

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Центр заочної, дистанційної та вечірньої форми навчання  
Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОЕКТУВАННЯ ПЛАТФОРМИ ДЛЯ  
КОРИСТУВАЧІВ КЛІЄНТ-ОРІЄНТОВАНОЇ АВТОМАТИЗОВАНОЇ  
СИСТЕМИ**

Здобувач освіти гр. ІН.мз-11с

Ілона БЄЛИМ

Науковий керівник,  
доцент, к.т.н.

Надія ТИРКУSOBA

В.о. Завідувача кафедри  
доцент, к.т.н.

Ігор ШЕЛЕХОВ

Суми 2022

# Сумський державний університет

(назва вузу)

Факультет ЕЛІТ Кафедра Комп'ютерних наук

Спеціальність 122 - Комп'ютерні науки

Затверджую:  
зав. кафедри \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Белим Ілоні Костянтинівни

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна технологія проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи

затверджую наказом по інституту від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Термін здачі студентом закінченого проекту (роботи) \_\_\_\_\_

3. Вхідні дані до проекту (роботи) \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, дослідження на актуальність системи, проектування та моделювання інформаційної технології проектування, розробка інформаційної технології проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність проекту, порівняння та аналіз сайтів-аналогів (переваги та недоліки), порівняння всіх можливих схожих та подібних рішень, функціональні та нефункціональні вимоги до системи, методи та технології розробки, опис бізнес-процесів системи у вигляді IDEF0, детальний опис бізнес-процесів з розбиттям на підфункції у вигляді IDEF1, Use-Case діаграма, схематична архітектура додатку, ER діаграма бази даних, створений дизайн додатку, висновки.

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_  
(підпис)

Завдання прийняв до виконання

\_\_\_\_\_  
(підпис)**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	Аналіз та розподіл запланованих робіт	До 15.09.2022	
2.	Написання технічного завдання	До 25.09.2022	
3.	Аналіз предметної області	До 30.09.2022	
4.	Проектування та моделювання інформаційної технології проектування	До 15.10.2022	
5.	Розробка автоматизованої системи	До 20.11.2022	
6.	Тестування інформаційної технології проектування	До 30.11.2022	
7.	Здача пояснювальної записки та файлів розробленого проекту	До 06.12.2022	

Студент – дипломник

\_\_\_\_\_  
(підпис)

Керівник проекту

\_\_\_\_\_  
(підпис)

## РЕФЕРАТ

Тема даної кваліфікаційної роботи магістра «Інформаційна технологія проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи».

Дана описова робота, тобто пояснювальна записка складається з таких блоків: вступ, 3 основних розділи – Аналіз предметної області, Моделювання та проектування, Розробка інформаційної технології, список використаних джерел та додатків.

Загальний обсяг роботи складає – 76 сторінок, що включають в себе 58 сторінок тексту основних розділів, 1 сторінка використаних джерел у роботі, яка складається з 15 пунктів, а також 16 сторінок додатків.

Під час розробки даної інформаційної технології виокремлено та дорозроблено такі частини:

- Форма редагування власного профілю відповідно до нових ролей;
- Форма планування онлайн уроків та додавання їх до системи;
- Форма редагування запланованих онлайн занять;
- Форма додавання відео-заняття;
- Форма редагування доданого відео-заняття;
- Форма перегляду всіх майбутніх онлайн занять;
- Форма перегляду всіх записаних онлайн занять(відео-уроків);
- Форма реєстрації на онлайн заняття;

Отже, вихідним результатом даної роботи є Інформаційна технологія проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи.

Ключові слова: інформаційна технологія, клієнт-орієнтована система, навчальна платформа, спортивно-бальні танці, онлайн навчання, онлайн заняття/уроки.

## ЗМІСТ

<b>ВСТУП</b> .....	6
<b>1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	7
<b>1.1 Огляд останніх досліджень та публікацій</b> .....	7
<b>1.2 Аналіз програмних продуктів – аналогів</b> .....	10
<b>1.3 Постановка задачі</b> .....	13
<b>2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ</b> .....	15
<b>2.1 Проектування інформаційної технології проектування</b> .....	15
<b>2.2 Проектування моделі бази даних</b> .....	18
<b>3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ</b> .....	22
<b>3.1 Архітектура програмного додатку</b> .....	22
<b>3.2 Програмна реалізація</b> .....	24
<b>3.3 Використання програмного продукту</b> .....	36
<b>ВИСНОВКИ</b> .....	58
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	59
<b>ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ</b> .....	60
<b>ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ</b> .....	68

## ВСТУП

Сьогодні кидає виклик кожному з нас, змушує змінюватись та пристосовуватись до різних умов життя. Карантин свого часу зробив свої корективи в повсякденне життя людей, перевівши майже всі аспекти життя в онлайн формат. Це довело всьому світу, що не важливо де ти знаходишся, якщо ти маєш Інтернет та гаджет ти можеш вільно навчатися, освоювати нові види діяльності, працювати, здійснювати покупки та займатися тренуванням з власним тренером у себе вдома використовуючи відео-зв'язок.

Великим потрясінням стала війна в Україні, безсумнівно це вплинуло на кожного з нас та на нашу культуру. Тому нашою головною задачею є бути сильними, стійкими та займатися тим, що може прославляти нашу країну та підтримувати її економіку. І не важливо якою сферою ти займаєшся, будь ти військовим, який захищає нашу землю, чи програміст, котрий розробляє додаток для телефона, чи професійний танцівник, який представляє Україну на міжнародній арені, все це допомагає країні розвиватися та писати свою неповторну історію.

Культура спорту завжди була і буде важливою, тому є дуже актуальним питанням пристосування її до сьогоденних реалій, щоб кожен спортсмен мав змогу продовжувати свою діяльність задля спадщини країни. Тому, клієнт-орієнтована автоматизована система дасть змогу кожному охочому здійснювати просту реєстрацію безпосередньо на сайті, переглядати перелік змагань, брати участь у заходах, проводити та відвідувати тренування в онлайн режимі, переглядати відео уроки, стати професійним викладачем, все що для цього необхідно це Інтернет та гаджет. Використання автоматизованої системи значно спростить життя як для танцівників, які палають бажанням продовжувати навчатися танцям, так і викладачів, які будуть проводити онлайн заняття, як і організаторів заходів, так і для учасників, представивши все необхідне в одній системі. Даний проект допоможе процвітанню спорту, як культури, задля того щоб вона не зникла з сторінок історії.

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень та публікацій

Світ змінюється, зовнішні фактори накладають значний відбиток в тенденції розвитку, та популярність до спортивно-бальних танців росте з кожним роком, охочих освоїти цей вид мистецтва збільшується з геометричною прогресією. Звісно, що це зумовило відкриття великої кількості клубів в різних містах України, та за її межами. Вони поповнюються новими перспективними та жагучими до перемоги танцівниками, котрі хочуть відобразити свої почуття та майстерність на паркеті. Щоб навчатися у кваліфікованих викладачів потрібно їздити у інші міста, а іноді й країни. Щоб взяти участь у змаганні необхідно виконати складний та довгий процес реєстрації, який вимагає очну присутність безпосередньо на місці проведення змагання. Та різні умови вносять корективи в життя людей, відбувається значна міграція людей по світу, що підштовхує до тенденцій переходу до онлайн платформ, котрі не мають привязки до географічного місцезнаходження.

Кожен професійний танцівник має свою «Танцювальну книгу» - документ, котрий зберігає інформацію по змаганням та результатами. При прибуттю на змагання в цей паспорт вносяться дані про сам захід, організаторів та класи, участь в яких приймаються, а вже після закінчення – кількість балів та результат, котрий необхідний для переходу до вищого танцювального класу.

В час цифровізації та розвитку інформаційних технології, всі паперові носії інформації поступово переходять на електронні, а очні заняття в онлайн, з використанням Інтернету. Тому є великим проривом створення системи, яка буде містити в собі можливість навчатися онлайн, переглядати відео-уроки у будь-який час та у будь-якому місці, ставати кваліфікованими викладачами, мати «Танцювальну книгу» професійного танцівника, можливість реєстрації та огляду змагань,

збереження та відображення всіх результатів, проведення онлайн тренувань та заходів.

Отже, проектом, котрий охоплює перелічені вище пункти буде розроблено програмний продукт - Інформаційна технологія проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи.

Дана система буде містити низку переваг та мати значний внесок в вирішенні багатьох аспектів, наприклад:

- Беручи до уваги соціальний аспект то завдяки продукту буде збільшена кількість танцівників, можливість навчатися онлайн з будь-якої точки світу, спрощений механізм реєстрацію на змагання, проведення та участь в онлайн танцювальних заходах.

- Для технічного аспекту дана система надає доступ до даних незважаючи на час та місце знаходження, та використання будь-якої платформи чи девайсу.

- Що стосується комерційного аспекту то даний продукт буде містити дані та актуальну інформацію всіх подій, як для викладачів, так і для студентів, як для учасників, так і для організаторів.

- Цінність рішення є в тому, що він буде незамінною системою для професійних танцівників, котрі хочуть взяти участь у змаганнях, або ж продовжити своє навчання онлайн, або стати кваліфікованим викладачем, з можливістю проведення всіх заходів онлайн, незважаючи на умови та географічне розташування.

Було проведено детальний аналіз розвитку спортивних-бальних танців в Україні за останні 5 років та як на цей розвиток вплинула пандемія та війна. Всі дані для наглядного вигляду було зібрано в таблицю 1.1. Дана таблиця відображає основні показники не тільки по Україні, а й в місті Суми. Тенденція розвитку була дуже стрімка, навіть за часів пандемії можна побачити, що розвиток продовжувався, затихав тільки на декілька місяців, коли був повний локдаун. Великого занепаду зазнав розвиток у цьому році, так як велика кількість людей виїхали за кордон, а для розвитку в танцях вони були прив'язані до місця. Тому дана система дасть можливість



продовжувати навчання з будь-якої точки світу та клуби спортивно-бальних танців будуть відроджуватися. [5, 6]

Таблиця 1.1 – Показники розвитку спортивно-бальних танців за останні 5 років

<b>Показники/Роки</b>	<b>2018</b>	<b>2019</b>	<b>2020</b>	<b>2021</b>	<b>2022</b>
Кількість клубів спортивно-бальних танців в Україні	87	101	128	140	99
Кількість клубів спортивно-бальних танців в Сумах	6	7	8	8	6
Кількість проведених змагань в Україні	121	156	95	139	37
Кількість проведених змагань в Сумах	10	12	16	17	4
Кількість відмінених змагань через пандемію/війну	-	-	13	7	77
Середня кількість танцівників в Україні	4350	5050	6400	7000	2970
Середня кількість танцівників в Сумах	300	350	450	500	200

## 1.2 Аналіз програмних продуктів – аналогів

Так як розроблювана інформаційна технологія проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи є частною клієнт-орієнтованої системи, то вона була порівняна по частинам з різними подібними сервісами, так щось схожого на разі не існує.

Раніше було зроблено порівняння з системами, які давали можливість співпрацювати зі змаганнями - web-сайти [audsf.com.ua](http://audsf.com.ua) та [flymark.com.ua](http://flymark.com.ua). На цей раз було проведено порівняння з онлайн системи, які дають можливість онлайн навчатися бальним танцям, тож було знайдено 2 віддалених аналогів - [super.prof.com.ua](http://super.prof.com.ua)(рис.1.1) та [buki.com.ua](http://buki.com.ua)(рис.1.2). Обидва рішення дуже подібні між собою та мають скудний функціонал.

Web-сайт [super.prof.com.ua](http://super.prof.com.ua) – це сервіс, який дає 2 опції для користувачів: зареєструватися як викладач та як студент, який буде навчатися. На платформі можна просто обрати одного з викладачів-репетиторів, які будуть проводити конкретно з одним студентом індивідуальні заняття. Не має можливості переглянути відео-уроки різних викладачів та можна записатися на заплановані онлайн заняття. Дизайн простий, але заважає влаштована реклама. Й звичайно ж весь функціонал який пов'язаний з змаганнями та «Танцювальною книгою» в даному сервісі відсутній.

Web-сайт [buki.com.ua](http://buki.com.ua) – сервіс подібний до попереднього, дає можливість бути викладцем та студентом, можна записатися на індивідуальні онлайн заняття, хоча для цього потрібно буде попрацювати з не дуже зручним UX дизайном. Сам сервіс більш орієнтований на шкільне навчання та підготовку до ЗНО та інших іспитів, тому ни про які змагання та «Танцювальні книги» мова не йде. Й звичайно ж переповнений різною влаштованою рекламою, яка заважає.

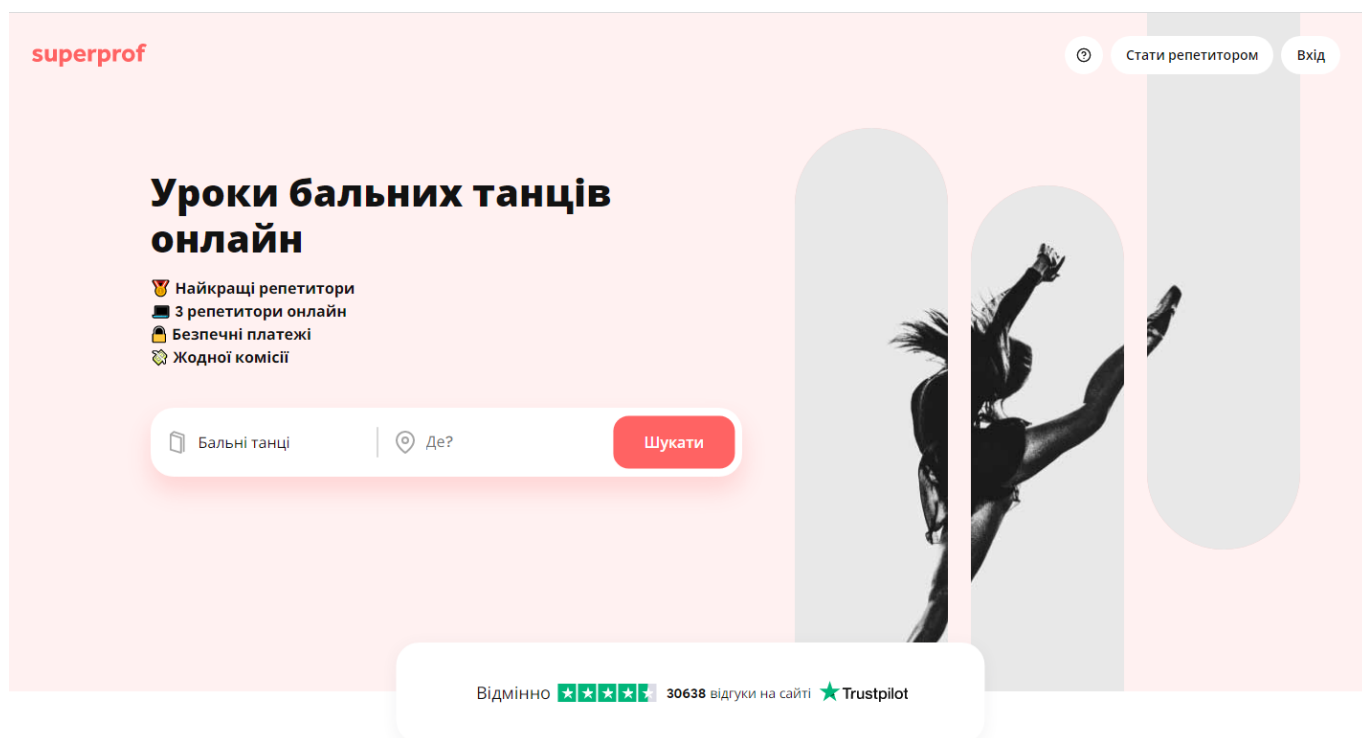


Рисунок 1.1 – Головна сторінка сайту super.prof.com.ua

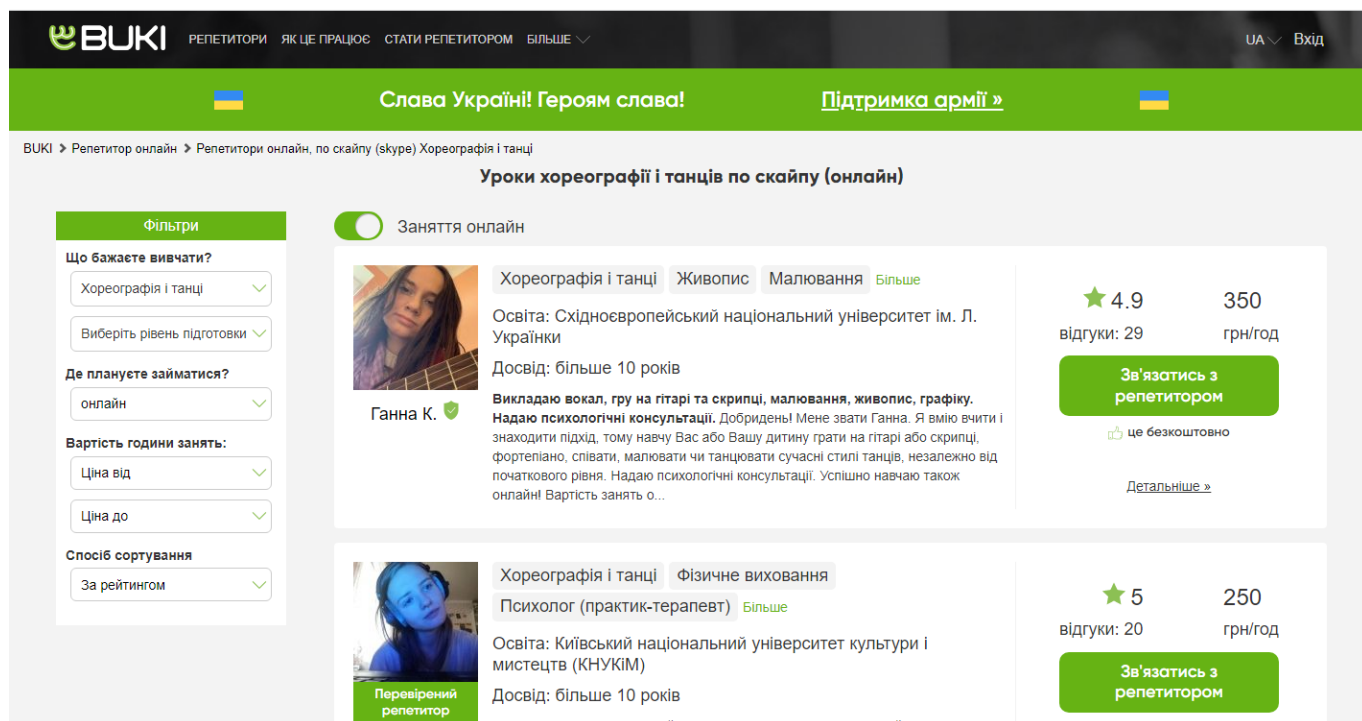


Рисунок 1.2 – Головна сторінка сайту buki.com.ua

Для детального, структурованого та чіткого розуміння можливостей майбутньої системи та сайтів-аналогів було сформовано порівняльну таблицю 1.2. Вона включає в себе сервіси-аналоги, які дають можливість зареєструватися на змаганні та переглядати результати минулих змагань в танцювальній книзі, а також новий функціонал – можливість онлайн навчання.

Таблиця 1.2 – Порівняльна таблиця основного функціонування сервісів-аналогів

<b>Характеристики</b>	<b>audsf.com.ua</b>	<b>flymark.com.ua</b>	<b>super.prof.com.ua</b>	<b>buki.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»	-	-	-	-	+
Наявність відео-уроків	-	-	-	-	+
Наявність запису на онлайн урок	-	-	+	+	+

Таблиця 1.2 – продовження порівняльної таблиці основного функціонування сервісів-аналогів

Характеристики	audsf.com.ua	flymark.com.ua	super.prof.com.ua	buki.com.ua	Створювана система
Можливість стати учнем	-	-	+	+	+
Можливість стати викладачем	-	-	+	+	+

### 1.3 Постановка задачі

Головною метою написання даної роботи є інформаційної технології проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи на базі вже створеної CRM-системи для управління змаганнями. Основними функціями даної платформи є:

- можливість навчатися у будь-який час та з будь-якої точки світу;
- можливість навчатися у реальному часі через онлайн уроки;
- можливість стати вчителем та проводити заняття;
- можливість мати доступ з будь-якої платформи та будь-якого девайсу;
- можливість доступу до платформи без встановлення будь-яких додатків,

а просто за допомогою браузера та інтернету.

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

- збір даних щодо актуальності розроблюваної системи;
- проведення аналізу схожих та подібних рішень;
- створення прототипів майбутнього рішення;
- проектування та безпосереднє створення додатку;

- розробка модуля редагування власного профілю відповідно до нових ролей;
- розробка модуля планування онлайн уроків та додавання їх до системи;
- розробка модуля редагування запланованих онлайн занять;
- розробка модуля додавання та редагування відео-заняття;
- розробка модуля перегляду всіх майбутніх онлайн занять або всіх записаних онлайн занять(відео-уроків);
- розробка модуля додавання на онлайн заняття до запланованих задач;
- проведення тестування веб-додатку;
- проведення виправлення знайдених багів;
- написання тестової документації.

## 2. МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

### 2.1 Проектування інформаційної технології проектування

Жодне проектування системи чи технології не розпочинається з безпосереднього написання коду. Після проведення детального аналізу, розбиття на функціональні та нефункціональні вимоги, наступним не менш важливим етапом є структуро-графічне відображення всіх вхідних та вихідних даних.

Для графічного відображення було обрано 2 типи діаграм, таких як IDEF(IDEF0) та UML. Перший тип був обраний для опису графічних й структурних процесів системи, другий – для показання відношень між акторами та прецедентами в системі.

Діаграма IDEF0 відображена на рисунку 2.1, вони демонструє загальні процеси системи та її головне призначення- функцію.

Діаграма IDEF – є розширеною версією першої, яка включає в себе інформаційні потоки, їх взаємодію між собою та аналіз структури. Цю контекстну діаграму можна переглянути на рисунку 2.2, який показує основний процес системи – «Навчання на онлайн платформі».

Діаграма UML – діаграма Варіантів-Використання – або ж діаграма прецедентів демонструє всі можливі варіанти, за допомогою яких можна використовувати систему. Об'єднує в собі всі функції CRM-систему для управління змаганнями та всі нові можливості для онлайн навчання бальним танцям. Дану діаграму можна переглянути на рисунку 2.3.

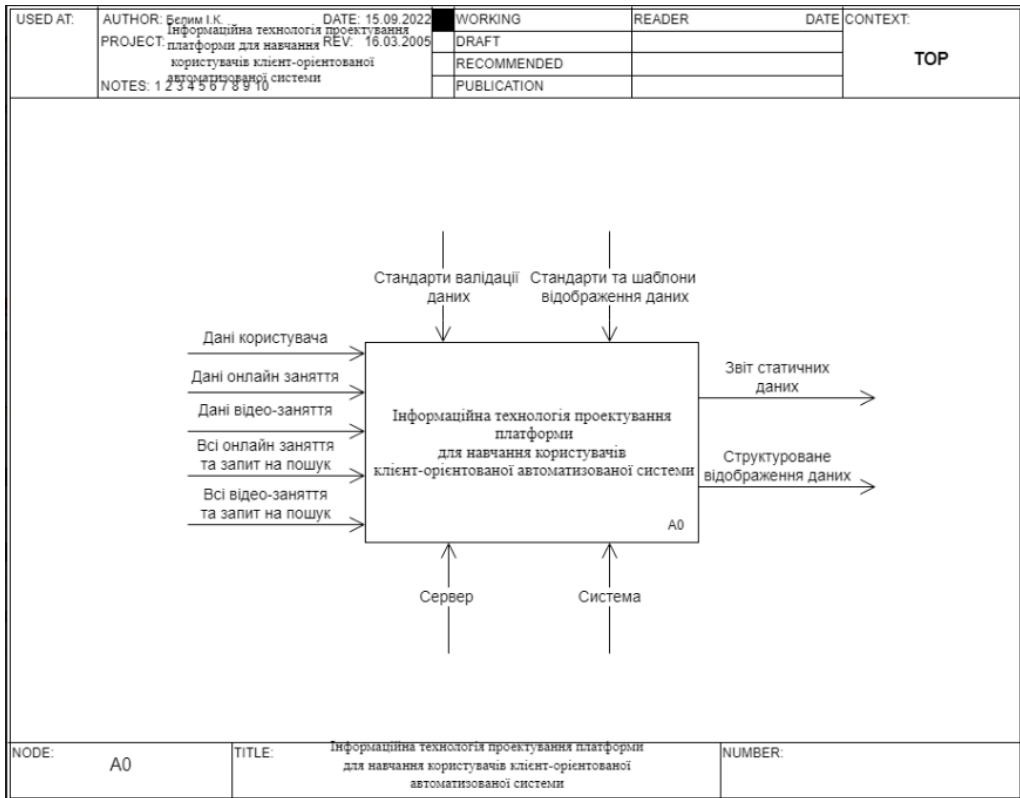


Рисунок 2.1 – IDEF0, контекстна діаграма першого рівня

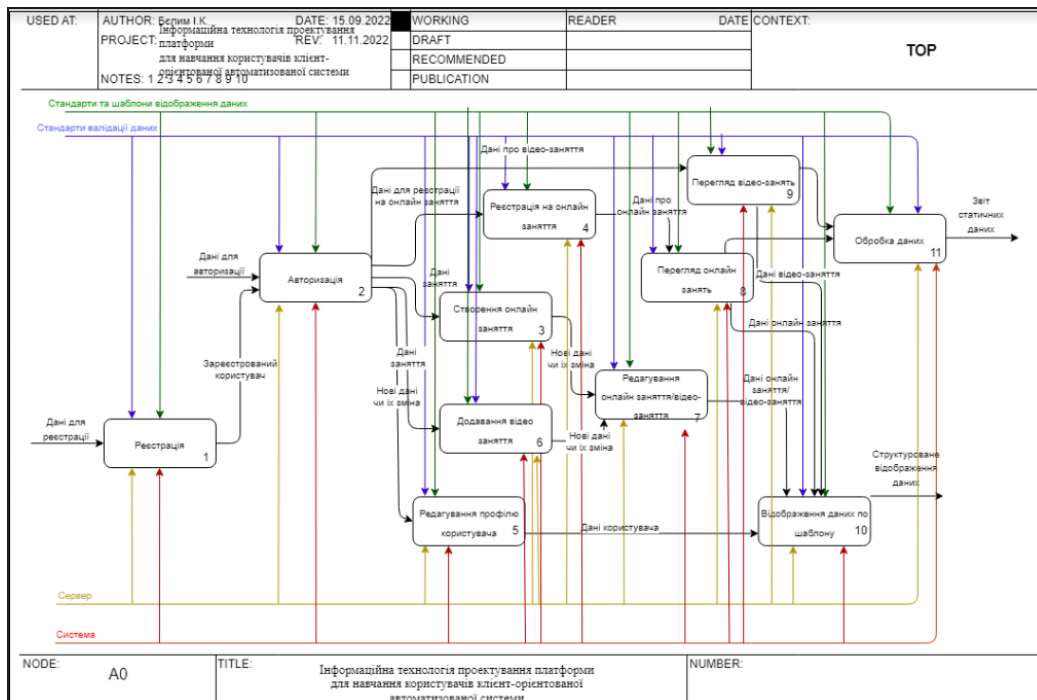


Рисунок 2.2 – IDEF1, декомповована контекстна діаграма першого рівня



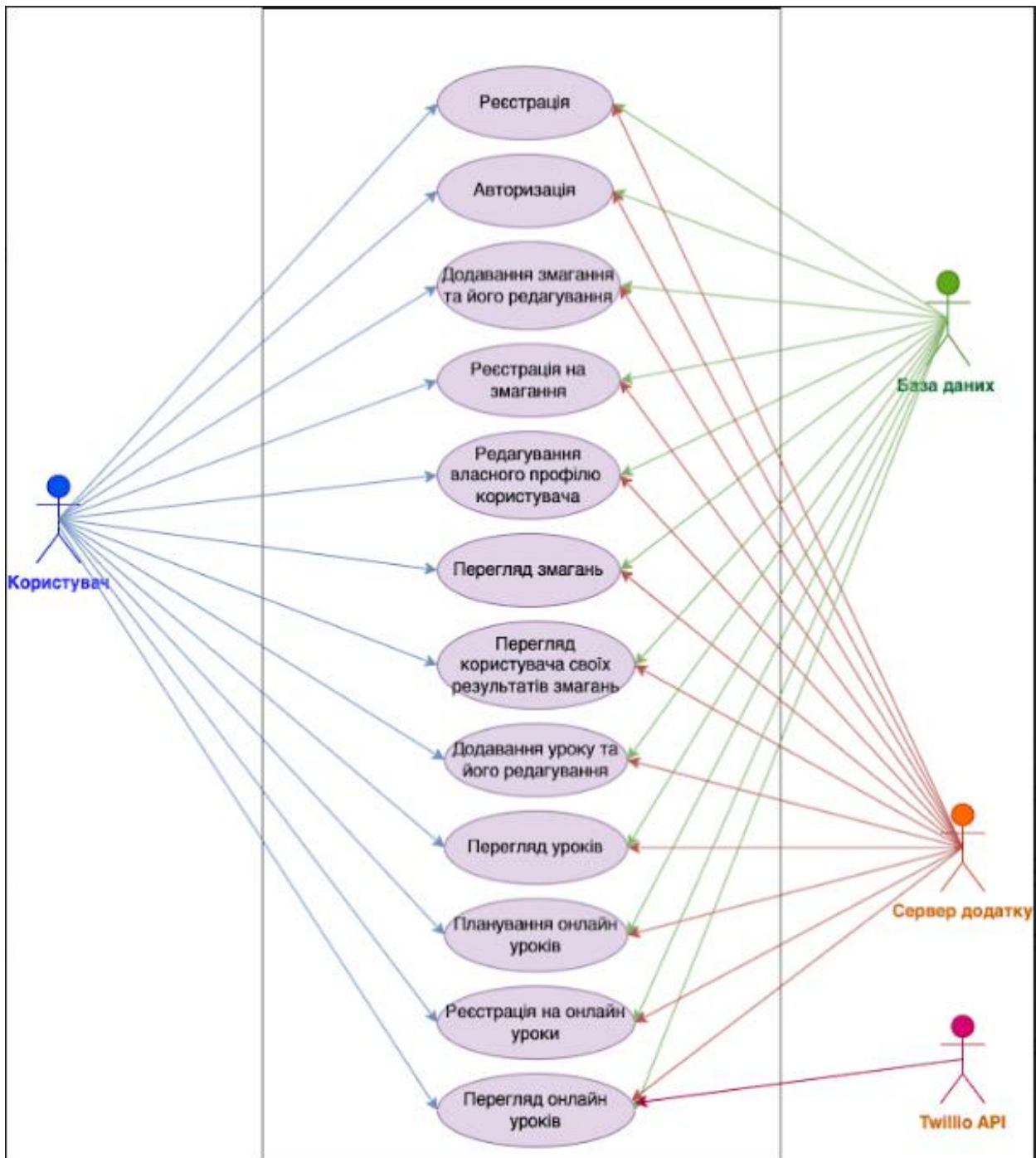


Рисунок 2.3 – Use-case діаграма

## 2.2 Проектування моделі бази даних

Так як дана інформаційна технологія проектування платформи для навчання розроблюється на базі вже існуючої системи, то відповідно до цього, базу даних довелося формувати на вже існуючих таблицях та проводити міграцію від старої версії до нової. Тобто вже існуючі атрибути та сутності було залишено та визначено нові необхідні для реалізації платформи. Дані визначені зв'язки далі було зібрано у ER-діаграму, щоб наглядно показати відношення між сутностями та їх атрибутами.

Для повноцінного функціонування системи було додано такі нові сутності:

- Студенти;
- Викладачі;
- Заняття;
- Онлайн заняття;
- Запит на навчання уроку;
- Запит на навчання онлайн уроку;
- Запит на створення уроку;
- Запит на створення онлайн уроку.

Відповідно до нових сутностей було зібрано нову таблицю 2.1, яка описує кожну сутність разом з її атрибутами.

Таблиця 2.1 – Опис додаткових сутностей БД та їх атрибутів

Сутність	Опис	Атрибут	Опис
students	Персональна інформація користувача, який є студентом	student_id	Ідентифікатор студента
		user_id	Ідентифікатор користувача системи
		s_class	Танцювальний клас студента
		s_club	Танцювальні клуби студента
teachers	Персональна інформація користувача, який є викладачем	teacher_id	Ідентифікатор викладача
		user_id	Ідентифікатор користувача системи
		experience	Досвід викладача
		t_club	Клуби викладача

Таблиця 2.1 – Продовження таблиці опису додаткових сутностей БД та їх атрибутів

Сутність	Опис	Атрибут	Опис
lessons	Інформація про уроки	lesson_id	Ідентифікатор уроку
		lesson_name	Назва уроку
		lesson_description	Опис уроку
		video_record	Відео запис уроку
online_lessons	Інформація про онлайн уроки	o_lesson_id	Ідентифікатор онлайн уроку
		o_lesson_name	Назва онлайн уроку
		o_lesson_description	Опис онлайн уроку
		o_lesson_date	Дата проведення онлайн уроку
		o_lesson_link	Посилання онлайн уроку
stud_request	Заявка студента на перегляд уроку	stud_request_id	Ідентифікатор заявки від танцівника на перегляд уроку
		lesson_id	Ідентифікатор уроку
		student_id	Ідентифікатор студента
tech_request	Створення уроку викладачем	tech_request_id	Ідентифікатор заявки створення уроку від викладача
		lesson_id	Ідентифікатор уроку
		tech_id	Ідентифікатор викладача
o_stud_request	Заявка студента на перегляд онлайн уроку	o_stud_request_id	Ідентифікатор заявки від студента на онлайн урок
		o_lesson_id	Ідентифікатор онлайн уроку
		stud_id	Ідентифікатор студента
o_tech_request	Створення онлайн уроку викладачем	o_tech_request_id	Ідентифікатор заявки створення онлайн уроку від викладача
		o_lesson_id	Ідентифікатор онлайн уроку
		tech_id	Ідентифікатор викладача

Ілюстрацією відповідної таблиці (також включаючи створені раніше таблиці в рамках кваліфікаційної роботи бакалавра) є реалізована ER-діаграма (рис. 2.4) сутностей, що відображає всі зв'язки таблиць-сутностей.

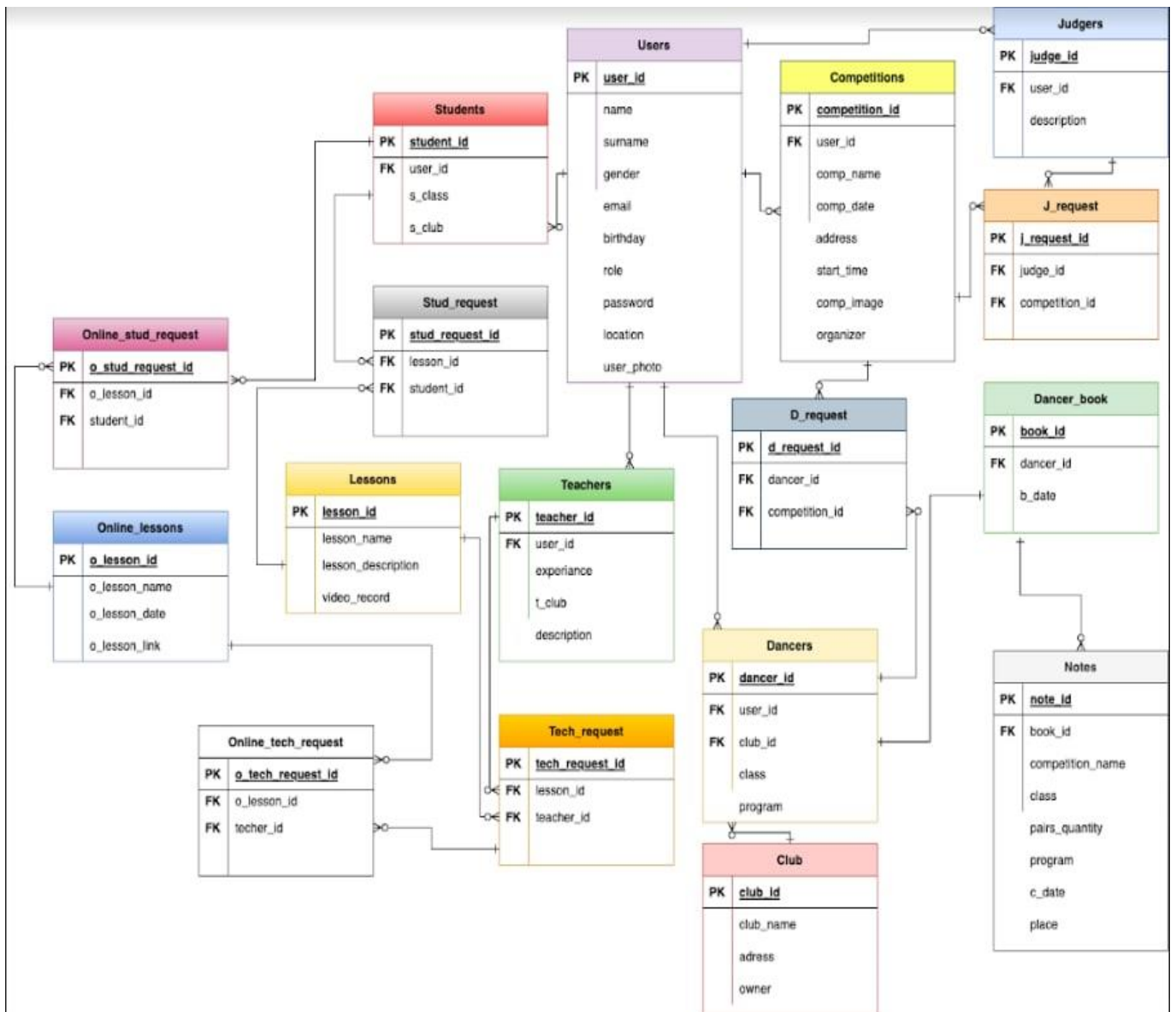


Рисунок 2.4 – ER діаграма

Відповідність вимальовується з аналогії зв'язків, які вже були «users» та «dancers», «users» та «judgers». Тож для таких сутностей як «users» та «students», «users» та «teachers» обрано зв'язок один-до-багатьох, що означає, що користувачі системи тепер можуть бути як студентами так і викладачами, але конкретний викладач або ж студент може мати тільки одного конкретного користувача.

Так само зв'язок один-до-багатьох «teachers» та «lessons», «teachers» та «online\_lessons», один викладач може додати багато уроків або провести багато

онлайн уроків, але у кожного уроку чи онлайн уроку є один конкретний створювач(тобто викладач).

Відношення один-до-багатьох також ідеально вписується в опис зв'язків між «students» та «stud\_request», «students» та «o\_stud\_request», «teachers» та «tech\_request», «teachers» та «o\_tech\_request», так як один конкретний студент може один раз відправити заявку на один і той самий урок чи один раз зареєструватися на один і той самий урок, так само як і викладач може один раз створити один і той самий урок чи один раз створити і провести один і той самий онлайн урок.

І відповідні зв'язки між «lessons» та «stud\_request», «lessons» та «tech\_request», «online\_lessons» та «o\_stud\_request», «online\_lessons» та «o\_tech\_request» паралельно говорять про те, що до одного уроку може мати велика кількість студентів доступ, як один викладач може створювати велику кількість уроків. Теж саме що багато студентів зможуть зареєструватися та перегляди один і той самий онлайн урок водночас, як і викладач може створювати та проводити безліч онлайн уроків.

### 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ

#### 3.1 Архітектура програмного додатку

При виконанні даної роботи, розробки інформаційної технології проектування платформи для навчання було використано принцип архітектури «клієнт- сервер».

Основними аспектами даної архітектури є те, що запити відправлені від певного користувача потрапляють на сервер та обробляються безпосередньо на ньому, після чого надсилається відповідь, яка вже принципами браузера перетворюється на зрозумілий та зовнішньо привабливий вигляд для користувача. Підґрунтям для даного механізму є мережа Інтернет, завдяки якій відбувається обмін даними. На рисунку 3.1 показаний процес взаємодії між клієнтом та сервером.

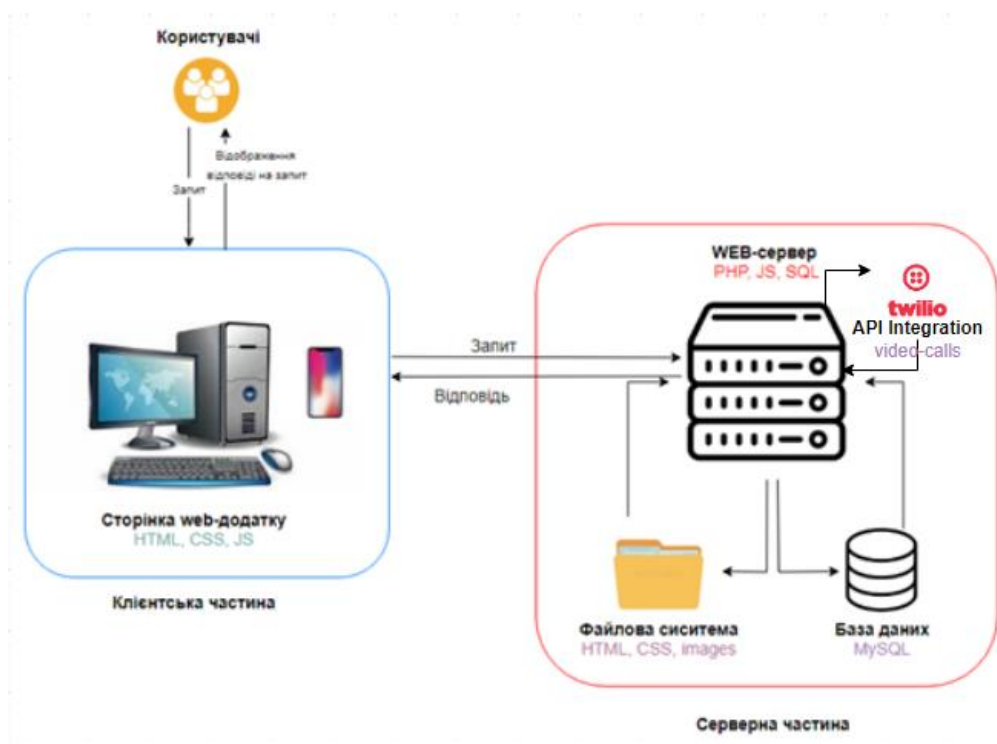


Рисунок 3.1 – Схематична архітектура системи

Система, що розроблюється є інформаційною технологією проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи, а отже правильне опрацювання всіх запитів від клієнтів та їх коректне представлення на сторінці є дуже важливим. При ситуації коли відповіддю на запит до сервера є пуста інформація або під час процесу виникла помилка, це все повинно бути оброблено і виведено на монітор користувача у зрозумілій формі.

Схематичне зображення відображає дану систему, та її складові, а саме два великих елемента – сервер та клієнт, вони виступають як окремі складники, але при цьому залежні один від одного та утворюють доцільний механізм. В середині якого існує велика кількість допоміжних та не менш важливих деталей.

Клієнтська частина системи за допомогою безлічі технологій, таких як JS, CSS та HTML виконують функцію відображення веб орієнтованої сторінки в браузері комп'ютерів чи інших пристроїв, адаптацію картинку для користувача в залежності від типу девайсів. Тобто, завдяки цієї частини користувач отримує відображення відповіді на запити [1, 9, 10].

Сервер або серверна частина складається з великої кількості компонентів, її функціонал дуже широкий, але головні функції, які виконує веб-сервер – обробляє запити, які надходять від клієнта, за допомогою PHP, JS, SQL [2, 3, 8]. До серверної частини також відноситься й файлова система, її задачею є робота з файлами каскадних стилів, що додають сторінці стилізований та структурований вигляд.

Серверна частина системи має велику кількість складових компонентів, котрі в свою чергу виконують широкий функціонал, головною функцією виступає обробка запитів, що надсилає користувач, вони виконуються завдяки SQL, PHP, JS [11, 12, 13]. Однією із елементів сервера є файлова система, задачею якої є надання сторінці структурований і стилізований зовнішній вигляд за допомогою файлів каскадного стилю.

Інтеграція з Twilio дозволить налагодити спілкування між користувачами в зручному для них просторі, це може бути як текстові повідомлення так і відео конференція. За допомогою цієї платформи будуть виконуватися онлайн заходи та заняття[4].

База даних, яка входить в серверну частину, представляю собою місце де зберігаються всі дані щодо користувача, заходів, уроків, вся ця інформація також має свій унікальний номер, що створюється автоматично, для подальшого швидкого розпізнання та пошуку в базі. Весь процес роботи можна описати так, аналізування запиту користувача, його перевірка на наявність, надсилання відповіді на даний запит та його подальше відображення клієнтською частиною у коректному вигляді на сторінці.

## 3.2 Програмна реалізація

Перед реалізацією інформаційної платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи був виконаний аналіз і порівняння безлічі технологій. Разом з цим опрацьовано помилки, які були допущені аналогами системи, та було виділено вимоги: зрозумілий, сучасний та простий для користувача, котрі відкривають його вперше, адаптивний до девайсів, як до комп'ютерів так і для мобільних телефонів різних платформ.

Тому, для реалізації даної системи було використано різні сторонні сервіси, що є в вільному доступі, а саме:

- Bootstrap – це фреймворк, який дає змогу користуватися готовими HTML-і CSS-шаблонами для простого створення та оформлення сторінки сайту[9];
- Google Fonts – це бібліотека від Google з вільним доступом для шрифтів[7];
- Font Awesome – це сервіс, який є частково вільним в доступі за для можливості додавати графічних елементів іконок до сторінки системи;
- Twilio – це платформа, що надає інструменти зв'язку для здійснення відео дзвінків та обміном текстових повідомлень[4].

Також для реалізації програми було використано релевантні технології, які наведені нижче:



- PHP (Hypertext Preprocessor) – скриптова мова, що використовується за для написання серверної частини коду в реалізовуваній системі[2];
- CSS (Cascading Style Sheets) – це мова, яка дозволяє HTML-сторінки мати стильове оформлення, коректувати розміри та кольори, а також розміщення блоків на сторінці в залежності від використовуваного девайсу[10];
- HTML (HyperText Markup Language) – це мова гіпертекстової розмітки та тегів, що надає сторінці структуру, блоки для розбиття і їх розміщення в певній послідовності[1];
- SQL (Structured Query Language) – це мова програмування, що використовується для управління даними, їх редагування, додавання, видалення та перегляд уже існуючих[3];
- MySQL – це реляційна система, що є вільною у доступі, за для управління базами даних, що була реалізована у даній системі[8];
- JS (JavaScript) – це мова, яка має широкий функціонал, вона є як об’єктно-орієнтована, так і динамічною та прототипною, в даному проекті була використана для інтерактивності сторінки та анімацій[14, 15];

Рисунок 3.2 є реалізацією бази даних в MySQL, що показує сутності, зв’язок між сутностями системи, всі типи даних атрибутів кожної сутності.

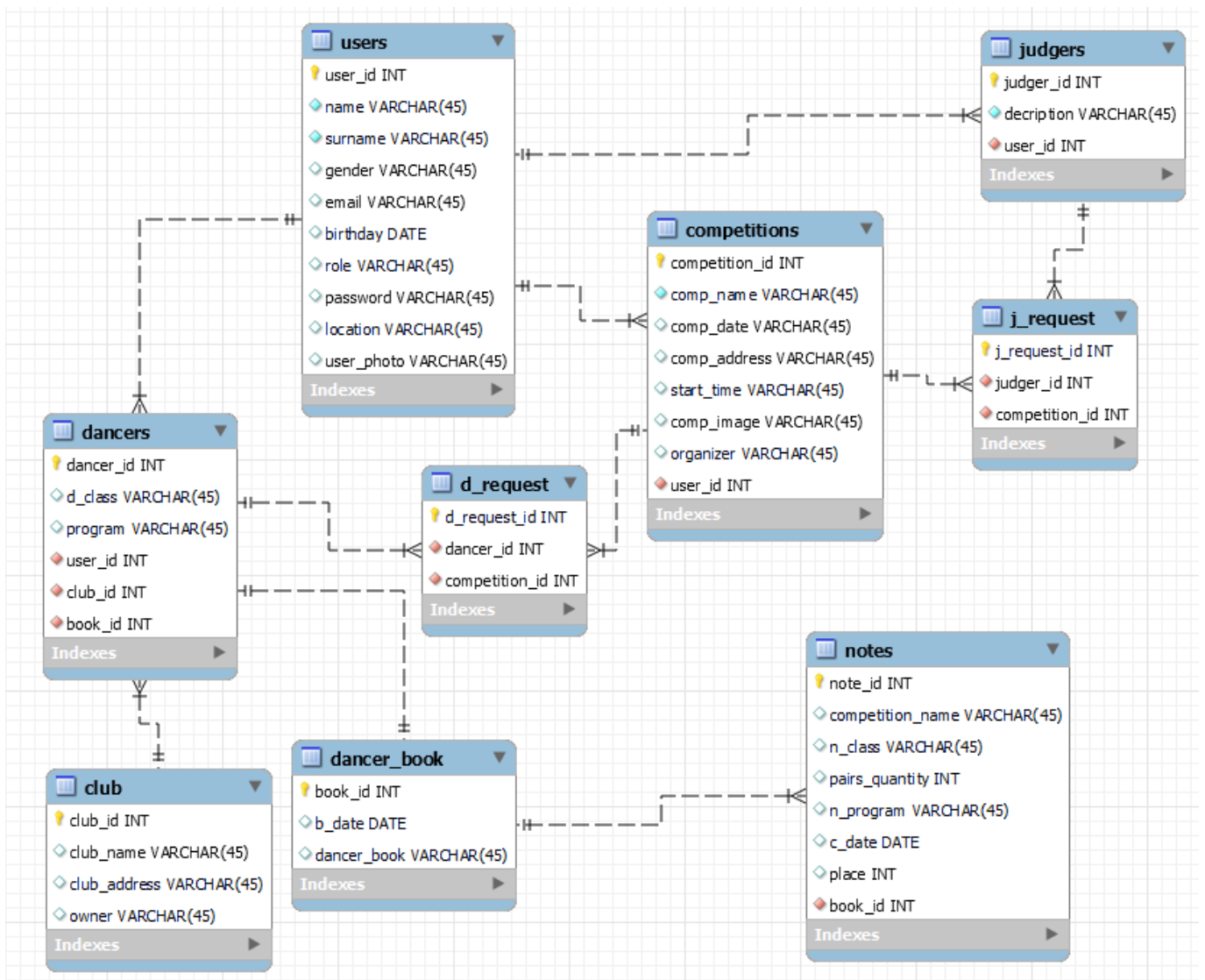


Рисунок 3.2 – Реалізація БД в MySQL

Рисунок 3.3 є відображенням залежностей та взаємодій модулів, що включають в себе основні функції.

Відповідно до схеми видно, що до основних модулів належать: можливість реєстрації, вхід до системи, додавання особистої інформації в профіль, редагування профілю, додавання та редагування відео-уроків, планування онлайн-уроків та редагування їх, проведення онлайн-уроків, додавання майбутніх уроків до задач.

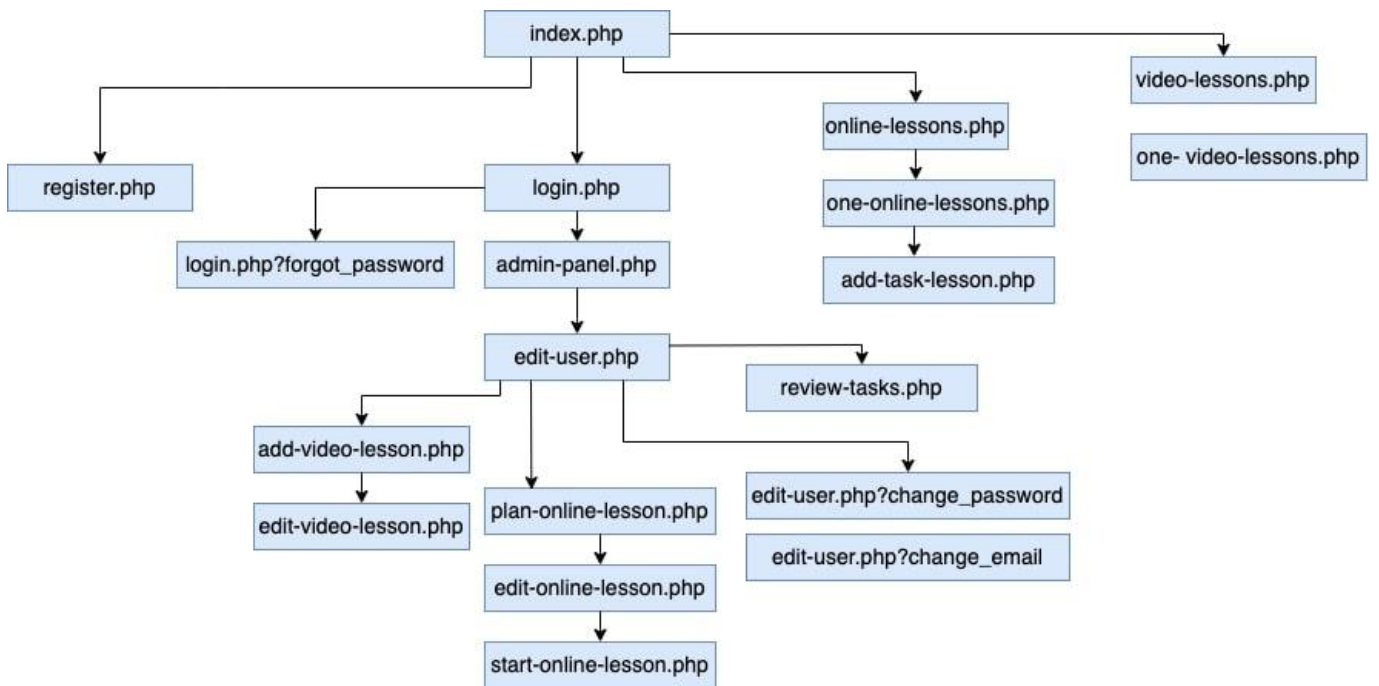


Рисунок 3.3 – Структура система в схематичному вигляді

### 3.2.1 Реєстрація користувача

Формат реєстрації нових користувачів залишився повністю незмінним, що дало можливість легко провести міграцію даних зі старої системи до нової. Тож для того, щоб стати повноцінним користувачем інформаційної системи потрібно заповнити 5 полів:

- Ім'я;
- Фамілія;
- Імейл;
- Пароль;
- Повторити введений пароль.

Всі ці введені дані користувачем заносяться в 5 окремих відповідних змінних, що зберігаються з методу POST. В цих змінних дані перевіряються, оброблюються чи відповідають вони вимогам полів, а тільки після цього ці дані заносяться до БД. Дану реалізацію можна подивитися на рисунку 3.4.

Змінні, які зберігають дані, що передалися по методу POST:

- `$n_name` – зберігає дані з поля Ім'я;
- `$n_surname` – зберігає дані з поля Фамілія;
- `$n_email` – зберігає дані з поля імейл;
- `$n_password` – зберігає дані з поля Пароль;
- `$n_password2` – зберігає дані з поля Повторити Пароль.

Як тільки-но користувач ввів дані та натиснув на кнопку «Зареєструватися» виконується перевірка:

- Чи немає вже в базі даних іншого користувача з таким же самим імейлом. Це виконується через порівняння двох змінних `check_email` – міна перевірки та `$n_email` – змінна з щойно введеними даними. Якщо знайдено помилку, то користувач бачить повідомлення «Помилка! Користувач з таким email вже зареєстрований!».
- Чи однакові паролі ввів користувач у поля Пароль та Повторити пароль. У негативному випадку він бачить повідомлення, що паролі не збігаються і потрібно перевірити це.

Якщо всі перевірки пройдено успішно, то змінна `$query` створює запит на створення нового запису в БД, друга змінна `$result` надсилає запит до БД. І вже після цього довгого процесу, який для користувача відбувається в лічені мілісекунди, він бачить повідомлення, що реєстрація пройшла успішно, і користувач може входити до системи з своїми занесеними кредами під час реєстрації.

```

<?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 3.4 – Процес реєстрації нового користувача

### 3.2.2 Додавання Відео-уроку викладачем

Тепер користувачі мають змогу під час редагування власного профілю обрати одну з чотирьох можливих ролей-професій, тобто чи є вони суддями, професійними танцівниками, студентами, які хочуть навчатися чи професійними викладачами, які хочуть навчати спортивно-бальним танцям.

Тож, якщо користувач відредагує свій профіль так, що буде обрано професію – викладач, то в нього з’явиться можливість додавати відео-уроки.

Звичайно перед тим, як у викладача з’явиться така можливість буде виконано перевірку чи це є реально профіль викладача, за допомогою змінної \$check\_role, якщо під час такої перевірки відповідь буде ні, то користувач побачить повідомлення «Помилка! Ви не є викладачем, тому ви не можете додавати Відео-уроки». У разі позивної відповіді від сервера під час перевірки Відео-урок буде додано и він буде відображатися в особистому кабінеті викладача та одразу ж з’явиться на загальній сторінці всіх відео-уроків. Також для того, щоб успішно створити відео-урок потрібно заповнити 2 поля – назва та опис і звичайно ж безпосередньо завантажити саме відео. Дану реалізацію можна переглянути на рисунку 3.5.

```
<?php
if(isset($_POST["lesson_name"])){$lesson_name = $_POST[lesson_name];}
if(isset($_POST["lesson_description"])){$lesson_description= $_POST[lesson_description];}
if(isset($_POST["lesson_video"])){$lesson_video= $_POST[lesson_video];}
if(isset($_POST["n_send"])){
    $check_email = mysql_query("SELECT 'email' FROM 'users' WHERE 'email' = '$c_email'");
    $check_role = mysql_query("SELECT 'role' FROM 'users' WHERE 'role' = '$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 викладачем, тому ви не можете додавати відео-уроки!
            <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) == 2){
        $date = date("m.d.y");
        $query = "INSERT INTO 'lessons' ('lesson_name', 'lesson_description', 'lesson_video')
        VALUES ('$lesson_name', '$lesson_description', '$lesson_video')";
        $result = mysql_query($query) or die ("Error: ".mysql_error() );
        if($result){
            echo '<div class="alert alert-info" role="alert">
            Відео-урок успішно додано, тепер його можуть переглядати студенти.</div>';
            exit();
        }
    }
}
?>
```

Рисунок 3.5 – Додавання «Відео-уроку» викладачем

### 3.2.3 Планування та створення онлайн-уроку викладачем

Планування онлайн заняття можливо так само як і створення відео-уроків тільки для користувачів з обраною професією під час редагування профілю – викладач. Для того щоб запланувати онлайн заняття необхідно заповнити 4 поля – назва, опис, обрати дату та час, а далі система одразу ж сама генерує посилання на майбутнє онлайн заняття, за допомогою інтеграції Twillio. За цим посиланням викладач буде переходити і образу ж буде давати доступ до аудіо- та відео- ресурсів, щоб можна було провести онлайн заняття. Звичайно ж збереженням запланованого онлайн уроку буде виконуватися перевірка чи є користувач викладачем, за допомогою змінної \$check\_role.

Якщо профіль користувача не є профілем викладача, то користувач побачить помилку «Помилка! Ви не є викладачем, тому ви не можете планувати та створювати Онлайн-уроки». У випадку успішного створення майбутнього Онлайн-уроку користувач побачить повідомлення «Онлайн-урок успішно створено та заплановано, тепер на нього можуть записуватися студенти». Дану реалізацію можна переглянути на рисунку 3.6.

```

<?php

if(isset($_POST["online_lesson_name"])){$online_lesson_name = $_POST[online_lesson_name];}
if(isset($_POST["online_lesson_description"])){$online_lesson_description= $_POST[online_lesson_description];}
if(isset($_POST["online_lesson_date"])){$online_lesson_date= $_POST[online_lesson_date];}
if(isset($_POST["online_lesson_time"])){$online_lesson_time= $_POST[online_lesson_time];}
if(isset($_POST["n_send"])){
    $check_email = mysql_query("SELECT 'email' FROM 'users' WHERE 'email' = '$c_email';");
    $check_role = mysql_query("SELECT 'role' FROM 'users' WHERE 'role' = '$c_role;");
    if(mysql_num_rows($check_email) != 0 && ($check_role) != 2){
        echo '
        <div class="alert alert-danger alert-dismissible fade show" role="alert">
        <strong>Помилка!</strong> Ви не є викладачем, тому ви не можете додавати відео-уроки!
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&time;</span>
        </button>
        </div>';
    }else (mysql_num_rows($check_role) == 2){
        $online_lesson_link = Twillio.generate_link();
        $date = date("m.d.y");
        $query = "INSERT INTO 'online_lessons' ('online_lesson_name', 'online_lesson_description', 'lesson_video', 'online_lesson_link')
        VALUES ('$lesson_name', '$lesson_description', '$lesson_video', '$online_lesson_link');";
        $result = mysql_query($query) or die ("Error: ".mysql_error() );
        if($result){
            echo '<div class="alert alert-info" role="alert">
            Онлайн-урок успішно створено та заплановано, тепер на нього можуть записуватися студенти.</div>';
            exit();
        }
    }
}
?>

```

Рисунок 3.6 – Планування «Онлайн уроку» викладачем

### 3.2.4 Додавання Онлайн уроку до своїх задач

Якщо користувач обрав собі професію – студент під час редагування профілю, то в нього є можливість переглядати всі відео-уроки та навчатися по ним, а також додавати собі до запланованих задач майбутні онлайн уроки, щоб точно їх не пропустити і мати швидких доступ до цих занять, так як вони додаються на особисту сторінку студента. Але звичайно з технічної сторони перед виконанням процесу додавання задачі(запланованого онлайн-уроку) до своїх задача, на свою особисту сторінку профіля виконується перевірка чи є цей користувач студентом, за допомогою все тієї ж змінної `$check_role`.

Звичайно користувача буде проінформовано, якщо його професія-роль не є студентом у цій системі, він побачить повідомлення «Помилка! Ви не є користувачем студентом, тому не можете додавати онлайн-уроки до задач». Якщо ж все пройде успішно, то користувач побачить «Онлайн-урок успішно доданий до вашої панелі задач.». Реалізація відображена на рисунку 3.7.

```

<?php
if(isset($_POST["online_lesson_name"])){$online_lesson_name = $_POST[online_lesson_name];}
if(isset($_POST["online_lesson_description"])){$online_lesson_description= $_POST[online_lesson_description];}
if(isset($_POST["online_lesson_date"])){$online_lesson_date= $_POST[online_lesson_date];}
if(isset($_POST["online_lesson_time"])){$online_lesson_time= $_POST[online_lesson_time];}
if(isset($_POST["n_send"])){
    $check_email = mysql_query("SELECT 'email' FROM 'users' WHERE 'email' = '$s_email';");
    $check_role = mysql_query("SELECT 'role' FROM 'users' WHERE 'role' = '$s_role';");
    if(mysql_num_rows($check_email) != 0 && ($check_role) != 3){
        echo '
        <div class="alert alert-danger alert-dismissible fade show" role="alert">
        <strong>Помилка!</strong> Ви не є користувачем студентом, тому ви не можете додавати онлайн-уроки до задач!
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
        </button>
        </div>';
    }else {mysql_num_rows($check_role) == 3){
        $date = date("m.d.y");
        $query = "INSERT INTO 'student_tasks' ('task_online_lesson_name', 'task_online_lesson_description', 'task_lesson_video', 'task_online_lesson_link')
        VALUES ('$lesson_name', '$lesson_description', '$lesson_video', '$online_lesson_link');";
        $result = mysql_query($query) or die ("Error: ".mysql_error() );
        if($result){
            echo '<div class="alert alert-info" role="alert">
            Онлайн-урок успішно доданий до вашої панелі задач.</div>';
            exit();
        }
    }
}
?>

```

Рисунок 3.7 – Додавання онлайн-уроку до запланованих задач



### 3.2.5 Редагування відео-уроку викладачем

Також реалізовано функціонал редагування відео-уроків. Викладача в своєму особистому кабінеті на рівні з тим, що можуть створювати, перегляди, так само можуть й редагувати вже створені відео-уроки. Цей функціонал зі сторони бекенду розбито на 2 функції – редагування прикріпленого відео-файлу(рис. 3.8) та редагування всієї інформації(рис. 3.9).

```

if (isset($_POST['del_video'])) {
    $sql_refresh = mysql_query("UPDATE 'lessons' SET 'c_video' = 'lesson.avi'
    WHERE 'user_id' = " . $_GET['red_id'], $link);
    echo '<meta http-equiv="Refresh" content="0; URL="add-lesson?
    user_id=' . $_GET['red_id'] . '">';
}

```

Рисунок 3.8 – Редагування відео у відео-уроці викладачем

Викладачі також можуть відредагувати повністю всі інші дані відповідно відео-уроку, ці всі дані були обов'язкові до заповнення, тобто його назву та опис(рис. 3.12), ця логіка реалізована таким чином: збираємо всю нову інформацію, що ввів користувач, за допомогою методу POST, далі всі ці дані записуємо в змінну \$sql\_refresh в запит до БД і далі безпосередньо надсилаємо запит до БД.

```

if (isset($_POST['refresh_submit'])){
    if($_FILES['red_video']['name'] != ''){
        $path = 'videos/';
        $tmp_path = 'temp/';
        //Массив допустимих значений для картинок
        $types = array('video/avi', 'video/mp4');
        //Максимальный объем данного файла
        $size = 4000000;
        $extension_vid = strtolower(substr(strrchr($_FILES['red_video']['name'], '.'),1));
        $name_vid = rand(1, 1000).'-' .md5($_FILES['red_video']['name']).'.'.$extension_vid;
        if ($_SERVER['REQUEST_METHOD'] == 'POST')
        {
            // 1) Проверка типа файла
            if (!in_array($_FILES['red_video']['type'], $types))
                die('Тип файла недопустимый. <a href="#">Загрузить другой?</a>'
                .$_FILES['red_video']['type']);
            // 2) Проверка размера файла
            if ($_FILES['red_video']['size'] > $size)
                die('Файл слишком большой. <a href="#">Загрузить другой?</a>');
            // 3) Загрузка подходящего файла и вывод оповещения
            if (!@copy($_FILES['red_video']['temp_name'], $path.$name_vid)){
                echo 'Что-то пошло не так.';
            }
        }
    }else { $name_video = $_result['c_video'];}
    $sql_refresh = mysql_query("UPDATE 'lessons' SET 'lesson_name' = '".$_POST['red_name']."',
    'lesson_description' = '".$_POST['red_description']."', 'lesson_description' = '".$_POST['red_description']."',
    'lesson_video' = '".$_POST['red_video']."',
if(!$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-lesson.php?
    user_id='.$_GET['red_id'].'">';
}
}
}

```

Рисунок 3.9 – Редагування відео-уроку

### 3.2.6 Редагування онлайн-уроку викладачем

Аналогічному тому, що викладачі мають змогу відредагувати додані ними відео-уроки, так само в них є можливість і відредагувати й заплановані онлайн-заняття, тільки до того моменту поки не настав час самого онлайн-заняття.

Найбільш вірогідними змінами у запланованому онлайн-уроці можуть бути зміни дати та часу його проведення. І дуже цікава річ, що була реалізована, за допомогою інтеграції Twillio, те що посилання в такому випадку змінюється, але для користувачів ці зміни взагалі непомітні. Дану реалізацію можна побачити на рисунку 3.10.

```

if (isset($_POST['refresh_submit'])) {
    $sql_refresh = mysql_query("UPDATE 'online-lessons' SET 'online_lesson_name' = '".$_POST['red_name']."',
    'online_lesson_description' = '".$_POST['red_description']."', 'online_lesson_date' = '".$_POST['red_date']."',
    'online_lesson_time' = '".$_POST['red_time']."', 'online_lesson_link' = '".$_POST['red_link']."'");
    if(!$sql_refresh) {
        echo '<p style="text-align:center; font-size^18px; color:red;">Успіх! '.';
    } else {
        echo '<meta http-equiv="Refresh" content = "0"; URL="add-online_lesson.php?
        user_id='.$_GET['red_id'].'">';
    }
}
}

```

Рисунок 3.10 – Редагування запланованого онлайн заняття

### 3.2.7 Twillio інтеграція

Можливість планування та проведення онлайн-уроків не можливо було реалізувати без використання інтеграції Twillio. Для того, щоб створити правильно зв'язок між двома сервісами потрібно було:

- Знайти в особистій акаунт консолі Twillio – Account SID;
- Далі потрібно було створити ключ – API Key також в консолі Twillio;
- І останнім пунктом є генерування Access Token використовуючи Twillio Library Helper.

За допомогою цих кроків було створено підключення до інтеграції(рис. 3.11). Так як на онлайн уроках буде присутня велика кількість користувачів, тобто не один викладач – один студент, а буде один викладач – група студентів, тому використовуються тільки групові кімнати для перегляду уроку – Group Rooms, і ці дзвінки зашифровані за допомогою технології E2E. Даний тип кімнат дозволяє доєднуватися в один онлайн-урок до 50 студентам за один раз, а якщо у випадку більшої кількості бажаючих студентів відвідати заняття, то тут все в порядку черги, тільки ті перші 50 осіб, які додали собі це заняття до задач зможуть потрапити на нього. Всіх інших бажаючих буде проінформовано, що кількість можливих людей вже вичерпана, спробуйте наступного разу.

```

require_once '/path/to/vendor/autoload.php'; // Loads the library
use Twilio\Jwt\AccessToken;
use Twilio\Jwt\Grants\VideoGrant;

// Required for all Twilio access tokens
$twilioAccountSid = getenv('TWILIO_ACCOUNT_SID');
$twilioApiKey = getenv('TWILIO_API_KEY');
$twilioApiSecret = getenv('TWILIO_API_KEY_SECRET');

// Required for Video grant
$roomName = 'cool room';
// An identifier for your app - can be anything you'd like
$identity = 'user_name';

// Create access token, which we will serialize and send to the client
$token = new AccessToken(
    $twilioAccountSid,
    $twilioApiKey,
    $twilioApiSecret,
    3600,
    $identity
);

// Create Video grant
$videoGrant = new VideoGrant();
$videoGrant->setRoom($roomName);

// Add grant to token
$token->addGrant($videoGrant);

// render token to string
echo $token->toJWT();

```

Рисунок 3.11 – Інтегрування до Twilio

### 3.3 Використання програмного продукту

Так як дана платформа для онлайн навчання спортивно-бальним танцям є частиною CRM-системи, заздалегідь вже є реалізований функціонал без якого жоден користувач не зможе перегляди та реєструватися на будь-які онлайн уроки є функціонал реєстрації. Дана форма після додавання нової навчальної платформи не зазнала змін, тому вона складається все з тих же 5 полів з валідацією:

- Ім'я
- Фамілія

- Імейл
- Пароль
- Повторити пароль

Рисунок 3.12 є відображенням форми реєстрації. З неї будь-який користувач може потрапити на сторінку авторизації. У випадку якщо поля при реєстрації заповнені невалідними даними, тобто значення не відповідають очікуванням(наприклад: пароль не збігається з повторним паролем), то користувач бачить відповідне повідомлення на екрані, у випадку успішної реєстрації користувач також бачить відповідне до цього повідомлення та з'являється можливість увійти до системи з відповідними до реєстрації кредитами(рис. 3.13).

Форма авторизації також залишилася незмінною, для того що увійти до системи достатньо просто пам'ятати два значення та заповнити їх у відповідні поля – це імейл та пароль. Форма входу-авторизації відображена на рисунку 3.14. Також у випадку якщо дані, з якими користувач буде авторизовуватись будуть не відповідати дійсності він побачить відповідний меседж(рис. 3.15).

Але навіть у випадку, якщо користувач забув пароль, то можливість його відновити залишилась не змінною, за допомогою імейлу, на який прийде лист-підтвердження, зі згенерованим автоматично рандомайзером пароль, який після авторизації можна буде зміни у налаштуваннях власного профілю. Рисунок 3.16 є відображенням форм відновлення паролю.

The screenshot shows the registration form for 'Ballroom dancing'. The form is titled 'Реєстрація' and is set against a background image of a ballroom dancer's legs. The form includes the following fields and elements:

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

Рисунок 3.12 – Форма реєстрації користувача

The figure displays three screenshots of the registration form showing different validation messages:

- Left screenshot:** Shows an error message: **Помилка!** Значення в полі Пароль та Повторити пароль не збігаються.
- Middle screenshot:** Shows an error message: **Помилка!** Користувач з таким імейлом вже зареєстрований.
- Right screenshot:** Shows a success message: **Реєстрація відбулася успішно.** Пройдіть [Авторизацію](#)

Рисунок 3.13 – Валідаційні меседжи під час реєстрації

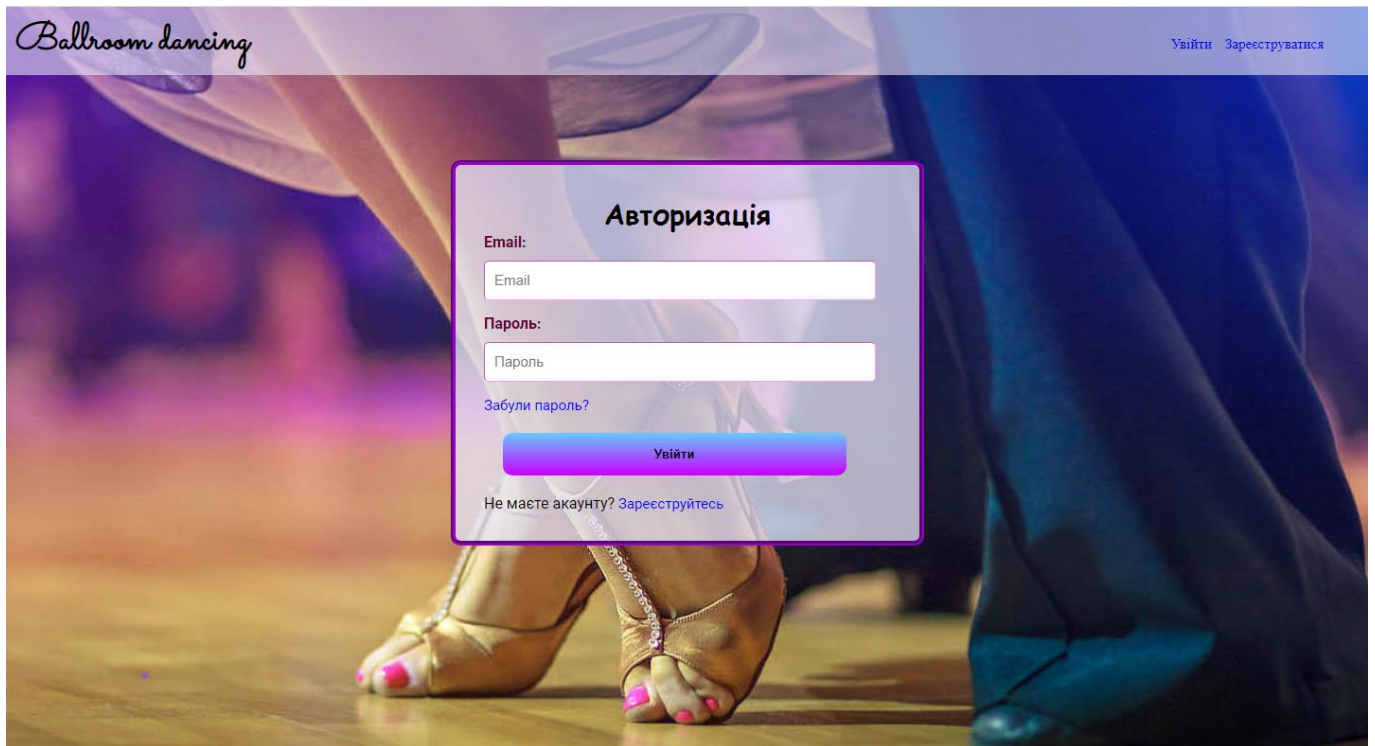


Рисунок 3.14 – Форма входу до системи користувача

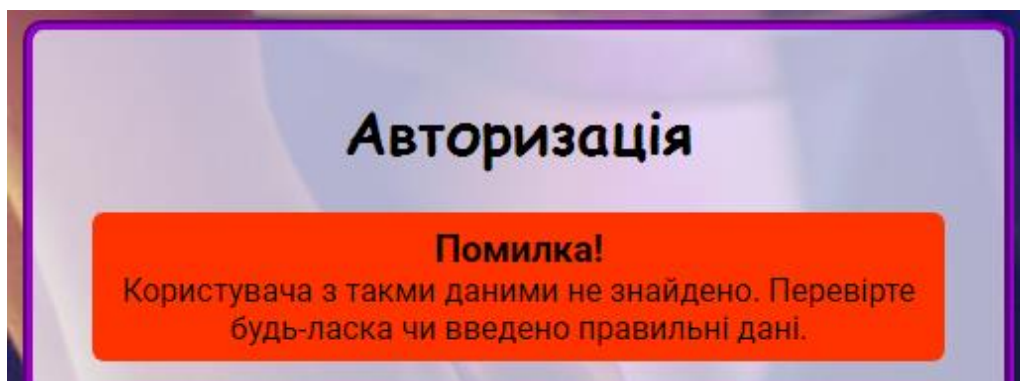


Рисунок 3.15 – Валідаційний меседж під час входу до системи



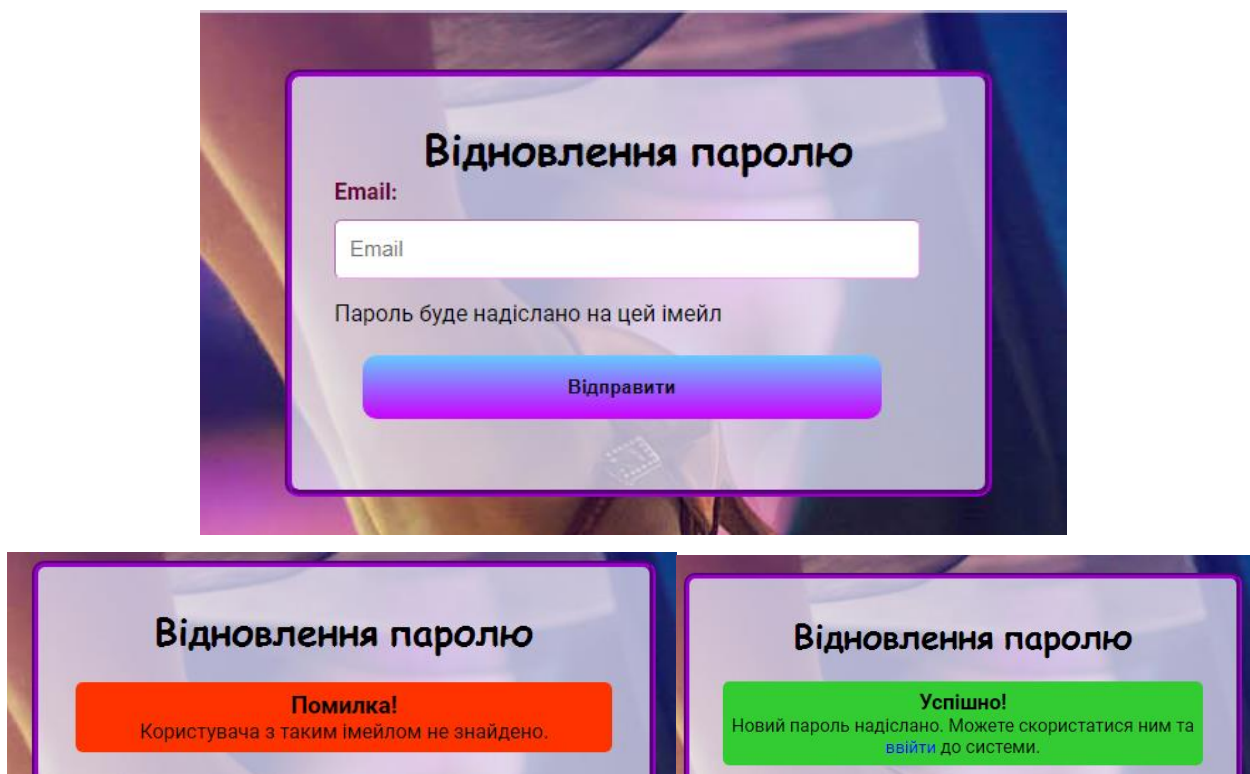


Рисунок 3.16 – Форма відновлення пароля з модальними вікнами

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 3.17.

Після проходження успішної реєстрації користувач потрапляє на сторінку власного профіля(рис. 3.17), де є коротка інформація, яка була заповнена під час реєстрації. З цієї сторінки в нього є можливість видалити профіль, відредагувати його і вже відповідно до відредагованої інформації профіль користувача набуде одного з 4 виглядів відповідно до обраних ролей під час реєстрації.



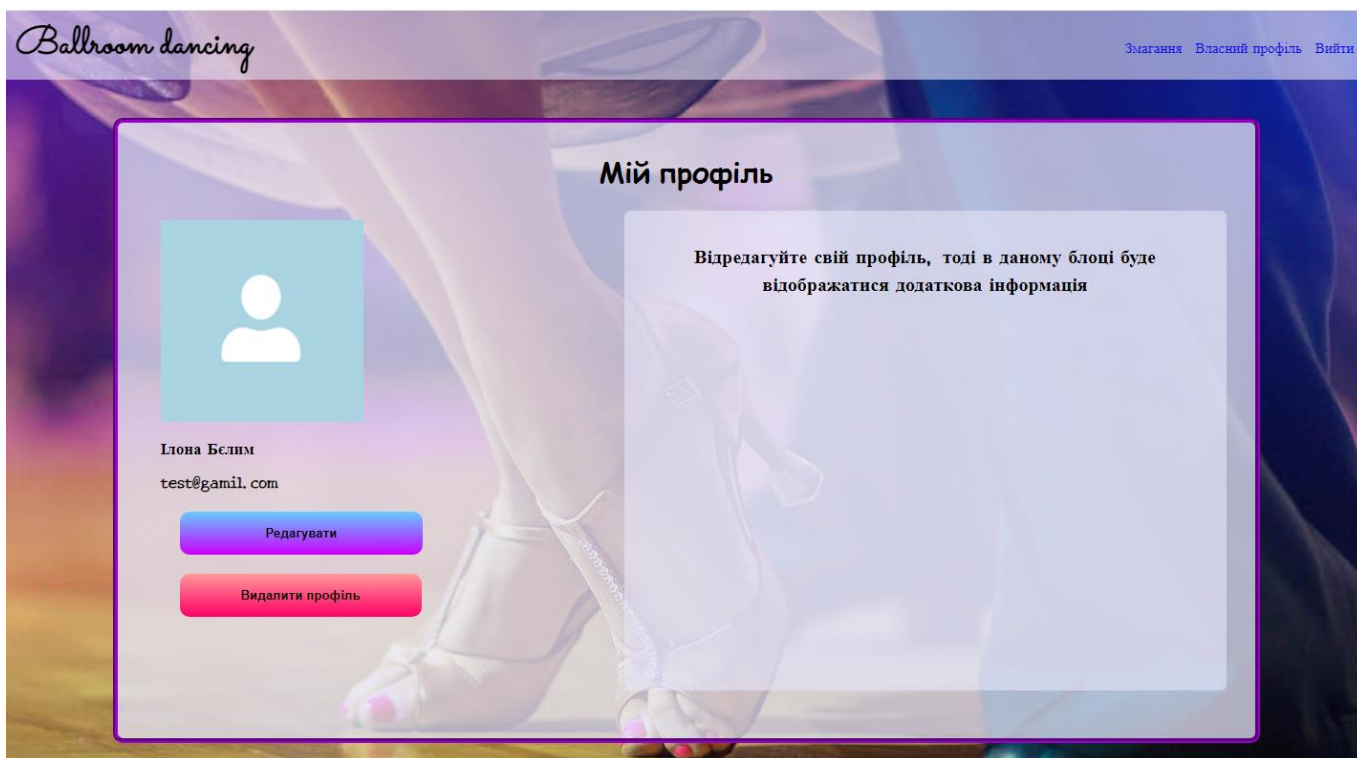


Рисунок 3.17 – Профіль щойно зареєстрованого нового користувача

Як тільки користувач потрапляє на сторінку редагування профілю, він може змінити всі поля заповнені під час реєстрації й звичайно ж електронну адресу й паролем. А також занести додаткову інформацію, яка ще не була занесена під час реєстрації, наприклад – фотокартка, день народження.

Раніше користувачами могли і повинні були ставати тільки ті користувачі, які є танцівниками, або ж суддями, а тепер ті хто хочуть навчатися на онлайн заняттях та відео-уроках – студенти та ті, хто хочуть проводити ці онлайн заняття та викладати відео-уроки – викладачі. Сторінка редагування профілю без обраних ще ролей відображена на рисунку 3.18.

**Відредагуйте потрібну інформацію**

Ім'я:

Фамілія:

Email:

День народження:

Пароль:

Місто:

Оберіть стать

Чоловіча

Жіноча

Фотокартка:  
 | Файл не вибран

Оберіть танцювальну сферу:

Суддя

Танцівник

Студент

Викладач

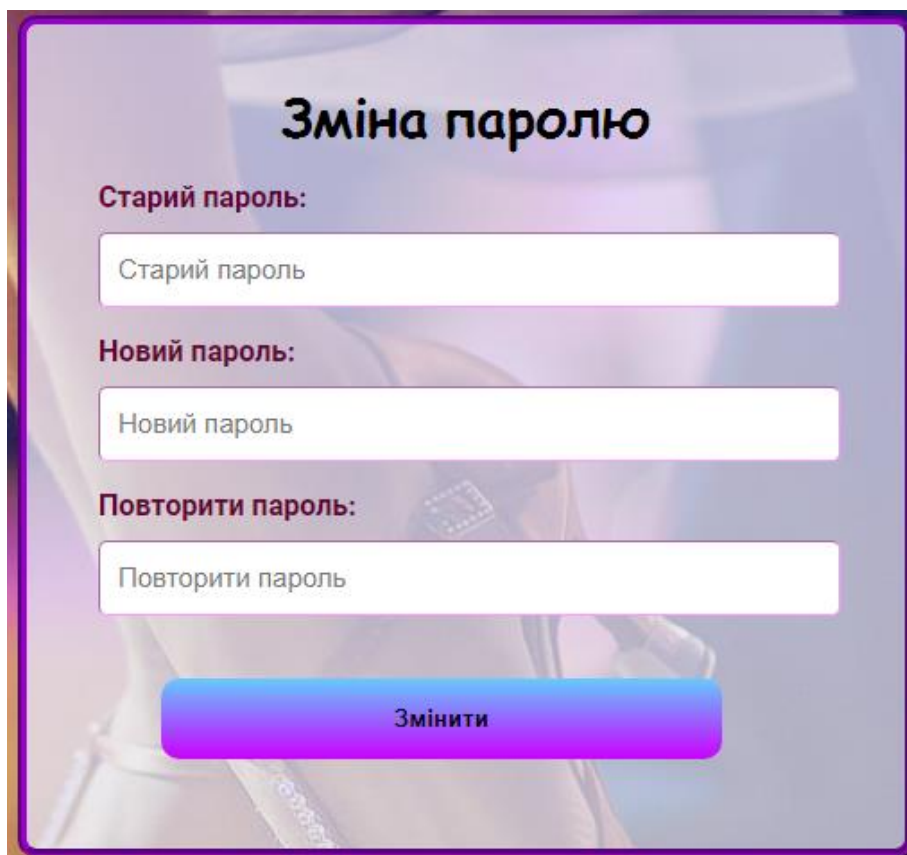
Рисунок 3.18 – Форма редагування звичайного профілю

Звичайно залишися незмінними форми зміни паролю та імейлу. Як і раніше зміни обох параметрів відбуваються на окремих сторінках, це можна переглянути на рисунку 3.19 та рисунку 3.21. Для того що змінити пароль необхідно заповнити 3 поля – старий пароль, новий пароль та повторити ще раз новий пароль, а для зміни електронної адреси достатньо просто вказати стару та нову.

Валідаційні меседжи залишися незмінними(рис. 3.20 та 3.22), 3 види:

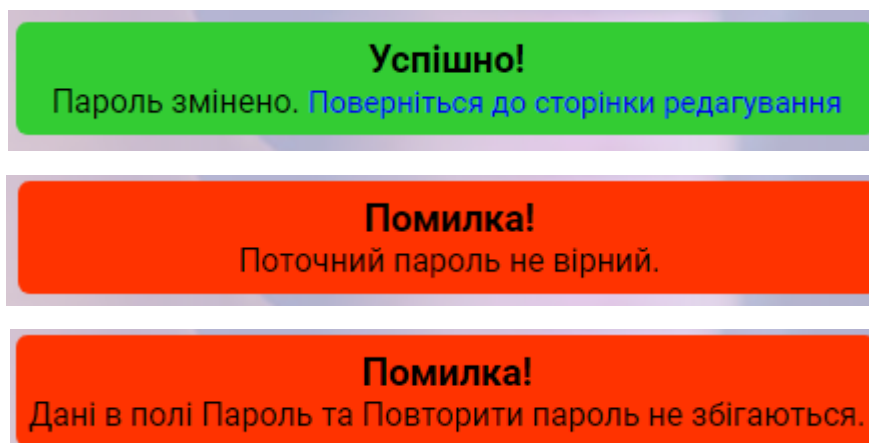
- Коли все пройшло успішно;
- Коли поточний імейл/пароль не збігаються

- Коли новий імейл вже існує в системі або новий пароль та повторений новий пароль не збігаються



The image shows a web form for changing a password. The title is 'Зміна паролю' (Change Password). It contains three input fields: 'Старий пароль:' (Old Password), 'Новий пароль:' (New Password), and 'Повторити пароль:' (Repeat Password). Each field has a placeholder text matching its label. Below the fields is a blue button labeled 'Змінити' (Change).

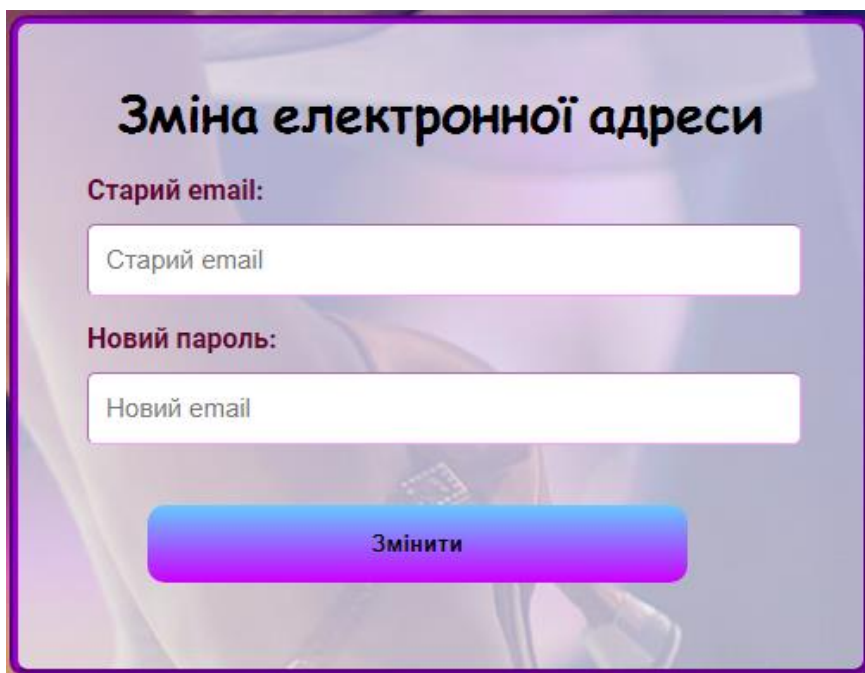
Рисунок 3.19 – Форма зміни паролю



The image displays three distinct validation messages in colored boxes:

- Успішно!** (Successful!) - Password changed. [Return to the editing page](#).
- Помилка!** (Error!) - Current password is incorrect.
- Помилка!** (Error!) - Data in the Password and Repeat password fields do not match.

Рисунок 3.20 – Валідаційні меседжи при зміні паролю



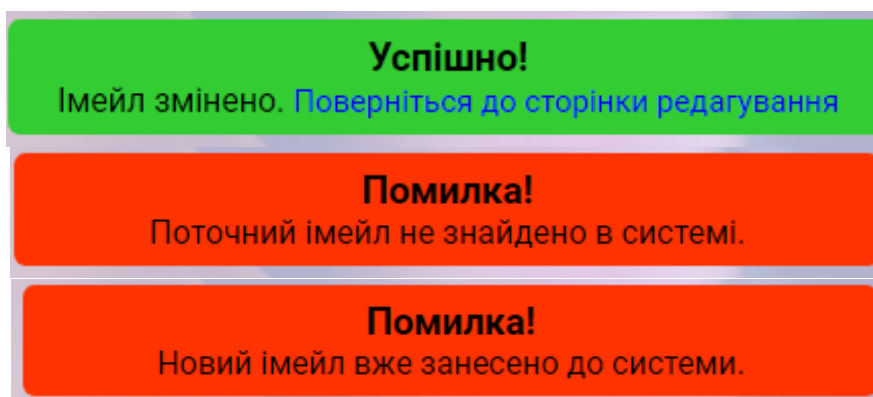
**Зміна електронної адреси**

Старий email:

Новий пароль:

**Змінити**

Рисунок 3.21 – Форма редагування імейлу



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

**Помилка!**  
Поточний імейл не знайдено в системі.

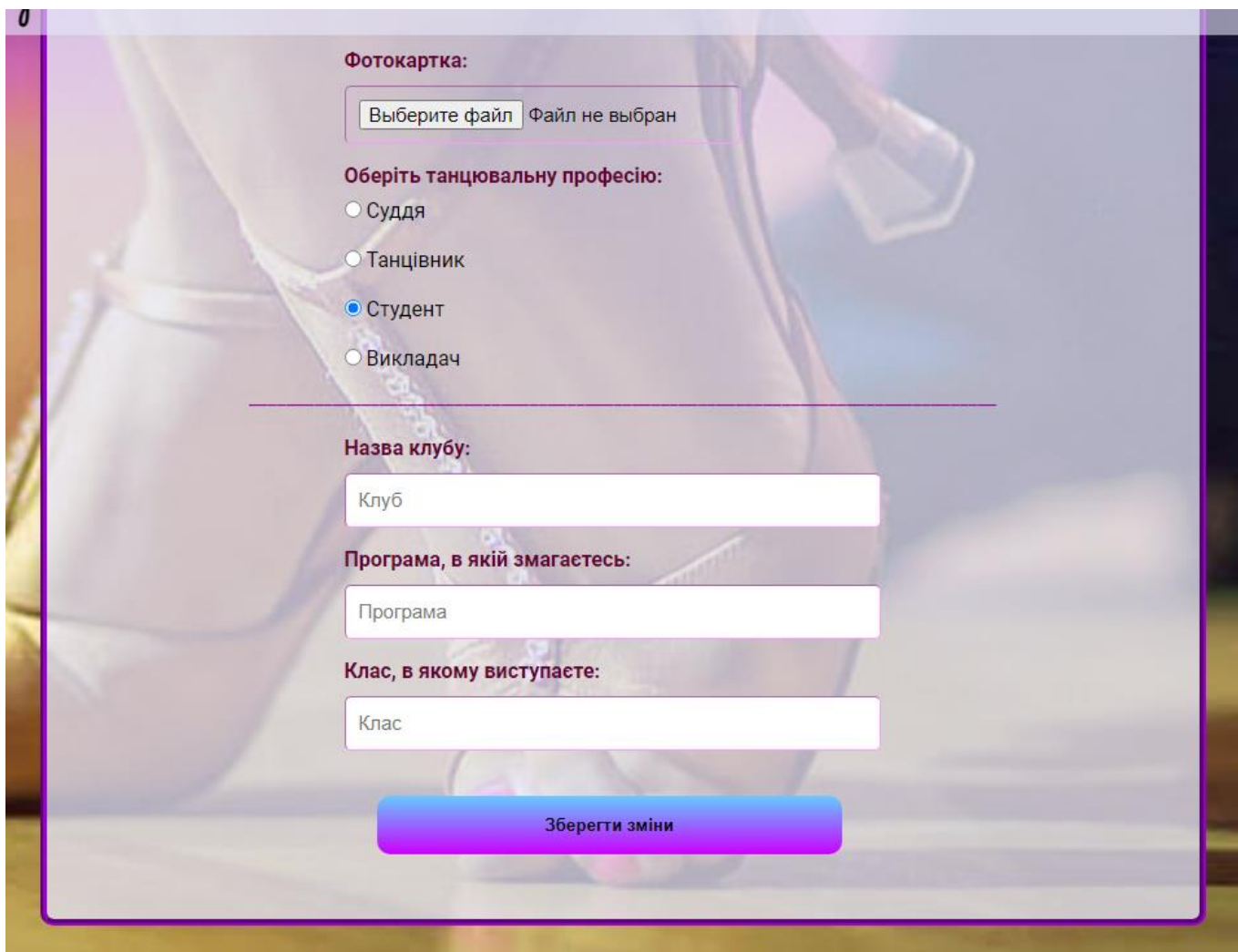
**Помилка!**  
Новий імейл вже занесено до системи.

Рисунок 3.22 – Модальні вікна під час зміни імейлу

Раніше, користувачі змінювали свій профіль відповідно до двох ролей, а зараз вже з'явилися нові 2 ролі, які в подальшому можна обирати разом з попередніми, які вже існували.

Тож, якщо зараз користувач обирає нову «Танцювальну професію», то в нього з'являються відповідні поля та власний профіль набуває змін також.

Якщо обрати танцювальну професію «Студент», то користувачу всього необхідно додатково заповнити 3 поля – назву клубу, танцювальну програму та клас, в якому займалися. Дану реалізацію можна побачити на рисунку 3.23.



**Фотокартка:**

Выберите файл    Файл не выбран

**Оберіть танцювальну професію:**

Суддя

Танцівник

Студент

Викладач

---

**Назва клубу:**

Клуб

**Програма, в якій змагаєтесь:**

Програма

**Клас, в якому виступаєте:**

Клас

Зберегти зміни

Рисунок 3.23 – Редагування профілю з вибором професії «Студент»

Якщо ж користувач обирає роль «Викладач», то в нього з'являються 3 інших додаткових поля, таких як – досвід(скільки років працює викладачем), клуб(або клуби в яких працювала людина) та звичайно написання резюме, короткого опису про себе. Це відображено на рисунку 2.24.

Танцівник  
 Студент  
 Викладач

---

**Досвід:**

Досвід

**Клуб:**

Клуб

**Напишіть своє резюме(додатковий опис):**

Резюме

Зберегти зміни

Рисунок 3.24 – Редагування профілю з вибором професії «Викладач»

Тепер вже після редагування власного профілю можливі аж 5 варіантів його вигляду, тобто що саме побачить користувач:

- Звичайний профіль користувача, який не обирає танцювальну професію, тому відображається коротка інформація (рис. 3.25);
- Профіль користувача-судді, з заповненим додатково резюме;
- Профіль користувача-танцівника, з доданою інформацією про клуб, танцювальну програму та клас;
- Профіль користувача-студента(рис.3.26), який додає коротку інформацію про свої танцювальні здібності, досвід;
- Профіль користувача-викладача(рис. 3.27), який описав свій досвід, клуби, де працював та додає своє коротке резюме.



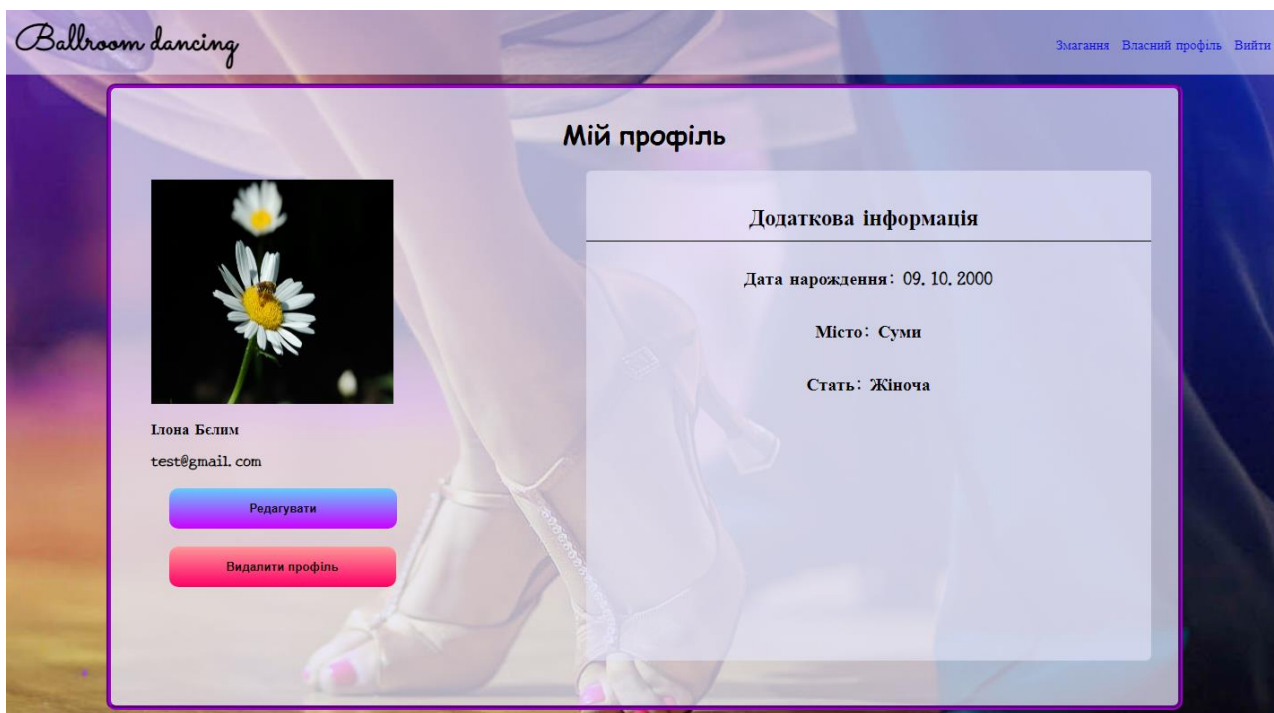


Рисунок 3.25 – Профіль звичайного користувача після першого редагування

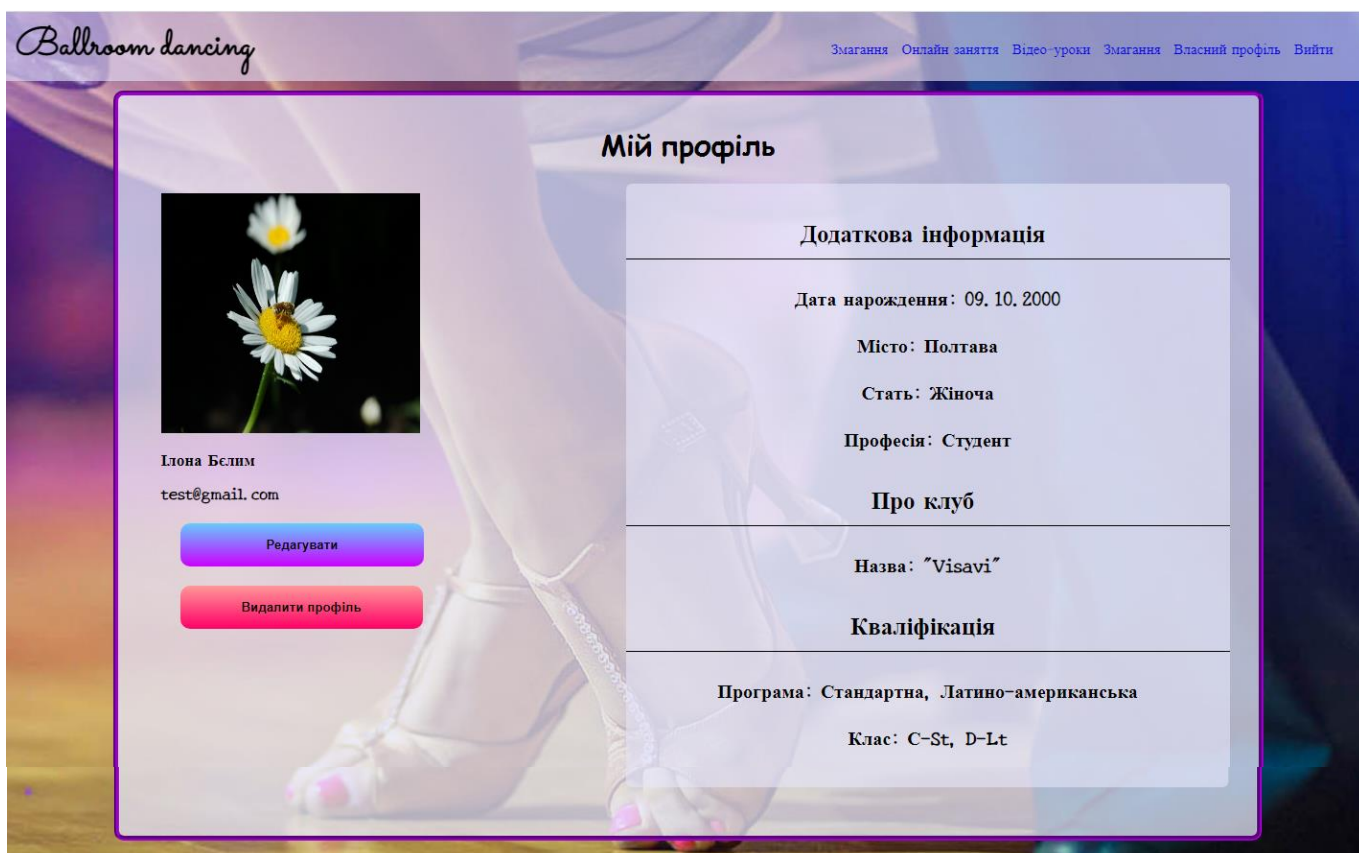



Рисунок 3.26 – Профіль користувача-студента після першого редагування

*Balroom dancing* Онлайн заняття | Відео-уроки | 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: Викладач

#### Кваліфікація

Досвід: 7 років викладання в 3 школах танців

Клуби: "Visavi", "Веселка", "Райдуга"

#### Резюме

З 5 років займалася бальними танцями, ще під час навчання почала викладати для молодших груп. На сьогоднішній день загалом викладаю 7 років, за плечима перемоги на Всеукраїнських та Міжнародних конкурсах

#### Додані відео-уроки

В вас поки ще немає доданих відео-уроків

[Додати відео-урок](#)

#### Заплановані онлайн уроки

В вас поки ще немає запланованих онлайн-уроків

[Запланувати онлайн-заняття](#)

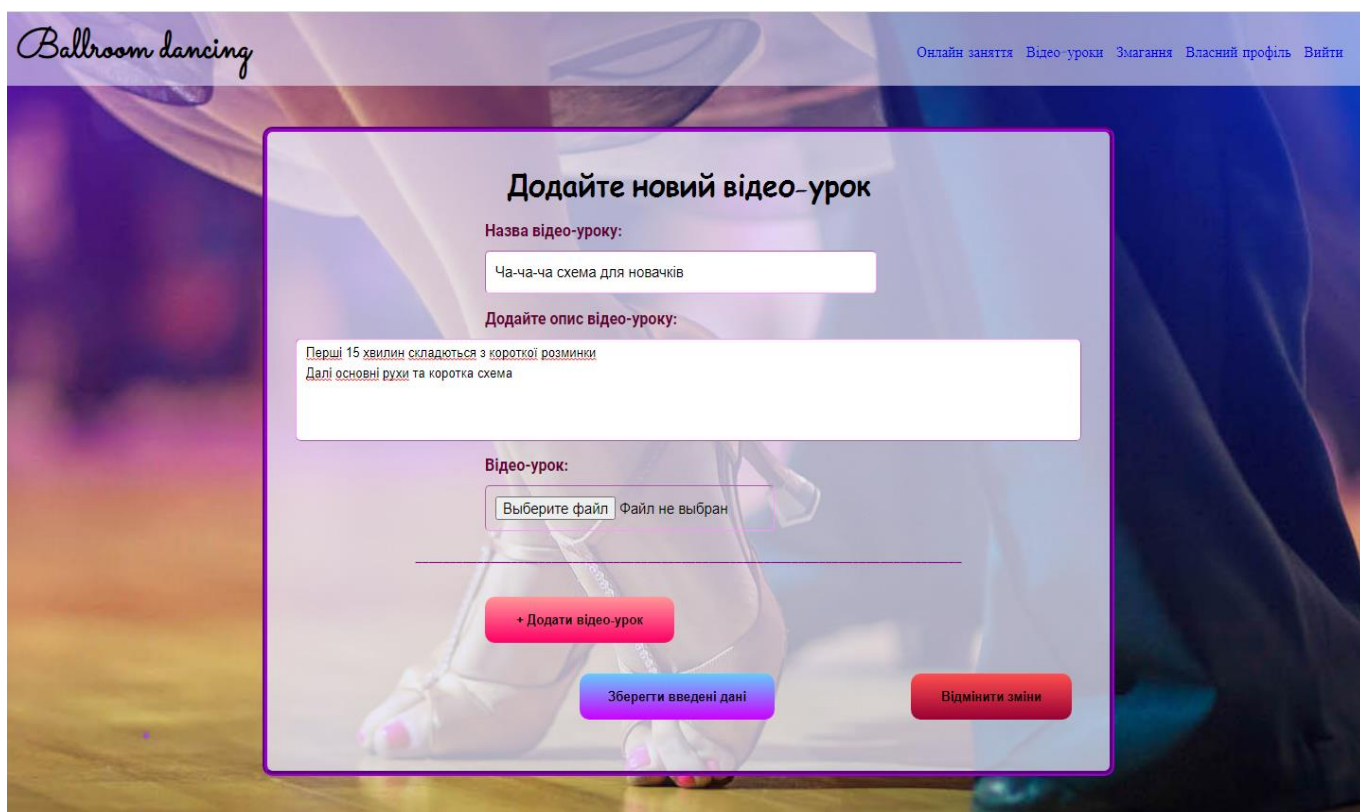
Рисунок 3.27 – Профіль користувача-викладача після редагування

Також вся ця інформація, яка була заповнена користувачами може бути відредагована, тобто звичайний профіль можна буде перетворити на один з чотирьох видів, а кожний з чотирьох видів можна буде зкомбінувати та перетворити на міксований профіль, наприклад, викладач-суддя, танцівник-студент.



Користувач-викладач має 2 додаткові можливості, додати відео-уроки з додатковим описом. Вони будуть з'являтися на сторінці усіх відео-уроків різних викладачів, які користувачі-студенти зможуть переглядати у будь-який час та маркувати як вподобані, щоб вони з'являлися у них у профілі для швидкого пошуку і перегляду.

За один раз викладач може додати як один так і декілька відео-уроків. Дана функція відображена на рисунку 3.28.



The screenshot shows a web interface for 'Ballroom dancing'. At the top left is the logo 'Ballroom dancing'. At the top right are navigation links: 'Онлайн заняття', 'Відео-уроки', 'Змагання', 'Власний профіль', and 'Вийти'. The main content is a modal form titled 'Додайте новий відео-урок'. The form contains the following elements:

- Назва відео-уроку:** A text input field containing 'Ча-ча-ча схема для новачків'.
- Додайте опис відео-уроку:** A text area containing the text: 'Перші 15 хвилин складються з короткої розминки' and 'Далі основні рухи та коротка схема'.
- Відео-урок:** A file selection area with a button 'Виберіть файл' and the text 'Файл не вибран'.
- At the bottom of the form are three buttons: '+ Додати відео-урок' (red), 'Зберегти введені дані' (blue), and 'Відмінити зміни' (red).

Рисунок 3.28 – Додавання одного нового відео-уроку викладачем

Ballroom dancing

Онлайн заняття Відео-уроки Змагання Власний профіль Вийти

### Додайте новий відео-урок

Назва відео-уроку:

Додайте опис відео-уроку:

Опис

Відео-урок:

 Файл не вибран

Рисунок 3.29 – Додавання одразу декількох відео-уроків викладачем

Також реалізовані можливість відредагувати додані відео-уроки, переглянувши їх детально та змінити необхідні відповідні дані. Після додавання хоча б одного відео-уроку строінка користувача-викладача набуває змін, у блоці відео-уроків починають відображатися прев'ю цих відео-уроків. Це можна переглянути на рисунку 3.30.

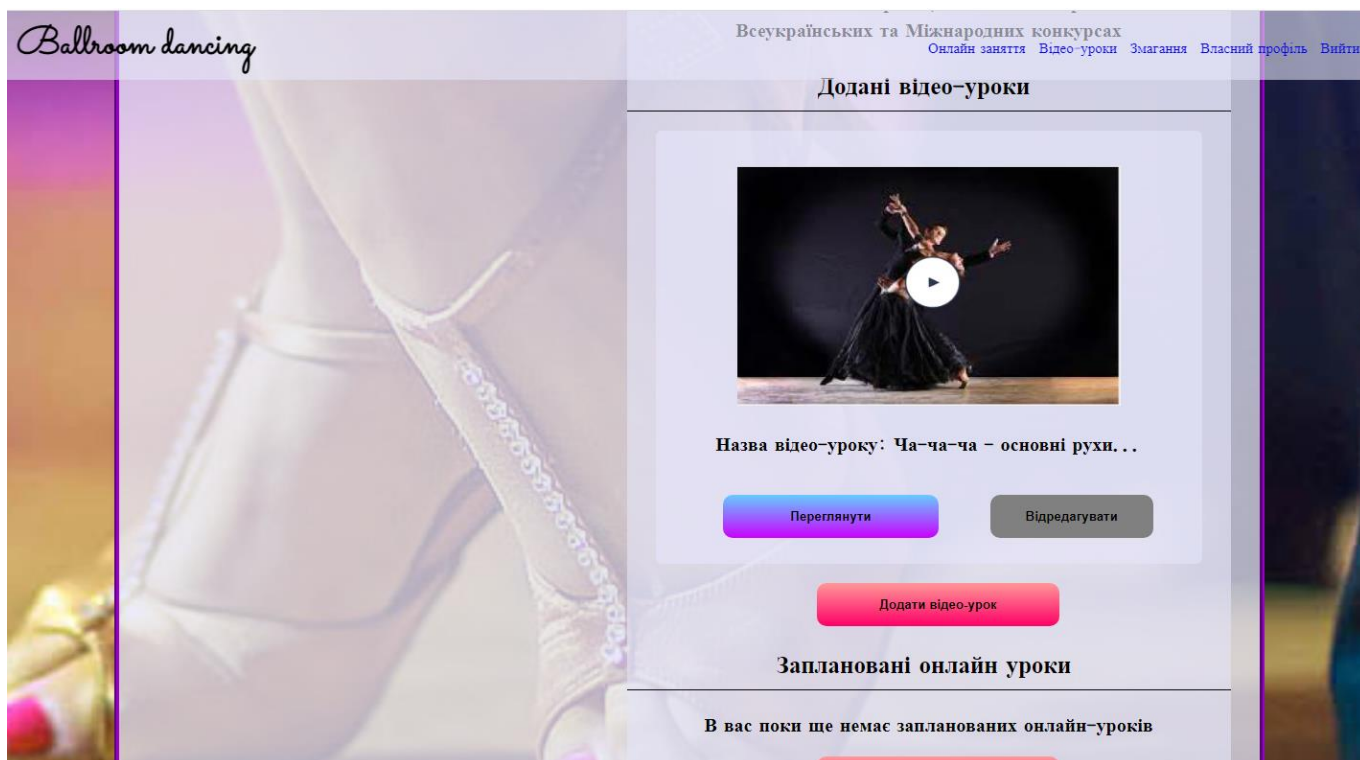


Рисунок 3.30 – Особиста сторінка користувача-викладача з доданим відео-уроком

Планування онлайн уроків відрізняється від додавання відео-уроків. Викладач заповнює поля:

- Назва онлайн-уроку;
- Опис онлайн-уроку;
- Обирають дату проведення онлайн-уроку;
- Обирають час проведення онлайн-уроку;

А далі автоматично система створює посилання, за яким викладач буде переходити в день уроку, щоб провести заняття. Це реалізовано, за допомогою інтеграції Twillio.

Переглянути приклад форми планування онлайн уроку можна на рисунку 3.31

*Ballroom dancing* Онлайн заняття | Відео-уроки | Змагання | Власний профіль | Вийти

### Заплануйте онлайн-урок

Назва онлайн-уроку:

Додайте опис онлайн-уроку:

Дата онлайн-уроку:

Час онлайн-уроку:

Згенероване посилання за яким переходити на онлайн-урок: [Lesson link](#)

Рисунок 3.31 – Форма планування онлайн уроку

*Ballroom dancing* Онлайн заняття | Відео-уроки | Змагання | Власний профіль | Вийти

Назва відео-уроку: Ча-ча-ча - основні рухи...

### Заплановані онлайн уроки

Назва онлайн-уроку: Ча-ча-ча для хобі класу

Посилання на урок: [Lesson link](#)

Запланована дата: 23. 02. 2023

Запланований час: 11:00

copyrightinh

Рисунок 3.32 – Особиста сторінка викладача із запланованим онлайн-уроком

Як тільки заплановано хоча б один майбутній онлайн-урок, то це одразу ж відображається на особистій сторінці викладача(рис. 3.32). Головним є те, що викладач бачить свої майбутні заплановані заняття та може за посиланням одразу ж переходити, щоб розпочати це заняття.

Звичайно у викладача є можливість зробити якісь із запланованими заняттями, наприклад змінити дату чи час або взагалі видалити план. І також звичайно може запланувати ще додаткові заняття.

Для студентів, які хочуть навчатися створено 2 окремі сторінки – «Онлайн заняття»(рис.3.33) та «Відео уроки»(3.34), для зручного доступу вони розміщені в меню-хедері. В залежності того, що користувач-студент хоче переглянути, він переходить на відповідні сторінки та або переглядає вже записані відео-уроки або ж записується на заплановані онлайн-заняття.

Для того, щоб почати переглядати відео-урок необхідно просто перейти на детальну сторінку одного відео-уроку(рис.3.35) за допомогою кнопки «Переглянути». Відкриється сторінка відео-уроку, де можна прочитати повністю опис уроку та побачити хто є автором цього уроку. Також за бажанням можна залишити коментар-відгук про заняття. Щоб відео розпочалось необхідно просто натиснути на кнопку «Play».

Для того, щоб подивитися повну інформацію про майбутній онлайн урок потрібно просто натиснути на кнопку «Переглянути» і ви потрапите на детальну сторінку одного майбутнього онлайн-заняття. Але онлайн-відео буде доступне до перегляду тільки в запланований день та час, коли викладач розпочне свій урок. Користувач-студент буде мати змогу його відкрити і дивитися у реальному часі. Також, для того, щоб студент не забув про заняття, в нього є змога додати його в закладки за допомогою кнопки «Додати до задач» і цей запланований онлайн-урок з'явиться в особистому кабінеті користувача-студента(рис.3.37).




Ballroom dancing

Онлайн заняття Відео-уроки Змагання Власний профіль Вийти

### Відео-уроки


#### Ча-ча-ча для хобі кл. . .



Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .

Переглянути


#### Самба для класу



Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .

Переглянути


#### Віденський вальс



Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .

Переглянути

#### Джайв N клас



Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .

Переглянути

< 1 >

Рисунок 3.33 – Загальна сторінка всіх відео-уроків

Ballroom dancing

Онлайн заняття Відео-уроки Змагання Власний профіль Вийти

### Заплановані онлайн-уроки

<p><b>Ча-ча-ча для хобі кл. . .</b></p> <p>Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .</p> <p>Викладач: Ілона Белім</p> <p>Дата: 23. 02. 2023</p> <p>Час: 11:00</p> <p><a href="#">Переглянути</a> <a href="#">Додати в задачі</a></p>	<p><b>Самба для класу</b></p> <p>Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .</p> <p>Викладач: Ілона Белім</p> <p>Дата: 23. 03. 2023</p> <p>Час: 12:00</p> <p><a href="#">Переглянути</a> <a href="#">Додати в задачі</a></p>
<p><b>Віденський вальс</b></p> <p>Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .</p> <p>Викладач: Ілона Белім</p> <p>Дата: 27. 02. 2023</p> <p>Час: 14:00</p> <p><a href="#">Переглянути</a> <a href="#">Додати в задачі</a></p>	<p><b>Джайв N клас</b></p> <p>Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. . .</p> <p>Викладач: Ілона Белім</p> <p>Дата: 30. 03. 2023</p> <p>Час: 11:00</p> <p><a href="#">Переглянути</a> <a href="#">Додати в задачі</a></p>


< 1 >

Рисунок 3.34 – Загальна сторінка всіх запланованих онлайн-уроків

Ballroom dancing


Онлайн заняття Відео-уроки Змагання Власний профіль Вийти

### Відео-урок



#### Ча-ча-ча для хобі класу

Викладач: Ілона Белім



Опис: Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. Звичайно буде розказано як правильно рахувати музику у цьому танці та як танцювати під неї. Заняття включас в себе рухи як для жіночої партії, так і для чоловічої. Впоратися з основними рухами зможуть всі!

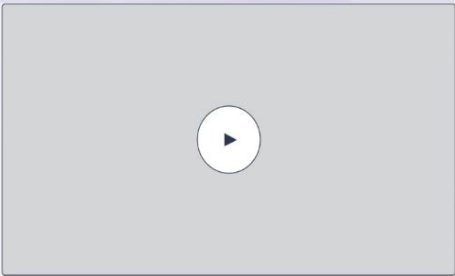
Залишити відгук

Рисунок 3.35 – Детальна сторінка одного відео-уроку

Ballroom dancing

Онлайн заняття Відео-уроки Змагання Власний профіль Вийти


### Онлайн-урок



Онлайн урок розпочнеться, коли настане його дата та час, поки не можна переглядати

#### Ча-ча-ча для хобі класу

Викладач: Ілона Белім



Дата: 23. 02. 2023

Час: 11:00

Опис: Перші 15 хвилин включають в себе коротку розминку, далі розтяжка і основні рухи танцю ча-ча-ча, такі як шосе, Нью-йорк. Звичайно буде розказано як правильно рахувати музику у цьому танці та як танцювати під неї. Заняття включас в себе рухи як для жіночої партії, так і для чоловічої. Впоратися з основними рухами зможуть всі!

Додати в закладки


Рисунок 3.36 – Детальна сторінка одного майбутнього онлайн-уроку



Ballroom dancing

Онлайн заняття | Відео уроки | Змагання | Власний профіль | Вийти

### Мій профіль



Ілона Белім  
test@gmail.com

Редагувати

Видалити профіль

#### Додаткова інформація

Дата народження: 09. 10. 2000

Місто: Полтава

Стать: Жіноча

Професія: Студент

#### Про клуб

Назва: "Visavi"

#### Кваліфікація

Програма: Стандартна, Латино-американська

Клас: C-St, D-Lt

#### Онлайн уроки, на які ви збираєтесь

Назва онлайн-уроку: Ча-ча-ча для хобі класу

Посилання на урок: [Lesson link](#)

Запланована дата: 23. 02. 2023

Запланований час: 11:00

Переглянути

Відмінити запис

Рисунок 3.37 – Профіль користувача-студента з відображеними уроками, на які він збирається

## ВИСНОВКИ

Під час виконання даної кваліфікаційної роботи був виконаний глибокий аналіз та дослідження усіх існуючих аспектів розробки інформаційної технології проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи.

Актуальність даної роботи зумовлена необхідністю підтримки та розвитку культури спорту в тяжкі часи в Україні, кількість професійних танцівників зростає, а охочих розпочати займатися танцями ще більше, тому дана система надає змогу користувачам розвивати себе, а в свою чергу і країну, незважаючи на зовнішні умови. Даний проект представляє собою цільну систему, в якій повністю реалізований процес необхідний для професійних танцівників, можливість реєстрації на змагання, перегляд результатів, онлайн заходи та тренування, це все надасть змогу продовжувати свою танцювальну діяльність, незважаючи на місце розташування користувача, використовуючи лише девайс та Інтернет.

При реалізації даної роботи було розглянуто такі питання:

- виконано глибокий аналіз продуктів-аналогів, знайдено їх недоліки та виправлено їх в даній системі, виявлені переваги та додані нові механізми та можливості до реалізованого проекту;
- створено велику кількість різних діаграм для відображення структурної архітектури системи та її функцій, розроблено діаграму варіантів-використання за для наглядності показу основних можливостей користувача, також було створено ER-діаграму для демонстрації зв'язків в БД;
- розроблено гнучкий та сучасний UI/UX дизайн, який запроваджено в дану систему.

Отже, інформаційна технологія проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи є наймовірним проектом, котрий робить свій вклад в розвиток спорту в Україні, надаючи можливість клубам створювати змагання та проводити їх, танцівникам вести «Танцювальну книгу», реєструватися на заходи та відвідувати їх онлайн, суддям оцінювати виступи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Довідник по HTML тегам: [Електронний ресурс]. URL: <https://css.in.ua/html/tags>
2. Підручник PHP: [Електронний ресурс]. URL: <http://www.apserver.org.ua/peregl.php?d=view&tid=6>
3. Бази даних MySQL. Навчальний посібник : [Електронний ресурс]. URL: <https://bohdan-books.com/catalog/book/98908/>
4. Twillio: [Електронний ресурс]. URL: <https://www.twilio.com/>
5. Проведені змагання зі спортивно-бальних танців: [Електронний ресурс]. URL: <https://www.audsf.org/competition/2022>
6. Статистика змагань зі спортивно-бальних танців: [Електронний ресурс]. URL: <https://flymark.com.ua/Competition>
7. Google Fonts: [Електронний ресурс]. URL: <https://fonts.google.com/>
8. MySQL: [Електронний ресурс]. URL: <https://metanit.com/sql/mysql/>
9. Bootstrap · The most popular HTML, CSS, and JS library in the world: [Електронний ресурс]. URL: <https://getbootstrap.com/>
10. CSS Tutorial - W3Schools: [Електронний ресурс]. URL: <https://www.w3schools.com/css/>
11. jQuery Introduction - W3Schools: [Електронний ресурс]. URL: [https://www.w3schools.com/jquery/jquery\\_intro.asp](https://www.w3schools.com/jquery/jquery_intro.asp)
12. jQuery UI: [Електронний ресурс]. URL: <https://jqueryui.com/>
13. jQuery: [Електронний ресурс]. URL: <https://jquery.com/>
14. Сучасний підручник з JavaScript: [Електронний ресурс]. URL: <https://uk.javascript.info/>
15. JavaScript підручник. Основи веб-програмування : [Електронний ресурс]. URL: <https://w3schoolsua.github.io/js/index.html#gsc.tab=0>

## **ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ**

### **ТЕХНІЧНЕ ЗАВДАННЯ**

**на створення програмного продукту «Інформаційної технології проектування платформи для навчання користувачів клієнт-орієнтованої автоматизованої системи»**

**Суми 2022**

# **1 Призначення й мета створення інформаційної технології проектування**

## **1.1 Призначення**

Даний програмний продукт розроблений для того щоб мати можливість віддалено навчатися та викладати спортивно-бальні танці, додавати відео-заняття та проводити онлайн заняття в реальному часі.

## **1.2 Мета створення**

Створення можливості навчатися з будь-якої точки світу та в будь-який час, дати можливість проводити заняття онлайн для професійних викладачів та мати змогу в реальному часі поспілкуватися на заняттях не виходячи з дому.

## **1.3 Цільова аудиторія**

Головними об'єктами для кого розроблена дана автоматизована система:

- Викладачі спортивно-бальних танців – додають відео-заняття з детальним описом для навчання студентів правильним рухам та їх назвам, планують та проводять онлайн заняття в реальному часі для студентів;
- Студенти спортивно-бальних танців - мають можливість переглядати та навчатися по доданим відео, також відвідувати онлайн заняття.

# **2 Вимоги до інформаційної технології проектування**

## **2.1 Загальні вимоги**

Розроблюваний сервіс має бути web-додатком, який буде складатися з взаємозалежних модулів, кожний з яких буде виконувати свої функціональні вимоги. UX/UI дизайн повинен бути інтуїтивно зрозумілим будь-якому звичайному користувачу мережі Інтернет, шрифт, розмір текстового наповнення, розміри та розміщення блоків мають бути приємними на вигляд та збалансованими на сторінці. Також сервіс має бути крос-платформним та крос-браузерним, і адаптивним.

## **2.2 Функціональні вимоги**

Кожен користувач має мати змогу зареєструватися в системі, далі відповідно під зареєстрованими кredsами мати змогу увійти до системи. Якщо забув пароль, то має бути реалізований функціонал відновлення паролю через пошту.

Наступною функцією є можливість відредагувати власний профіль, змінити дані занесені під час реєстрації, а також заповнити додаткові поля відповідно до обраної ролі – викладач чи студент.

Якщо обрано роль – викладач, то далі у цього користувача має бути змога додати відео-урок з описом, а також відредагувати його. Й звичайно має бути можливість запланувати та провести онлайн-заняття в обраний день та час, його можна відредагувати до початку заняття тільки за декілька днів, не пізніше.

Якщо ж обрано роль – студент, то даний тип користувачів має змогу переглядати відео-уроки, додавати собі в заплановані задачі майбутні онлайн-заняття та переглядати безпосередньо онлайн-заняття в реальному часі.

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

## 2.3.1 Відображення сторінки зі всіма запланованими майбутніми уроками

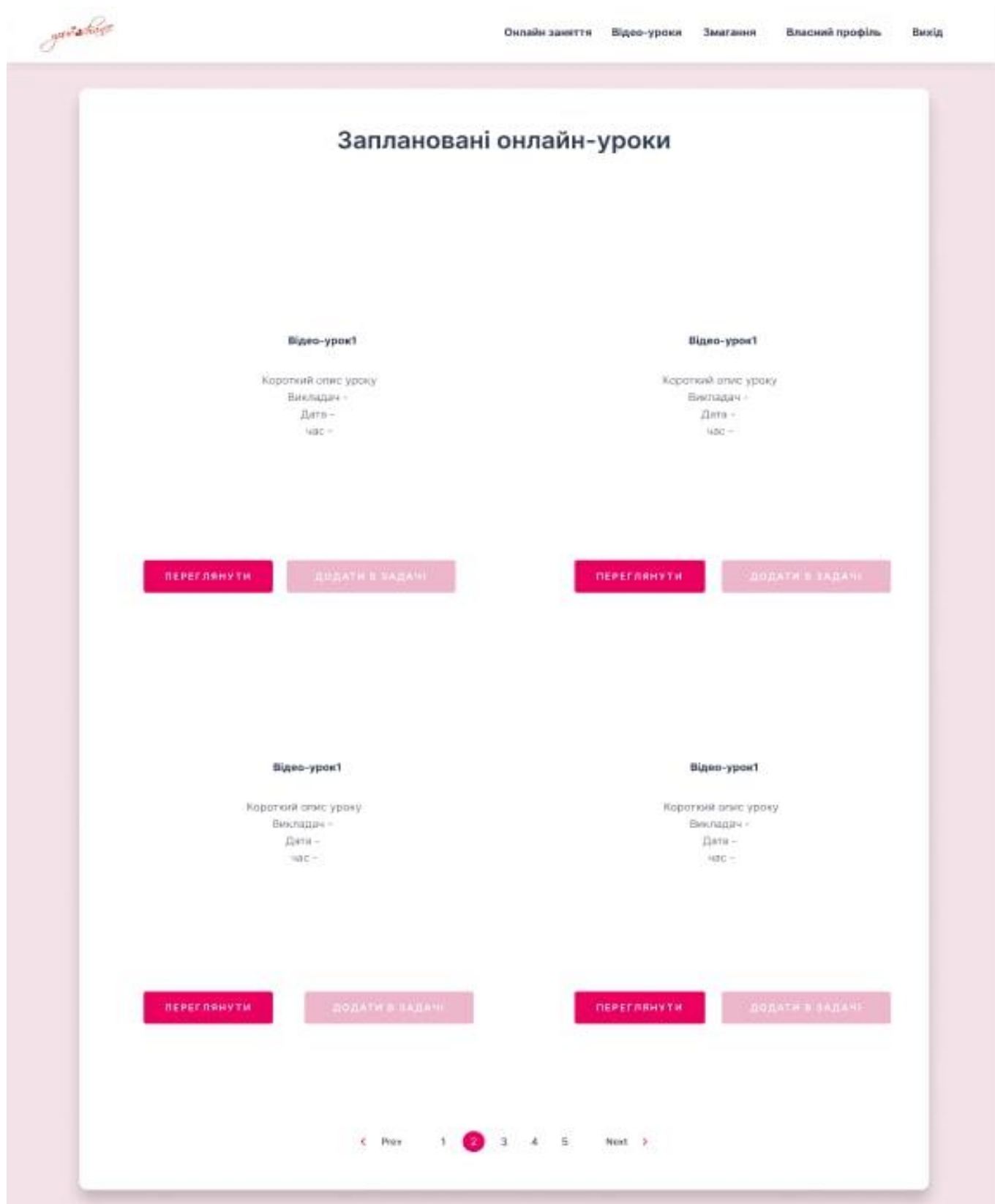


Рисунок 2.3.1.1 – Дизайн сторінки зі запланованими онлайн заняттями

## 2.3.2 Відображення сторінки зі всіма доданими відео-уроками

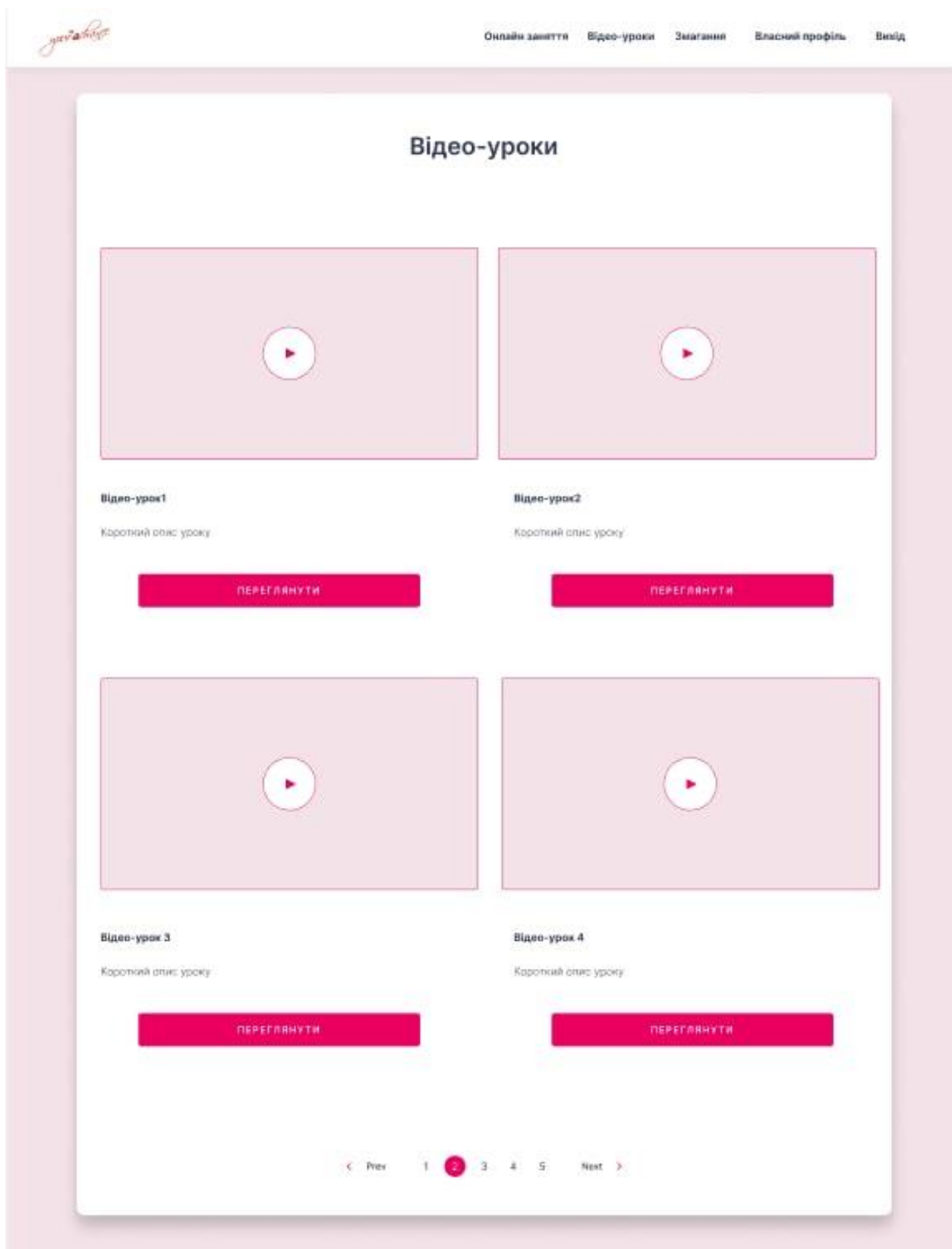


Рисунок 2.3.2.1 – Дизайн всіх доданих відео-уроків







### 3 Склад та зміст необхідних робіт з реалізації інформаційної технології проектування

Таблиці 3.1 є відображенням запланованих робіт, які необхідні для створення вихідного продукту, та час, який ці роботи займуть.

Таблиця 3.1 – План роботи та необхідні етапи

№	Склад та зміст робіт	Строк розробки
1	Design(проектування): Розробка структури БД та інформаційної технології.	4 днів
2	Розробка інтерфейсу: Продумування зручного UX/UI дизайну, вибір кольорової гами, верстання та підв'язка стилів для кожної сторінки.	9 дні
3	Програмування модулів та підв'язка їх з бековою частиною :  <ul style="list-style-type: none"> <li>• розробка модуля редагування власного профілю відповідно до нових ролей;</li> <li>• розробка модуля планування онлайн уроків та додавання їх до системи;</li> <li>• розробка модуля редагування запланованих онлайн занять;</li> <li>• розробка модуля додавання та редагування відео-заняття;</li> <li>• розробка модуля перегляду всіх майбутніх онлайн занять або всіх записаних онлайн занять(відео-уроків).</li> </ul>	3 дні  4 дні  2 дні  2 дні  3 дні  4 дні

4	Модульне, adhoc та інтеграційне тестування. УАТ тестування	7 днів
	<b>Загальний час на реалізацію</b>	<b>38 днів</b>

## ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ

SQL запит на створення нової 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;

```

```

Create table `students` (
  `student_id` Int NOT NULL,

```

```

    `user_id` Int NOT NULL,
    `s_club` Varchar(500),
    `s_class` Varchar(500),
    Primary Key (`student_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `teachers` (
    `teacher_id` Int NOT NULL,
    `user_id` Int NOT NULL,
    `t_club` Varchar(500),
    `experience` Varchar(500),
    `description` Varchar(500),
    Primary Key (`teacher_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `lessons` (
    `lesson_id` Int NOT NULL,
    `lesson_name` Varchar(40) NOT NULL,
    `lesson_description` Varchar(20),
    `video` Varchar(100) DEFAULT NULL,
    Primary Key (`lesson_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `online_lessons` (
    `o_lesson_id` Int NOT NULL,
    `o_lesson_name` Varchar(40) NOT NULL,
    `o_lesson_description` Varchar(20),
    `o_lesson_date` Date,
    `o_lesson_time` Time,
    `o_lesson_link` Varchar(100) DEFAULT NULL,
    Primary Key (`o_lesson_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `stud_request` (
    `stud_request_id` Int NOT NULL,
    `student_id` Int NOT NULL,
    `lesson_id` Int NOT NULL,
    Primary Key (`stud_request_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `tech_request` (
    `tech_request_id` Int NOT NULL,
    `teacher_id` Int NOT NULL,
    `lesson_id` Int NOT NULL,
    Primary Key (`tech_request_id`))
ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

Create table `online_stud_request` (
    `o_stud_request_id` Int NOT NULL,
    `student_id` Int NOT NULL,
    `o_lesson_id` Int NOT NULL,
    Primary Key (`o_stud_request_id`))

```

```
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
Create table `online_tech_request` (
  `o_tech_request_id` Int NOT NULL,
  `teacher_id` Int NOT NULL,
  `o_lesson_id` Int NOT NULL,
  Primary Key (`o_tech_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 `competitons` (`competition_id`) on delete restrict on update restrict;
```

```
Alter table `j_request`
add Foreign Key (`competition_id`)
references `competitons` (`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;
```

```
Alter table `students`
add Foreign Key (`user_id`)|
```

```
references `users` (`user_id`) on delete restrict on update restrict;
```

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

```
Alter table `stud_request`  
add Foreign Key (`lesson_id`)  
references `competitons` (`lesson_id`) on delete restrict on update restrict;
```

```
Alter table `stud_request`  
add Foreign Key (`student_id`)  
references `competitons` (`student_id`) on delete restrict on update restrict;
```

```
Alter table `o_stud_request`  
add Foreign Key (`o_lesson_id`)  
references `competitons` (`lesson_id`) on delete restrict on update restrict;
```

```
Alter table `o_stud_request`  
add Foreign Key (`student_id`)  
references `competitons` (`student_id`) on delete restrict on update restrict;
```

```
Alter table `tech_request`  
add Foreign Key (`lesson_id`)  
references `competitons` (`lesson_id`) on delete restrict on update restrict;
```

```
Alter table `tech_request`  
add Foreign Key (`teacher_id`)  
references `competitons` (`student_id`) on delete restrict on update restrict;
```

```
Alter table `o_tech_request`  
add Foreign Key (`o_lesson_id`)  
references `competitons` (`lesson_id`) on delete restrict on update restrict;
```

```
Alter table `o_tech_request`  
add Foreign Key (`tech_id`)  
references `competitons` (`tech_id`) on delete restrict on update restrict;
```

```
COMMIT;
```



Лістинг коду сторінки всіх відео-уроків:

```

<?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" xml: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"/>

    <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-video-lessons.css">

</head>
<header>

    <ul>
        <li> <a class = "h1">Ballroom dancing</a></li>
        <li> <a class= "nav-menu" href="#features">Онлайн заняття</a></li>
        <li> <a class= "nav-menu" href="#features">Відео-уроки</a></li>
        <li> <a class= "nav-menu" href="#features">Змагання</a></li>
        <li> <a class= "nav-menu" href="#order">Власний профіль</a></li>
        <li> <a class= "nav-menu" href="#order">Вийти</a></li>
    </ul>

    <?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>
    </ul>
} 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>';
}
?>

</header>

<body>

    <section>
        <form action="" name="ordering" method="GET">

            <div>
                <h2>Відео-уроки</h2>

                <div id="menu">
                    <div class="comp">
                        <p class="inform1">Ча-ча-ча для хобі кл...</p>
                        </img>
                        <p class="inform">Перші 15 хвилин включають в себе коротку
розминку, далі розтяжка і основні рухи танцю ча-ча-ча,
такі як шосе, Нью-Йорк...</p>
                        <button class="submit" type="submit">Переглянути</button>
                    </div>

                    <div class="comp">
                        <p class="inform1">Віденський вальс</p>
                        </img>
                        <p class="inform">Перші 15 хвилин включають в себе коротку
розминку, далі розтяжка і основні рухи танцю ча-ча-ча,
такі як шосе, Нью-Йорк...</p>
                        <button class="submit" type="submit">Переглянути</button>
                    </div>

                    <!--<button class="delete" type="submit">Переглянути
змагання</button-->
                </div>
            </form>

```

```

        </div>
    </div>
    <div id="content">
        <!--<div class="super">-->
        <!--<h3>Майбутні змагання</h3>-->

                <div class="comp">
                    <p class="inform1">Самба для класу</p>
                    </img>
                    <p class="inform">Перші 15 хвилин включають в себе коротку
розминку, далі розтяжка і основні рухи танцю ча-ча-ча,
                    такі як шосе, Нью-йорк...</p>
                    <button class="submit" type="submit">Переглянути</button>
                </div>

                <div class="comp">
                    <p class="inform1">Джайв N класс</p>
                    </img>
                    <p class="inform">Перші 15 хвилин включають в себе коротку
розминку, далі розтяжка і основні рухи танцю ча-ча-ча,
                    такі як шосе, Нью-йорк...</p>
                    <button class="submit" type="submit">Переглянути</button>
                </div>

                </div>

        </div>

        <!--</div>-->
    </div>
    </div>
    <p class="count">< 1 ></p>

</form>

</section>
<?php
    $sql_note = mysql_query("SELECT `lesson_name`, `lesson_description`, `lesson_teacher`,
`lesson_image` from `lessons`
                                Order by `lessonettition_id` DESC Limit 10");
    while ($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-lesson"></i> <a
href="lessons.php?lesson_id=.$result_note["lesson_id"].">.$result_note["lesson_name"].'
.$result_note["lesson_date"].</a> </p>
                                                                    <p><i class="fas
fa-lesson"></i> <a
href="lessons.php?lesson_id=.$result_note["lesson_id"].">.$result_note["lesson_name"].'
.$result_note["lesson_date"].</a> </p>
                                                                    <p><i class="fas
fa-lesson"></i> <a
href="lessons.php?lesson_id=.$result_note["lesson_id"].">.$result_note["lesson_name"].'
.$result_note["lesson_date"].</a> </p>
                                                                    <p><i class="fas fa-lesson"></i> <a
href="lessons.php?lesson_id=.$result_note["lesson_id"].">.$result_note["lesson_name"].'
.$result_note["lesson_date"].</a> </p>
                                                                    </div>
                                                                    </div>
                                                                    </li>
                                                                    </ul>
                                                                    </div>;
                                                                    ?>

<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> <p>copyrightinh</p></div>

</footer>
</html>

```