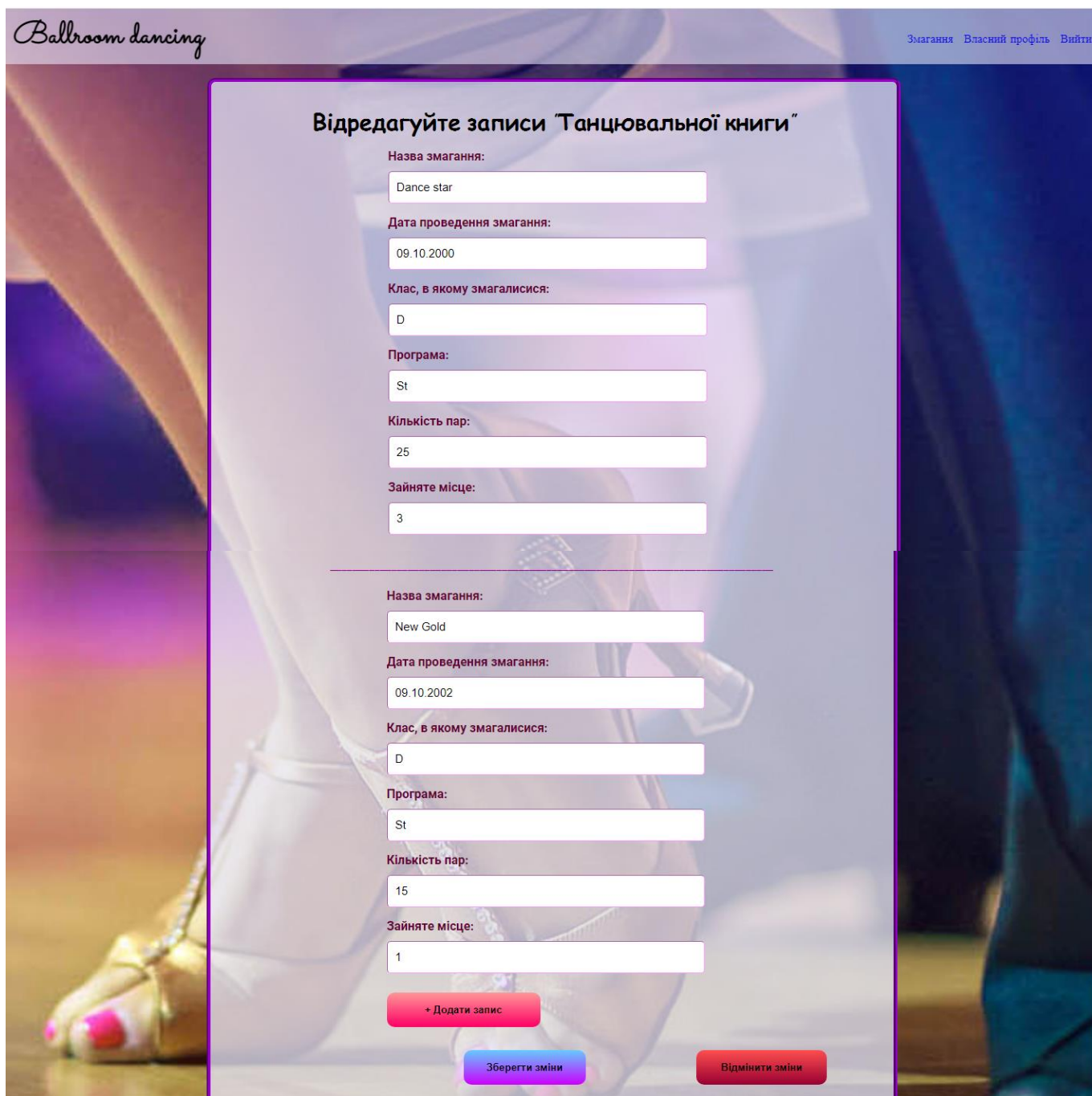


Рисунок 3.34 – Профіль користувача-танцівника і дані «Танцювальної книги»

Отже, для користувача з роллю – танцівник, обраною після редагування власного профілю реалізовано більш широкий функціонал для того, щоб він зміг відійти від звичайної паперової книги, яка є у кожного танцівника, до електронної для більш зручного і швидкого моніторингу своїх досягнень та аналізу своїх можливостей.

Рисунок 3.35 є сторінкою редагування танцювальної книги, де занесено 2 записи, а також можна додати ще один або декілька і зберегти все водночас.



*Ballroom dancing* Змагання Власний профіль Вийти

### Відредагуйте записи «Танцювальної книги»

Назва змагання:

Дата проведення змагання:

Клас, в якому змагалися:

Програма:

Кількість пар:

Зайняте місце:

---

Назва змагання:

Дата проведення змагання:

Клас, в якому змагалися:

Програма:

Кількість пар:

Зайняте місце:

Рисунок 3.35 – Сторінка редагування «Танцювальної книги»

Ще в системі є користувачі з правами адміністратора, їх профіль (рис. 3.36) виглядає подібно до профілю звичайного користувача, який щойно зареєструвався в системі, їм не потрібно редагувати власні дані. Тільки адміністратори можуть створювати нові змагання (рис. 3.37), після чого вони будуть доступні для перегляду для всіх користувачів системи.



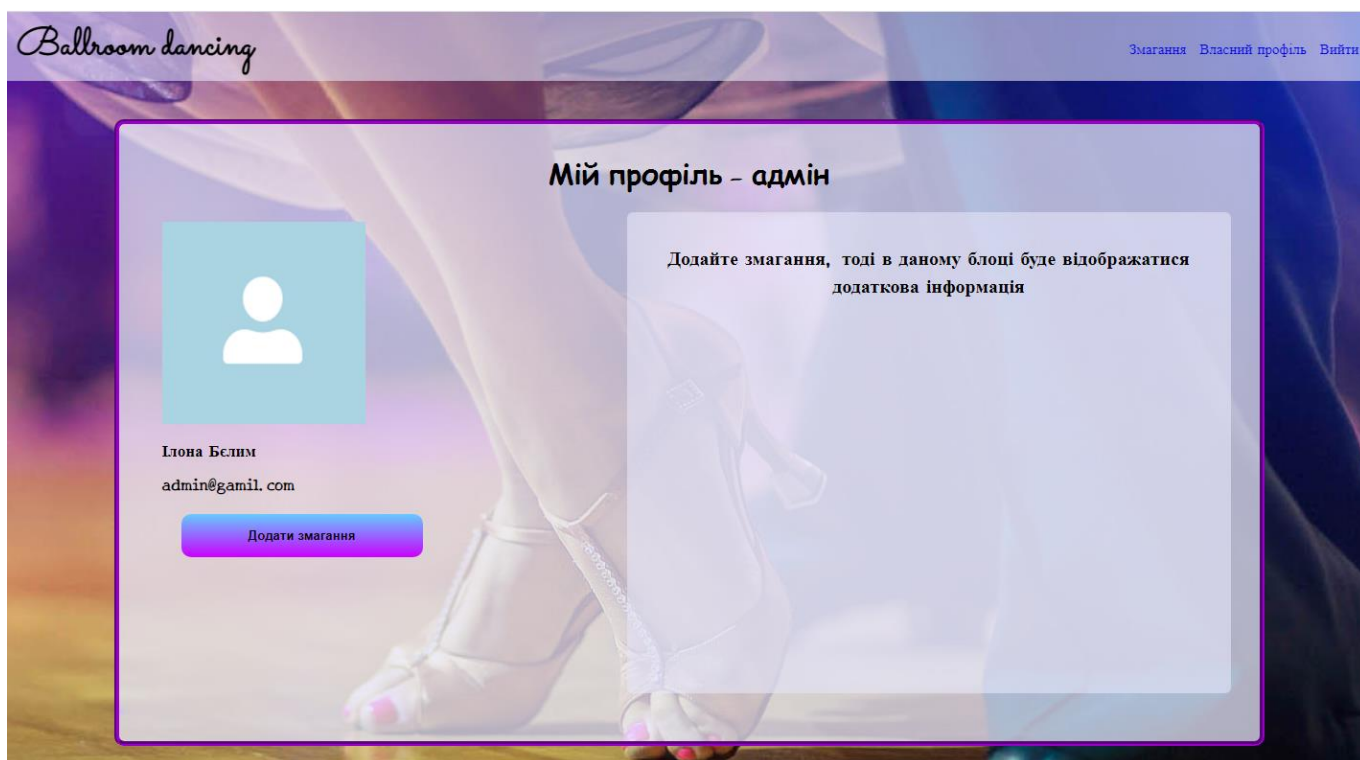


Рисунок 3.36 – Сторінка профілю користувача-адміністратора

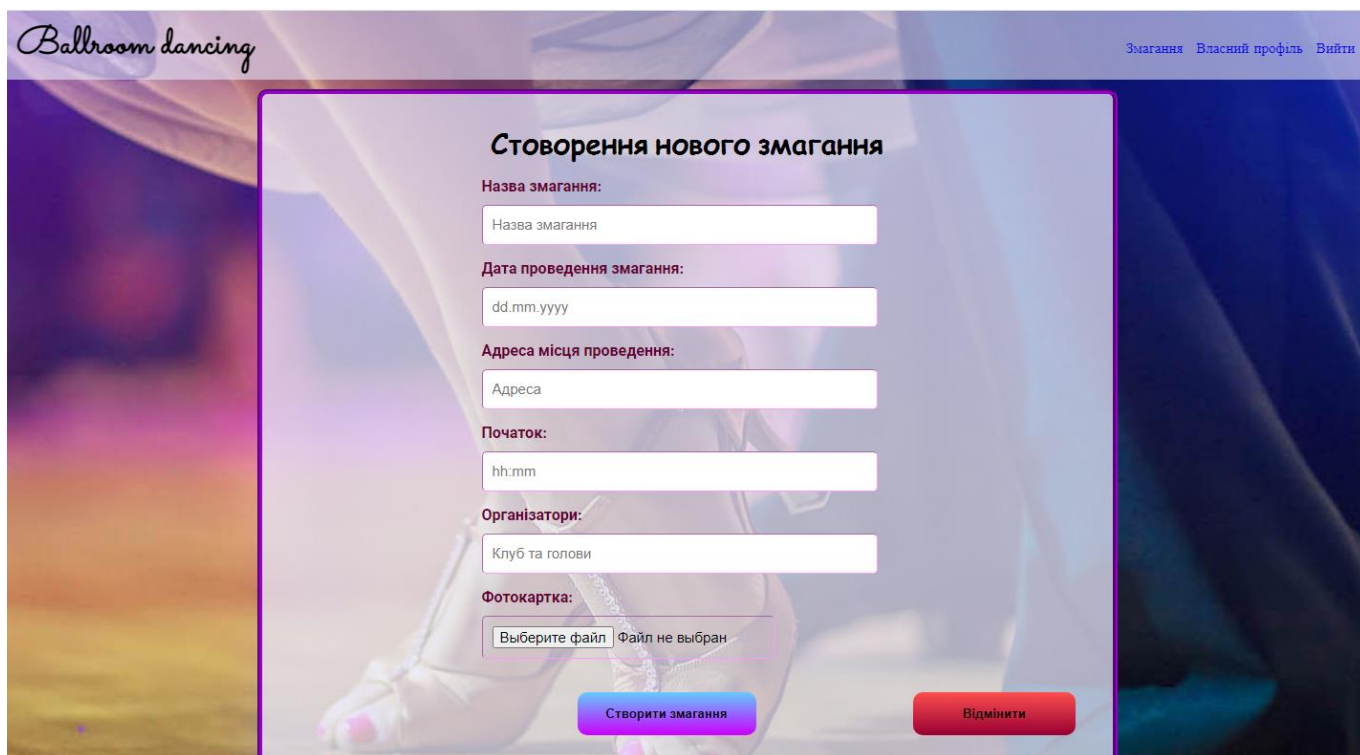
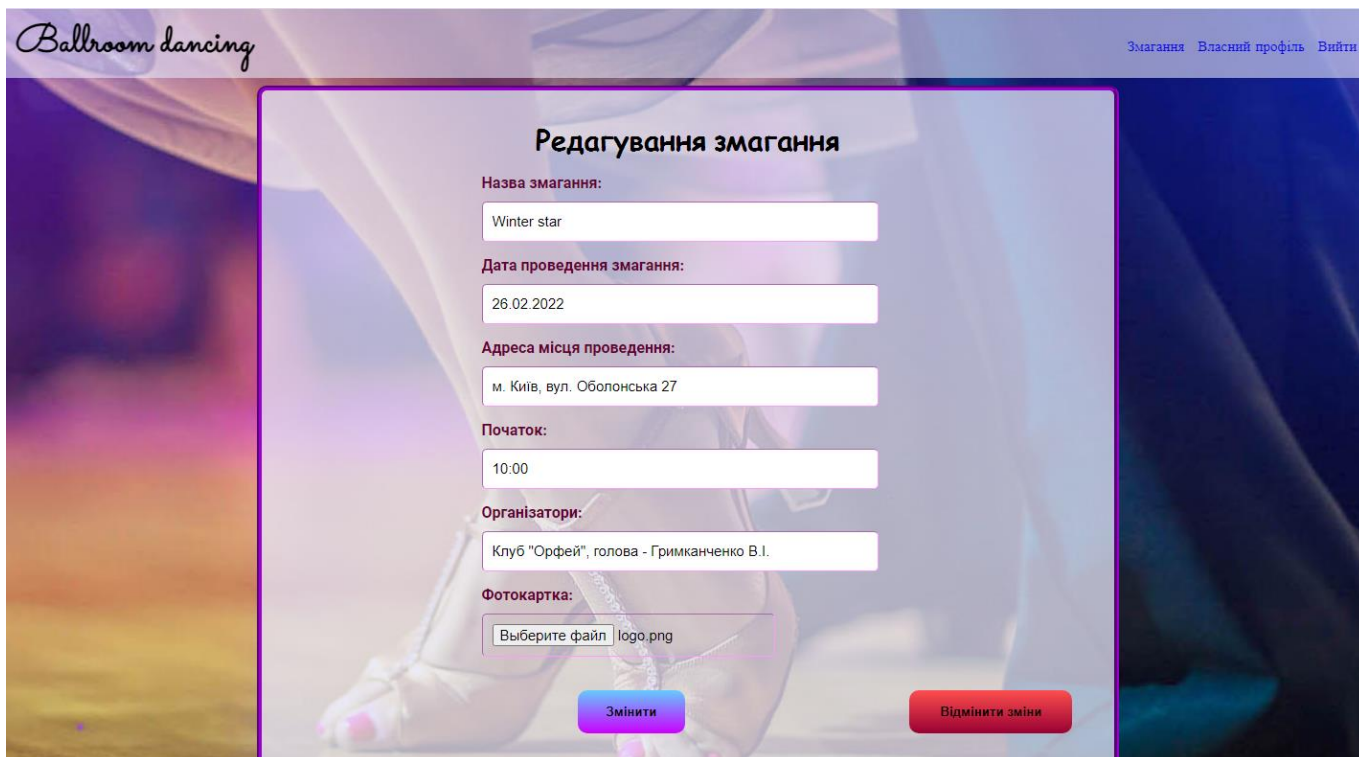


Рисунок 3.37 – Сторінка створення нового змагання

Звичайно адміністратор може редагувати занесені ним в систему змагання (рис.3.38), але тільки до того моменту, поки ніхто з користувачів-танцівників ще не встиг на них зареєструватися або користувачі-судді відправити резюме на суддівство даного змагання.



The screenshot shows a web interface for editing a competition. The page title is 'Ballroom dancing' and the navigation menu includes 'Змагання', 'Власний профіль', and 'Вийти'. The main heading is 'Редагування змагання'. The form contains the following fields:

- Назва змагання:** Winter star
- Дата проведення змагання:** 26.02.2022
- Адреса місця проведення:** м. Київ, вул. Оболонська 27
- Початок:** 10:00
- Організатори:** Клуб "Орфей", голова - Гримканченко В.І.
- Фотокартка:** Выберите файл logo.png

At the bottom of the form, there are two buttons: 'Змінити' (Save) and 'Відмінити зміни' (Cancel Changes).


Рисунок 3.38 – Сторінка редагування створеного змагання

На рисунку 3.39 представлено сторінку адміністратора з занесеними ним змаганнями в систему, біля стислої інформації по змаганням адміністратору дана можливість переглянути змагання або ж відредагувати. Якщо на змагання вже зареєструвалися, то він кнопка «Відредагувати» буде неактивною. Також створена пагінація, щоб користувач міг переходити по сторінкам та передивлятися всі змагання.

Ballroom dancing

Змагання Власний профіль Вийти

### Мій профіль – адмін




Глона Белім  
admin@gamil.com

Додати змагання

#### Майбутні змагання

##### Змагання "Winter star"




Дата проведення: 26. 02. 2022

Організатори: Клуб "Орфей", голова – Гримканченко В. І.

Переглянути Відредагувати

##### Змагання "Rudaga plus"



Дата проведення: 26. 11. 2022

Організатори: Клуб "Орфей", голова – Гримканченко В. І.

Переглянути Відредагувати

< 1 >

Рисунок 3.39 – Профіль адміністратора зі списком змагань

Після того, як адміністратор додав змагання в систему, воно відображається на сторінці «Змагання» (рис. 3.40), де вже танцівники та судді можуть переглянути




кожне змагання окремо та більш детально та зареєструватися чи податися, в якості судді.

*Ballroom dancing* Змагання Власний профіль Вийти

### Перелік змагань

**Змагання "Gold Autumn"**




Дата проведення: 25. 12. 2021

Організатори: Клуб "Smaft", голова – Перемішківська С. В.

Переглянути
Зареєструватися

**Змагання "Viva Winners"**




Дата проведення: 29. 12. 2021

Організатори: Клуб "New almond", голова – Стерепанченко С. В.

Переглянути
Зареєструватися

**Змагання "Winter star"**




Дата проведення: 26. 02. 2022

Організатори: Клуб "Орфей", голова – Гримканченко В. І.

Переглянути
Зареєструватися

**Змагання "Rudaga plus"**



Дата проведення: 26. 11. 2022

Організатори: Клуб "Орфей", голова – Гримканченко В. І.

Переглянути
Зареєструватися

< 1 >

Рисунок 3.40 – Сторінка зі змаганнями

Натиснувши на кнопку «Переглянути» користувачі потрапляють на сторінку, де відображається конкретне змагання та вся інформація по ньому (рис. 3.41). З цієї сторінки користувачі-танцівники можуть зареєструватися на змагання, а користувачі-судді відправити резюме.

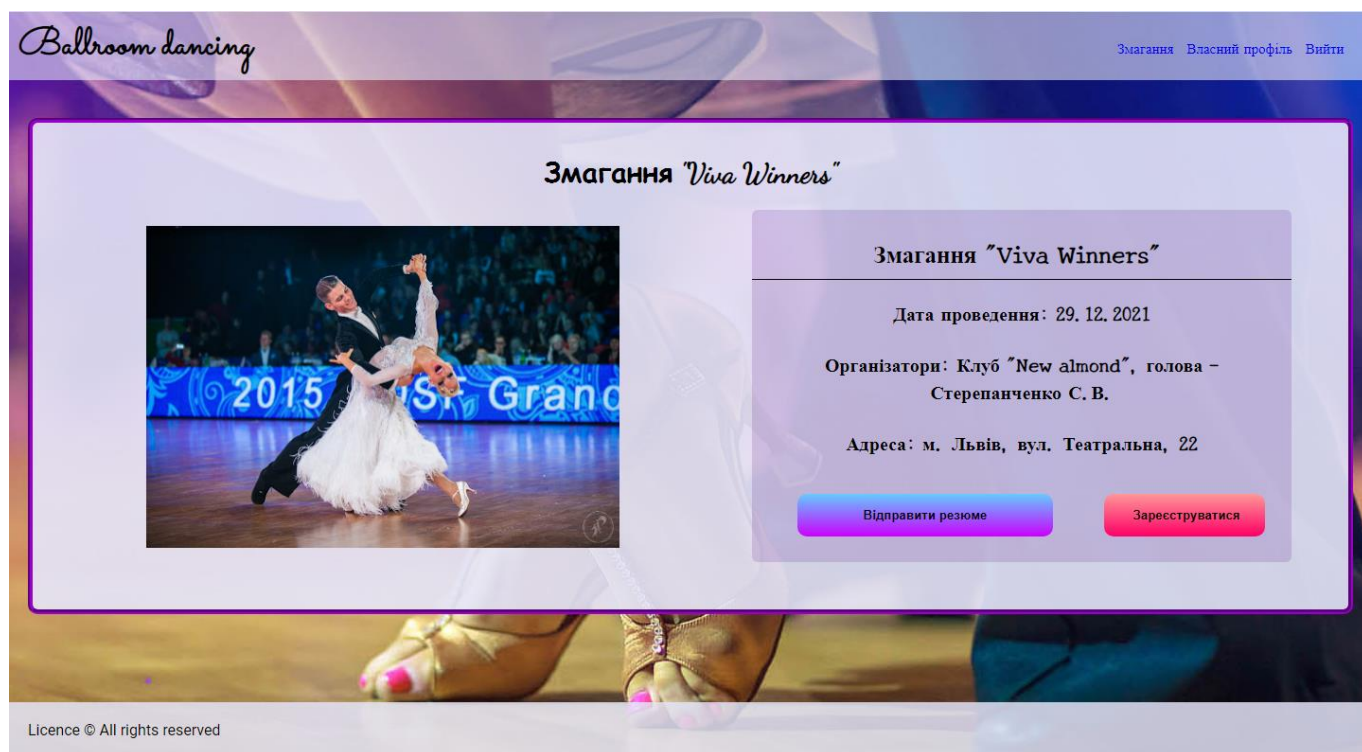
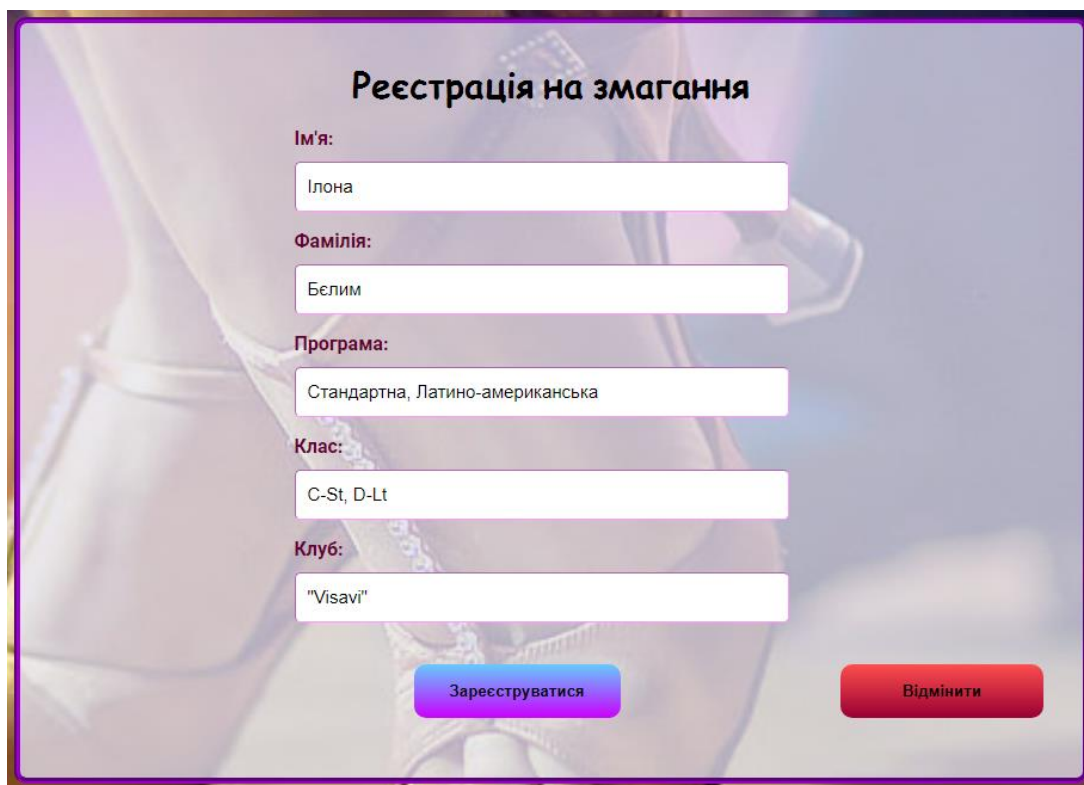


Рисунок 3.41 – Сторінка одного змагання

Якщо користувач-танцівник натискає на кнопку «Зареєструватися» під блоком з даними про змагання відкривається форма для реєстрації (рис. 3.42) із вже заповненими даними, які підтягнулися із даних власного профілю, але ці дані, які будуть відправлятися на реєстрацію можна відредагувати одразу.

Аналогічний процес проходить, якщо користувач-суддя натискає на кнопку «Відправити резюме», відкривається форма для заповнення та відправки резюме (рис. 3.43) із авто-заповненими даними з власного профілю, які також можна одразу тут відредагувати.



**Реєстрація на змагання**

Ім'я:

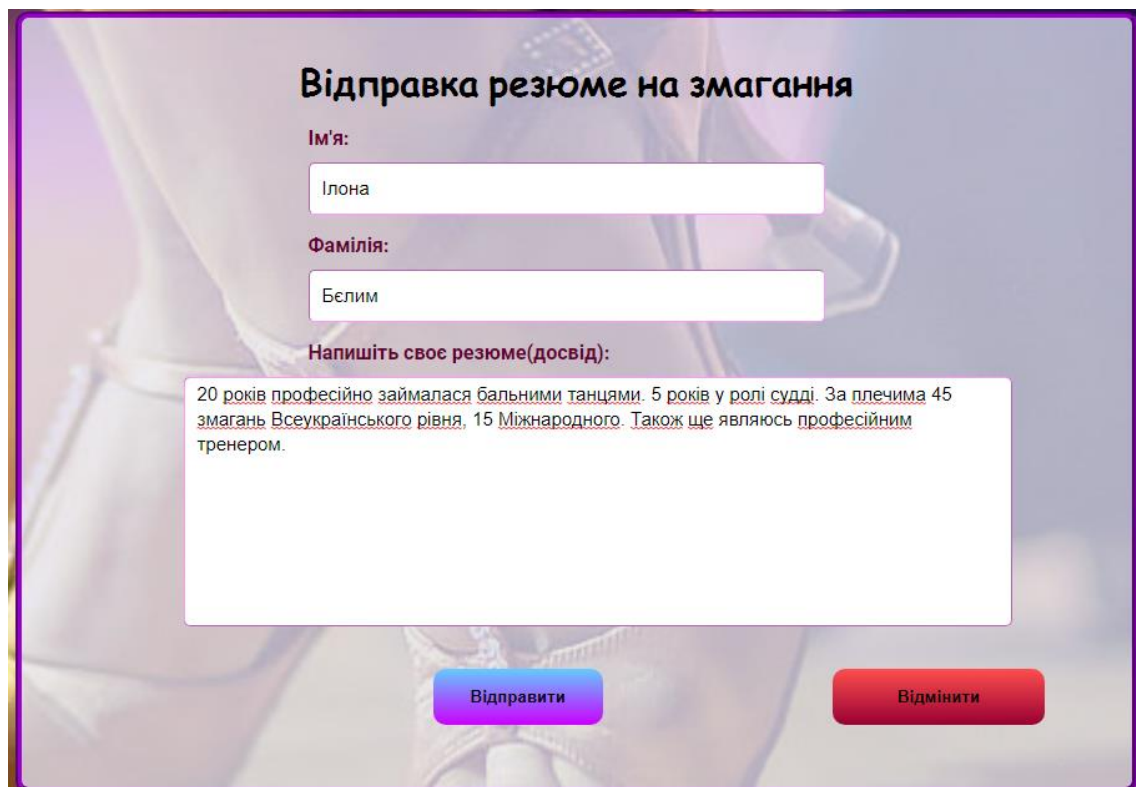
Фамілія:

Програма:

Клас:

Клуб:

Рисунок 3.42 – Форма реєстрації танцівника на змагання



**Відправка резюме на змагання**

Ім'я:

Фамілія:

Напишіть своє резюме(досвід):

Рисунок 3.43 – Форма відправки резюме на змагання

## ВИСНОВКИ

Під час розроблення даної кваліфікаційної роботи було зроблено глибокий аналіз та дослідження всіх моментів стосовно розробки WEB-орієнтованої CRM-системи управління змаганнями клубів спортивно-бальних танців.

Актуальність даної системи з кожним днем тільки збільшується, бо кількість клубів спортивно-бальних танців зростає, а кількість танцівників, які хочуть реалізувати себе, в цій сфері стрімко стрибає вгору. Реєстрація на змагання потребує багато часу, якщо це робити онлайн, «Танцювальні книги» зберігаються в паперовому вигляді, а дана система все це автоматизує.

У ході виконання даної роботи було розглянуто ряд таких питань:

- проведено глибокий аналіз продуктів-аналогів, виявлено їх недоліки та викреслено їх з даної системи, а також виокремлено всі переваги та додано нові можливості до реалізованої системи;
- розроблено чималу кількість діаграм для структурованого відображення архітектури системи та функцій, також створено діаграму варіантів-використання для відображення основних можливостей користувача в системі, для відображення зв'язків в БД розроблено ER-діаграму;
- створено сучасний та гнучкий UI/UX дизайн, який впроваджено в даний web-орієнтований додаток;
- проведено смоук, модульне та інтеграційне, UAT тестування для перевірки працездатності та адаптивності системи.

Отже, дана web-орієнтована CRM-система є неймовірною знахідкою для клубів, які створюють змагання, танцівників, які хочуть брати в них участь та вести свою «Танцювальну книгу» онлайн, а також для суддів бальних танців, які хочуть знати про актуальні майбутні змагання та виступати на них у ролі суддів.

## СПИСОК ЛІТЕРАТУРИ

1. Frain B. Responsive Web Design with HTML5 and CSS / B. Frain // Packt Publishing, 2020. - 330 с.
2. Coulson L., Jephson B., Larsen R., Park M., Zburlea M. The HTML and CSS Workshop / L. Coulson, B. Jephson, R. Larsen, M. Park, M. Zburlea // Packt Publishing, 2019. – 110, 552-557 с.
3. Google Fonts: [Електронний ресурс]. URL:<https://fonts.google.com/>
4. Bramer M. Web Programming with PHP and MySQL: A Practical Guide / M. Bramer // Springer, 2015. – 187-190 с.
5. Справочник по JavaScript: [Електронний ресурс]. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference>
6. Руководство по MySQL: [Електронний ресурс]. URL: <https://metanit.com/sql/mysql/>
7. Flanagan D. JavaScript: The Definitive Guide, 7th Edition / D. Flanagan // O'Reilly, 2020. – 155-170 с.
8. Bootstrap · The most popular HTML, CSS, and JS library in the world: [Електронний ресурс]. URL: <https://getbootstrap.com/>
9. CSS Tutorial - W3Schools: [Електронний ресурс]. URL: <https://www.w3schools.com/css/>
10. PHP, MySQL и другие веб-технологии: [Електронний ресурс]. URL: <http://www.php.su/>
11. jQuery Introduction - W3Schools: [Електронний ресурс]. URL: [https://www.w3schools.com/jquery/jquery\\_intro.asp](https://www.w3schools.com/jquery/jquery_intro.asp)
12. Никольский А.П. Javascript на примерах. Практика, практика и только практика / А.П. Никольский // Наука и Техника, 2018. – 187 с.



13. Бибо Б., Кац И., Роза А. jQuery в действии / Б. Бибо, И. Кац, А. Роза // Питер, 2017. – 77 с.
14. Браун Э. Изучаем JavaScript: руководство по созданию современных веб-сайтов / Э. Браун // ООО Альфа-книга, 2017. – 57-59 с.
15. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов / Д. Скляр // Альфа-книга, 2017. – 335-339 с.
16. Флэнаган Д. JavaScript Полное руководство / Д. Флэнаган // Диалектика, 2021. – 174-179 с.
17. Кузнецов М., Симдянов И. Самоучитель PHP 7 + код / М. Кузнецов, И. Симдянов // БХВ-Петербург, 2018. – 143-150 с.

## **ДОДАТКИ**

### **Додаток А**

#### **ТЕХНІЧНЕ ЗАВДАННЯ**

**на створення програмного додатку «WEB-орієнтована CRM-система  
управління змаганнями клубів спортивно-бальних танців»**

**Суми 2021**

## **1 Призначення й мета створення Web-орієнтованої системи**

### **1.1 Призначення**

Розроблюваний програмний продукт створений для реєстрації на змагання зі спортивно-бальних танців, створення цих змагань та збереження даних в «Танцювальну книгу».

### **1.2 Мета створення**

Надання можливості онлайн зареєструватися на змагання, перетворити «Танцювальну книгу» з паперової в електронну та мати доступ до неї з будь-якої точки світу.

### **1.3 Цільова аудиторія**

Основними об'єктами на кого орієнтована дана система є:

- Танцівники – реєструються онлайн на змагання зі спортивно-бальних танців, зберігають дані про минулі змагання в електронній танцювальній книзі;
- Судді спортивно-бальних танців – мають можливість переглянути майбутні змагання та відправити своє резюме для суддівства на змаганні;
- Адміністратор – має можливість створити змагання.

## **2 Вимоги до Web-орієнтованої системи**

### **2.1 Загальні вимоги до додатку**

Даний сервіс необхідно реалізувати, як web-сайт. Також він повинен містити в собі взаємопов'язані підсистеми розподілені по конкретним окремим функціям. Щодо зовнішнього вигляду, то UI/UX повинен бути зрозумілим будь-якому користувачу без додаткових зусиль, також кольори та розміри блоків повинні бути гармонійно підібрані. Даний сервіс повинен працювати в незалежності від операційної системи та девайсу.

### **2.2 Вимогу до функцій додатку**

Першою функцією даного сервісу є можливість реєстрації нового користувача, далі використовуючи дані під яким був зареєстрований – можливість авторизації в системі.

Також зареєстрований користувач має можливість переглянути свій власний профіль, відредагувати дані, доповнити додатковою інформацією. В залежності від того, як буде відредагований профіль, він перетвориться на профіль 2 типів – танцівник або суддя.

Для користувачів-танцівників реалізована можливість створення «Танцювальної книги» та занесення в неї результатів минулих змагань, які можна відредагувати.

Для користувачів-адміністраторів реалізований функціонал додавання нового змагання та його редагування, але тільки до того моменту, поки на нього ще не зареєструвалися танцівники або не відправили резюме судді.

Користувачі-танцівники та користувачі-судді мають можливість переглядати майбутні змагання та зареєструватися на них та відправити резюме на суддівство відповідно.

### ***2.3 Візуальне представлення додатку***

## 2.3.1 Відображення сторінки за всіма змаганнями в системі – головної сторінки

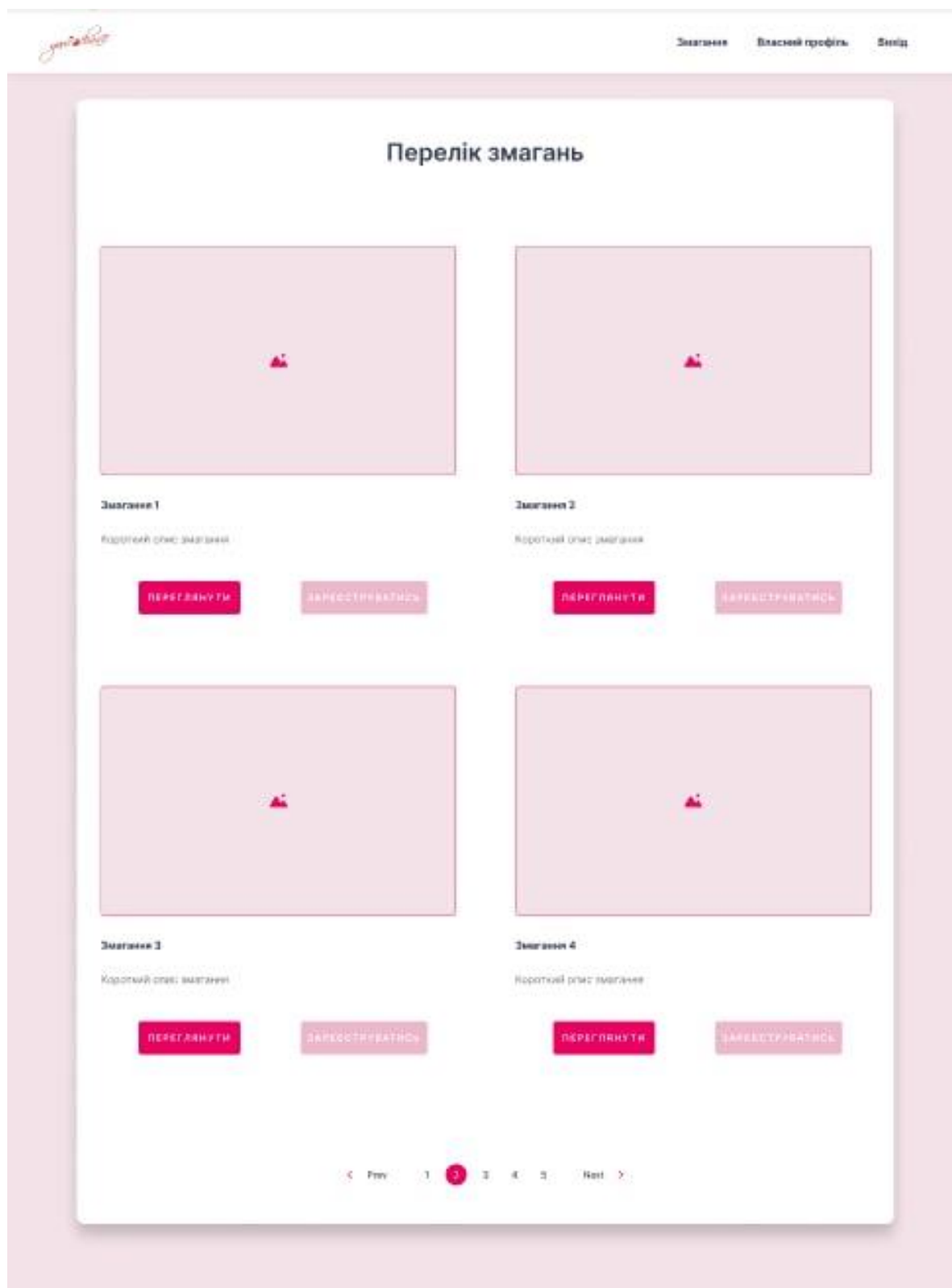
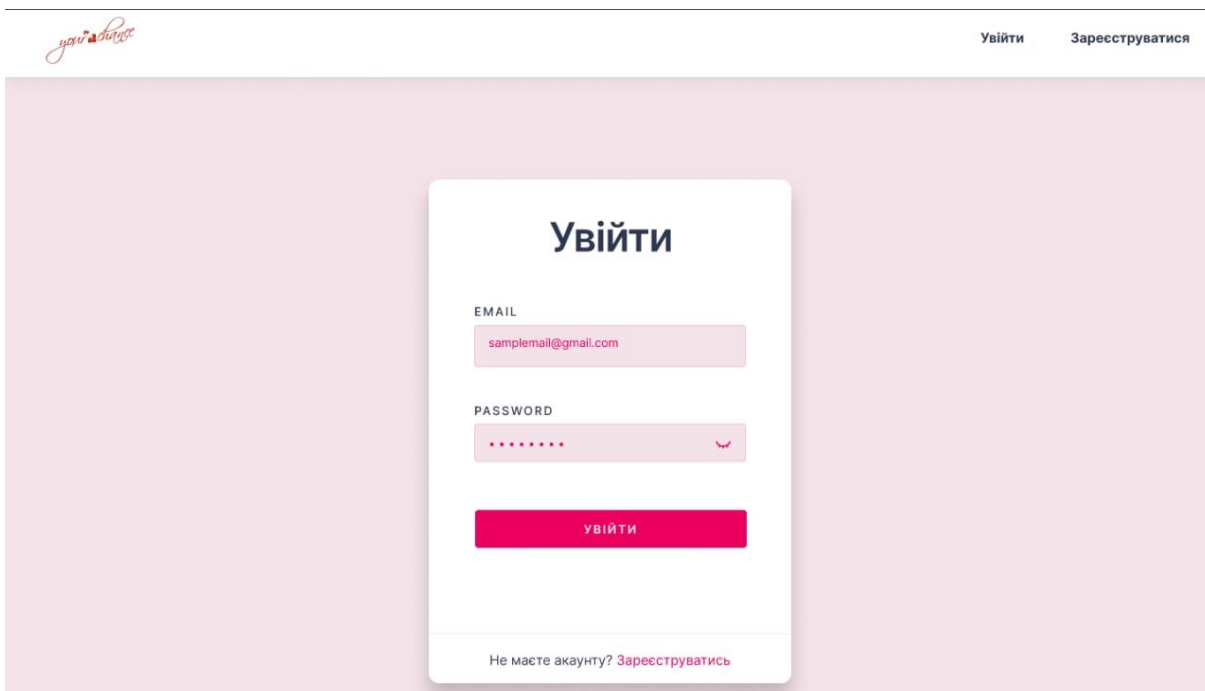


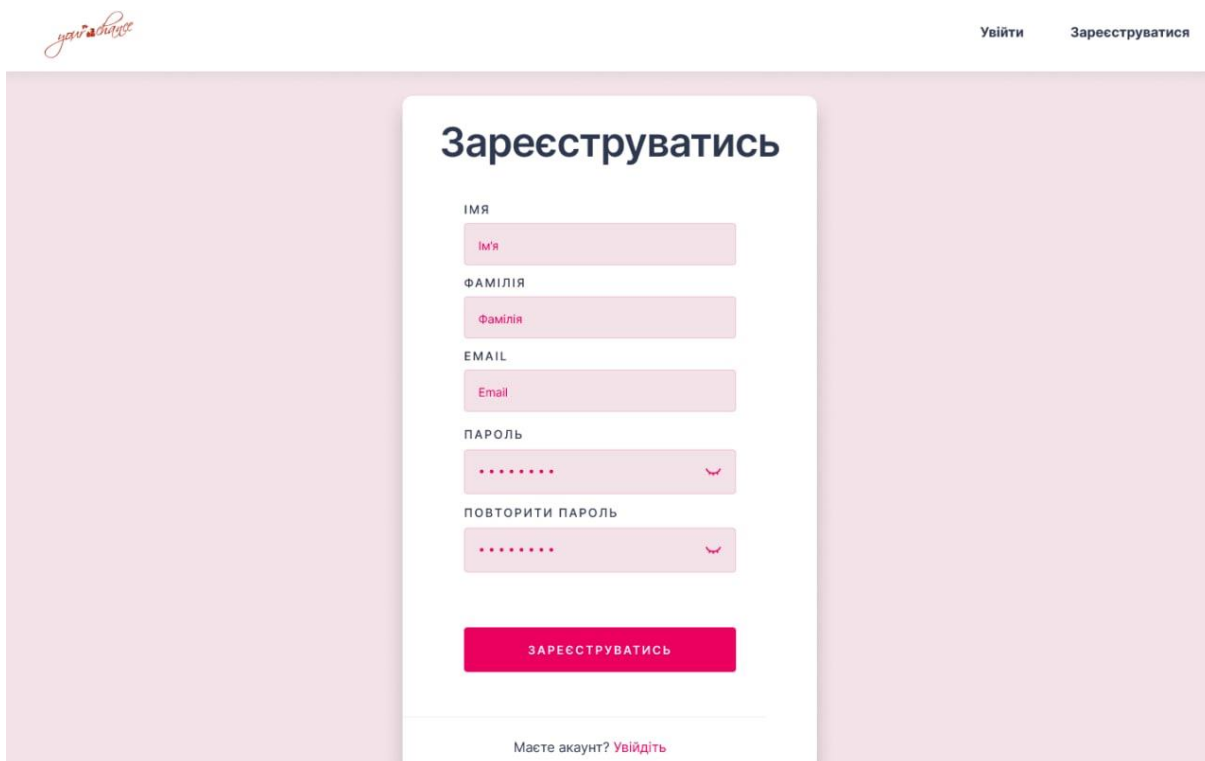
Рисунок 2.3.1.1 – Дизайн сторінки зі всіма змаганнями – головної сторінки

## 2.3.2 Відображення форми авторизації та реєстрації



The image shows a login form titled "Увійти" (Login) centered on a light pink background. At the top left is a logo with the text "улюблені" in a cursive font. At the top right are two links: "Увійти" and "Зареєструватися". The form itself is white and contains the following elements: a title "Увійти", an "EMAIL" label above a text input field containing "samplemail@gmail.com", a "PASSWORD" label above a password input field with seven dots and a visibility toggle icon, a large pink button labeled "УВІЙТИ", and a footer link "Не маєте акаунту? Зареєструватися".

Рисунок 2.3.2.1 – Дизайн сторінки входу



The image shows a registration form titled "Зареєструватися" (Register) centered on a light pink background. At the top left is a logo with the text "улюблені" in a cursive font. At the top right are two links: "Увійти" and "Зареєструватися". The form itself is white and contains the following elements: a title "Зареєструватися", an "ІМЯ" label above a text input field containing "Імя", a "ФАМІЛІЯ" label above a text input field containing "Фамілія", an "EMAIL" label above a text input field containing "Email", a "ПАРОЛЬ" label above a password input field with seven dots and a visibility toggle icon, a "ПОВТОРИТИ ПАРОЛЬ" label above a second password input field with seven dots and a visibility toggle icon, a large pink button labeled "ЗАРЕЄСТРУВАТИСЬ", and a footer link "Маєте акаунт? Увійдіть".

Рисунок 2.3.2.2 – Дизайн сторінки реєстрації

### 2.3.3 Сторінка з усіма змаганнями

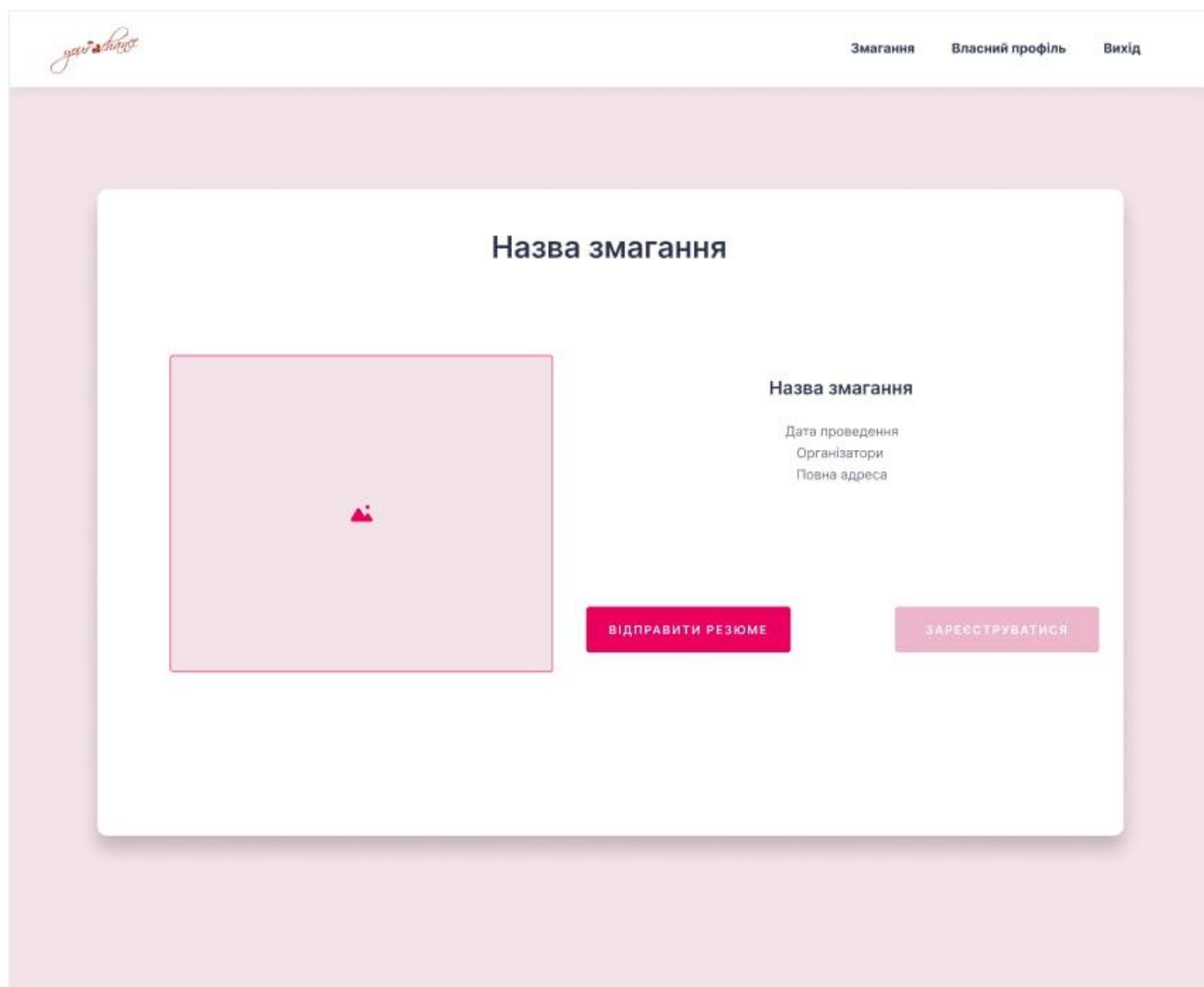


Рисунок 2.3.3.1 – Дизайн сторінки перегляду одного змагання

## 2.3.4 Відображення сторінки власного профілю користувача з роллю танцівник

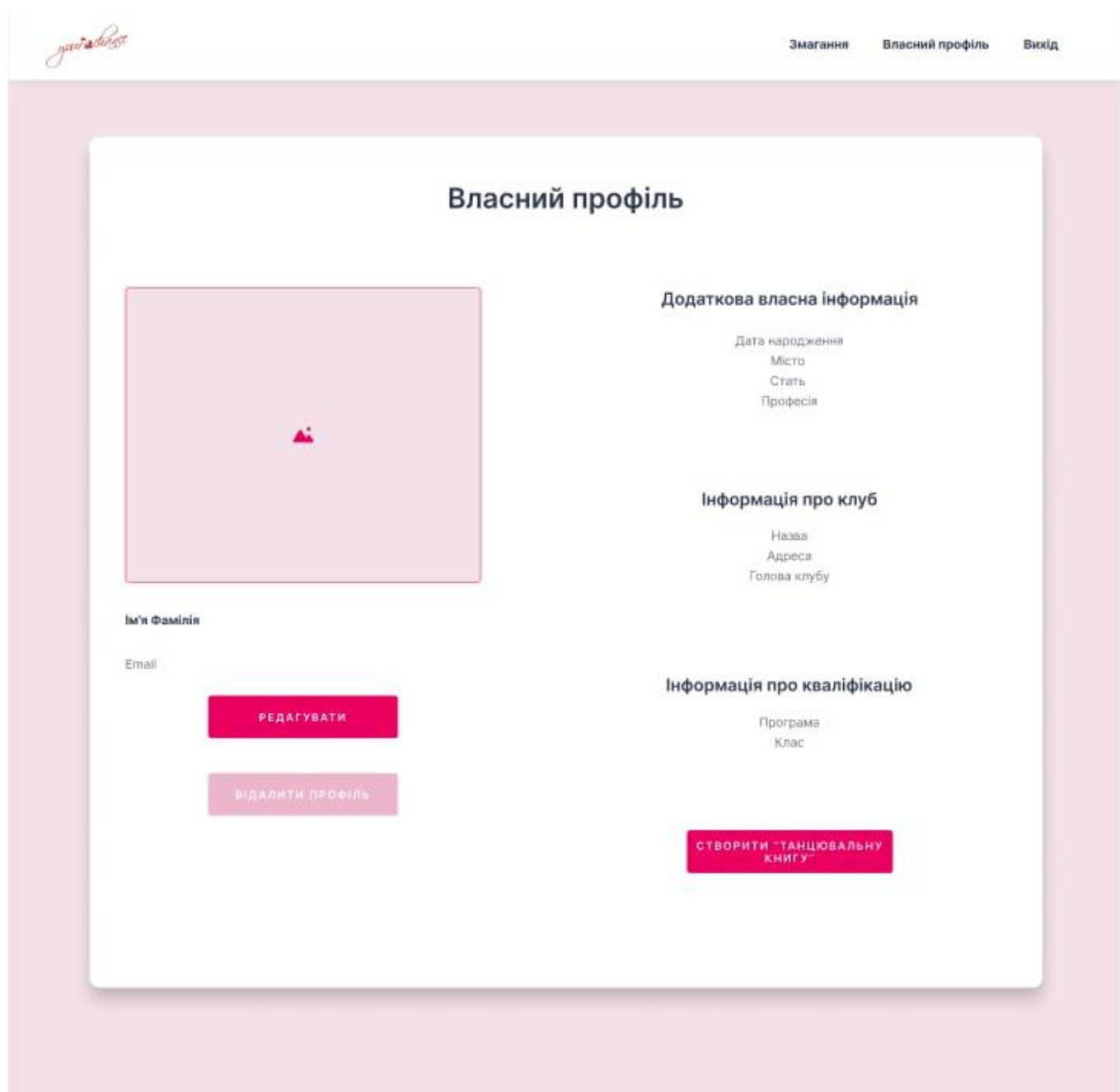


Рисунок 2.3.4.1 – Дизайн сторінки профілю користувача-танцівника



### 3 Склад і зміст робіт зі створення додатку

В таблиці 3.1 відображено, які роботи необхідно виконати та скільки часу вони займуть.

Таблиця 3.1 – Етапи та кроки розробки системи

№	Склад і зміст робіт	Строк розробки
1	Проектування: Створення структури БД та логічне проектування всієї системи.	5 днів
2	Розробка інтерфейсу: Підбір кольорової гами, верстання та стильове наповнення окремих сторінок.	7 дні
3	Програмування окремих функціональних модулів : <ul style="list-style-type: none"> <li>• Форми реєстрації та входу;</li> <li>• Форма редагування власного профілю;</li> <li>• Форма створення «Танцювальної книги» та додавання записів до неї;</li> <li>• Форма редагування записів «Танцювальної книги»;</li> <li>• Форма створення змагання;</li> <li>• Форма редагування змагання;</li> <li>• Форма перегляду всіх змагань;</li> <li>• Форма реєстрації на змагання;</li> <li>• Форма подання заявки на суддівство на змагання.</li> </ul>	2 дні 2 дні 1 день 2 дні 1 день 2 дні 2 дні 2 дні 1 день
4	Модульне та інтеграційне тестування. UAT тестування	6 днів
	<b>Весь затрачений час для робіт</b>	<b>33 днів</b>

## Додаток Б

### Лістинг програмного коду

SQL запит на створення БД:

```
Create table `users` (
  `user_id` Int NOT NULL,
  `name` Varchar(40) DEFAULT NULL,
  `surname` Char(40) DEFAULT NULL,
  `gender` Varchar(20) DEFAULT NULL,
  `email` Varchar(100) NOT NULL,
  `birthday` Date NOT NULL,
  `role` Int NOT NULL,
  `password` Varchar(100) NOT NULL,
  `location` Varchar(100) NOT NULL DEFAULT NULL,
  `user_photo` Varchar(100) DEFAULT NULL,
  Primary Key (`user_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
Create table `competitions` (
  `competition_id` Int NOT NULL,
  `user_id` Int NOT NULL,
  `comp_name` Varchar(100) NOT NULL,
  `comp_date` Date NOT NULL,
  `comp_address` Varchar(150) NOT NULL,
  `start_time` Varchar(50) NOT NULL,
  `comp_image` Varchar(100),
  `organizer` Varchar(100) NOT NULL,
  Primary Key (`competition_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
Create table `judgers` (
  `judge_id` Int NOT NULL,
  `user_id` Int NOT NULL,
  `description` Varchar(500),
  Primary Key (`judge_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
Create table `dancers` (
  `dancer_id` Int NOT NULL,
  `user_id` Int NOT NULL,
  `club_id` Int NOT NULL,
  `d_class` Varchar(20) NOT NULL,
  `program` Varchar(20) NOT NULL,
  Primary Key (`dancer_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```

Create table `clubs` (
  `club_id` Int NOT NULL,
  `club_name` Varchar(100) NOT NULL,
  `club_adress` Varchar(100) NOT NULL,
  `owner` Varchar(100) NOT NULL,
  Primary Key (`club_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `dancer_book` (
  `book_id` Int NOT NULL,
  `dancer_id` Int NOT NULL,
  `b_date` Date NOT NULL,
  Primary Key (`book_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `notes` (
  `note_id` Int NOT NULL,
  `book_id` Int NOT NULL,
  `competition_name` Varchar(100) NOT NULL,
  `n_class` Varchar(20) NOT NULL,
  `pairs_quantity` Int NOT NULL,
  `n_program` Varchar(20) NOT NULL,
  `c_date` Date NOT NULL,
  `place` Varchar(20) NOT NULL,
  Primary Key (`note_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `d_request` (
  `d_request_id` Int NOT NULL,
  `dancer_id` Int NOT NULL,
  `competition_id` Int NOT NULL,
  Primary Key (`d_request_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `j_request` (
  `j_request_id` Char(20) NOT NULL,
  `judge_id` Int NOT NULL,
  `competition_id` Int NOT NULL,
  Primary Key (`j_request_id`)) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Alter table `competitons`
add Foreign Key (`user_id`)
references `users` (`user_id`) on delete restrict on update restrict;

```

```

Alter table `judgers`
add Foreign Key (`user_id`)
references `users` (`user_id`) on delete restrict on update restrict;

```

```

Alter table `dancers`
add Foreign Key (`user_id`)
references `users` (`user_id`) on delete restrict on update restrict;

```

```

Alter table `d_request`
add Foreign Key (`competition_id`)
references `competitions` (`competition_id`) on delete restrict on update restrict;
Alter table `j_request`
add Foreign Key (`competition_id`)
references `competitions` (`competition_id`) on delete restrict on update restrict;

Alter table `j_request`
add Foreign Key (`judge_id`)
references `judgers` (`judge_id`) on delete restrict on update restrict;

Alter table `d_request`
add Foreign Key (`dancer_id`)
references `dancers` (`dancer_id`) on delete restrict on update restrict;

Alter table `dancer_book`
add Foreign Key (`dancer_id`)
references `dancers` (`dancer_id`) on delete restrict on update restrict;

Alter table `dancers`
add Foreign Key (`club_id`)
references `clubs` (`club_id`) on delete restrict on update restrict;

Alter table `notes`
add Foreign Key (`book_id`)
references `dancer_book` (`book_id`) on delete restrict on update restrict;

COMMIT;

```

ЛІСТИНГ КОДУ СТОРІНКИ ВСІХ ЗМАГАНЬ:

```

<?php
session_start();
require 'scripts/connect.php';
if (isset($_GET["is_exit"])) {
    if ($_GET["is_exit"] == 1) {
        unset($_SESSION['user_name']);
        session_destroy();
        header("Location: ?is_exit=0");
    }
}
}?>
<!DOCTYPE html>
<html lang="ua" xmlns:lang="ua" xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="UTF-8">
    <title>All about ballroom dancing</title>
    <meta name="description" content="Ballroom dancing"/>
    <meta property="og:title" content="Ballroom competitions"/>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

```

```

<link rel='stylesheet prefetch' href='http://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-
awesome.min.css'>
  <link rel='stylesheet prefetch'
href='http://fonts.googleapis.com/css?family=Roboto:400,100,300,500,700,900&subset=latin,latin-ext'>
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Dancing+Script&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Sacramento&display=swap"
rel="stylesheet">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=New+Tegomin&display=swap"
rel="stylesheet">
  <link href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" rel="stylesheet">

<link rel="stylesheet" href="css/all-competitions.css">
</head>
  <header>
    <ul>
      <li> <a class = "h1">Ballroom dancing</a></li>
      <li>
        <a class= "nav-menu" href="all-competition.php">Змагання</a>
      </li>
      <li>
        <a class= "nav-menu" href="profile.php">Власний профіль</a>
      </li>
      <li>
        <a class= "nav-menu" href="singin.php">Вийти</a>
      </li>
      <?php
        if (isset($_SESSION['user_name'])) {
          echo '
<li class="nav-item">
  <a class="nav-link" href="admin-panel.php?user_id='.$_SESSION['user_id'].'"><span>Адмін-
панель</span></a>
</li>
<li class="nav-item">
  <a class="nav-link" href="?is_exit=1"><span>Вихід</span> <i class="fas fa-sign-out-
alt"></i></a>
</li>
          ';
        }
        else {
          echo '
<li class="nav-item">
  <a class="nav-link" href="login.php"><span>Вхід</span> <i class="fas fa-sign-in-alt"></i></a>
</li>';
        }
      ?>
    </ul>
  </header>
<body>

```

```

<section>
  <section action="" name = "ordering" method = "GET">
    <div>
      <h2>Перелік змагань</h2>
      <div id="menu">
        <div class = "comp">
          <p class = "inform1">Змагання "Gold Autumn"</p>
          </img>
          <p class = "inform">Дата проведення: 25.12.2021</p>
          <p class = "inform">Організатори: Клуб "Smaft", голова -
Перемішковська С.В.</p>
          <button class = "view" type="button" >Переглянути</button>
          <button class = "sendCV" type="submit">Зареєструватися</button>
        </div>

        <div class = "comp">
          <p class = "inform1">Змагання "Winter star"</p>
          </img>
          <p class = "inform">Дата проведення: 26.02.2022</p>
          <p class = "inform">Організатори: Клуб "Орфей", голова -
Гримканченко В.І.</p>
          <button class = "view" type="button" >Переглянути</button>
          <button class = "sendCV" type="submit">Зареєструватися</button>
        </div>
      </div>

      <div id="content">
        <div class = "comp">
          <p class = "inform1">Змагання "Viva Winners"</p>
          </img>
          <p class = "inform">Дата проведення: 29.12.2021</p>
          <p class = "inform">Організатори: Клуб "New almond", голова -
Стерепащенко С.В.</p>
          <button class = "view" type="button" >Переглянути</button>
          <button class = "sendCV" type="submit">Зареєструватися</button>
        </div>

        <div class = "comp">
          <p class = "inform1">Змагання "Rudaga plus"</p>
          </img>
          <p class = "inform">Дата проведення: 26.11.2022</p>

```

```

        <p class = "inform">Організатори: Клуб "Орфей", голова -
Гримканченко В.І.</p>
        <button class = "view" type="button" >Переглянути</button>
        <button class = "sendCV" type="submit">Зареєструватися</button>
        </div>
    </div>
</div>
<?php
    $sql_note = mysql_query("SELECT `comp_name`, `comp_date`, `comp_address`,
`start_time`, `comp_image` from `competitions`
        Order by `competition_id` DESC Limit 10");
    while ($result_note = mysql_fetch_array($sql_note)) {
        echo '
        <div class="col-xl-8 col-lg-10 col-md-10 col-sm-11">
        <ul class="list-unstyled">
        <li class="media">
            
            <div class="media-body">
                <h5 class="mt-0 mb-1"><i class="fas fa-heading"></i> <a href="some-
note.php?note_id=.$result_note["note_id"].">'.$result_note["title"].'</a></h5>
                <hr/>
                <div class="d-flex justify-content-between">
                    <p><i class="fas fa-competition"></i> <a
href="competitions.php?comp_id=.$result_note["comp_id"].">'.$result_note["comp_name"].'
'.$result_note["comp_date"].'</a> </p>
                                                                <p><i class="fas
fa-competition"></i> <a
href="competitions.php?comp_id=.$result_note["comp_id"].">'.$result_note["comp_name"].'
'.$result_note["comp_date"].'</a> </p>
                                                                <p><i class="fas
fa-competition"></i> <a
href="competitions.php?comp_id=.$result_note["comp_id"].">'.$result_note["comp_name"].'
'.$result_note["comp_date"].'</a> </p>
                <p><i class="fas fa-competition"></i> <a
href="competitions.php?comp_id=.$result_note["comp_id"].">'.$result_note["comp_name"].'
'.$result_note["comp_date"].'</a> </p>
                </div>
            </div>
        </li>
        </ul>
        </div>';
    }
?>
</section>
</section>
<script src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>
<script src="js/index.js"></script>
</body>
<footer>
    <div class="footer"> <p>Licence © All rights reserved</p></div>
</footer>
</html>

```