

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

**Інститут (факультет) Автоматизації і комп'ютерних систем  
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки**

**«До захисту в ЕК»**  
Директор інституту(декан факультету)  
Андрій ФОРСЮК  
(підпис) (ім'я та прізвище)

« 2 » червня 2025р.

**«До захисту допущено»**  
Завідувач кафедри  
Сергій ГРИБКОВ  
(підпис) (ім'я та прізвище)

« 2 » червня 2025р.

**КВАЛІФІКАЦІЙНА РОБОТА  
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Інформаційні системи та штучний інтелект

на тему: Розроблення вебсайту для спортивно-культурного комплексу  
"Софіївський"

Виконав: здобувач 4 курсу, групи КН-4-4

Височанський Даниїл Ігорович  
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник Харкянен Олена Валеріївна  
(прізвище, ім'я, по батькові повністю) (підпис)

Консультанти \_\_\_\_\_  
(ім'я та прізвище) (підпис)

\_\_\_\_\_ (ім'я та прізвище) (підпис)

\_\_\_\_\_ (ім'я та прізвище) (підпис)

Рецензент \_\_\_\_\_  
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач \_\_\_\_\_  
(підпис)

Київ - 2025р.

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем  
 Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки  
 Освітній ступінь бакалавр  
 Спеціальність 122 «Комп'ютерні науки»  
 Освітньо-професійна програма Інформаційні системи та штучний інтелект

## ЗАТВЕРДЖУЮ

Завідувач кафедри  
Інформаційних технологій, штучного  
інтелекту і кібербезпеки  
Сергій ГРИБКОВ

“ 28 ” квітня 2025 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

**Височанського Даниїла Ігоровича**

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення вебсайту для спортивно-культурного комплексу  
"Софіївський"

керівник роботи **Харкянен Олена Валеріївна, к.т.н., доцент**  
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 28 квітня 2025 року № 254-кс

2. Строк подання здобувачем роботи 30.05.2025 р.

3. Вихідні дані до роботи

Інформація про СКК «Софіївський», організаційна структура, посадові  
обов'язки працівників, шаблони документів реєстрації, розкладу та  
звернень

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження предметної області та постановка задачі

2. Технічне завдання та проектування

3. Проектування, створення та апробація програмного забезпечення

5. Перелік графічного матеріалу

1. Схема бази даних.

2. Архітектура вебсайту.

3. Діаграма Ганта.

4. Скріншоти інтерфейсу системи.

5. Функціональна модель діяльності СКК «Софіївський».

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Харкянен Олена Валеріївна		
2	Харкянен Олена Валеріївна		
3	Харкянен Олена Валеріївна		

7. Дата видачі завдання 28.04.2025**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі. Визначення цілей, завдань та вимог до системи	28.04.2025 – 01.05.2025	Виконано
2	Аналіз предметної області. Огляд систем-аналогів	02.05.2025– 07.05.2025	Виконано
3	Реалізація основних модулів системи та проектування бази даних	08.05.2025– 11.05.2025	Виконано
4	Підготовка пояснювальної записки, презентації	12.05.2025– 17.05.2025	Виконано
5	Оформлення відгуку, рецензії, подання роботи	19.05.2025– 22.05.2025	Виконано

Здобувач

(підпис)

Керівник роботи

(підпис)

Височанський Д.І.

(прізвище та ініціали)

Харкянен О. В.

(прізвище та ініціали)

## АНОТАЦІЯ

У кваліфікаційній роботі здійснено розробку вебсайту для Спортивно-Культурного Комплексу «Софіївський», що забезпечує ефективну цифрову комунікацію між закладом і користувачами. Вміщує 3 розділи загальним обсягом 77 сторінок, містить 32 рисунки, 3 таблиці та 30 посилань на використані джерела та два додатки.

У ході дослідження проаналізовано цільову аудиторію, визначено функціональні та нефункціональні вимоги, побудовано архітектуру системи, реалізовано користувацький інтерфейс та адміністративну частину.

Вебсайт адаптовано до мобільних пристроїв, передбачено механізми зворотного зв'язку та онлайн-запису. Проведено тестування працездатності системи у реальних умовах.

**Ключові слова:** ВЕБСАЙТ, ІНФОРМАЦІЙНА СИСТЕМА, СПОРТИВНО-КУЛЬТУРНИЙ КОМПЛЕКС, ЦИФРОВІЗАЦІЯ, АДАПТИВНИЙ ІНТЕРФЕЙС.

## ABSTRACT

The project presents the development of a website for the Sofiivskyi Sports and Cultural Complex that enables effective digital communication between the institution and its users. comprises 3 chapters spanning a total of 77 pages. It incorporates 32 figures, 3 tables, and 30 cited references and two additional chapters.

The study includes an analysis of the target audience, identification of functional and non-functional requirements, system architecture design, implementation of both the user interface and the administrative panel.

The website is adapted for mobile devices and includes feedback mechanisms and online registration features. The system was tested under real-world conditions.

**Keywords:** WEBSITE, INFORMATION SYSTEM, SPORTS AND CULTURAL COMPLEX, DIGITALIZATION, ADAPTIVE INTERFACE.

## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1. Аналіз діяльності СКК «Софіївський» .....	9
1.2. Аналіз цільової аудиторії та її інформаційних потреб.....	14
1.3. Аналіз нинішнього стану комп'ютеризації СКК «Софіївський».....	23
1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів.....	25
1.5. Аналіз аналогічних веб-рішень і порівняння їх функціоналу .....	28
1.6. Розрахунок техніко-економічного обґрунтування впровадження створюваного програмного забезпечення .....	33
1.7. Обґрунтування необхідності проектування та розробки вебсайту для СКК «Софіївський».....	35
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ.....	37
2.1. Загальна характеристика системи .....	37
2.2. Функціональні вимоги до веб-сайту .....	38
2.3. Нефункціональні вимоги .....	42
2.4. Архітектурні обмеження та середовище виконання .....	44
2.5. Склад і зміст робіт по створенню системи .....	46
2.6. Вимоги до структури та дизайну інтерфейсу користувача.....	47
2.7. Порядок контролю і приймання системи .....	49
2.8. Вимоги до складу і змісту робіт із підготовки до введення системи в дію .....	50
2.9. Вимоги до документації .....	50
2.10. Джерела розробки .....	50
РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	52

	7
3.1. Вибір інструментальних засобів розробки .....	52
3.2. Проєктування архітектури сайту .....	53
3.3. Розробка основних модулів веб-сайту .....	56
3.4. Інструкція користувача.....	58
3.5. Тестування програмного продукту .....	65
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	77

## ВСТУП

Актуальність теми дослідження зумовлена зростаючою потребою у цифровізації інформаційного простору закладів культурно-спортивного спрямування, які дедалі активніше впроваджують сучасні засоби комунікації для взаємодії з громадськістю. У цьому контексті створення вебсайту для Спортивно-Культурного Комплексу "Софіївський" (СКК "Софіївський") є не лише інструментом популяризації його діяльності, а й необхідним кроком до підвищення ефективності внутрішньої організації, інформування відвідувачів та залучення нової аудиторії. Вебресурс має стати єдиним цифровим середовищем для презентації комплексу, публікації актуальних новин, розкладів занять, інформації про секції, а також можливості електронного бронювання й зворотного зв'язку.

Метою роботи є розробка функціонального, зручного та адаптивного вебсайту для Спортивно-Культурного Комплексу "Софіївський", який забезпечить інтерактивну взаємодію користувачів з інформаційними ресурсами установи, сприятиме її відкритості та інформативності. Для досягнення цієї мети необхідно реалізувати інтерфейс користувача з доступом до структурованої інформації про діяльність комплексу, передбачити механізми адміністрування контенту, а також забезпечити коректну роботу сайту на різних пристроях.

Об'єктом дослідження виступає процес цифрового представлення діяльності культурно-спортивних закладів засобами веб-технологій. Предметом дослідження є веб-сайт як інформаційна система, що об'єднує технічні, дизайнерські та функціональні рішення з метою ефективною презентації та керування інформацією про комплекс "Софіївський".

Завдання дослідження включають аналіз потреб цільової аудиторії, вибір відповідних інструментів та технологій веб-розробки, проектування архітектури вебсайту, створення графічного інтерфейсу, реалізацію функціональних модулів та тестування працездатності сайту в реальних умовах.

## РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1. Аналіз діяльності СКК «Софіївський»

Спортивно-Культурний Комплекс «Софіївський» – це сучасний багатофункціональний спортивний заклад у селі Софіївська Борщагівка, передмісті Києва. Комплекс діє як комунальна установа, заснована Борщагівською сільською радою (Бучанський район Київської області). Офіційно заклад зареєстровано у травні 2022 року, а фізично відкриття відбулося раніше – 18 жовтня 2020 року відзначено урочисте відкриття новозбудованого комплексу для громади. Відтоді «Софіївський» позиціонується як один з найкращих спортивних комплексів Київщини, пропонуючи мешканцям широкі можливості для занять спортом і культурного розвитку.

Для інформування громадськості комплекс не має окремого веб-сайту, натомість активно представлений у соціальних мережах. Офіційна сторінка СКК «Софіївський» у Facebook регулярно публікує новини, анонси заходів та фотозвіти (станом на 2025 рік сторінка налічує близько 1,2 тис. підписників). Також діє аккаунт в Instagram, де публікуються світлини з тренувань і змагань. Контактна інформація закладу (адреса: вул. Соборна, 114-Б; телефон; email) та посилання на Facebook вказані на офіційному сайті сільської ради, що підтверджує статус цієї сторінки як основного онлайн-ресурсу комплексу. Таким чином, жителі можуть дізнаватися про діяльність «Софіївського» саме через соцмережі, де заклад підтримує інформаційно-розважальний формат спілкування з аудиторією.

Інфраструктура СКК «Софіївський» включає як відкриті спортивні майданчики, так і сучасні криті зали. На території комплексу облаштовано повнорозмірне футбольне поле із штучним покриттям, глядацькими трибунами та професійним освітленням, що дозволяє проводити тренування і матчі у будь-який час доби та за будь-якої погоди. Окрім цього, зведено спортивні зали для різних видів спорту: зали спортивної та художньої гімнастики оснащені сучасним обладнанням, а окремі спеціалізовані зали призначені для занять бойовими

мистецтвами (бокс, кікбоксинг, боротьба) і укомплектовані всім необхідним спорядженням. Для загальної фізичної підготовки і відпочинку на свіжому повітрі комплекс має тренажерний майданчик (outdoor gym) та столи для настільного тенісу.

Помітною особливістю закладу є поєднання спортивної та культурно-освітньої складових. На базі комплексу планується робота бібліотеки та музею, а також інших осередків виховного дозвілля. Таким чином, «Софіївський» виконує не лише суто спортивну функцію, а й слугує центром культурного життя громади. Включення бібліотеки і музею до структури комплексу покликане забезпечити різносторонній розвиток відвідувачів – від фізичного загартування до інтелектуального дозвілля. За своїм статусом заклад є комунальною неприбутковою установою, тому його інфраструктура доступна для широких верств населення на некомерційній основі. Організаційно комплекс підпорядковується відділу освіти, культури, молоді та спорту місцевої ради, що здійснює управління та фінансування діяльності. Керівником (директором) СКК «Софіївський» від заснування призначено Космачевську Ольгу Леонідівну, яка відповідає за розвиток матеріально-технічної бази та координацію роботи секцій.

Перелік секцій охоплює різні напрями: командні види спорту (футбол, регбі), єдиноборства (бокс, кікбоксинг, дзюдо, таеквон-до), гімнастичні види (спортивна й художня гімнастика, акробатика), а також черлідінг.

Одним із головних завдань комплексу є розвиток дитячого і юнацького спорту. Від початку 2022 року СКК «Софіївський» оголосив масштабний набір дітей до низки спортивних секцій різного профілю[1]. На сьогодні при комплексі діють секції для дітей і підлітків із таких видів спорту:

- Футбол – тренування з футболу на сучасному полі, участь у дитячо-юнацьких змаганнях;
- Регбі – секція з регбі для молоді, один із небагатьох осередків розвитку цього виду спорту в області;
- Бокс – школа боксу, де займаються під керівництвом досвідчених тренерів;

- Таеквон-до (ITF) – бойове мистецтво таеквон-до (версія ITF), вихованці вивчають техніку і беруть участь у турнірах;
- Дзюдо – секція дзюдо, що виховує юних борців та готує їх до змагань;
- Черлідінг – група черлідінгу (спортивно-танцювальний напрям), де діти опановують акробатичні та танцювальні елементи;
- Спортивна гімнастика – секція спортивної гімнастики (опорні стрибки, вправи на снарядах тощо) з належним обладнанням;
- Спортивна акробатика – акробатичні вправи в команді та парах, що розвивають силу і координацію;
- Художня гімнастика – секція художньої гімнастики (вправи зі стрічками, обручами тощо) для дівчат;
- Кікбоксинг – секція кікбоксингу, де діти вивчають техніку ударних єдиноборств.

Як видно з переліку, комплекс охоплює ігрові види спорту, силові та бойові дисципліни, а також гімнастично-акробатичний напрям і танцювальний спорт. Така різноманітність секцій дає можливість кожній дитині обрати заняття до душі. Тренування проводять кваліфіковані тренери; для зручності батьків контакти тренерів кожної секції були оприлюднені у рекламному оголошенні (рисунок 1.1) та на офіційних сторінках.

ОГОЛОШУЄТЬСЯ НАБІР ДІТЕЙ У СПОРТИВНІ СЕКЦІЇ		
Футбол 063 978 57 31	Регбі 067 307 09 93	Бокс 067 366 40 73
Таеквон-до ITF 063 102 61 88	Дзюдо 067 156 95 37	Черлідінг 095 365 64 84
Спортивна гімнастика 097 852 89 80	Спортивна акробатика 067 402 83 29	
Художня гімнастика 099 379 07 88 / 093 465 12 27		Кікбоксинг 066 745 71 31

Софіївська Борщагівка, Соборна, 114Б

Рисунок 1.1 – Рекламне оголошення комплексу «Софіївський»

Окрім спортивних секцій, «Софіївський» надає громаді й інші послуги у сфері фізичного виховання та дозвілля. Зокрема, комплекс може

використовуватися для проведення масових спортивних заходів, турнірів, показових виступів. Передбачено можливість оренди приміщень чи майданчиків для проведення зборів, змагань інших організацій або клубів – статут закладу дозволяє надавати такі послуги на договірних засадах. Також, відповідно до зареєстрованих видів діяльності, у структурі комплексу можливе створення гуртків за інтересами, груп денного догляду за дітьми тощо. Наприклад, бібліотека при комплексі може стати місцем проведення культурних заходів – виставок, лекцій, зустрічей для дітей і молоді. Таким чином, СКК «Софіївський» функціонує як центр, що надає населенню повний спектр послуг: від спортивних тренувань і змагань до освітньо-культурного дозвілля.

СКК «Софіївський» має статус комунальної установи, підзвітної місцевій владі. Засновником і власником комплексу є Борщагівська сільська рада, яка фінансує його діяльність з місцевого бюджету та визначає стратегію розвитку[2]. Керівництво здійснюється директором закладу (Ольга Космачевська) та адміністративною командою, до складу якої входять методисти, інструктори й тренери секцій. Оскільки «Софіївський» підпорядковано відділу освіти, культури, молоді та спорту, його діяльність тісно інтегрована в загальну систему роботи з молоддю громади. Комплекс співпрацює зі спортивними федераціями та клубами регіону. Це проявляється, зокрема, у спільному проведенні змагань: на базі СКК регулярно відбуваються відкриті турніри за участі дитячо-юнацьких спортивних шкіл Києва та області.

Для розвитку окремих видів спорту комплекс налагоджує партнерські зв'язки зі спеціалізованими школами. Наприклад, акробатичні турніри у «Софіївському» організовуються спільно зі столичними школами спортивної акробатики, що підвищує рівень змагальної практики вихованців[3]. У футболі комплекс взаємодіє з місцевими клубами: є співпраця з громадською організацією СК «Софія» (Софіївська Борщагівка) – цей клуб бере участь у обласних змаганнях і може використовувати поле комплексу для тренувань та матчів. Також поруч діє відомий футбольний клуб «Чайка» (Петропавлівська Борщагівка); хоча він є окремою

структурою, наявність сучасної інфраструктури «Софіївського» створює передумови для проведення спільних спортивних заходів у громаді.

Варто відзначити, що комплекс відкритий до ініціатив мешканців: адміністрація співпрацює з батьківськими комітетами, громадськими об'єднаннями та спонсорами щодо підтримки спортивних проєктів. Наприклад, турніри з футболу серед аматорів (R-Cup та інші) проходять за партнерства місцевого бізнесу, що надає призи та інформаційну підтримку. Така кооперація зміцнює зв'язки комплексу з громадою і сприяє популяризації здорового способу життя[4].

За час роботи СКК «Софіївський» став центром проведення гучних спортивних подій та свят для населення. Традиційним стало відзначення Дня фізичної культури і спорту – зокрема, у вересні проводяться дні відкритих дверей для всіх охочих. Так, 12 вересня на території комплексу відбувся святковий захід з нагоди Дня фізкультури: для дітей і дорослих організували майстер-класи від тренерів секцій, показові виступи вихованців, працювали розважальні локації (частування солодкою ватою та морозивом для малечі тощо). Під час цього заходу відвідувачі могли безкоштовно спробувати свої сили в різних видах спорту, дізнатися про розклад гуртків та записатися до секцій. Такі публічні свята вже стали доброю традицією комплексу, підвищуючи його популярність серед мешканців [5].

Крім внутрішніх заходів, «Софіївський» регулярно приймає змагання різного рівня. На його базі проводяться відкриті першості з гімнастики та єдиноборств. Приміром, у квітні 2023 року комплекс приймав відкритий чемпіонат зі спортивної акробатики «Весняна Зірка», куди з'їхалися команди дитячо-юнацьких шкіл м. Києва та області – подібні турніри дозволяють юним спортсменам громади випробувати свої навички на змагальному рівні у рідних стінах. Вихованці СКК «Софіївський» також успішно виступають на виїзних турнірах: так, у грудні 2023 року юні дзюдоїсти комплексу вибороли перше та друге місця на відкритому турнірі з дзюдо в місті Тернополі. Регулярно надходять новини про досягнення

спортсменів «Софіївського» на обласних і всеукраїнських змаганнях, що свідчить про якісний тренувальний процес[6].

Відгуки користувачів про діяльність комплексу загалом позитивні, хоча як відносно новий заклад (діє кілька років) «Софіївський» ще збирає свою репутацію. На профільному ресурсі Mixsport комплекс названо «сучасним спортивним комплексом, одним з найкращих у Київській області», що підкреслює високий рівень його оснащення. У соціальних мережах батьки дітей, які відвідують секції, відзначають професіоналізм тренерів та хороші умови для занять. Формальних онлайн-рейтингів наразі небагато (наприклад, на платформі Mixsport користувачі ще не залишили оцінок), але жива активність громади – велика кількість відвідувачів заходів, фотозвіти, подяки в коментарях – свідчить про довіру та інтерес до комплексу[7].

Спортивно-культурний комплекс «Софіївський» реально існує і успішно функціонує як ключовий спортивний осередок Софіївсько-Борщагівської громади. Він має розвинену матеріальну базу, різнопрофільні спортивні секції, підтримку місцевої влади та позитивні відгуки від користувачів. На сьогодні «Софіївський» не має однойменних аналогів – ця назва стосується саме комплексу в Софіївській Борщагівці, що вирізняє його серед інших закладів. Отже, СКК «Софіївський» впевнено займає своє місце в житті громади, популяризуючи спорт і здоровий розвиток та слугуючи прикладом вдалого поєднання спортивних і культурних ініціатив на місцевому рівні.

## **1.2. Аналіз цільової аудиторії та її інформаційних потреб**

Соціально-демографічна характеристика цільової аудиторії веб-сайту Спортивно-Культурного Комплексу «Софіївський» охоплює широкий спектр мешканців міста та прилеглих населених пунктів, які потенційно взаємодіють із комплексом у різних ролях – як активні учасники заходів, глядачі, батьки дітей, що відвідують секції, або як представники адміністрації й організатори культурно-масових подій. Основними демографічними групами є діти віком від 5 до 14 років, підлітки та молодь до 25 років, дорослі особи працездатного віку (переважно 26–

55 років), а також люди похилого віку, які можуть бути залучені до оздоровчих програм чи відвідування культурних заходів. Рівень освіти більшості користувачів середній або вищий, що передбачає загальну базову обізнаність із цифровими технологіями, хоча серед старшого населення ця компетенція може бути нижчою.

У соціальному розрізі спостерігається перевага сімей із дітьми, що активно шукають можливості для всебічного розвитку та дозвілля своїх дітей, а також молодих людей, які зацікавлені у спортивній активності, групових заняттях, відвідуванні культурних подій або участі в конкурсах і виставках. Значну частину аудиторії становлять жінки, особливо у сегменті батьків дітей, що потребують детальної та оперативної інформації про розклад занять, вартість абонементів, спеціальні пропозиції або зміни у функціонуванні секцій. У той самий час чоловіки частіше цікавляться спортивними секціями, турнірами або можливістю оренди спортивних об'єктів. Географічна структура аудиторії зосереджена переважно в межах міста, однак частина користувачів може приїздити з інших регіонів у рамках участі в змаганнях або фестивалях, що зумовлює потребу в окремих сервісах, зокрема навігації, контактній інформації, умовах проживання та логістики.

Таким чином, соціально-демографічна структура аудиторії є достатньо різномірною, що обумовлює необхідність створення веб-сайту з адаптивним інтерфейсом, інтуїтивною навігацією, багатомовною підтримкою, а також гнучкою структурою подання контенту, яка дозволяє кожному сегменту користувачів швидко отримувати актуальну для нього інформацію. Урахування цих характеристик має суттєве значення для подальшого проектування як інформаційної архітектури ресурсу, так і його функціональних компонентів.

У межах функціонування Спортивно-Культурного Комплексу «Софіївський» можна виокремити кілька груп користувачів, що відрізняються за характером і глибиною залучення до діяльності установи, а також за типами взаємодії з інформаційною системою. Насамперед, до базової цільової аудиторії належать регулярні відвідувачі спортивних секцій, гуртків і тренувань, зокрема діти, підлітки та молодь. Їхнє залучення має постійний характер, вони зазвичай користуються послугами комплексу декілька разів на тиждень, тож потребують оперативного

доступу до актуального розкладу, змін у графіку занять, а також до інформації про тренерський склад і нові програми. Водночас надзвичайно важливо враховувати інформаційні потреби батьків цих дітей, які часто виступають як посередники між закладом та самими учасниками секцій. Вони потребують детальної інформації про якість послуг, сертифікацію персоналу, доступність місць для запису, умови оплати та механізми зворотного зв'язку.

Окрему групу формують відвідувачі культурних заходів, які можуть бути менш регулярними у своєму залученні, але водночас відіграють важливу роль у публічному житті комплексу. Це мешканці мікрорайону, шанувальники театрального, музичного чи виставкового мистецтва, а також залучені громадські активісти, які можуть брати участь у тематичних подіях, форумах або ярмарках. Їхня потреба у взаємодії з сайтом полягає у швидкому пошуку афіші, бронюванні місць, ознайомленні з архівами минулих подій і перегляді медіаматеріалів, які висвітлюють діяльність комплексу. Такі користувачі особливо чутливі до естетичного вигляду та презентаційної якості контенту, адже їх мотивація відвідування часто ґрунтується на емоційному сприйнятті.

Ще однією категорією користувачів є адміністративно-організаційний персонал комплексу, який, хоч і не є зовнішнім користувачем, проте має потребу в окремому рівні доступу до функцій сайту. Ця група взаємодіє з інформаційною системою з боку управління розкладом, модерації контенту, обробки запитів, реєстрацій, проведення аналітики відвідуваності та ведення внутрішньої документації. Залежно від ступеня залучення, може виникати потреба в багаторівневій системі авторизації, ролях, доступі до адміністративної панелі або звітності, що вимагає побудови веб-системи з гнучкою моделлю прав користувачів.

Крім основних груп, існують також потенційні користувачі, які лише планують відвідати заклад, і тому їхній запит переважно інформаційно-ознайомчий. Їм важливо швидко знайти контактну інформацію, фото- та відеогалерею, опис послуг і їхню вартість. Саме ця група користувачів є найменш стійкою до технічних недопрацювань або неінтуїтивного інтерфейсу — навіть короткий негативний досвід взаємодії може призвести до втрати клієнта. З огляду

на це, створення сайту повинно охоплювати як активних і постійних відвідувачів, так і користувачів з епізодичною чи перспективною зацікавленістю, для кожного з яких необхідно реалізувати відповідні інформаційні канали, сервіси та дизайн-рішення.

Мотивація користування сайтом Спортивно-Культурного Комплексу «Софіївський» формується під впливом поєднання практичних потреб відвідувачів та загальної зацікавленості у подієвому, тренувальному та адміністративному житті комплексу. У більшості випадків користувачі заходять на сайт із чітко визначеною метою — отримати актуальну інформацію про графік роботи спортивних залів, розклад занять у секціях, реєстрацію на заходи або деталі проведення майбутніх культурно-масових подій. Батьки дітей, які займаються у гуртках, зазвичай цікавляться умовами запису, наявністю вільних місць, цінами, а також розкладом та змінами в роботі секцій. Інші групи користувачів, зокрема спортсмени, тренери або організатори подій, прагнуть швидко отримати інформацію про доступність інфраструктури, умови оренди приміщень, технічні характеристики залів чи навіть правила користування обладнанням.

Інформаційна поведінка таких користувачів характеризується орієнтацією на функціональний і лаконічний контент, де пріоритет мають простота доступу до потрібного розділу, наявність чітко структурованої інформації, а також відсутність зайвого навантаження рекламними або вторинними елементами. Типовим є сценарій використання, при якому користувачі не прагнуть довготривалого перебування на сайті, а натомість швидко переходять до розділів, пов'язаних із конкретною дією — запис, перегляд афіші, завантаження документа, підтвердження участі або звернення до адміністрації. Це свідчить про настанову на швидке задоволення конкретної інформаційної потреби, що потребує інтерфейсу з мінімальною кількістю кліків і високою інтуїтивністю навігації.

Окрему категорію становлять користувачі, які заходять на сайт з ознайомлювальною метою, наприклад, потенційні відвідувачі, які раніше не мали досвіду взаємодії з комплексом. Для них важливо сформувати позитивне враження, надати вичерпну інформацію про переваги, спектр послуг, професіоналізм

персоналу, фотогалереї подій та візуальні уявлення про внутрішній простір закладу. Також варто враховувати, що користувачі часто не використовують традиційну навігацію через меню, а натомість застосовують внутрішній пошук або переходи за посиланнями з соціальних мереж чи месенджерів. Це змінює структуру інформаційних потоків на сайті, акцентуючи необхідність логічного взаємопов'язання між сторінками, візуальної привабливості головної сторінки та оптимізації мобільної версії.

У підсумку, мотивація користувачів визначається прагненням оперативно отримати релевантну інформацію та зручно здійснювати взаємодію з адміністрацією або сервісами комплексу, що обумовлює специфіку їх інформаційної поведінки — короткі, цілеспрямовані сеанси з перевагою функціональності над естетикою, за винятком ознайомлювальних візитів, які вимагають глибшого змістового та емоційного занурення.

Рівень технічної обізнаності та цифрової компетентності користувачів є визначальним чинником при формуванні архітектури веб-сайту Спортивно-Культурного Комплексу «Софіївський», оскільки саме від нього залежить зручність користування, доступність функціоналу та ефективність взаємодії відвідувачів з інформаційними сервісами. Аналіз аудиторії показує, що серед користувачів переважають декілька категорій — це діти шкільного віку, підлітки, молоді батьки, а також особи пенсійного віку або старшого покоління, які беруть участь у культурних або оздоровчих заходах. Кожна з цих груп має різний рівень володіння цифровими технологіями, що вимагає адаптивного підходу до розробки інтерфейсу.

Для молоді характерне інтуїтивне користування сучасними пристроями, навички швидкої навігації в мобільних застосунках, здатність працювати з інтерактивними елементами, формами зворотного зв'язку, онлайн-оплатою, бронюванням тощо. Вони очікують наявності мобільної адаптації сайту, швидкого доступу до актуальної інформації, інтеграції з соціальними мережами, push-сповіщень та динамічного контенту. Водночас користувачі старшого віку значно рідше користуються новітніми інструментами інтернет-комунікації, вони часто

обмежуються переглядом базової інформації, і потребують простоти навігації, чіткого візуального поділу елементів, логічно побудованих розділів, великих кнопок і мінімуму інтерактивності, яка може викликати плутанину або технічні труднощі.

Тому технічна реалізація сайту повинна передбачати мультиплатформену адаптивність, дві версії інтерфейсу — стандартну для більш досвідчених користувачів і спрощену для тих, хто потребує мінімального навантаження на цифрові навички. Крім того, важливо забезпечити стабільну роботу на всіх популярних браузерях, підтримку слабких пристроїв з обмеженими ресурсами, а також максимально знизити бар'єр входу для нових відвідувачів, які раніше не користувалися подібними системами. Висока цифрова різноманітність цільової аудиторії зумовлює необхідність гармонійного поєднання доступності, надійності та інтуїтивної зрозумілості веб-ресурсу, щоб забезпечити комфортний досвід для кожного користувача незалежно від його віку, технічного досвіду чи рівня цифрової компетентності.

Потреба в доступності інформації в режимі реального часу у контексті функціонування веб-сайту Спортивно-Культурного Комплексу «Софіївський» зумовлена як специфікою діяльності самого закладу, так і очікуваннями користувачів, що взаємодіють із ним через цифрове середовище. Сутність цієї потреби полягає в тому, що відвідувачі, батьки дітей, які займаються в спортивних секціях, учасники культурних заходів, а також адміністративні працівники повинні мати змогу миттєво отримувати актуальну, перевірену інформацію щодо графіків, змін у розкладі, повідомлень про відміну або перенесення подій, а також про наявність вільних місць для участі в гуртках чи оренди приміщень. Особливо це стосується тих ситуацій, коли події відбуваються часто, а розклад зазнає змін унаслідок зовнішніх чинників: погодних умов, службових змін тренерського складу чи технічних обмежень.

Оперативність оновлення інформації виступає чинником формування довіри до цифрового інструменту з боку кінцевого користувача. Якщо сайт не забезпечує своєчасного інформування, це спричиняє фрустрацію, зменшення зацікавленості у

взаємодії та повертає користувачів до телефонних дзвінків або фізичних візитів до закладу, що нівелює саму ідею автоматизації комунікацій. В умовах сучасного інформаційного простору, де люди звикли до негайного отримання відповідей у режимі 24/7, саме затримка в оновленні або відсутність можливості дізнатися про зміни в режимі реального часу перетворюється на системну ваду інформаційної системи. Публікація новин, розсилки змін у розкладі, повідомлення про акційні пропозиції чи нові послуги повинні відбуватися негайно, із забезпеченням стабільного каналу комунікації між адміністрацією комплексу та його цільовою аудиторією.

Для реалізації такої функціональності веб-сайт має бути технічно здатним обробляти та відображати змінні дані без значних затримок. Це передбачає наявність інтегрованого інтерфейсу для адміністраторів, які зможуть легко оновлювати контент, системи сповіщення користувачів про зміни (через push-нотифікації, електронну пошту, SMS чи месенджери), а також можливість кешування контенту лише на обмежений термін, щоб завжди показувати найбільш актуальні версії сторінок. Оскільки користувачі дедалі частіше заходять на сайт із мобільних пристроїв, бажано забезпечити синхронізацію з мобільною версією або окремим застосунком, що дозволить отримувати інформацію про зміни навіть у дорозі. Таким чином, доступність у реальному часі має бути не лише бажаною функцією, а фундаментальною вимогою, що визначає якість взаємодії користувача з цифровим простором комплексу.

У контексті розробки веб-сайту для Спортивно-Культурного Комплексу «Софіївський» мовні та комунікаційні потреби цільової аудиторії виступають суттєвим чинником, що визначає загальний рівень доступності ресурсу для різних соціальних груп. У більшості випадків очікується, що сайт матиме повноцінну україномовну версію, яка не лише відповідатиме сучасним вимогам законодавства щодо мови обслуговування, а й забезпечуватиме комфорт користувачів, які переважно є мешканцями України та споживають інформаційний контент саме українською мовою. Водночас, існує ймовірність, що серед користувачів будуть присутні носії інших мов, зокрема відвідувачі з інших країн або іноземні тренери

та учасники культурних обмінів, що зумовлює необхідність реалізації щонайменше англomовного дубля інтерфейсу. Забезпечення багатомовності має бути виконано не механічним перекладом основних розділів, а із врахуванням культурних відмінностей і специфіки сприйняття, що дозволить забезпечити глибшу залученість користувачів.

Крім мовної доступності, ключовою є побудова ефективної комунікаційної структури, що дозволить забезпечити безперервний зворотний зв'язок між адміністрацією комплексу та користувачами. У цифрових умовах взаємодія не повинна обмежуватися лише інформаційною подачею — важливо забезпечити можливість користувачу не лише прочитати новини чи ознайомитись із розкладом, а й звернутись із запитом, подати заявку на участь у заході, забронювати зал або залишити відгук. Із цією метою необхідно реалізувати кілька каналів комунікації: інтеграцію з електронною поштою, месенджерами, форму зворотного зв'язку без обов'язкової реєстрації, а також передбачити наявність чат-бота або онлайн-чату для вирішення типових питань у реальному часі. Такий підхід дозволяє не лише оперативно реагувати на потреби користувачів, а й формувати довіру до установи як до відкритої й сучасної організації, що прагне до постійної взаємодії із громадою.

Крім того, комунікаційна модель повинна враховувати специфіку цільової аудиторії щодо каналів сприйняття. Молодші користувачі більш схильні сприймати інформацію через візуальний контент — зображення, відео, короткі анонси в соціальних мережах. Відповідно, веб-сайт має бути пов'язаний із соціальними платформами комплексу, надаючи змогу швидко ділитись подіями, переглядати фотозвіти чи трансляції. Старші користувачі, натомість, більше орієнтовані на текстову інформацію, зрозумілу навігацію та можливість отримати консультацію телефоном чи за допомогою листування. Таким чином, реалізація мовних і комунікаційних потреб не є лише технічною деталлю, а формується як стратегічний напрям, що визначає ступінь інклюзивності ресурсу, ефективність обслуговування та загальний імідж Спортивно-Культурного Комплексу «Софіївський» у цифровому середовищі.

У сучасних умовах цифрової взаємодії зростає значущість надання користувачеві індивідуального досвіду при взаємодії з інформаційними системами. Для веб-сайту Спортивно-Культурного Комплексу «Софіївський» потреба в персоналізації та автоматизації сервісів зумовлена бажанням підвищити якість обслуговування відвідувачів, зменшити часові витрати на типові дії та забезпечити комфортний доступ до функціоналу з урахуванням індивідуальних особливостей кожного користувача. Персоналізація у цьому контексті передбачає створення особистого кабінету, у якому відвідувач має змогу переглядати історію своїх візитів, обрані секції, графік тренувань або заходів, а також отримувати рекомендації, сформовані на основі попередньої активності. Таке середовище забезпечує не лише зручність, а й сприяє формуванню емоційного зв'язку між користувачем і закладом, оскільки надає відчуття індивідуального підходу.

Автоматизація сервісів, у свою чергу, слугує засобом оптимізації рутинних процесів як для кінцевого користувача, так і для адміністрації комплексу. Зокрема, мова йде про впровадження онлайн-реєстрації на події або гуртки, автоматичного формування нагадувань про майбутні заходи, підтвердження бронювання, оплати через інтеграцію з платіжними системами, генерації електронних квитанцій і звітів. Для адміністративного персоналу автоматизовані сервіси дозволяють спростити облік відвідувачів, управління розкладом, контроль завантаженості приміщень і навіть верифікацію учасників заходів. Завдяки такій автоматизації знижується ймовірність людських помилок, пришвидшується обробка даних і водночас підвищується рівень сервісу.

Поєднання персоналізації та автоматизації забезпечує динамічний і адаптивний характер функціонування веб-сайту, що дозволяє ефективно відповідати на змінні запити користувачів і водночас зберігати високу продуктивність системи. У рамках спортивно-культурного середовища це відкриває перспективи не лише для інформування, а й для активного залучення аудиторії до процесів планування, запису, зворотного зв'язку та участі в житті комплексу. Саме такий підхід забезпечує конкурентну перевагу та сприяє

формуванню сучасного, клієнтоорієнтованого образу Спортивно-Культурного Комплексу «Софіївський».

### **1.3. Аналіз нинішнього стану комп'ютеризації СКК «Софіївський»**

В сучасних реаліях комп'ютеризація закладу – фундамент ефективної праці, комфорту для клієнтів та відповідності цифровим сподіванням цільової аудиторії. Стосовно Спортивно-Культурного Комплексу «Софіївський», комп'ютеризація включає впровадження технічних та програмних інструментів для забезпечення організаційної діяльності, обміну інформацією та комунікації з відвідувачами.

#### **1.3.1. Інформаційна інфраструктура**

На даний момент інформаційна інфраструктура СКК представлена наступною технікою:

- один стаціонарний комп'ютер для адміністратора;
- ноутбук у керівника;
- принтер/сканер для друку документів;
- Wi-Fi роутер з доступом до Інтернету (швидкість приблизно 30–50 Мбіт/с);
- особисті мобільні пристрої співробітників, які час від часу застосовуються для комунікації.

Наведене обладнання обслуговує тільки базові операційні потреби: друк графіків, створення документів в Word/Excel, ведення списків відвідувачів та розкладів в ручному режимі.

#### **1.3.2. Програмне забезпечення**

На комп'ютерах установи встановлено стандартні офісні програми (Microsoft Office або LibreOffice), які використовуються для підготовки розкладів, листів, наказів. Спеціалізоване програмне забезпечення для управління секціями, користувачами чи внутрішніми процесами — відсутнє. Вся інформація зберігається як окремі файли, що не синхронізуються між собою та не мають захищеного доступу.

### **1.3.3. Онлайн-присутність**

СКК «Софіївський» не забезпечений власним веб-сайтом, що відчутно обмежує його цифрову доступність. Єдина присутність в цифровому просторі — це сторінки у соціальних мережах (Facebook, Instagram), які оновлюються вручну, епізодично та не містять всієї необхідної інформації. Користувачі позбавлені можливості:

- ознайомитись з розкладом занять онлайн;
- подати заявку на участь у секції дистанційно;
- знайти контакти або опис секцій у структурованому вигляді.

Таким чином, основна взаємодія між закладом та користувачами відбувається в офлайн-режимі або телефоном, що не відповідає вимогам сучасного підходу до обслуговування.

### **1.3.4. Інформаційна безпека**

Захист даних в установі забезпечений лише частково. Використовуються елементарні паролі на комп'ютерах та антивірусне ПЗ. Резервне копіювання документів здійснюється несистематично, а централізована система зберігання інформації відсутня. Це створює загрози втрати важливих даних, особливо у випадку технічної несправності або помилки користувача.

### **1.3.5. Рівень ІТ-компетентності персоналу**

Цифрова грамотність працівників знаходиться на середньому рівні:

- адміністративний персонал володіє базовими навичками роботи з комп'ютером;
- керівництво застосовує електронну пошту, текстові редактори, іноді — хмарні сервіси;
- тренери та інші фахівці переважно не використовують цифрові засоби у своїй діяльності.

Відсутність досвіду роботи з сучасними веб-технологіями, платформами для онлайн-реєстрації чи електронної комунікації вимагає або проведення спеціального навчання, або залучення сторонніх спеціалістів для підтримки майбутнього веб-сайту.

## 1.4. Функціональне моделювання та аналіз існуючих бізнес-процесів

### 1.4.1. Функціональна модель діяльності СКК «Софіївський»

Функціональна модель діяльності Спортивно-Культурного Комплексу «Софіївський» демонструє основні управлінські та операційні процеси, що забезпечують планування, втілення та координацію подій, спілкування з громадськістю та інформування населення. Для створення функціональної моделі використано методологію IDEF0 (Integration Definition for Function Modeling), котра дозволяє подати систему у вигляді блоків функцій з визначенням входів, виходів, контролюючих впливів та механізмів реалізації.

Контекстна діаграма A-0 (Рис. 1.2) узагальнює загальну функцію «Організувати діяльність СКК «Софіївський». Основними входними даними є звернення громадян, план роботи та відомості про доступні ресурси. Контролюючі фактори включають нормативну базу, фінансування та накази керівництва. Механізми реалізації - адміністрація, тренерський склад, технічні засоби та канали комунікації. На виході - розклад подій, відповіді на звернення, інформація для населення.

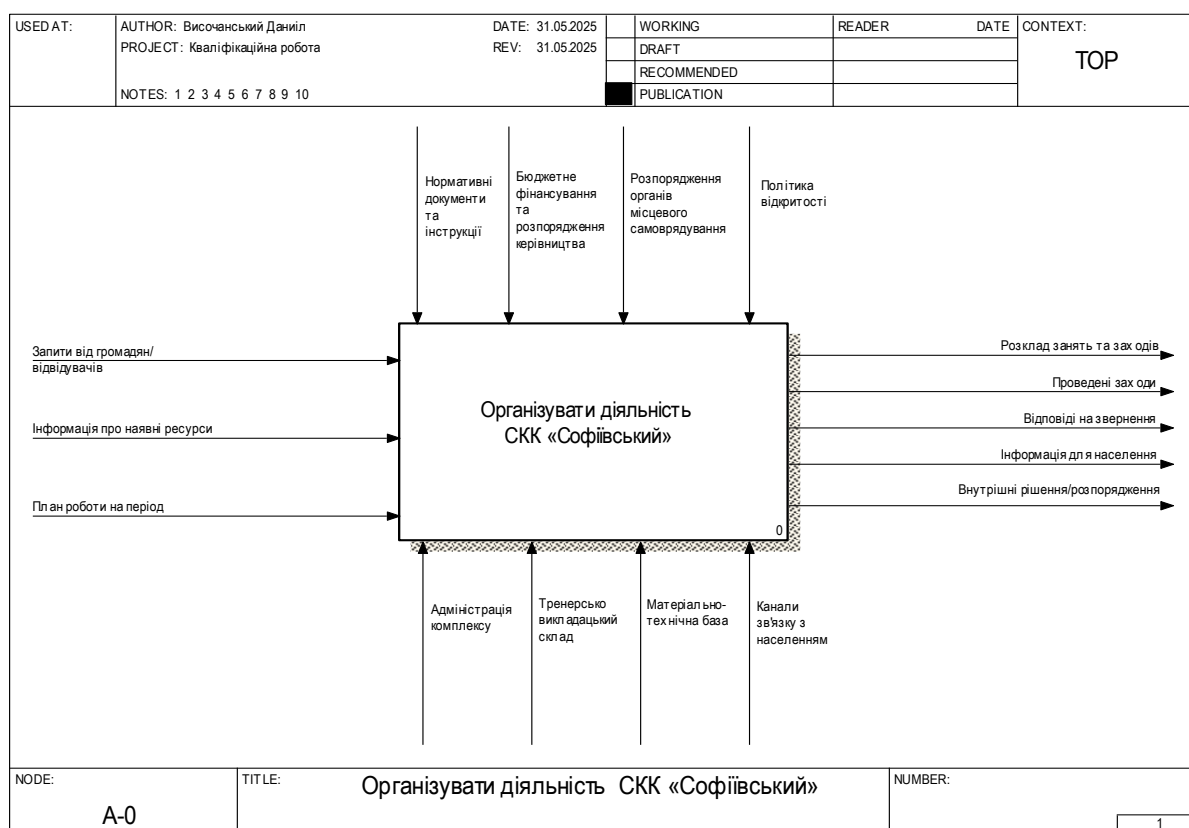


Рисунок 1.2 – Контекстна діаграма функціональної моделі

На діаграмі А0 (Рис. 1.3) реалізована декомпозиція функції А-0 на п'ять підфункцій:

- А1 – Планування діяльності комплексу — формування плану роботи на основі запитів і ресурсів;
- А2 – Проведення спортивних і культурних заходів — виконання затверджених активностей;
- А3 – Взаємодія з відвідувачами — прийом та обробка звернень.
- А4 – Адміністративне управління — підтримка внутрішньої координації та контролю;
- А5 – Інформування населення — розповсюдження публічної інформації через доступні канали.

Усі ці процеси тісно взаємопов'язані та формують єдину систему функціонування СКК як організаційної структури, орієнтованої на надання соціокультурних послуг населенню.

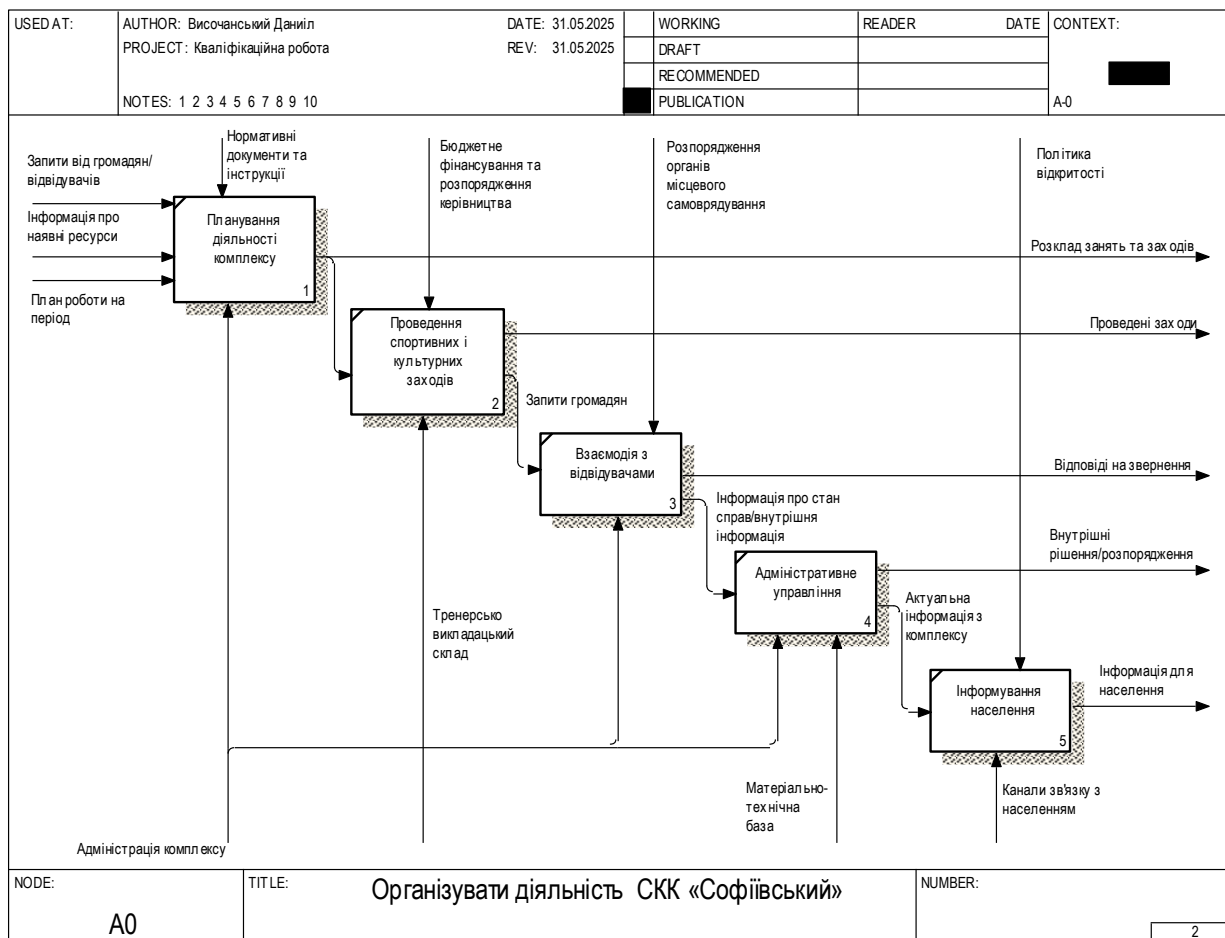


Рисунок 1.3 – Перший рівень декомпозиції моделі діяльності СКК «Софіївський»

### **1.4.2. Виявлені проблеми**

Проаналізувавши функціональну модель, було виявлено низку перешкод, які ускладнюють ефективне виконання процесів у межах СКК:

1. Відсутність уніфікованої цифрової системи обліку та координації діяльності. Організаційна робота здебільшого відбувається вручну або в офлайн-режимі, що ускладнює оновлення даних та формування звітності.

2. Розрізненість взаємодії з відвідувачами.

Запити та звернення надходять через різні, несистематизовані канали зв'язку — телефонні дзвінки, паперові заяви, месенджери, без подальшої централізації або архівації.

3. Недостатній рівень оперативної інформатизації населення.

Оголошення публікуються без певної системи, відсутня єдина онлайн-платформа з афішами, розкладами та актуальними новинами.

4. Відсутність інструментів внутрішнього моніторингу.

Адміністрація не має оперативного доступу до даних про поточний стан справ: відвідуваність, графік роботи, навантаження на інструкторів, використання ресурсів.

### **1.4.3. Пропозиції щодо вирішення виявлених проблем**

Для оптимізації наявних бізнес-процесів та усунення вказаних недоліків, пропонується оновити функціональну модель шляхом переходу до стану "ТО ВЕ". Цей стан передбачає впровадження цифрової інформаційної системи управління діяльністю комплексу.

Ключові пропозиції:

- Створення єдиного веб-ресурсу Спортивного Комплексу (СКК), що включатиме: актуальний розклад занять і подій, зручну форму для звернень, стрічку новин та важливих повідомлень;
- Впровадження онлайн-реєстрації на секції та різноманітні заходи з можливістю збереження історії поданих заявок;
- Забезпечення електронного документообігу в адміністрації для автоматизації звітності та ведення обліку;

- Використання адаптивної платформи інформування населення, що дозволить оперативно розповсюджувати інформацію через різні канали (веб-сайт, соціальні мережі, електронна пошта);
- Створення панелі моніторингу для адміністрації комплексу – дашборду зі статистикою подій, звернень, навантаження, візуалізованими графіками тощо.

Відмінності між функціональною моделлю статусу «ТО ВЕ» та поточною ситуацією (модель статусу «AS IS») полягають у переході від ручного та паперового управління до цифрового, автоматизованого та інтегрованого процесу з прозорим обміном інформацією. Це підвищить ефективність управління, оперативність реагування на запити та загальну відкритість комплексу для громади.

Перспективи: впровадження оновленої моделі бізнес-процесів створить сприятливі умови для подальшого розвитку цифрових сервісів, таких як мобільні додатки, системи оповіщення, рейтингові анкети, що значно покращить ефективність роботи та клієнтоорієнтованість закладу.

### 1.5. Аналіз аналогічних веб-рішень і порівняння їх функціоналу

Розглянемо сайти-аналоги та здійснимо їх порівняльний аналіз.

Сайт Спортивного комплексу НТУУ «КПІ» [8] є прикладом веб-ресурсу, орієнтованого на студентську та загальноосвітню аудиторію з метою інформування про діяльність спортивних секцій, подій та доступності об'єктів інфраструктури (Рис. 1.4).

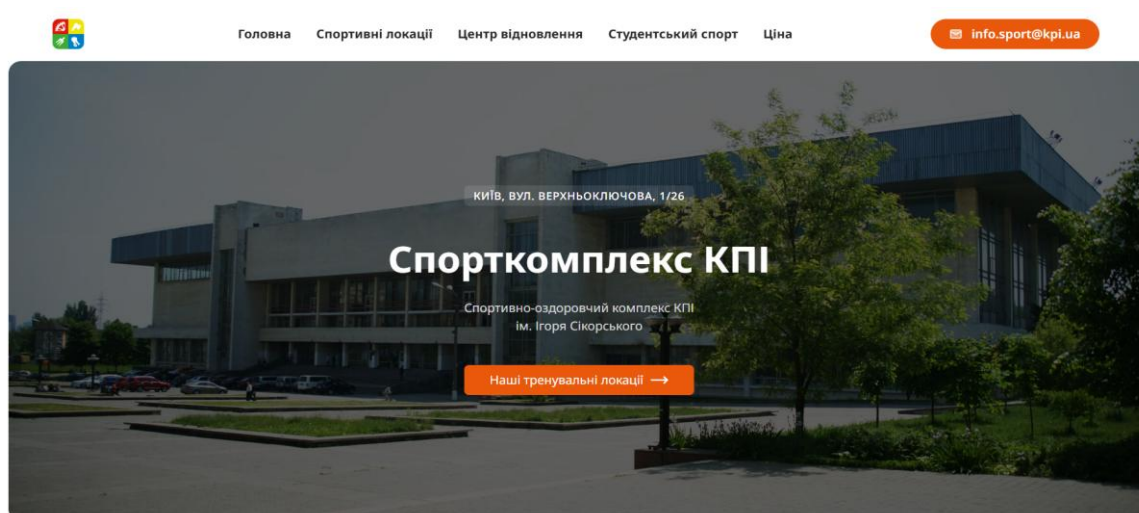


Рисунок 1.4 - Веб-сайт спортивного комплексу НТУУ «КПІ»

Його функціональність зосереджена переважно на наданні статичної інформації щодо розкладу занять, опису спортивних секцій, умов відвідування, контактних даних, а також окремих новин та анонсів. Сайт має розділи з фотогалереєю, переліком інструкторів і тренерів, а також можливістю завантаження документів (наприклад, правил користування спортивною базою). Присутній блок новин, що дозволяє частково інформувати користувачів про поточні події, хоча оновлення здійснюється нерегулярно.

До переваг сайту слід віднести лаконічну та інтуїтивно зрозумілу структуру, що дозволяє швидко знайти потрібну інформацію. Наявність окремих сторінок для кожного виду спорту спрощує навігацію та допомагає користувачам зорієнтуватися у спектрі запропонованих послуг. Сайт адаптований до мобільних пристроїв, має досить швидке завантаження сторінок, що позитивно впливає на користувацький досвід. Крім того, перевагою є наявність контактної інформації з точним місцем розташування споруд, що важливо для нових відвідувачів.

Серед недоліків ресурсу варто відзначити відсутність інтерактивних функцій: немає можливості онлайн-реєстрації на секції, бронювання спортивних об'єктів, персонального кабінету користувача чи інтеграції з платіжними або календарними системами. Контент представлений переважно у вигляді статичних текстів без динамічного завантаження чи фільтрації за інтересами. Візуальний стиль сайту є застарілим, що знижує привабливість ресурсу для молодшої аудиторії. Відсутність багатомовної підтримки обмежує потенційне охоплення, особливо для іноземних студентів. Сайт також не передбачає елементів доступності для користувачів з особливими потребами, що знижує рівень інклюзивності.

Узагальнюючи, сайт Спортивного комплексу НТУУ «КПІ» виконує базову інформативну функцію, однак потребує модернізації для відповідності сучасним вимогам користувацького досвіду, зокрема в аспектах інтерактивності, персоналізації та візуальної привабливості.

Сайт Київського міського палацу дитячої та юнацької творчості [9] являє собою веб-ресурс культурно-освітнього спрямування, що орієнтований на

інформування батьків і дітей про наявні гуртки, творчі студії, події та умови участі у позашкільній діяльності (Рис. 1.5).

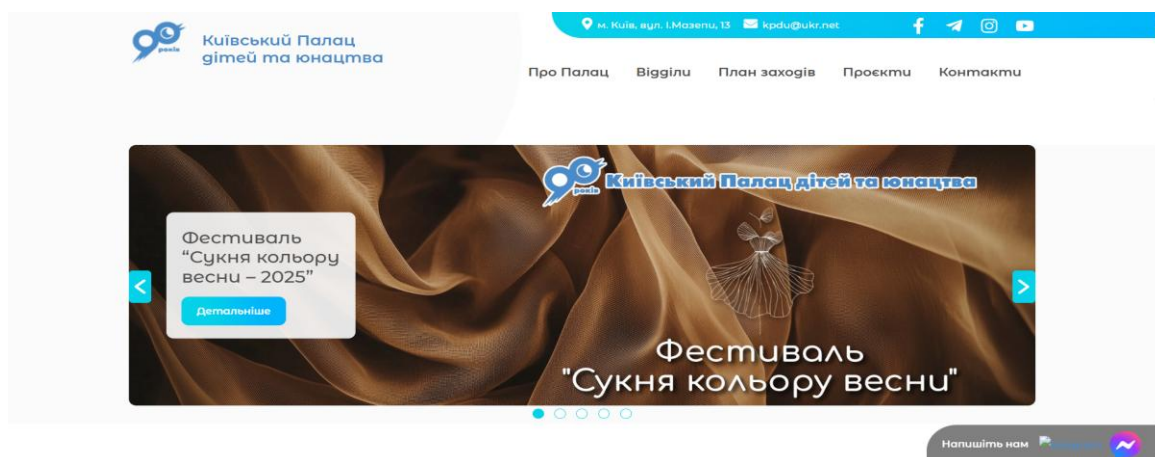


Рисунок 1.5 – Сайт Київського міського палацу дитячої та юнацької творчості

Функціональність сайту зосереджена передусім на презентації структури установи, окремих напрямів роботи, оголошеннях та медіаматеріалах, що висвітлюють результати виховного процесу. Основні функціональні модулі включають розділи про історію та місію закладу, новини, фотогалереї, опис гуртків та секцій, контактну інформацію, а також можливість переглядати розклад подій. Крім того, передбачено окремий блок для педагогів, адміністративну інформацію, документи, що регламентують роботу, та посилання на соціальні мережі.

До переваг даного ресурсу можна віднести тематичну структурованість, що відповідає реальній організаційній моделі установи, а також наявність вичерпної інформації про діяльність різних відділів і підрозділів. Інтерфейс сайту загалом зрозумілий, доступ до контенту не вимагає реєстрації, а навігація реалізована через зрозуміле меню. Наявність фотогалерей та звітів про проведені заходи дозволяє батькам і громадськості візуально оцінити активність установи. Додатковою перевагою є те, що сайт підтримує мобільну адаптацію та оперативно оновлює контент, особливо новини.

Водночас ресурс має низку недоліків, які обмежують його ефективність як інтерактивного інструмента взаємодії з аудиторією. Сайт не має функціоналу для електронної реєстрації до гуртків або подій, не передбачає створення особистих кабінетів для батьків чи учнів, відсутня можливість персоналізованого сповіщення про зміни в розкладі або актуальні події. Крім того, пошуковий механізм

реалізовано формально, без глибокої фільтрації за категоріями чи датами, що ускладнює швидкий доступ до цільової інформації. Також відчувається деяка застарілість дизайну та нестача сучасних інтерактивних елементів (відеопрезентації, інтеграція з картами, календар подій тощо), що могло б підвищити залученість користувачів.

Таким чином, сайт виконує свою інформативну функцію, однак потребує розширення функціональних можливостей і модернізації інтерфейсу, щоб відповідати сучасним очікуванням користувачів і створити повноцінний канал електронної взаємодії між закладом та його цільовою аудиторією.

Веб-сайт мережі фітнес-клубів Sport Life [10] є прикладом комплексного цифрового рішення для спортивно-оздоровчого бізнесу, яке поєднує в собі багатофункціональність, інформативність та інтерактивність (рисунок 1.6).

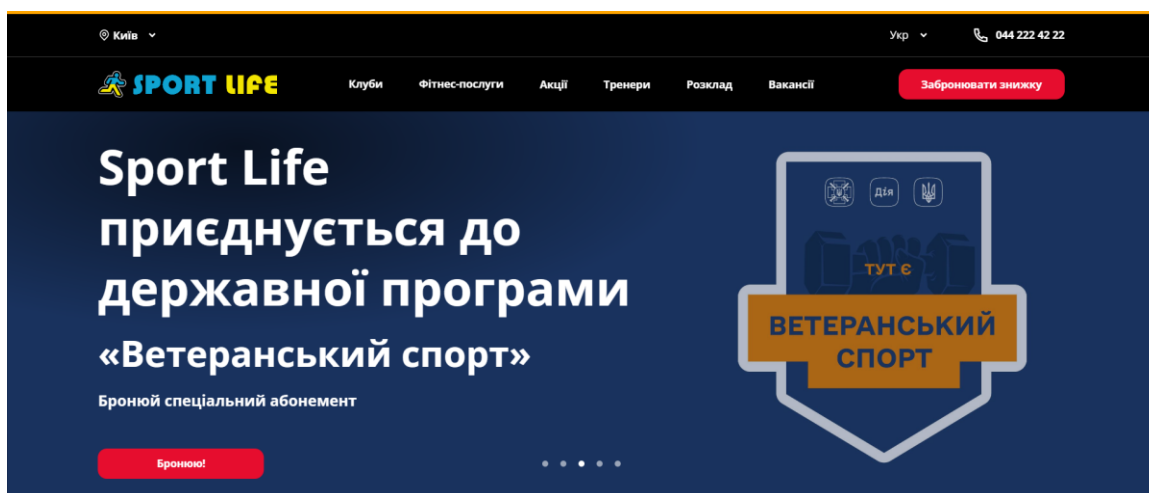


Рисунок 1.6 – Веб-сайт Sport Life

Його функціональність охоплює широкий спектр можливостей для користувачів: від ознайомлення з послугами та розкладом занять до онлайн-бронювання абонементів і персоналізованого управління тренуваннями.

Серед ключових функцій сайту можна виокремити інтерактивну карту клубів по всій Україні, що дозволяє користувачам швидко знаходити найближчий фітнес-центр та ознайомлюватися з його специфікою. Кожен клуб має окрему сторінку з детальним описом послуг, розкладом занять, інформацією про тренерський склад та фотогалереєю, що сприяє прозорості та довірі з боку потенційних клієнтів. Також сайт пропонує можливість онлайн-купівлі абонементів з різними опціями,

включаючи акційні пропозиції та спеціальні програми, такі як «Ветеранський спорт».

Перевагами веб-сайту є його зручна навігація та адаптивний дизайн, що забезпечує комфортне користування як на стаціонарних комп'ютерах, так і на мобільних пристроях. Інтеграція з соціальними мережами та наявність блогу з корисними порадами щодо фітнесу та здорового способу життя підвищують залученість користувачів та створюють додаткову цінність. Крім того, сайт підтримує багатомовність, що розширює його доступність для різноманітної аудиторії.

Проте, незважаючи на численні переваги, сайт має і певні недоліки. Зокрема, відсутність особистого кабінету з можливістю відстеження прогресу тренувань, історії відвідувань та взаємодії з тренерами обмежує рівень персоналізації сервісу. Також деякі користувачі можуть відчувати труднощі з навігацією через велику кількість інформації та функцій, що потребує більш інтуїтивно зрозумілого інтерфейсу. Крім того, відсутність онлайн-чату або швидкої підтримки може ускладнити отримання оперативної допомоги або консультації.

Загалом, веб-сайт Sport Life є ефективним інструментом для залучення та обслуговування клієнтів, проте для досягнення вищого рівня користувацького досвіду доцільно впровадити додаткові функції персоналізації та покращити інтерфейс для спрощення навігації.

Порівняльна таблиця наведених вище аналогічних веб-рішень наведена у таблиці 1.1.

Таблиця 1.1. Порівняльна таблиця альтернативних веб-рішень

Критерій / Сайт	Спорткомплекс НТУУ «КПІ»	Київський палац творчості	Sport Life
Цільова аудиторія	Студенти	Діти, батьки	Широка аудиторія
Основна мета	Інформування	Інформування	Інформування, сервіс
Онлайн-реєстрація / бронювання	Немає	Немає	Є
Особистий кабінет	Немає	Немає	Обмежено
Медіаматеріали (фото/відео)	Фото	Фото	Фото, відео
Адаптивність	Є	Є	Є
Мультимовність	Немає	Немає	Є
Оновлення контенту	Нерегулярне	Регулярне	Регулярне
Дизайн	Застарілий	Простий	Сучасний

### 1.6. Розрахунок техніко-економічного обґрунтування впровадження створюваного програмного забезпечення

Для оцінки ефективності створення веб-платформи для Спортивно-Культурного Комплексу «Софіївський» було проведено аналіз витрат та очікуваного економічного ефекту, враховуючи етапи розробки, впровадження та подальшої експлуатації системи.

#### Аналіз вимог:

трудомісткість: 10 люд.-днів;

команда: 2 особи;

тривалість етапу: 5 днів;

витрати: 8000₴ (заробітна плата) + 2000₴ (аналітика, консультації);

загалом: 10000₴.

**Проектування:**

трудомісткість: 12 люд.-днів;

команда: 2 особи;

тривалість: 4 дні;

витрати: 6000₴ (зарплата) + 1500₴ (графічні матеріали, прототипи);

загалом: 7500₴.

**Розробка:**

трудомісткість: 24 люд.-дні;

команда: 2 особи;

тривалість: 8 днів;

витрати: 12000₴ (заробітна плата) + 2000₴ (бібліотеки, базове ПЗ);

загалом: 14000₴.

**Тестування:**

трудомісткість: 6 люд.-днів;

команда: 1 особа;

тривалість: 3 дні;

витрати: 3000₴ (зарплата) + 500₴ (засоби для перевірки);

загалом: 3500₴.

**Інтеграція та публікація:**

трудомісткість: 4 люд.-дні;

команда: 1 особа;

тривалість: 2 дні;

витрати: 2000₴ (зарплата) + 1500₴ (хостинг, домен, SSL);

загалом: 3500₴.

**Підтримка та супровід:**

трудомісткість: 8 люд.-днів (орієнтовно 2-3 год/тиждень);

витрати: 5000₴/рік (оновлення контенту, технічна підтримка).

Сукупні витрати на розробку та запуск: 38500₴

**Оцінка економічного ефекту:**

зменшення часу обробки заявок та інформування громадян – 12000€ / рік;  
покращення іміджу та залучення нових відвідувачів – 25000€ / рік;  
оптимізація внутрішньої комунікації та скорочення паперового документообігу – 8000€ / рік.

Загальний економічний ефект: 45000€ на рік.

Термін окупності:  $38500€ / 45000€ \approx 0,85$  року (приблизно 10 місяців).

Таким чином, створення веб-платформи для СКК «Софіївський» демонструє здатність окупитися протягом менше ніж року після запуску. Це не лише забезпечує економію ресурсів, а й сприяє підвищенню якості сервісу, більшій прозорості та полегшенню взаємодії між керівництвом комплексу та населенням.

### **1.7. Обґрунтування необхідності проектування та розробки вебсайту для СКК «Софіївський»**

Проведений аналіз показав, що на даний момент Спортивно-Культурний Комплекс «Софіївський» не має сучасної веб-платформи. Вона необхідна для того, щоб відвідувачі могли зручно отримувати інформацію про послуги, розклад занять, актуальні новини закладу, а також попередньо записуватися на секції чи заходи. Відсутність такого онлайн-інструменту ускладнює комунікацію з цільовою аудиторією, негативно впливає на впізнаваність бренду та гальмує цифровий розвиток.

В процесі аналізу поточного стану автоматизації (див. пункт 1.3) було встановлено, що більшість процесів взаємодії з клієнтами, як-от запис, бронювання залів, інформування про зміни у розкладі, здійснюються вручну або по телефону. Це не відповідає сучасним вимогам користувачів. Також розглянуто, як зміниться робота бізнесу після впровадження веб-сайту. Планується спростити процедуру запису на заняття, автоматизувати інформування відвідувачів про новини та зміни, зменшити навантаження на адміністративний персонал, а також організувати онлайн збір зворотного зв'язку.

У пункті 1.3 було проаналізовано веб-рішення, які успішно використовуються в інших спортивних закладах, таких як фітнес-клуби чи

спортивні академії. Ці приклади демонструють позитивний вплив наявності якісного сайту на організацію роботи та рівень обслуговування клієнтів. Водночас, типові рішення не враховують специфіку діяльності СКК «Софіївський» і не передбачають можливості масштабувати функціонал відповідно до потреб комплексу.

Розробка власного веб-сайту дає змогу створити платформу, яка повністю відповідатиме структурі, розкладу та послугам СКК. Планується впровадити такі функції: перегляд розкладу секцій, онлайн реєстрація, зворотний зв'язок, доступ до новин, фотогалереї та інтеграція з соціальними мережами. Використання сучасного технологічного стеку (наприклад, React + Node.js) забезпечить гнучкість у подальшому розвитку сайту - як в аспекті дизайну, так і функціонально.

Таким чином, розробка веб-сайту для СКК «Софіївський» є логічним і стратегічно обґрунтованим рішенням. Це не просто цифровий ресурс – це зручний інструмент для покращення взаємодії з відвідувачами, підвищення прозорості діяльності комплексу, зміцнення довіри та розширення аудиторії.

## РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

### 2.1. Загальна характеристика системи

Система, що розробляється, є веб-сайтом для Спортивно-Культурного Комплексу «Софіївський» і виконує функцію багаторівневої інформаційної платформи, яка має на меті не лише представлення установи в цифровому просторі, а й забезпечення ефективної взаємодії з цільовою аудиторією. У контексті сучасного інформаційного суспільства, коли цифрові канали комунікації є домінантними у сфері поширення інформації, популяризації послуг та організації внутрішніх і зовнішніх процесів, потреба у такій системі є обґрунтованою та невідкладною. Комплекс «Софіївський» активно розвивається як центр спорту, освіти і дозвілля для дітей, молоді та широкої громадськості, однак станом на час розробки не має власного офіційного сайту, що обмежує його можливості в інформаційному обміні, ускладнює процедури запису до секцій, популяризацію заходів, комунікацію з батьками, тренерами та партнерами.

Запропонована система повинна вирішити кілька практичних і комунікаційних задач одночасно. Насамперед, вона має забезпечити централізоване зберігання й упорядкування інформації про секції, тренерів, розклад занять, досягнення вихованців, фотозвіти з заходів та новини комплексу. Завдяки цьому система зменшує залежність від соціальних мереж як основного інформаційного каналу, уникаючи втрати даних у разі блокувань чи змін алгоритмів, і забезпечує довгострокову цифрову стабільність установи. Крім того, реалізація функціоналу зворотного зв'язку, онлайн-заявок на запис до секцій, інтеграції карти місцезнаходження, календаря заходів та новинного блоку сприяє підвищенню прозорості та зручності для потенційних і постійних відвідувачів.

У технічному сенсі система буде побудована за принципами адаптивної верстки, що дозволяє коректно відображати сайт на різних пристроях – смартфонах, планшетах, настільних ПК. Такий підхід забезпечує доступність незалежно від технічного оснащення користувача і враховує актуальні тенденції у поведінці споживачів цифрових продуктів. Структура сайту буде сформована з

урахуванням логіки розміщення контенту, де кожна сторінка матиме чітке функціональне призначення: головна сторінка для презентації комплексу, сторінки секцій з описами та графіками, сторінка новин для інформування, контактна сторінка для інтеграції засобів зв'язку тощо.

У межах системи також буде передбачено розділ адміністрування, що дозволить адміністрації закладу самостійно змінювати текстову і візуальну інформацію, додавати новини, розклад занять, оголошення, без потреби залучення розробників. Це сприятиме автономності закладу в цифровому просторі й зменшить витрати на зовнішню технічну підтримку. Така автономія в управлінні контентом є особливо важливою для комунальних установ, де питання економії ресурсів стоїть особливо гостро.

Розробка цієї системи є інструментом цифрової трансформації діяльності СКК «Софіївський» і водночас – засобом формування сучасного іміджу закладу в очах громадськості. Сайт виступатиме своєрідною вітриною, яка демонструє як матеріально-технічні можливості комплексу, так і його соціально-культурну значущість. Таким чином, інформаційна система, що створюється, є не лише технічним продуктом, а й складовою частиною стратегічного розвитку установи, що відповідає вимогам часу, очікуванням споживачів і стандартам прозорості комунікацій у сфері державного і муніципального управління.

## **2.2. Функціональні вимоги до веб-сайту**

Функціональні вимоги до веб-сайту Спортивно-Культурного Комплексу «Софіївський» охоплюють сукупність характеристик, які визначають поведінку системи під час взаємодії з користувачами та адміністрацією, а також описують, які саме операції система має бути здатною виконувати у відповідь на дії відвідувача або адміністративного персоналу. Ці вимоги формуються з урахуванням поточних потреб комплексу, особливостей його цільової аудиторії, а також загальноприйнятих стандартів побудови інформаційних систем у сфері спорту та дозвілля.

Перш за все, веб-сайт має виконувати інформативну функцію, тобто надавати користувачам доступ до структурованих і завжди актуальних даних про комплекс, його місцезнаходження, графік роботи, інфраструктуру, наявні спортивні та культурні секції, а також умови участі. Інформація про кожну секцію повинна бути представлена у вигляді окремої сторінки або блоку, що включає короткий опис напрямку, перелік тренерів із контактами, графік занять та інші специфічні характеристики (вікові обмеження, рівень підготовки, наявність вільних місць тощо). Для досягнення повної інформативності сайт має також містити розділ з новинами та анонсами подій, у якому регулярно публікуватимуться повідомлення про майбутні турніри, дні відкритих дверей, святкові заходи, зміни в розкладі тощо. Це забезпечить актуальність ресурсу і сприятиме залученню відвідувачів.

Крім інформативності, важливим функціональним компонентом є можливість інтерактивної взаємодії. Сайт має дозволяти користувачам залишати заявки на вступ до секцій або запис на консультацію з адміністрацією. Такий функціонал передбачає наявність інтерактивних форм, що збирають необхідні контактні дані користувача, обраний напрямок секції, вік дитини (для дитячих секцій) або бажану дату зустрічі. Для зручності ці форми повинні мати базову валідацію введених даних і формувати підтвердження відправлення. У разі потреби можна передбачити механізм автоматичного повідомлення адміністратора про нові заявки через email або внутрішню адмін-панель. Таким чином, інтерактивність сайту сприятиме автоматизації рутинних адміністративних завдань і покращенню комунікації з громадськістю.

Ще одним суттєвим функціональним аспектом є адаптивність і мультимовність. Сайт має коректно відображатися на мобільних пристроях, планшетах і десктопах, незалежно від операційної системи або браузера. Це означає, що всі елементи інтерфейсу (меню, зображення, текстові блоки, кнопки тощо) повинні динамічно змінювати своє розташування та розмір відповідно до параметрів екрану. Адаптивність напряму впливає на зручність користування і є стандартною вимогою для сучасних інформаційних систем. Мультимовність дозволить охопити не лише україномовну аудиторію, а й користувачів, які надають

перевагу іншій мові, наприклад англійській, що є актуальним для іноземців або партнерських організацій, які можуть брати участь у міжнародних заходах.

З огляду на потреби адміністрації закладу, веб-сайт повинен містити функціонал для керування контентом, що реалізується через систему адміністрування. Панель адміністратора має забезпечувати доступ авторизованим особам до редагування текстових розділів, додавання новин, оновлення розкладів занять, публікації фотозвітів та відео з заходів, а також перегляд отриманих заявок від користувачів. Для зручності адміністрування повинна бути передбачена можливість попереднього перегляду контенту перед публікацією, а також функції фільтрації, сортування та пошуку серед заявок. Важливо, щоб доступ до адмін-панелі був захищеним: передбачалося введення логіна й пароля, а також можлива двофакторна аутентифікація для забезпечення безпеки.

Окремо слід зазначити, що сайт має включати блок мультимедійних матеріалів — галереї з фотографіями та відео, які ілюструють життя комплексу, його заходи, тренувальні процеси тощо. Такий блок виконує функцію візуального представлення активностей, формує позитивний імідж закладу і дозволяє користувачам оцінити атмосферу та рівень організації ще до фізичного візиту. Крім того, корисною є інтеграція карти з точним геолокаційним маркером комплексу, що спрощує пошук маршруту для нових відвідувачів. Для зворотного зв'язку варто передбачити форму повідомлень або контактні кнопки з можливістю прямого переходу до месенджерів чи email, що відповідає сучасним вимогам комунікативної зручності.

У результаті реалізації всіх вищенаведених функціональних вимог веб-сайт стане повноцінним інформаційно-комунікаційним середовищем для Спортивно-Культурного Комплексу «Софіївський», яке не лише репрезентуватиме установу онлайн, а й значною мірою автоматизує її щоденну діяльність, сприятиме ефективному поширенню інформації, підвищенню прозорості управління та загальному зростанню суспільної зацікавленості до спортивного й культурного життя громади.

Система мусить втілювати всі основні функції, необхідні для продуктивної взаємодії з користувачами та керування спортивними секціями. Окрему увагу приділено функціям адміністрування: управлінню розкладом занять, базою тренерів і користувачів, а також обробці звернень. Це забезпечує повноцінну підтримку організаційної діяльності Спортивно-Культурного Комплексу «Софіївський». Перелік функцій наведено у таблиці 2.1.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

№	Найменування функції	Вхідна інформація	Вихідна інформація
1	Реєстрація клієнта	Ім'я, прізвище, email, номер телефону, пароль	Дані збережено, створено обліковий
2	Авторизація користувача	Email, пароль	Доступ до системи, токен
3	Перегляд розкладу секцій	Категорії секцій, пошукові фільтри (за віком, напрямом)	Список доступних секцій
4	Запис на секцію	ID секції, ID користувача	Запис збережено, підтвердження участі
6	Керування секціями (адміністратор)	Назва, опис, кількість місць, тренер	Оновлений список секцій
7	Керування тренерами (адміністратор)	Ім'я, прізвище, email, пароль	Додано/редаговано тренера
8	Керування користувачами (адміністратор)	Дані користувача	Список користувачів, оновлення даних
9	Обробка звернень користувачів	Повідомлення, контактні дані	Список звернень, статус обробки
10	Формування списків учасників секцій для тренера	Параметри звіту (період днів)	Кількість записів, нових користувачів

### 2.3. Нефункціональні вимоги

Нефункціональні вимоги до веб-сайту Спортивно-Культурного Комплексу «Софіївський» визначають ті характеристики системи, які не пов'язані безпосередньо з функціональністю, але істотно впливають на якість її роботи, зручність використання, безпечність, масштабованість та витрати на обслуговування. Ці вимоги охоплюють технічні, організаційні й експлуатаційні аспекти, які забезпечують стійку й надійну роботу системи у реальних умовах її використання як адміністративним персоналом, так і широким колом користувачів. Зокрема, вони формують основу для досягнення високих стандартів доступності, ефективності, адаптивності та підтримованості веб-продукту.

Одним з ключових нефункціональних критеріїв є надійність, яка передбачає, що веб-сайт має стабільно функціонувати протягом тривалого часу без збоїв або непередбачуваних зупинок. Це означає, що реалізація системи повинна враховувати механізми обробки помилок, належну структуру логіки запитів, перевірку на наявність несправностей у процесі завантаження сторінок та виведення даних. Важливо, щоб навіть у випадку часткової відмови сервера або мережевих неполадок користувачі могли бачити коректне повідомлення про тимчасову недоступність або обмеження функцій, що запобігатиме виникненню негативного враження від системи.

Також важливою є швидкодія системи, тобто час реакції сайту на дії користувача має бути мінімальним. Усі сторінки повинні завантажуватися протягом 2–3 секунд, що досягається за рахунок оптимізації графічних ресурсів, використання сучасних форматів стиснення, кешування даних та ефективного написання програмного коду. Враховуючи, що сайт орієнтований на широку аудиторію, включаючи батьків і дітей, доступ до інформації має бути максимально швидким і лаконічним. Крім того, продуктивність сайту повинна залишатися стабільною при підвищеному навантаженні, наприклад, під час масового перегляду розкладу занять у початку сезону або у дні відкритих заходів.

Невід'ємною вимогою до сучасних веб-продуктів є адаптивність інтерфейсу, що гарантує коректне відображення вмісту незалежно від типу пристрою

(смартфон, планшет, ноутбук, десктоп) і його роздільної здатності. Це передбачає гнучке верстання компонентів сайту, уникнення фіксованих розмірів елементів та використання медіа-запитів. Особливо критичною адаптивність є для користувачів мобільних пристроїв, які складають переважну частину трафіку для локальних установ. Таким чином, адаптивний дизайн дозволяє не лише забезпечити технічну доступність контенту, а й створює позитивний користувацький досвід, що є визначальним фактором повторного відвідування сайту.

Окрема категорія нефункціональних вимог стосується безпеки системи. Веб-сайт має бути захищеним від типових вразливостей, таких як SQL-ін'єкції, міжсайтове виконання скриптів (XSS), атаки типу CSRF тощо. Для цього необхідно впровадити базові політики безпеки: використання HTTPS для шифрування трафіку, валідацію всіх вхідних даних, захист форм від спаму через CAPTCHA або інші методи, обмеження доступу до адміністративної панелі лише авторизованим користувачам. Збереження особистих даних користувачів (у випадку подачі онлайн-заявок) також повинно відповідати принципам конфіденційності, що передбачає їх шифрування або обмежене використання лише для цілей зворотного зв'язку.

До нефункціональних вимог також належить зручність у підтримці та розширенні системи. Архітектура сайту повинна бути модульною, з чітко структурованими файлами, щоб у майбутньому можна було без зайвих труднощів оновлювати дизайн, змінювати логіку окремих сторінок або додавати новий функціонал (наприклад, електронну оплату за участь у секціях або інтеграцію з месенджерами). Це значно знижує витрати часу та коштів на подальше обслуговування ресурсу. Важливим є також створення технічної документації, яка міститиме опис структури проекту, налаштування середовища, інструкції для адміністратора, що дає змогу закладу самостійно адмініструвати сайт без звернення до розробника.

Ще однією групою вимог є юзабіліті, або зручність користування. Навігація по сайту має бути інтуїтивно зрозумілою, структура меню логічною, а кількість кроків до отримання потрібної інформації – мінімальною. Користувачі мають

швидко знаходити розклад занять, опис секцій, контактні дані, форму для подачі заявки тощо. Текстовий контент повинен бути чітким, зручним для читання, з дотриманням контрасту і шрифтів, що легко сприймаються. Також необхідно передбачити наявність пошукової системи або фільтраційних функцій для полегшення доступу до розділів при великому обсязі контенту.

Таким чином, нефункціональні вимоги визначають загальну якість реалізації веб-сайту Спортивно-Культурного Комплексу «Софіївський» і мають критичне значення для сприйняття ресурсу як сучасного, надійного та ефективного. Їх дотримання не лише забезпечує довговічність і безпеку роботи сайту, а й істотно підвищує його соціальну й організаційну ефективність, сприяючи активнішому залученню мешканців до спортивного життя громади.

#### **2.4. Архітектурні обмеження та середовище виконання**

Архітектурні обмеження та середовище виконання веб-сайту Спортивно-Культурного Комплексу «Софіївський» визначають ті технічні рамки, в межах яких буде здійснюватися розробка, розгортання, підтримка та подальше масштабування інформаційної системи. Обмеження накладаються як вимогами до типу платформи, так і технічними передумовами експлуатації на рівні хостингу, інтеграції з базами даних та реалізації клієнтського інтерфейсу. Ці фактори є визначальними для вибору технологічного стека, логіки розділення функціональних компонентів системи, методів доступу до даних і способів обміну інформацією між користувачем і серверною частиною.

Передусім визначальним архітектурним фактором є те, що система є веб-сайтом, тобто вона реалізується як клієнт-серверна веб-програма, до якої користувачі отримують доступ через веб-браузери. Це означає, що як фронтенд (інтерфейсна частина), так і бекенд (серверна логіка, робота з базою даних) повинні бути сумісні з протоколом HTTP/HTTPS, використовувати стандартні веб-технології та бути оптимізованими для взаємодії через інтернет без необхідності інсталяції додаткового ПЗ[11]. Сайт буде розгортатися на публічному хостингу або

VPS-сервері, який підтримує сучасні веб-фреймворки, мови програмування та засоби керування реляційними базами даних.

Друге обмеження — відсутність прив'язки до конкретної мови програмування — надає широкі можливості для вибору технологій, однак накладає відповідальність за забезпечення сумісності компонентів[12]. У зв'язку з цим можливими є реалізації бекенду мовами PHP, Python (наприклад, Flask або Django), JavaScript (Node.js), або навіть Java чи C#, залежно від кваліфікації розробника та умов хостингу. Фронтенд, як правило, розробляється із застосуванням HTML5, CSS3 та JavaScript, що є стандартом для побудови інтерфейсів. Також можливе використання фреймворків (Bootstrap, React, Vue.js), які забезпечують компонентну структуру інтерфейсу та зручність адаптивної верстки[13].

Третє обмеження вимагає наявності графічного інтерфейсу користувача, тобто система повинна мати повноцінну візуальну оболонку, що забезпечує просту, доступну й інтуїтивно зрозумілу взаємодію з відвідувачем сайту. Цей інтерфейс реалізується як набір візуальних компонентів: меню навігації, форми введення, таблиці даних, фотогалереї, інтерактивні кнопки тощо. Усі компоненти повинні бути адаптивними, підтримувати взаємодію через мишу або сенсорні екрани, а також відповідати стандартам веб-доступності (зокрема, контрастність, масштабування, альт-теги до зображень тощо). Інтерфейс є не лише естетичною складовою, а й логікою доступу до функцій: саме через нього користувачі отримують інформацію, надсилають заявки, переглядають графіки та новини.

Останнє архітектурне обмеження стосується використання SQL як базової технології для зберігання, обробки та пошуку даних. Це передбачає реалізацію реляційної моделі бази даних, у якій зберігатимуться такі типи інформації, як: дані про секції (назви, розклад, опис, тренери), новини, заявки користувачів, облікові записи адміністраторів. У зв'язку з цим доцільним є використання систем керування базами даних, які підтримують стандарт SQL: це можуть бути MySQL, PostgreSQL, SQLite або інші еквівалентні СКБД, залежно від масштабу та хостингових можливостей. Важливо, щоб доступ до бази даних був захищеним і

структурованим, із застосуванням параметризованих запитів або ORM-інструментів для уникнення вразливостей типу SQL-ін'єкцій.

Таким чином, розробка системи повинна здійснюватися з урахуванням зазначених обмежень: вона має бути веб-застосунком із повноцінним візуальним інтерфейсом, реалізованим на будь-якій з сумісних мов програмування, з підтримкою адаптивності, доступності та використанням SQL-орієнтованої бази даних для зберігання й обробки контенту. Такий підхід забезпечить масштабованість, технічну стабільність, функціональну повноту і відповідність сучасним вимогам до інформаційних систем публічного призначення.

## 2.5. Склад і зміст робіт по створенню системи

Стадії створення системи та терміни виконання наведені у таблиці 2.2.

*Таблиця 2.2. Найменування робіт при створенні системи*

№	Найменування робіт	Строки виконання
1	Передпроектний аналіз діяльності СКК “Софіївський”	28.04.2025 – 30.04.2025
2	Розробка технічного завдання	01.05.2025 – 03.05.2025
3	Проектування структури сайту та бази даних	04.05.2025 – 06.05.2025
4	Розробка дизайну інтерфейсу сайту	07.05.2025 – 08.05.2025
5	Підключення та налаштування XAMPP та phpmyadmin	08.05.2025 – 09.05.2025
6	Розробка серверної частини (Node.js)	10.05.2025 – 11.05.2025
7	Розробка клієнтської частини (React)	12.05.2025 – 13.05.2025
8	Реалізація особистого кабінету та адмін-панелі	14.05.2025 – 15.05.2025
9	Тестування функціоналу та виправлення помилок	15.05.2025 – 17.05.2025
10	Підготовка користувацької та технічної документації	17.05.2025 – 18.05.2025

Далі на буде наведено діаграму Ганта (рисунок 2.1), яка була створена на основі термінів зазначених у таблиці 2.2

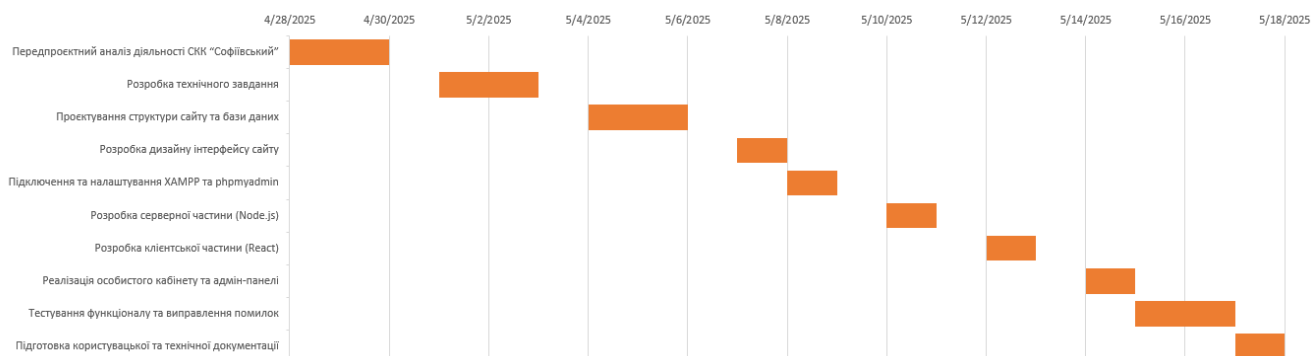


Рисунок 2.1 – Діаграма Ганта

## 2.6. Вимоги до структури та дизайну інтерфейсу користувача

Вимоги до структури та дизайну інтерфейсу користувача веб-сайту Спортивно-Культурного Комплексу «Софіївський» формуються з урахуванням необхідності забезпечення високої інформативності, інтуїтивної зручності, візуальної послідовності, доступності для різних категорій користувачів, а також відповідності функціональним завданням системи. Інтерфейс має бути адаптивним, тобто автоматично підлаштовуватися під розміри екрана пристрою, з якого здійснюється доступ, зберігаючи при цьому повноцінну логіку навігації та відображення вмісту. Дизайн повинен підтримувати єдину кольорову палітру, читабельну типографіку та чітку ієрархію елементів, що сприятиме швидкому сприйняттю інформації та уникненню перевантаженості сторінок.

На головній інформаційній сторінці сайту передбачається статичний блок із стислим описом діяльності СКК, історією створення, місією, напрямками роботи, контактною інформацією та інтегрованою картою місцезнаходження. Цей блок повинен бути розміщений у верхній частині структури сайту і доступний без авторизації. Додатково на сторінці має бути розділ новин або подій, що виводить анонси майбутніх заходів або оголошення. У нижній частині розміщується футер із контактами, посиланнями на соціальні мережі, політикою конфіденційності та можливістю швидкого переходу до форм запису.

Форма для запису на секції має бути компактною та доступною як зі сторінки кожної секції, так і окремо через загальне меню. Вона повинна містити стандартні поля: ПІБ студента, вік, контактні дані батьків (у разі, якщо студент — дитина), вибір секції зі списку (динамічно оновлюваного на основі даних бази), бажаний час або дні занять (із врахуванням наявного розкладу), а також коментар (необов'язковий). Після надсилання даних, користувач має отримувати візуальне підтвердження (success message) і при потребі лист на електронну пошту. Важливо реалізувати валідацію обов'язкових полів та захист від спаму.

Форма звернення у відділ підтримки має бути доступна як окрема сторінка з меню або як вікно модального типу. Доцільно реалізувати її у вигляді простої контактної форми, де користувач зазначає тему звернення, своє ім'я, електронну пошту, а також текст повідомлення.

Можливість реєстрації має бути реалізована як на стартовій сторінці, так і в окремому пункті головного меню. Процес реєстрації повинен передбачати заповнення базових даних користувача: ім'я, електронна адреса, пароль, а також можливу роль (студент або тренер/адміністрація). Для студентів, що реєструються, система повинна автоматично створювати особистий кабінет, доступ до якого відкривається лише після успішної автентифікації.

Особистий кабінет студента повинен реалізовуватися як окремий інтерфейс з вертикальним меню, де користувач зможе переглядати поточний розклад занять, список секцій, на які він записаний, а також записатися на нові секції за аналогічною формою, описаною вище. Тут же має відображатися будь-яка індивідуальна інформація або примітки від тренера (наприклад, щодо досягнень, необхідності відпрацювання, рекомендацій). Також передбачається реалізація перегляду історії заявок, можливість редагувати особисті дані, змінювати пароль і виходити з кабінету.

Особистий кабінет адміністрації (тренерів, методистів, координаторів) має включати інструменти для перегляду повного списку учасників відповідних їм секцій, можливість сортування й пошуку учасників, а також створення індивідуальних приміток для кожного студента. Ці примітки повинні

відображатися в особистому кабінеті студента. Додатково тренер має мати можливість формувати звітність — зокрема, експортувати список учасників, відмітки про відвідування, статистику по секції тощо. Для реалізації таких звітів може бути передбачено створення у форматах CSV або PDF з попереднім переглядом і фільтрацією даних. Увесь адміністративний функціонал має бути захищений правами доступу, а дії тренерів логуватись для забезпечення прозорості.

Загальна концепція інтерфейсу передбачає логічне групування функціоналу відповідно до ролі користувача, чітку навігацію, мінімалістичний і не перевантажений стиль, а також застосування візуальних маркерів (іконок, кольорових індикаторів, статусів), які допомагають швидко орієнтуватися в системі. Це забезпечує високий рівень юзабіліті для обох категорій користувачів — як для нових відвідувачів, які шукають загальну інформацію про комплекс, так і для постійних учасників секцій та адміністративного персоналу, які щоденно взаємодіють із системою.

## **2.7. Порядок контролю і приймання системи**

2.7.1 Розроблений вебсайт для Спортивно-Культурного Комплексу «Софіївський» вступає в дію на основі погодженого технічного завдання. Перед стартом система мусить пройти приймальні випробування згідно з вимогами ДСТУ 3974-2000.

2.7.2 Випробування здійснюються розробником спільно з представниками адміністрації СКК «Софіївський». Вони містять перевірку працездатності ключових функцій: реєстрації користувачів, запису на секції, роботи адмін-панелі, захисту персональних даних. Програму перевірки готує розробник, а затверджує її відповідальна особа від СКК.

2.7.3 Сайт передається в дослідну експлуатацію на підставі технічного завдання та інструкції користувача. В процесі тестового використання виявляються можливі недоліки, за результатами чого формується перелік доопрацювань із зазначенням рекомендованих термінів їхнього виконання.

2.7.4 Після завершення дослідної експлуатації та усунення виявлених недоліків сайт вводиться в постійну експлуатацію. Факт прийняття системи оформлюється актом здачі-приймання між розробником та замовником.

## **2.8. Вимоги до складу і змісту робіт із підготовки до введення системи в дію**

Для запуску вебсайту в роботу замовник виконує низку підготовчих дій, що охоплюють:

- забезпечення наявності технічних засобів (комп'ютери, мобільні пристрої з доступом до інтернету) для роботи з сайтом;
- організацію навчання персоналу, відповідального за адміністрування сайту, з ознайомленням із функціоналом та інструкцією користувача;
- проведення дослідної експлуатації для перевірки працездатності головних функцій та збору відгуків від працівників;
- після успішного тестування — ухвалення рішення про повноцінне введення сайту в експлуатацію.

## **2.9. Вимоги до документації**

Для збудованої інформаційної системи у вигляді веб-сайту СКК "Софіївський" створюється збірка проектної документації, котра вміщує технічне завдання та технічний проект.

Вся документація оформлюється згідно з вимогами Державних стандартів серій:

- ДСТУ 19 – Єдина система програмної документації
- ДСТУ 24 – Єдина система стандартів автоматизованих систем управління

## **2.10. Джерела розробки**

При розробленні технічного завдання на систему використано наступні документи:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;

- ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом;
- ДСТУ ISO/IEC 27001:2015 Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вимоги;
- ДСТУ ISO/IEC TR 29110-5-1-2:2015 Інженерія програмного забезпечення. Профілі життєвого циклу для малих організацій;
- ДСТУ ISO/IEC/IEEE 29119-1:2015 Тестування програмного забезпечення. Частина 1. Поняття і визначення;
- ДСТУ ISO 9241-210:2019 Ергономіка взаємодії людина-система. Частина 210. Людиноцентричне проектування інтерактивних систем.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1. Вибір інструментальних засобів розробки

Користувацький інтерфейс розробляється за допомогою бібліотеки React, яка дозволяє створити компоненти для кожного функціонального елементу — окрема сторінка для інформації про комплекс, форма запису на секцію, форма звернення, сторінки реєстрації та входу, а також особисті кабінети студентів і тренерів[14]. Всі компоненти стилізуються за допомогою чистого CSS: стилі створюються вручну для кожного елементу, без використання сторонніх бібліотек або фреймворків. Це дозволяє повністю контролювати зовнішній вигляд інтерфейсу, мінімізує обсяг коду, зменшує залежності та дає можливість адаптувати дизайн під конкретні потреби замовника[15]. Для організації стилів можуть використовуватись CSS-модулі або окремі файли для кожного компонента.

Серверна частина реалізується на Node.js з використанням Express.js як мінімалістичного веб-фреймворку для обробки HTTP-запитів[16]. Сервер відповідає за обробку запитів з React-додатку: надсилання форм, автентифікацію користувачів, взаємодію з базою даних, логіку для кабінетів, формування відповідей API. Авторизація користувачів реалізується через сесії або токени (наприклад, JWT), а всі запити маршрутизуються через Express[17].

Дані зберігаються в MySQL, де створюється повноцінна реляційна база з таблицями: користувачі (із поділом на студентів і тренерів), секції, розклад, заявки, примітки тренерів, повідомлення[18]. З Node.js взаємодія з базою здійснюється через бібліотеку mysql2 або ORM-інструмент Sequelize (якщо потрібна вища абстракція)[19]. Структура БД передбачає зв'язки один-до-багатьох (наприклад, тренер – багато секцій; секція – багато студентів), а також таблиці для логування звернень і звітів.

Фронтенд (React) і бекенд (Node.js) можуть бути розгорнуті окремо. Наприклад, фронтенд — на Netlify або Vercel, бекенд — на Render, Railway або

Heroku, база — на PlanetScale (MySQL-сумісна) або локальному VPS. Передача даних здійснюється через REST API, формат — JSON[20].

Таким чином, обраний стек — React + CSS + Node.js + MySQL — забезпечує достатню функціональність, контроль над усіма етапами розробки, мінімальні витрати на залежності та повну адаптованість до специфіки невеликої публічної установи без ускладнень, пов'язаних із перевантаженими фреймворками чи інструментами.

### **3.2. Проєктування архітектури сайту**

Наведена архітектурна діаграма веб-сайту Спортивно-Культурного Комплексу «Софіївський» (Див. додаток А.1.) побудована за принципами трирівневої архітектури, яка передбачає розділення системи на клієнтський рівень (інтерфейс користувача), серверну логіку (обробка запитів та бізнес-логіка) і рівень бази даних (зберігання та управління інформацією). Такий підхід забезпечує модульність, масштабованість, повторне використання компонентів, а також дозволяє розробляти та супроводжувати окремі частини системи незалежно одна від одної, що відповідає сучасним принципам побудови веборієнтованих інформаційних систем.

Клієнтська частина (React) реалізує інтерфейс взаємодії користувачів з системою. До її складу входять компоненти, що забезпечують доступ до функціональних можливостей сайту відповідно до ролей користувачів. Форма запису на секції надає можливість подання заявки до обраного напрямку занять; ця форма є доступною без попередньої автентифікації та є основним засобом взаємодії для нових користувачів. Головна сторінка виконує інформативну функцію — надає загальні відомості про комплекс, секції, тренерський склад, місцезнаходження та контактну інформацію. Форма звернення у підтримку забезпечує комунікацію користувача з адміністративним персоналом або технічною підтримкою — ця форма дозволяє надсилати запити, звернення або скарги. Реєстрація / Вхід реалізує автентифікацію користувача і доступ до захищених частин системи — особистих кабінетів. Після входу користувач або потрапляє до Особистого кабінету студента,

де має змогу переглядати розклад, список секцій, отримувати повідомлення від тренерів і записуватись на інші секції, або до Особистого кабінету адміністрації, у якому тренери можуть бачити списки учасників, формувати примітки до студентів і створювати звітності за секціями.

Серверна частина (Node.js + Express) виступає проміжною ланкою між клієнтом і базою даних, реалізуючи логіку обробки запитів. Кожен контролер API відповідає за певну групу функціональних можливостей. API контролер заявок обробляє дані з форми запису на секцію та зберігає їх у відповідну таблицю бази даних (applications). API контролер секцій забезпечує доступ до інформації про доступні секції, яку запитує головна сторінка або особистий кабінет користувача. API контролер користувачів відповідає за створення, редагування, читання та обробку повідомлень у системі звернень до підтримки, а також керування профілями. API контролер автентифікації реалізує процес реєстрації, входу та автентифікації користувача, перевіряючи дані проти таблиці users. API контролер розкладу дозволяє отримати або змінити дані про графік занять, які зберігаються у таблиці schedule, і є доступними як студенту (для перегляду), так і тренеру (для редагування). API контролер приміток / звітів дозволяє адміністрації формувати примітки до кожного студента (наприклад, коментарі щодо відвідування чи поведінки), а також створювати звіти, які можуть бути експортовані у визначеному форматі.

Рівень бази даних (MySQL) зберігає структуровану інформацію, необхідну для забезпечення цілісності та логіки роботи системи. Таблиця users містить облікові записи як студентів, так і тренерів, із вказанням ролі. Таблиця sections містить інформацію про спортивні секції, до яких можна записатися. Таблиця applications зберігає подані заявки на участь у секціях. Таблиця schedule містить детальний розклад занять за секціями, днями та часом. Таблиця notes використовується для зберігання особистих приміток, які тренер залишає студенту. Таблиця messages зберігає повідомлення, надіслані через форму підтримки.

Всі взаємодії в системі реалізуються через RESTful API, де клієнтська частина надсилає HTTP-запити до відповідного контролера серверної частини,

який, у свою чергу, взаємодіє з базою даних, отримуючи, змінюючи або додаючи інформацію. Така архітектура дозволяє легко масштабувати систему, додавати нові функції (наприклад, календар подій, реєстрація на змагання, аналітика), розгортати окремі частини на різних серверах, а також забезпечити ефективну багаторольову модель взаємодії з єдиною централізованою системою.

Таким чином, представлена архітектура є цілісним рішенням для забезпечення повноцінної цифрової присутності Спортивно-Культурного Комплексу «Софіївський», поєднуючи прозору структуру з чітко визначеними відповідальностями, що гарантує якісне користувацьке обслуговування, автоматизацію процесів керування, та надійне зберігання даних.

Структура бази даних, зображена на діаграмі (Див. додаток А.2.), представляє собою реляційну модель, орієнтовану на організацію навчального процесу, зокрема — на керування секціями (групами), записами користувачів, плануванням занять, а також системою підтримки та нотаток. У моделі використано одинадцять таблиць, які між собою пов'язані зовнішніми ключами. Наведено детальний опис основних таблиць і їхніх зв'язків.

Центральною таблицею є `enrollments`, що відіграє роль зв'язувальної таблиці між студентами (`users`), тренерами (також у `users`) та секціями (`sections`). У таблиці зберігаються записи про зарахування користувачів до певних секцій, зазначено статус (наприклад, `active` або `cancelled`), час сесії та коментарі.

Таблиця `users` містить інформацію про всіх користувачів системи. У ній зберігаються такі поля, як ім'я, прізвище, електронна пошта, хеш пароля, а також зовнішній ключ `role_id`, що вказує на відповідну роль користувача в таблиці `roles`. Це дає змогу диференціювати права доступу або функціональні можливості, наприклад, між студентами та тренерами.

Таблиця `sections` є логічною сутністю, яка описує конкретну секцію або групу занять. Вона включає назву, опис, максимальну кількість учасників (`capacity`) та зовнішній ключ `trainer_id`, який посилається на таблицю `users` і вказує на відповідального тренера.

Таблиця `section_schedules` деталізує розклад секцій. Вона містить посилання на `section_id` та описує, у який день тижня та в який час (початок та кінець) проходять заняття. Це дозволяє реалізувати багаторазові (періодичні) сесії без дублювання секцій.

Таблиця `sessions` конкретизує окремі заняття в межах записів у `enrollments`. Вона містить посилання на `enrollment_id`, дату та час сесії, а також тренера, який її проводить. Це дає змогу вести облік відвідувань або подій у рамках навчального процесу.

Таблиця `notes` зберігає текстові нотатки, які прикріплені до конкретного запису користувача в секції (тобто `enrollment_id`). Також є автор нотатки (через `author_id`, що вказує на `users`) і час створення. Це може використовуватись для зворотного зв'язку або ведення педагогічних спостережень.

Таблиця `support_requests` реалізує систему звернень до технічної підтримки. Вона пов'язана з користувачами через `user_id`, включає тему звернення, повідомлення, час створення та статус (`open` або `closed`). Це забезпечує можливість взаємодії користувачів із підтримкою напряму.

Таблиця `roles` є довідником ролей у системі. Вона містить лише ідентифікатор ролі (`role_id`) та її назву. Через цю таблицю реалізовано базову рольову модель авторизації, яку використовує таблиця `users`.

Таким чином, структура бази даних є повністю нормалізованою, з чітко визначеними зв'язками між таблицями. Вона дозволяє відслідковувати участь користувачів у секціях, управляти розкладами, взаємодією через технічну підтримку, а також містить механізми для занотовування прогресу або особливостей проходження занять. Логіка побудови дає змогу масштабувати систему, впроваджувати багаторівневий контроль доступу та вести облік діяльності без втрати цілісності даних.

### **3.3. Розробка основних модулів веб-сайту**

Розробка основних модулів веб-сайту ґрунтується на розмежуванні обов'язків між інтерфейсом користувача, бізнес-логікою та обробкою даних. У

межах модуля автентифікації реалізовано механізм, який дозволяє користувачам реєструватися та входити в систему. Для цього у React створюється форма, де значення з полів відслідковуються через хук `useState`, а запит надсилається методом `POST` до бекенду. Наприклад, у компоненті входу реалізовано:

```
<input type="email" value={email} onChange={e => setEmail(e.target.value)} />
```

Це дозволяє оновлювати локальний стан форми під час введення даних, а після натискання кнопки викликається `handleLogin`, у якому формується запит до `API /api/auth/login`, що передає `email` та пароль у форматі `JSON`.

На серверному рівні відповідний маршрут обробляє ці дані, перевіряє їх в базі `MySQL` і у випадку успішної аутентифікації формує токен доступу:

```
const token = jwt.sign({ id: user.id, role: user.role }, process.env.JWT_SECRET);
```

Цей токен зберігається у клієнта і використовується для захисту подальших запитів.

У модулі запису на секцію користувач у фронтенді вибирає напрямок занять із випадуючого списку, вводить ім'я й надсилає заявку через `HTTP`-запит. Наприклад, вибір секції реалізується елементом:

```
<select onChange={e => setSectionId(e.target.value)}>
  <option value="1">Футбол</option>
  <option value="2">Бокс</option>
</select>
```

На сервері ці дані обробляються маршрутом `Express`, який виконує `SQL`-запит до таблиці `applications`, зберігаючи нову заявку. Таким чином, кожен запит із `React` додає новий рядок до таблиці заявок без необхідності перезавантаження сторінки.

Особистий кабінет студента реалізовано як окремий `React`-компонент, який після автентифікації завантажує персоналізовані дані через захищені `API`-запити. Для прикладу, запит на отримання приміток тренера виглядає так:

```
useEffect(() => {
  fetch('/api/student/notes', {
    headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }
  })
```

```
}).then(res => res.json()).then(data => setNotes(data));
}, []);
```

Сервер при цьому перевіряє токен у заголовку, розшифровує ідентифікатор користувача, і відповідно до ролі обирає дані з таблиці notes. Кожен запис передається клієнту, де рендериться у вигляді списку приміток або таблиці, з прив'язкою до відповідних секцій або тренерів.

У свою чергу, особистий кабінет адміністрації дозволяє тренеру бачити списки учасників секцій, робити текстові примітки та формувати звітність. Примітка створюється шляхом заповнення форми, яка після надсилання генерує POST-запит до API. На боці сервера дані зберігаються у таблиці notes, а відображаються студенту у кабінеті в розділі «Мої примітки».

Усі модулі взаємодіють із базою даних через чітко структуровані SQL-запити. Наприклад, додавання нової заявки реалізується простим рядком:

```
await db.query('INSERT INTO applications (name, section_id) VALUES (?, ?)',
[name, sectionId]);
```

Таким чином, основні модулі вебсайту сформовані як повноцінні функціональні блоки, що інтегрують логіку на стороні клієнта й сервера з централізованим зберіганням інформації у базі даних, забезпечуючи наскрізну взаємодію між користувачами, адміністраторами й даними системи.

### 3.4. Інструкція користувача

Після успішного завантаження сайту Користувач потрапляє на головну сторінку (Рис. 3.1.) де він може побачити головну сторінку та натиснути на одну з двох кнопок:

- увійти (якщо користувач має акаунт);
- реєстрація (якщо користувач не має акаунту) .

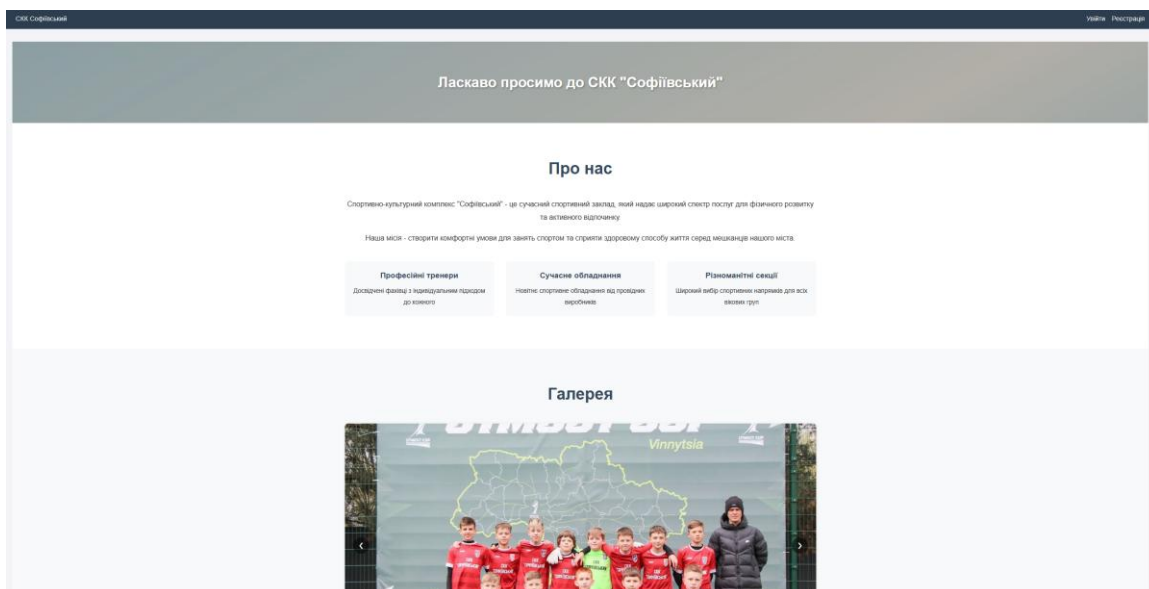


Рисунок 3.1 – Головна сторінка сайту

Також на головній сторінці користувач може:

- Ознайомитися з **інформацією про заклад**
- Переглянути **географічне розташування на карті**
- Залишити запит через **форму зворотного зв'язку**

(Рис. 3.2)

### Зв'язатися з нами

Ваш телефон

Текст звернення

Надіслати

### Як нас знайти

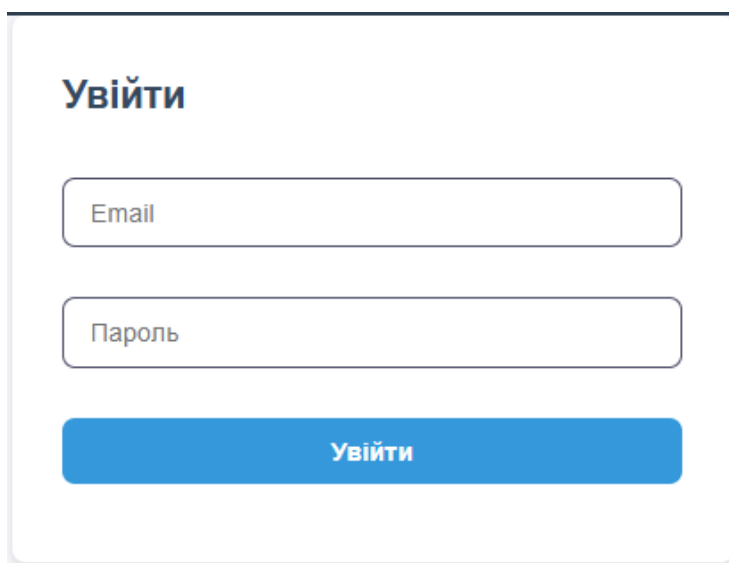
СКК "Софіївський" розташований за адресою:  
вул. Соборна, 114б  
с. Софіївська Борщагівка  
Київська область

Ми знаходимося в зручному місці з хорошою транспортною розв'язкою. Поруч є зупинки громадського транспорту та місця для паркування.



Рисунок 3.2 – Головна сторінка зі зворотнім зв'язком та можливістю зворотнього зв'язку

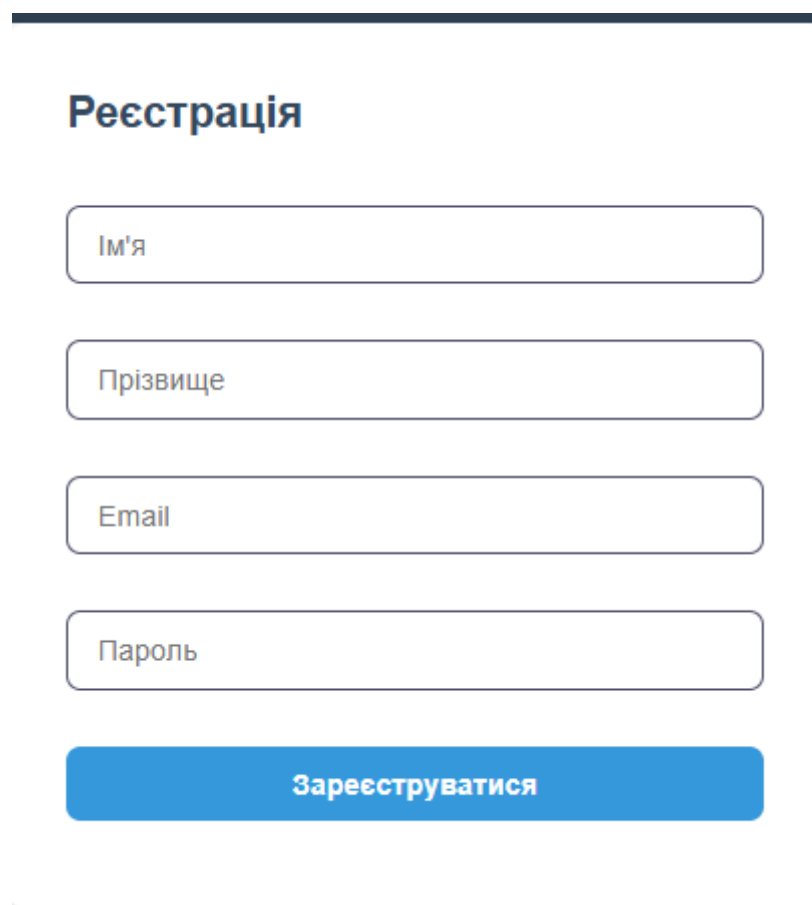
Якщо користувач натиснув на кнопку увійти (Рис. 3.2), він потрапляє у меню, де може ввести свою пошту та пароль після чого потрапить у свій кабінет.



The image shows a login form with the title "Увійти" (Login) in bold blue text. Below the title are two input fields: "Email" and "Пароль" (Password). At the bottom of the form is a blue button with the text "Увійти" (Login).

Рисунок 3.3 – Авторизація на сайт

Якщо користувач натиснув на кнопку реєстрації (Рис. 3.3), він потрапляє у меню, де може ввести своє ім'я, прізвище, свою пошту та пароль після чого потрапить у свій кабінет.



The image shows a registration form with the title "Реєстрація" (Registration) in bold blue text. Below the title are four input fields: "Ім'я" (Name), "Прізвище" (Surname), "Email", and "Пароль" (Password). At the bottom of the form is a blue button with the text "Зареєструватися" (Register).

Рисунок 3.4 – Реєстрація на сайт

Після успішного входу в систему користувач опиняється в особистому кабінеті, що відкриває перед ним такі можливості:

- запис до секцій через інтуїтивно зрозумілу форму;
- перегляд розкладу занять, прив'язаних до конкретних секцій;
- отримання важливих сповіщень;

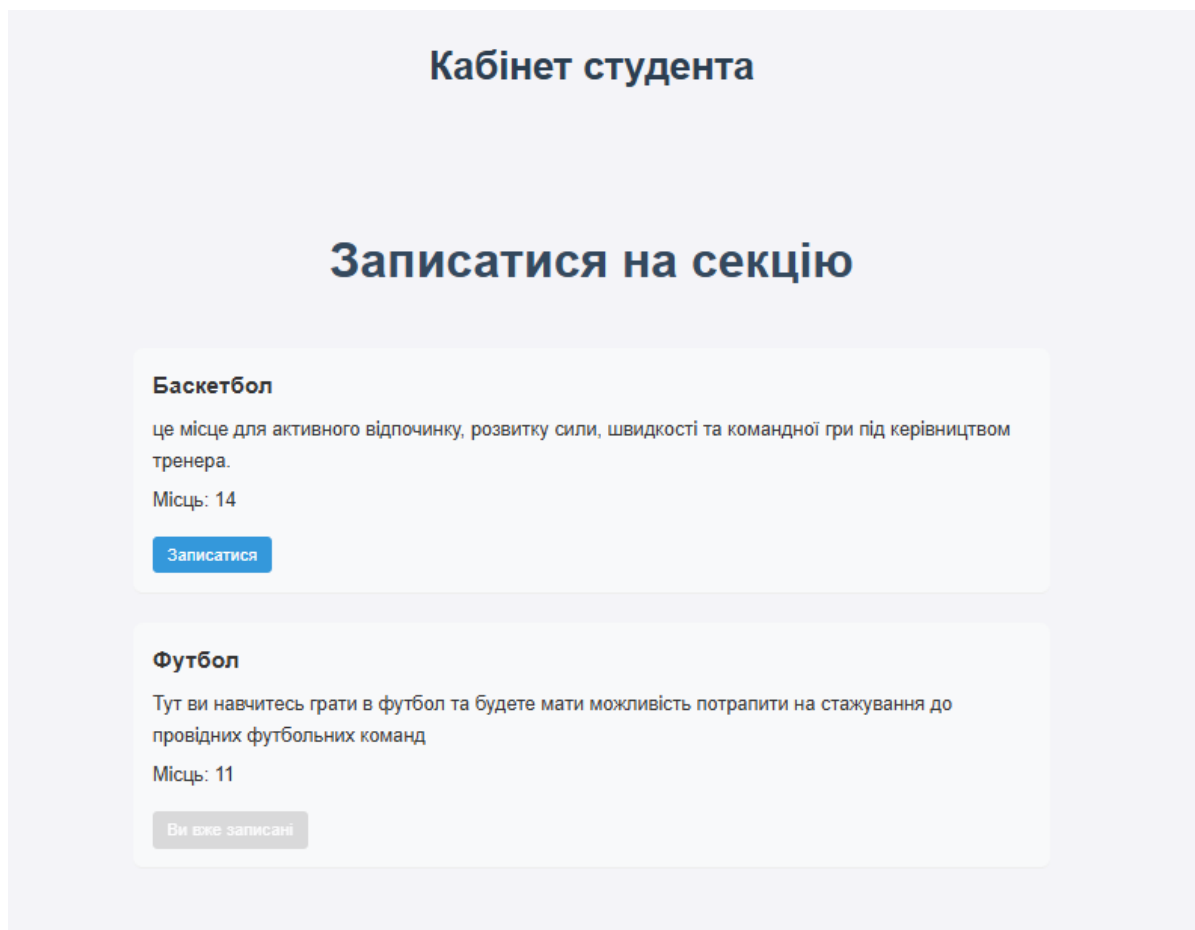


Рисунок 3.5 – Кабінет студента, запис на секції

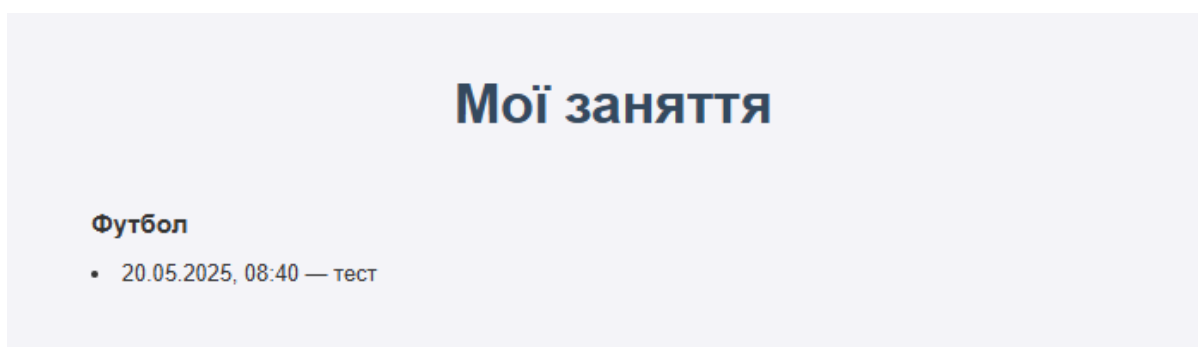


Рисунок 3.6 – Заняття студента по секціях

Після авторизації користувач з правами адміністратора потрапляє до особистого кабінету адміністратора, де має повний доступ до функцій керування сайтом.

Один із ключових функціональних модулів — обробка звернень користувачів через форму зворотного зв'язку.

Адміністратор сайту може:

- переглядати перелік отриманих запитів;
- надати відповідь користувачу (наприклад, за вказаним номером телефону);
- закрити звернення після його обробки

Описані функції наведені на рисунку 3.7.

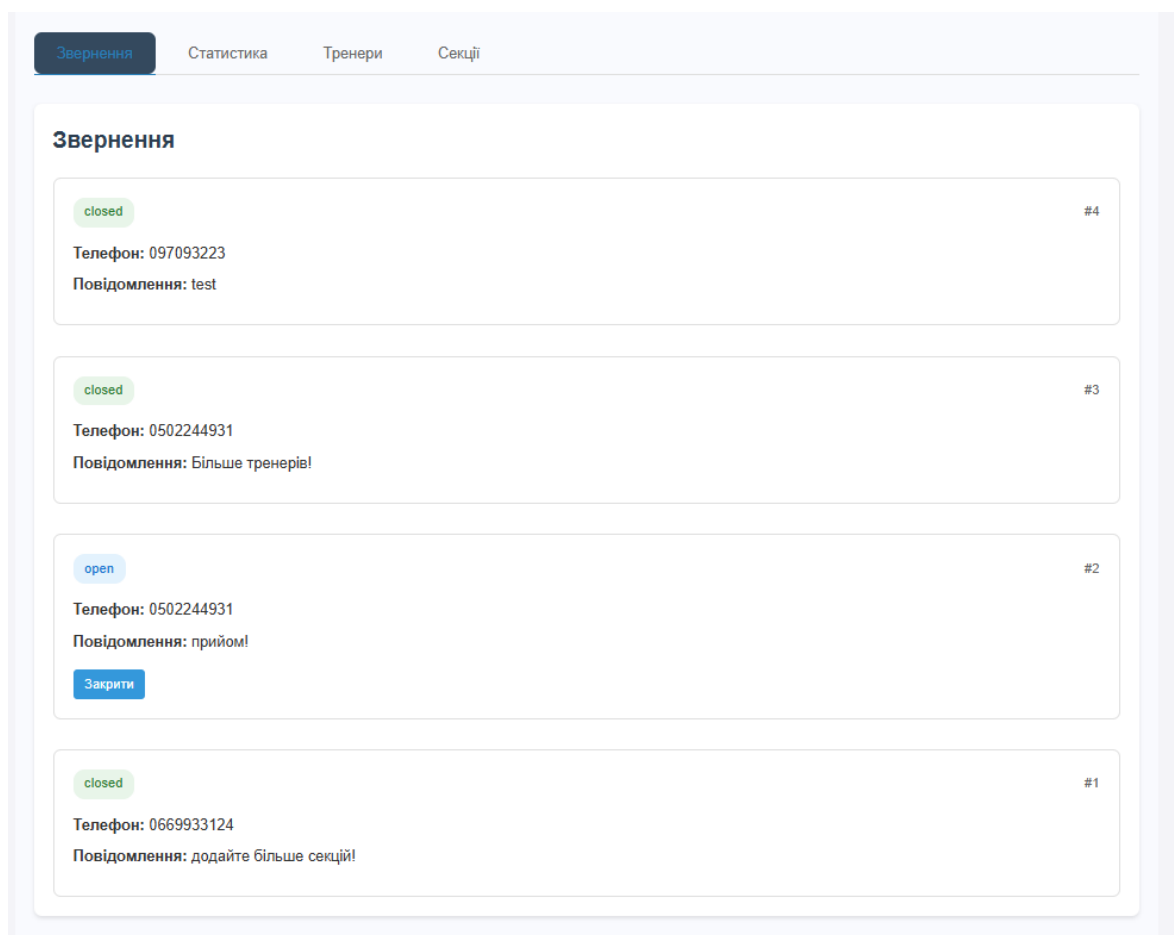


Рисунок 3.7 – Особистий кабінет адміністратора, модуль звернення

Окрім звернень, у кабінеті доступна **панель аналітики**, де відображається:

- статистика відвідуваності сайту;
- кількість реєстрацій;
- активність користувачів за періодами.

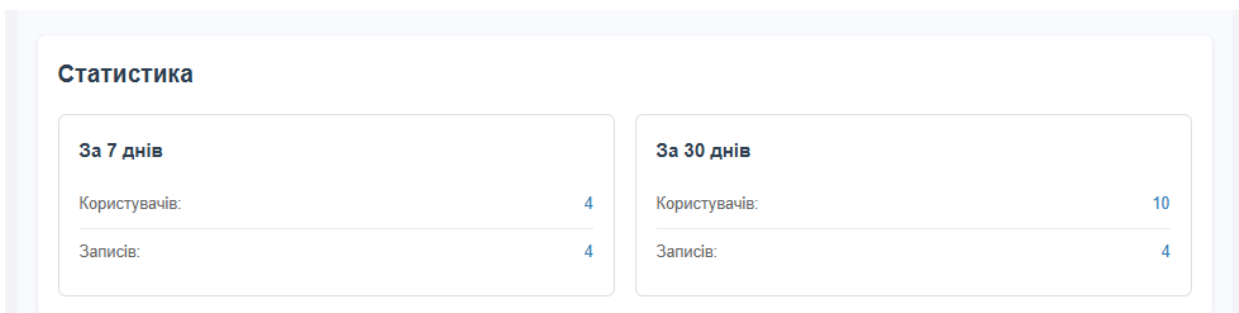


Рисунок 3.8 – Особистий кабінет адміністратора, з переглядом активності

Ще одна важлива функція адміністратора — керування персоналом. У відповідному модулі адміністратор може:

- додавати нових тренерів;
- редагувати інформацію про наявних інструкторів;
- видаляти тренерів за потреби.

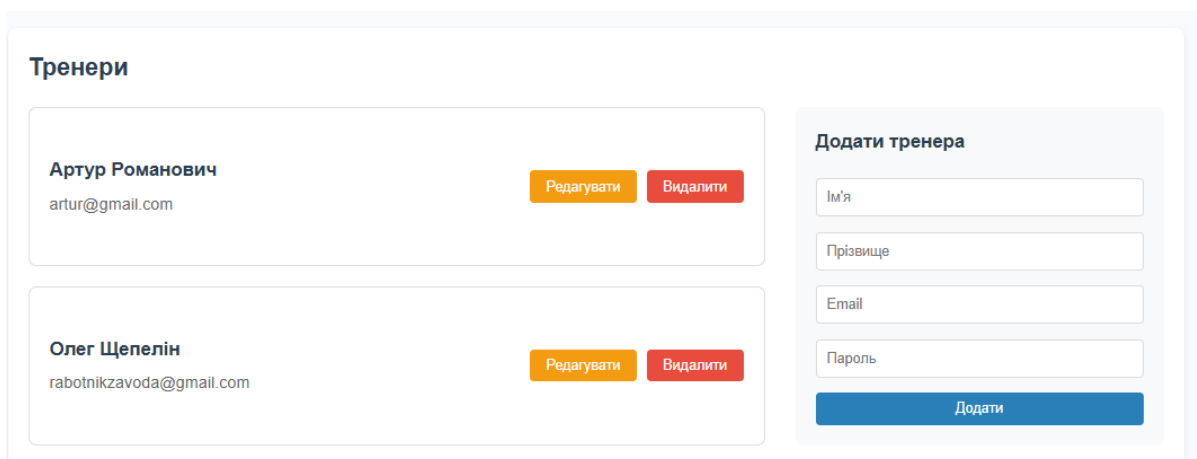


Рисунок 3.9 – Особистий кабінет адміністратора, модуль редагування та додавання тренерів

Також у сусідній вкладці реалізовано функціонал **керування секціями**, де доступні операції:

- додавання нових секцій;
- редагування параметрів існуючих секцій;
- видалення секцій, що втратили актуальність.

**Секції**

**Футбол**  
Тут ви навчитесь грати в футбол та будете мати можливість потрапити на стажування до провідних футбольних команд  
Місце: 11 Тренер: Олег Щепелін  
[Редагувати](#) [Видалити](#)

**Баскетбол**  
це місце для активного відпочинку, розвитку сили, швидкості та командної гри під керівництвом тренера.  
Місце: 14 Тренер: Артур Романович  
[Редагувати](#) [Видалити](#)

**Додати секцію**

Назва

Опис

Кількість місць

Оберіть тренера

[Додати](#)

Рисунок 3.10 – Особистий кабінет адміністратора, модуль редагування та додавання секцій

Після входу до системи користувач з роллю **тренера** потрапляє до власного кабінету, де має можливість:

- створювати заняття для закріплених секцій із зазначенням дати, часу та теми;
- редагувати або видаляти попередні або скасовані заняття;
- переглядати список студентів, які зареєструвались на відповідну секцію.

**Додати групове заняття**

Оберіть секцію

дд.мм.гггг --:--

Коментар до заняття

[Додати заняття](#)

Секція	Час	Коментар	Дії
Баскетбол	дд.мм.гггг --:-- <input type="text"/>	тест <input type="text"/>	<a href="#">Скасувати</a>

Рисунок 3.11 – Особистий кабінет тренера, модуль додавання заняття для секції

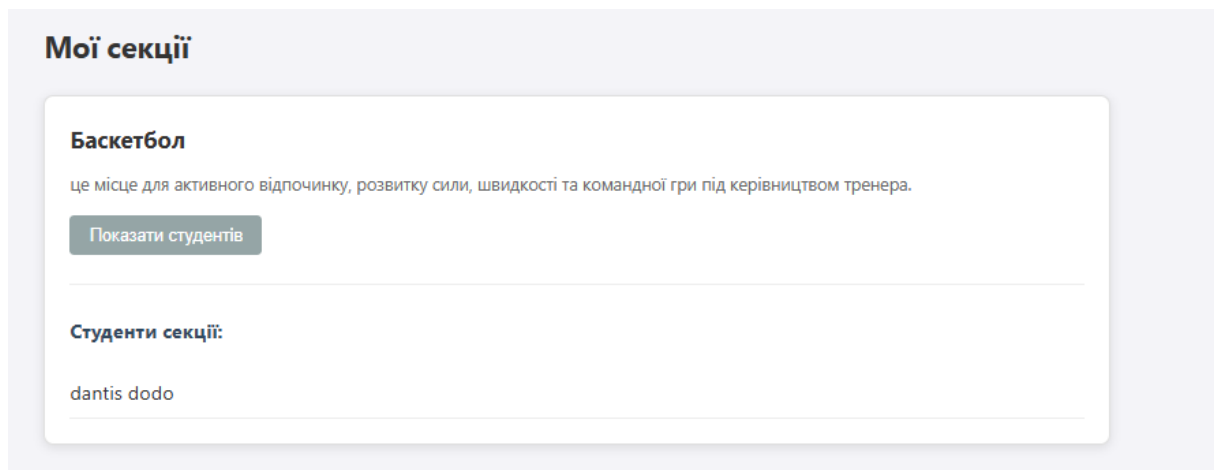


Рисунок 3.12 – Особистий кабінет тренера, модуль для перегляду студентів секції

### 3.5. Тестування програмного продукту

Процес тестування веб-сайту для управління секціями та тренерами було проведено з метою перевірки коректності функціонування усіх ключових елементів системи. Перше тестування охоплювало форму звернення до адміністрації, де перевірялися правильність введення номера телефону та тексту повідомлення, коректне надсилання запиту, його збереження у базі даних, а також відображення у розділі перегляду запитів (Рис. 3.13 – 3.14).

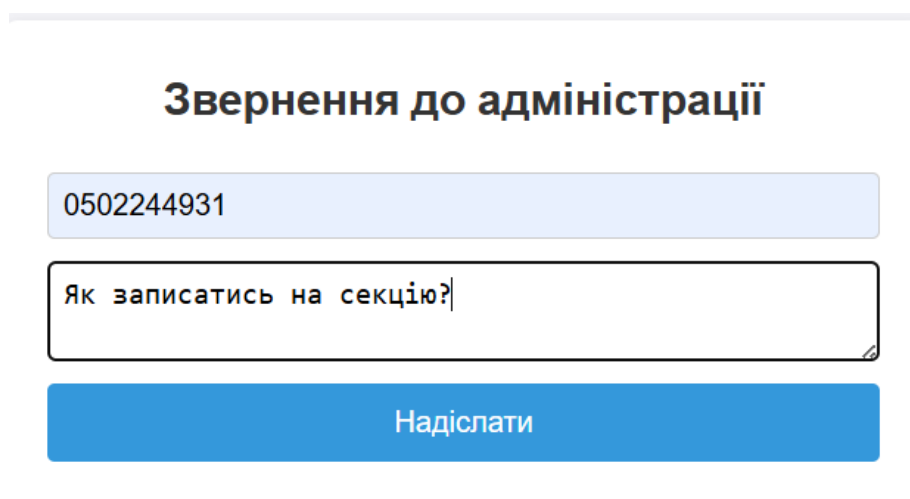


Рисунок 3.13 – Форма для звернення до адміністрації

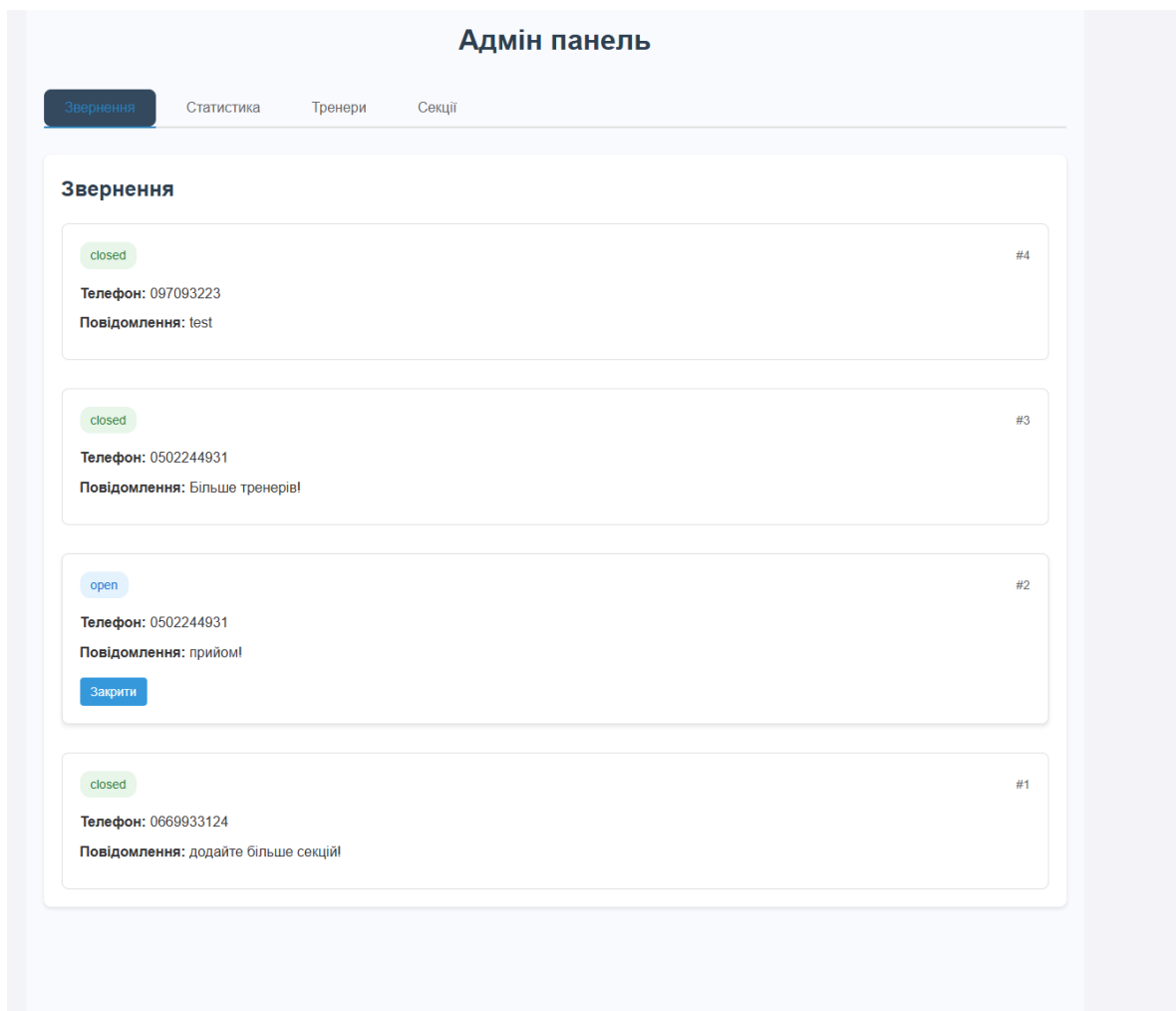


Рисунок 3.14 – Зареєстровані звернення

На цьому етапі було зафіксовано, що звернення надсилаються лише після перевірки формату введених даних, а також наявності тексту звернення.

Наступним етапом було тестування звітності, зокрема формування звітів за користувачами у розрізі семиденних і тридцятиденних періодів (Рис. 3.15).

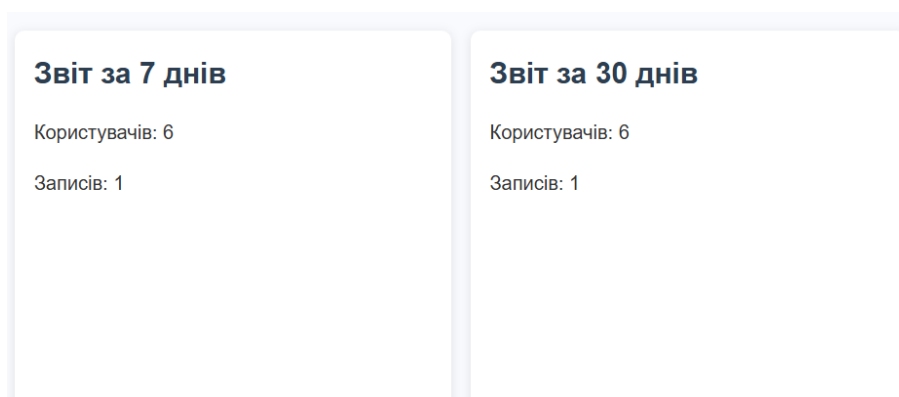
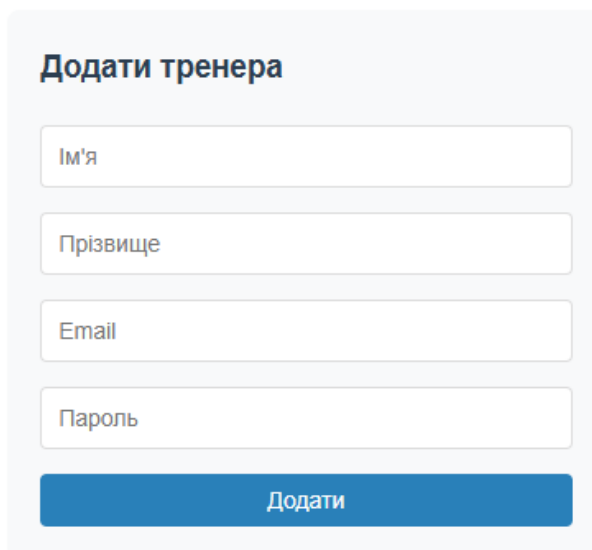


Рисунок 3.15 – Звіт по користувачах

Було перевірено, що система коректно рахує кількість запитів, їхню хронологію та виводить точну інформацію за відповідними фільтрами. Це

дозволило переконатися в правильності реалізації аналітичних функцій веб-застосунку.

Тестування модуля тренерів включало перевірку можливості додавання (Рис. 3.16), редагування та видалення записів про тренерів (Рис. 3.17).



**Додати тренера**

Ім'я

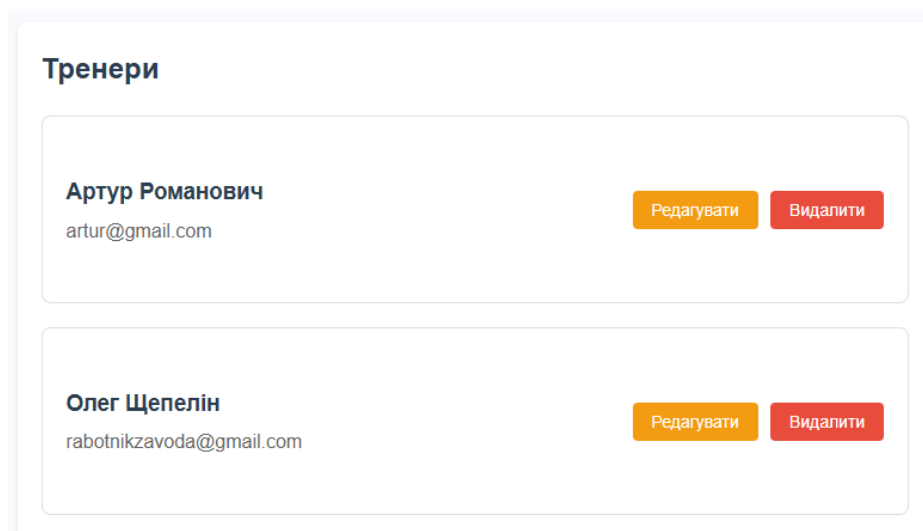
Прізвище

Email

Пароль

Додати

Рисунок 3.16 – Форма для додавання тренера



**Тренери**

**Артур Романович**  
artur@gmail.com

Редагувати Видалити

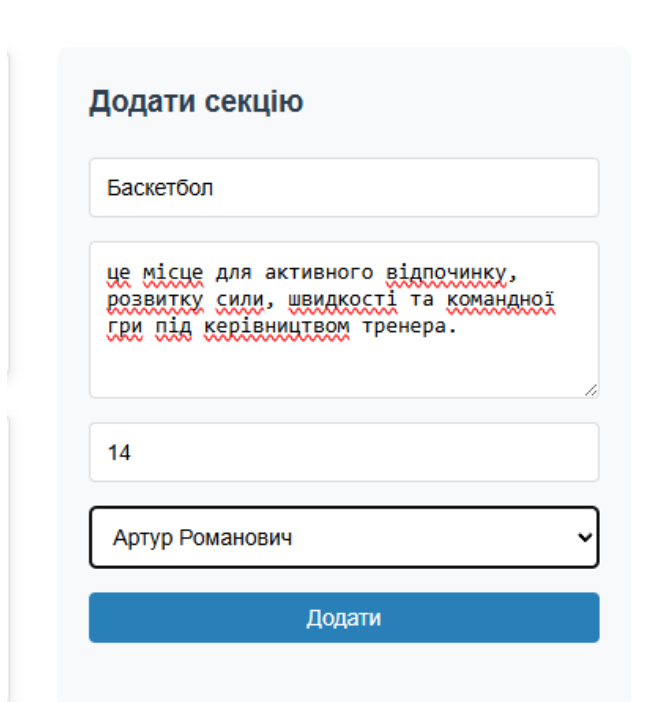
**Олег Щепелін**  
robotnikzavoda@gmail.com

Редагувати Видалити

Рисунок 3.17 – Редагування тренера

Було підтверджено, що при введенні неповних або некоректних даних система блокує дію і видає відповідне повідомлення. Збереження нових тренерів відбувається лише при наявності валідної інформації в усіх полях. Також перевірялося, що після редагування зміни з'являються у списку тренерів без потреби оновлення сторінки вручну.

Функціонал керування секціями також був протестований, зокрема додавання нових секцій, вибір тренера зі списку, а також відображення секцій на загальному списку (Рис. 3.18).



**Додати секцію**

Баскетбол

це місце для активного відпочинку, розвитку сили, швидкості та командної гри під керівництвом тренера.

14

Артур Романович

Додати

Рисунок 3.18 – Додавання секції

Було перевірено, що при спробі створити секцію без назви або опису виводиться повідомлення про помилку, а введення дублікату секції неможливе. Також перевірено функціональність редагування даних про секцію (Рис. 3.19), зокрема зміну назви, опису та прикріпленого тренера.

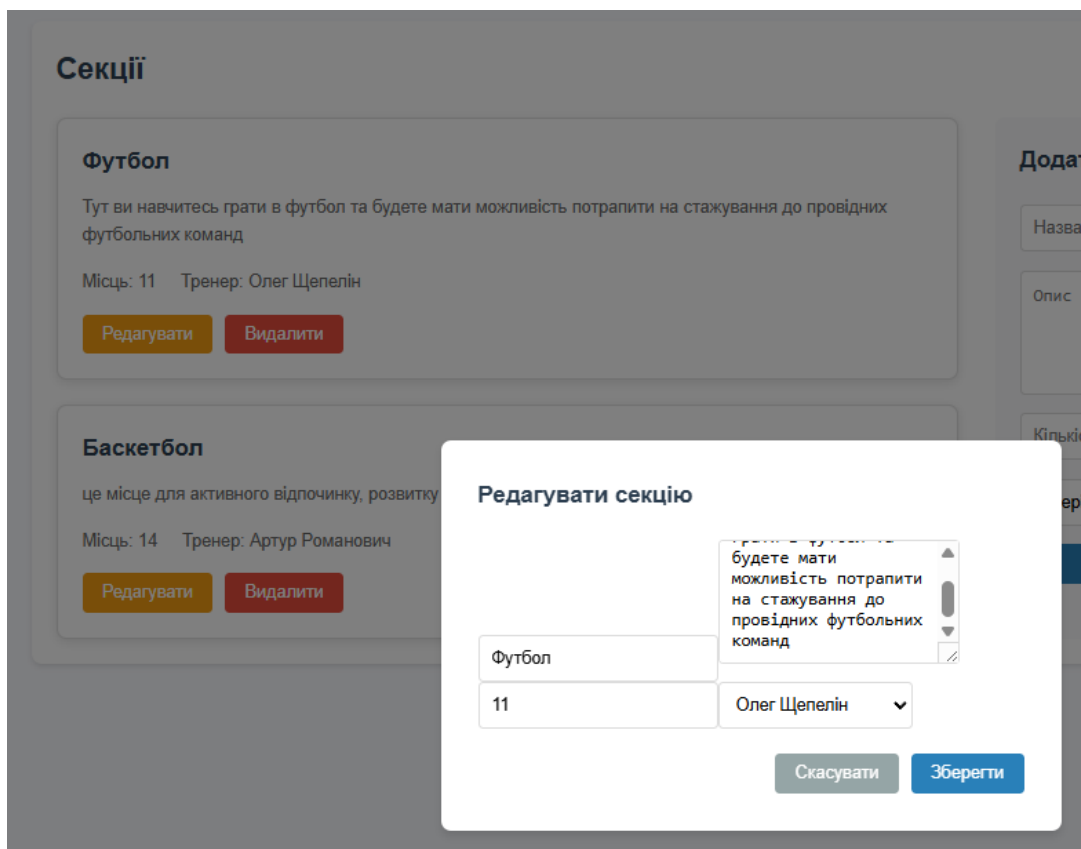


Рисунок 3.19 – Редагування секції

Окрема увага приділялась процесу реєстрації користувачів, де тестувались усі валідаційні механізми форми: перевірка наявності імені, пошти та унікальності логіна (Рис. 3.20).

Рисунок 3.20 – Реєстрація користувача

У випадках порушення вимог з'являлись відповідні повідомлення, що забезпечує захист від некоректного введення.

Функціонал запису на секції було протестовано як з точки зору користувача, так і з боку тренера. На рисунку 3.21 відображено інтерфейс, через який студент обирає секцію, а система відображає доступні варіанти відповідно до наявних тренерів та розкладу.

**Записатися на секцію**

**Баскетбол**  
це місце для активного відпочинку, розвитку сили, швидкості та командної гри під керівництвом тренера.  
Місце: 14  
**Записатися**

**Футбол**  
Тут ви навчитесь грати в футбол та будете мати можливість потрапити на стажування до провідних футбольних команд  
Місце: 11  
Ви вже записані

Рисунок 3.21 – Запис на секцію

Також протестовано інтерфейс додавання занять тренером (Рис. 3.22), де перевірялось, чи можна вказати дати, часи та обрати секцію.

**Додати заняття**

user user — Баскетбол

22.05.2025 15:37

Купити собі форму та м'яч

**Додати**

Рисунок 3.22 – Додавання заняття тренером

У разі відсутності необхідних даних або вибору неіснуючої секції, дія не виконується. Після збереження інформації дані коректно відображаються в особистому кабінеті користувача.

Завершальним етапом було тестування відображення розкладу у кабінеті студента та повідомлень від тренера (Рис. 3.23).

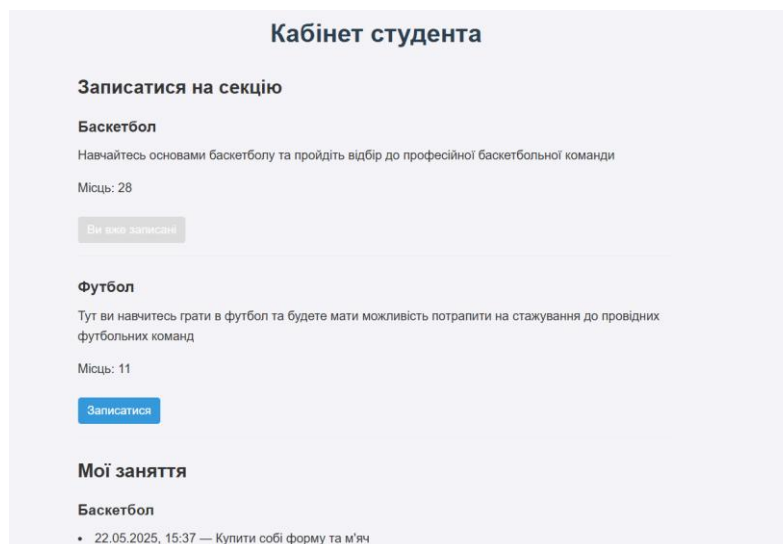


Рисунок 3.23 – Повідомлення від тренера та призначений час заняття

Важливою перевіркою було відображення тільки актуальних занять та індивідуальних повідомлень, а також відповідність часу й секції. Система успішно розділяє майбутні й минулі заняття, коректно виводить повідомлення, що гарантує належну комунікацію між студентом і тренером.

У підсумку, проведене тестування показало, що всі модулі веб-сайту функціонують згідно з очікуваннями, інтерфейси є зрозумілими та логічно побудованими, а система ефективно обробляє дані користувачів, тренерів і секцій. Жодних критичних збоїв або помилок виявлено не було, що дозволяє стверджувати про готовність системи до використання в реальному середовищі.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто поставленої мети – створено повноцінний, функціонально завершений та адаптивний веб-сайт для Спортивно-Культурного Комплексу «Софіївський». Було проведено всебічний аналіз предметної області, що дозволив виявити особливості діяльності комплексу, специфіку його цільової аудиторії, а також окреслити вимоги до інформаційної системи, яка покликана задовольнити потреби як відвідувачів, так і адміністративного персоналу. Проведене дослідження існуючих аналогічних рішень засвідчило, що більшість наявних веб-ресурсів у сфері спортивно-культурної діяльності або обмежені у функціональності, або не мають інтерактивних компонентів. Це обґрунтувало доцільність розробки сайту, який би поєднував інформативність, інтуїтивну навігацію, адаптивний дизайн і засоби інтерактивної взаємодії з користувачем.

На основі сформульованих функціональних і нефункціональних вимог було спроектовано архітектуру веб-сайту, яка включає багаторівневу систему доступу, інтерфейс з розділами про секції, новини, контактну інформацію, а також модулі для запису на заняття, зворотного зв'язку й адміністрування контенту. Розробка інтерфейсу здійснювалася з урахуванням адаптивності, що забезпечило коректне відображення контенту на мобільних пристроях. Впроваджено інтерактивні механізми, які дозволяють користувачам подавати заявки онлайн, переглядати розклад занять, отримувати повідомлення та взаємодіяти з тренерами, що значно підвищило зручність користування сайтом.

Технічна реалізація передбачала створення клієнтської частини на базі сучасних веб-технологій (HTML, CSS, JavaScript, React), а також серверної логіки з використанням Node.js, Express та MySQL для збереження структурованих даних. Архітектура реалізована відповідно до принципів REST, що забезпечило модульність, масштабованість та підтримку багаторольової взаємодії. Особливу увагу було приділено безпеці обробки даних, що включало автентифікацію, захист від типових веб-уразливостей та контроль доступу до панелі адміністратора.

У завершальному етапі було проведено тестування функціональних модулів, зокрема механізмів реєстрації, запису до секцій, відображення розкладу, взаємодії з повідомленнями та особистими примітками тренерів. Усі модулі працювали відповідно до очікувань, критичних помилок не виявлено. Веб-сайт продемонстрував стабільність роботи в реальному середовищі та готовність до використання з боку громади.

Таким чином, у результаті виконаної роботи було реалізовано повноцінну інформаційну систему цифрової взаємодії для Спортивно-Культурного Комплексу «Софіївський», що підвищує ефективність комунікації з відвідувачами, автоматизує частину адміністративних процесів і сприяє формуванню позитивного іміджу закладу у цифровому просторі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

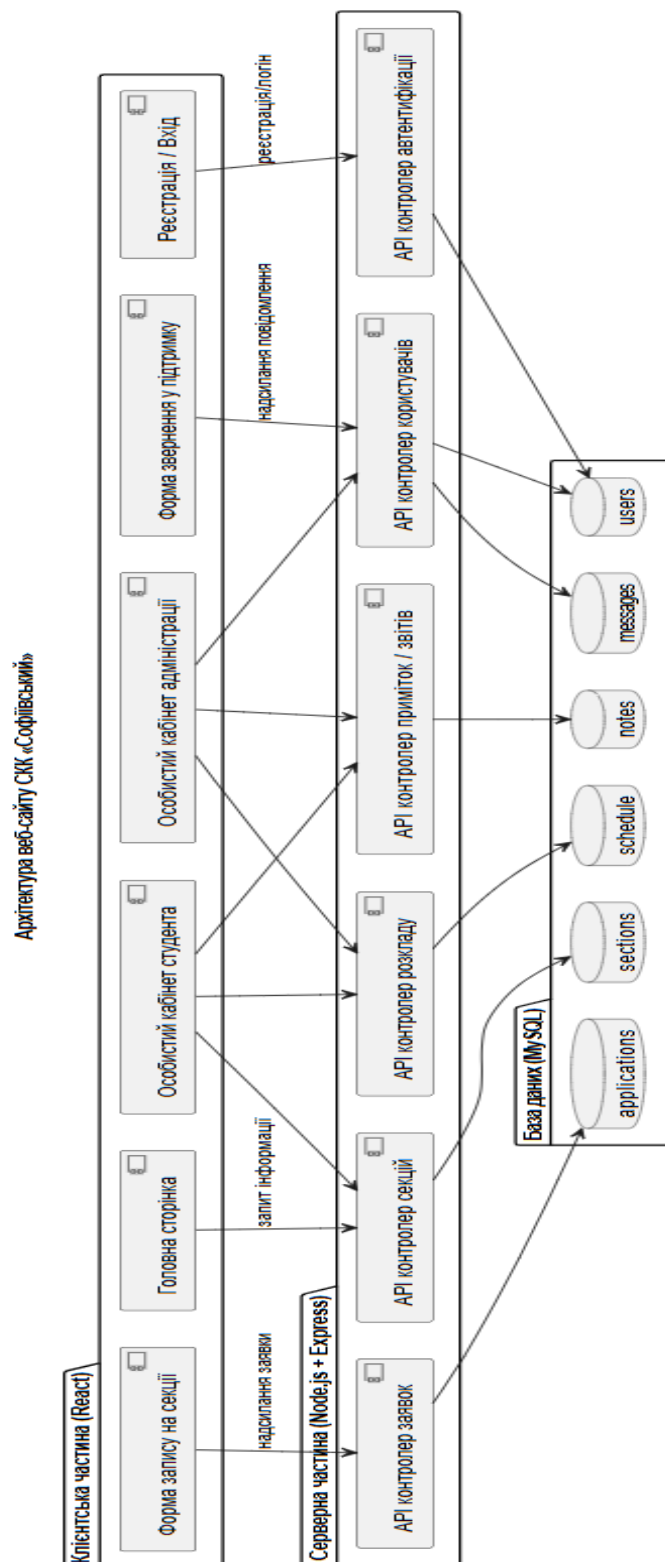
1. Борщагівська сільська рада. До уваги мешканців громади! Спортивно-культурний комплекс «Софіївський» запрошує діток у спортивні секції. URL: <https://brada.gov.ua/society/do-uvahy-meshkantsiv-hromady/> (дата звернення: 10.05.2025).
2. Борщагівська сільська рада. Рішення 7 сесії VIII скликання про реорганізацію комунального підприємства «Спортивно-культурний комплекс «Софіївський». URL: <https://brada.gov.ua/wp-content/uploads/2021/10/Про-реорганізацію-комунального-підприємства-СПОРТИВНО-КУЛЬТУРНИЙ-КОМПЛЕКС-СОФІЇВСЬКИЙ.pdf> (дата звернення: 10.05.2025).
3. Оpendатабот. КУ «СКК «Софіївський» – реєстраційні дані. URL: <https://opendatabot.ua/c/43864224> (дата звернення: 10.05.2025).
4. YouControl. КУ «СКК «Софіївський» – інформація про компанію. URL: [https://youcontrol.com.ua/catalog/company\\_details/43864224/](https://youcontrol.com.ua/catalog/company_details/43864224/) (дата звернення: 10.05.2025).
5. Управління освіти, культури, молоді та спорту Борщагівської сільської ради. Мережа закладів. URL: [https://bsr-vokmc.gov.ua/?page\\_id=34](https://bsr-vokmc.gov.ua/?page_id=34) (дата звернення: 10.05.2025).
6. Mixsport. Спортивно-культурний комплекс «Софіївський». URL: <https://mixsport.pro/places/sportivno-kulturnij-kompleks-sofiivskij> (дата звернення: 10.05.2025).
7. Facebook. Офіційна сторінка СКК «Софіївський». URL: <https://www.facebook.com/sccsofiivsky/> (дата звернення: 10.05.2025).
8. Спортивний комплекс НТУУ «КПІ». URL: <https://sport.kpi.ua/> (дата звернення: 10.05.2025).
9. Київський міський палац дитячої та юнацької творчості. URL: <https://palace.kiev.ua/> (дата звернення: 10.05.2025).
10. Sport Life – мережа фітнес-клубів. URL: <https://sportlife.ua/> (дата звернення: 10.05.2025).

11. Маслов М. Стандарт розробки React для початківців. DOU. URL: <https://dou.ua/forums/topic/44659/> (дата звернення: 15.05.2025).
12. DreamHost. Як вивчити React для початківців у 2024 році. URL: <https://www.dreamhost.com/blog/uk/vivchennia-react/> (дата звернення: 15.05.2025).
13. W3Schools українською. React Форми. URL: [https://w3schoolsua.github.io/react/react\\_forms.html](https://w3schoolsua.github.io/react/react_forms.html) (дата звернення: 15.05.2025).
14. DAN IT Education. Що таке React JS і для чого він потрібен? URL: <https://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen/> (дата звернення: 15.05.2025).
15. DreamHost. Що таке Node.js? Всебічне введення + Приклади використання. URL: <https://www.dreamhost.com/blog/uk/shcho-take-nodejs/> (дата звернення: 15.05.2025).
16. JavaScript.org.ua. Основи Node.js: Посібник для початківців. URL: <https://javascript.org.ua/osnovi-node-js-posibnik-dlya-pochatkivciv/> (дата звернення: 15.05.2025).
17. Guru99. Підручник Node.js для початківців: навчіться крок за кроком за 3 дні. URL: <https://www.guru99.com/uk/node-js-tutorial.html> (дата звернення: 15.05.2025).
18. FoxmindEd. Багатопоточність у Node.js: основи та приклади. URL: <https://foxminded.ua/bahatopotocnist-u-node-js/> (дата звернення: 15.05.2025).
19. CyberBionic Systematics. Як стати FullStack Node.js розробником. URL: <https://edu.cbsystematics.com/ua/blog/how-become-fullstack-nodejs> (дата звернення: 15.05.2025).
20. DOU. Співбесіда з Node.js розробником. 255 запитань для Junior, Middle і Senior. URL: <https://dou.ua/lenta/articles/interview-node-js/> (дата звернення: 15.05.2025).
21. Пасічник, В. В. Веб-технології : підручник / В. В. Пасічник, О. В. Пасічник, Д. І. Угрин. – Львів : Магнолія 2006, 2024. – 336 с. – (Комп'ютинг)

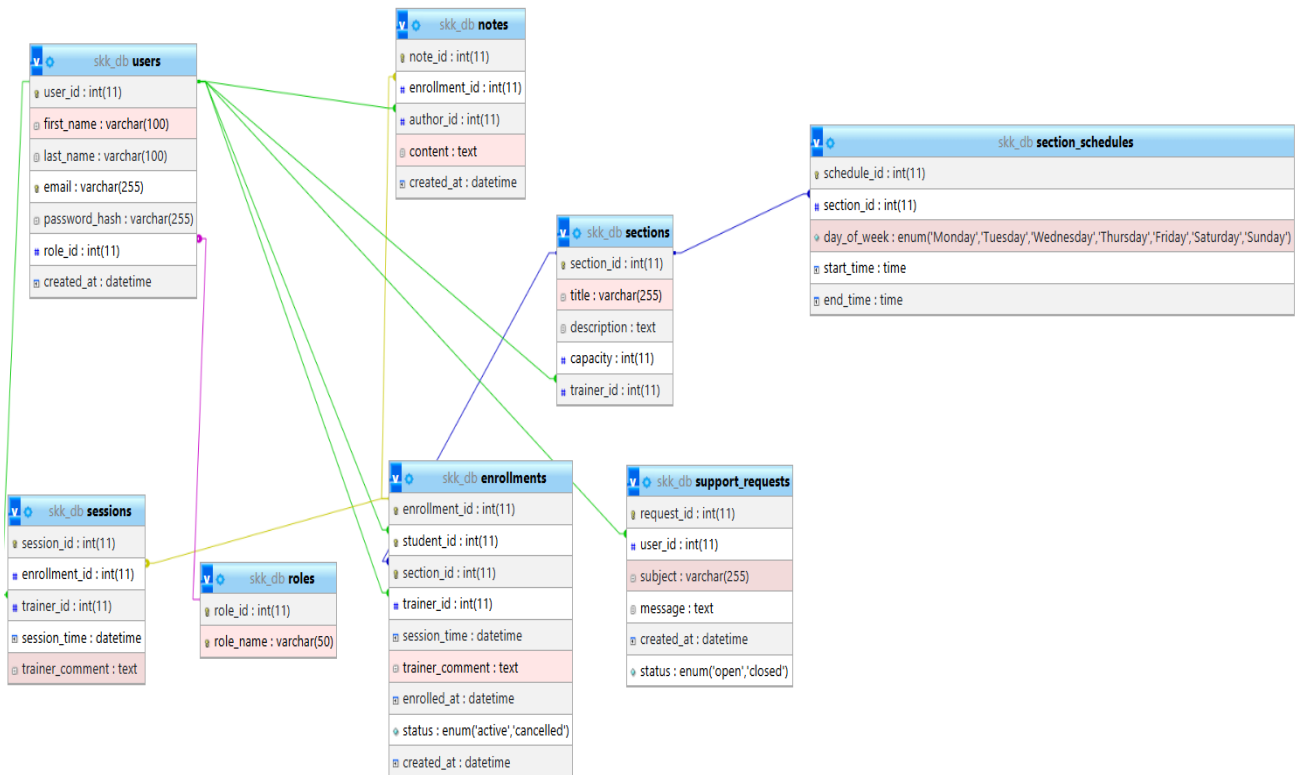
22. Пастернек, І. І. Інструментальні засоби веб-технологій : навч. посібник / І. І. Пастернек, А. Т. Костик ; Національний університет "Львівська політехніка". – Львів : Магнолія 2006, 2024. – 198 с.
23. Основи інформаційних технологій : навч. посібник / А. М. Гуржій, Л. І. Возненко, Н. І. Поворознюк, В. В. Самсонов. – Київ : Літера ЛТД, 2023. – 288 с.
24. Сьєрра, К. Head First. Java : пер. з англ. / К. Сьєрра, Б. Бейтс. – 2-ге вид. – Харків : Фабула, 2023. – 720 с.
25. Фрімен, Е. Head First. Програмування на JavaScript : пер. з англ. / Е. Фрімен, Е. Робсон. – Харків : Фабула, 2022. – 672 с.
26. Мартін, Р. Чистий код: створення і рефакторинг за допомогою Agile = Clean code: a handbook of Agile software craftsmanship : пер. з англ. / Р. Мартін. – Харків : Фабула, 2021. – 448 с.
27. Крамаренко С. П. "Проектування інформаційних систем". Харків: ХНУРЕ, 2020. 172 с.
28. Буйницька, О. П. Інформаційні технології та технічні засоби навчання : навч. посібник / О. П. Буйницька ; Київ. ун-т ім. Б. Грінченка. – Київ : ЦУЛ, 2022. – 240 с. УДК 004.38:37.013.42:159.9(075.8)
29. Кузьмичов, А. І. Аналітика мережевих структур. Моделювання засобами WinQSB та MS Excel : практикум / А. І. Кузьмичов ; Ін-т проблем реєстрації інформ. НАН України. – Київ : Ліра-К, 2021. – 208 с. УДК 004.4.
30. Теоретичні та практичні аспекти розвитку Інтернет-економіки : навч. посібник / О. О. Шультга, О. І. Вівчар, М. Ф. Зайлик [та ін.] ; за наук. ред. : І. Л. Татомир, Л. Г. Квасній ; Прикарпат. ін-т ім. М. Грушевського ПрАТ ВНЗ "МАУП". – Дрогобич-Трускавець : Просвіт, 2021. – 386 с. УДК 338.001.36:004.738.52

## ДОДАТКИ

Додаток А. Архітектурно-структурна модель веб-сайту СКК «Софіївський»



Додаток А.1 – Архітектура веб-сайту СКК “Софіївський”



Додаток А.2 – Схема бази даних

**Додаток Б. Лістинг програмного коду**

```
import React, { useState, useEffect } from 'react'; // ← додали useEffect
import {
  BrowserRouter as Router,
  Routes,
  Route,
  Link,
  Navigate,
  useNavigate
} from 'react-router-dom';
import axios from 'axios';
import { jwtDecode } from 'jwt-decode';

import Login      from './components/Login';
import Register   from './components/Register';
import StudentDashboard from './components/StudentDashboard';
import TrainerDashboard from './components/TrainerDashboard';
import AdminDashboard  from './components/AdminDashboard';

import './App.css';

// Встановлюємо базовий URL для axios
axios.defaults.baseURL = 'http://localhost:4000';

function NavBar() {
  const navigate = useNavigate();
  const token    = localStorage.getItem('token');
  let greeting   = null;
```

```

if (token) {
  try {
    const { role_id, first_name } = jwtDecode(token);
    if (role_id === 3) {
      greeting = 'Вітаю, адміністраторе!';
    } else if (role_id === 2) {
      greeting = `Вітаю, тренере!`;
    } else if (role_id === 1) {
      greeting = `Вітаю, студенте!`;
    }
  } catch {
    localStorage.removeItem('token');
  }
}

```

```

const handleLogout = () => {
  localStorage.removeItem('token');
  navigate('/');
  window.location.reload();
};

```

```

return (
  <nav className="navbar">
    <div className="navbar-left">
      <Link to="/">СКК Софіївський</Link>
    </div>
    <div className="navbar-right">
      {!token ? (

```

```

    </>
    <Link to="/login">Увійти</Link>
    <Link to="/register">Реєстрація</Link>
  </>
): (
  </>
  <span className="greeting">{greeting}</span>
  <button className="btn-logout" onClick={handleLogout}>
    Вихід
  </button>
</>
)}
</div>
</nav>
);
}

```

```

function Homepage() {
  const images = [
    '/images/football.jpg',
    '/images/karate.jpg',
    '/images/team.jpg',
    '/images/football.jpg',
    '/images/karate.jpg',
    '/images/team.jpg'
  ];

  const [phone, setPhone] = useState("");
  const [message, setMessage] = useState("");

```

```

const handleSupport = async e => {
  e.preventDefault();
  try {
    await axios.post('/api/support', { phone, message });
    alert('Звернення надіслано');
    setPhone("");
    setMessage("");
  } catch (err) {
    console.error(err);
    alert('Помилка надсилання звернення');
  }
};

return (
  <div className="home">
    <h1>Ласкаво просимо до СКК “Софіївський”</h1>

    <div className="carousel">
      <div className="carousel-track">
        {images.map((src, idx) => (
          <div className="carousel-slide" key={idx}>
            <img src={src} alt={`slide-${idx}` } />
          </div>
        ))}
      </div>
    </div>

    <div>
      <form className="support-form" onSubmit={handleSupport}>

```

```

<h2>Звернення до адміністрації</h2>
<input
  type="text"
  placeholder="Ваш телефон"
  value={phone}
  onChange={e => setPhone(e.target.value)}
  required
/>
<textarea
  placeholder="Текст звернення"
  value={message}
  onChange={e => setMessage(e.target.value)}
  required
/>
<button type="submit">Надіслати</button>
</form>
</div>
);
}

export default function App() {
  // При першому запуску підставляємо токен у заголовки
  useEffect(() => {
    const token = localStorage.getItem('token');
    if (token) {
      axios.defaults.headers.common['Authorization'] = `Bearer ${token}`;
    }
  }, []);
}

```

```

return (
  <Router>
    <NavBar />
    <main className="main-content">
      <Routes>
        <Route path="/" element={<Homepage />} />
        <Route path="/login" element={<Login />} />
        <Route path="/register" element={<Register />} />
        <Route path="/student" element={<StudentDashboard />} />
        <Route path="/trainer" element={<TrainerDashboard />} />
        <Route path="/admin" element={<AdminDashboard />} />
        <Route path="*" element={<Navigate to="/" replace />} />
      </Routes>
    </main>
  </Router>
);
} import React, { useEffect, useState } from 'react';
import axios from 'axios';
import dayjs from 'dayjs';
import './StudentDashboard.css';

export default function StudentDashboard() {
  const [sections, setSections] = useState([]);
  const [sessions, setSessions] = useState([]);
  const [error, setError] = useState(null);

  useEffect(() => {
    fetchSections();
    fetchSessions();
  });

```

```
}, []);
```

```
// 1) Список секцій + статус
```

```
async function fetchSections() {  
  try {  
    const { data } = await axios.get('/api/student/enrollments', {  
      headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }  
    });  
    setSections(data);  
  } catch (err) {  
    console.error('fetchSections error', err);  
    setError('Не вдалося завантажити секції');  
  }  
}
```

```
// 2) Ваші заняття
```

```
async function fetchSessions() {  
  try {  
    const { data } = await axios.get('/api/student/sessions', {  
      headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }  
    });  
    setSessions(data);  
  } catch (err) {  
    console.error('fetchSessions error', err);  
    setError('Не вдалося завантажити заняття');  
  }  
}
```

```
// 3) Записатися на секцію
```

```

const handleEnroll = async sectionId => {
  try {
    await axios.post(
      '/api/student/enroll',
      { section_id: sectionId },
      { headers: { Authorization: `Bearer ${localStorage.getItem('token')}` } }
    );
    fetchSections(); // оновлюємо статус
  } catch (err) {
    console.error('enroll error', err);
    setError(err.response?.data?.message || 'Не вдалося записатися');
  }
};

```

```

return (
  <div className="student-dashboard">
    <h1>Кабінет студента</h1>
    {error && <div className="alert error">{error}</div>}
    <section className="enroll-section">
      <h2>Записатися на секцію</h2>
      {sections.length === 0 ? (
        <p>Секції не знайдені.</p>
      ) : (
        <ul className="section-list">
          {sections.map(s => (
            <li key={s.section_id} className="section-item">
              <h3>{s.title}</h3>
              <p>{s.description}</p>
            )}
          )}
        </ul>
      )}
    </section>
  </div>
)

```

```

<p>Місце: {s.capacity}</p>
{s.status === 'open' ? (
  <button
    className="btn btn-primary"
    onClick={() => handleEnroll(s.section_id)}
  >
    Записатися
  </button>
) : (
  <button className="btn btn-disabled" disabled>
    Ви вже записані
  </button>
)}
</li>
)}}
</ul>
)}
</section>

<section className="my-sessions">
  <h2>Мої заняття</h2>
  {sessions.length === 0 ? (
    <p>У вас ще немає призначених занять.</p>
  ) : (
    // Групуємо сесії за секцією
    [...new Map(
      sessions.map(s => [s.section_id, s])
    ).values()].map(group => {
      const same = sessions.filter(s => s.section_id === group.section_id);

```

```

return (
  <div key={group.section_id} className="sessions-card">
    <h3>{group.section_title}</h3>
    <ul>
      {same.map(s => (
        <li key={s.session_id}>
          {dayjs(s.session_time).format('DD.MM.YYYY, HH:mm')}
          {s.trainer_comment && ` — ${s.trainer_comment}`}
        </li>
      ))}
    </ul>
  </div>
);
})
}
</section>
</div>
);
} import React, { useState, useEffect } from 'react';
import axios from 'axios';
import './AdminDashboard.css';

export default function AdminDashboard() {
  const [support, setSupport] = useState([]);
  const [report7, setReport7] = useState({ users_count: 0, enroll_count: 0 });
  const [report30, setReport30] = useState({ users_count: 0, enroll_count: 0 });
  const [trainers, setTrainers] = useState([]);
  const [sections, setSections] = useState([]);

```

```

const [newTrainer, setNewTrainer] = useState({ first_name:"", last_name:"", email:"",
password:" });

const [editTrainer, setEditTrainer] = useState(null);

const [newSection, setNewSection] = useState({ title:"", description:"", capacity:"",
trainer_id:" });

const [editSection, setEditSection] = useState(null);

const [msg, setMsg]          = useState(null);
const [err, setErr]          = useState(null);

useEffect(() => {
  const token = localStorage.getItem('token');
  if (token) axios.defaults.headers.common['Authorization'] = `Bearer ${token}`;
  fetchSupport(); fetchReport(7,setReport7); fetchReport(30,setReport30);
  fetchTrainers(); fetchSections();
}, []);

// Звернення
const fetchSupport = async () => {
  try {
    const { data } = await axios.get('/api/admin/support');
    setSupport(data);
  } catch {
    setErr('Не вдалося завантажити звернення');
  }
};

const closeSupport = async id => {
  try {
    await axios.post('/api/admin/support/respond',{ request_id:id });
    setMsg('Звернення закрито');
  }
};

```

```

    fetchSupport();
  } catch {
    setErr('Не вдалося закрити звернення');
  }
};

// Звіти
const fetchReport = async (days, setter) => {
  try {
    const { data } = await axios.get(`/api/admin/report?days=${days}`);
    const row = Array.isArray(data)? data[0]||{}: data;
    setter({ users_count: row.users_count||0, enroll_count: row.enroll_count||0 });
  } catch {
    setErr(`Не вдалося завантажити звіт за ${days} днів`);
  }
};

// Тренери CRUD
const fetchTrainers = async () => {
  try {
    const { data } = await axios.get('/api/admin/trainers');
    setTrainers(data);
  } catch {
    setErr('Не вдалося завантажити тренерів');
  }
};

const addTrainer = async e => {
  e.preventDefault(); setErr(null); setMsg(null);
  try {

```

```

    await axios.post('/api/admin/trainers',newTrainer);

    setMsg('Тренера додано'); setNewTrainer({ first_name:", last_name:", email:",
password:" });

    fetchTrainers();

  } catch {

    setErr('Не вдалося додати тренера');

  }

};

const startEditTrainer = t => setEditTrainer({ ...t });
const saveEditTrainer = async () => {

  try {

    await axios.put(`/api/admin/trainers/${editTrainer.user_id}`,editTrainer);

    setMsg('Тренера оновлено'); setEditTrainer(null); fetchTrainers();

  } catch {

    setErr('Не вдалося оновити тренера');

  }

};

const deleteTrainer = async id => {

  try {

    await axios.delete(`/api/admin/trainers/${id}`);

    setMsg('Тренера видалено'); fetchTrainers();

  } catch {

    setErr('Не вдалося видалити тренера');

  }

};

// Секції CRUD (з JOIN на trainer_name)
const fetchSections = async () => {

  try {

```

```

const { data } = await axios.get('/api/admin/sections');
setSections(data);
} catch {
  setErr('Не вдалося завантажити секції');
}
};

const addSection = async e => {
  e.preventDefault(); setErr(null); setMsg(null);
  try {
    await axios.post('/api/admin/sections', {
      ...newSection,
      capacity: Number(newSection.capacity) });
    setMsg('Секцію додано');
    setNewSection({ title: "", description: "", capacity: "", trainer_id: "" });
    fetchSections();
  } catch {
    setErr('Не вдалося додати секцію');
  }
};

const startEditSection = s => setEditSection({ ...s });
const saveEditSection = async () => {
  try {
    const { section_id, title, description, capacity, trainer_id } = editSection;
    await axios.put(`/api/admin/sections/${section_id}`, {
      title, description,
      capacity: Number(capacity) });
    if (trainer_id) await axios.post('/api/admin/assign-trainer', { section_id, trainer_id });
    setMsg('Секцію оновлено'); setEditSection(null); fetchSections();
  } catch {
    setErr('Не вдалося оновити секцію');
  }
}

```

```

};

const deleteSection = async id => {
  try {
    await axios.delete(`/api/admin/sections/${id}`);
    setMsg('Секцію видалено'); fetchSections();
  } catch {
    setErr('Не вдалося видалити секцію');
  }
};

return (
  <div className="admin-dashboard">
    <h1>Адмін панель</h1>
    {msg && <div className="alert success">{msg}</div>}
    {err && <div className="alert error">{err}</div>}

    <div className="cards">

      {/* Звернення */}
      <div className="card">
        <h2>Звернення</h2>
        {support.length===0 && <p>Немає звернень</p>}
        {support.map(r=>(
          <div key={r.request_id} className="support-item">
            <p><strong>ID:</strong> {r.request_id}</p>
            <p><strong>Телефон:</strong> {r.phone||r.subject||'-'}</p>
            <p><strong>Повідомлення:</strong> {r.text||r.message||'-'}</p>
            <p><strong>Статус:</strong> {r.status}</p>
            {r.status==='open' && (

```

```

    <button onClick={()=>closeSupport(r.request_id)}>Закрити</button>
  )}
  <hr/>
</div>
)}}
</div>

{/* Звіти */}
<div className="card">
  <h2>Звіт за 7 днів</h2>
  <p>Користувачів: {report7.users_count}</p>
  <p>Записів: {report7.enroll_count}</p>
</div>
<div className="card">
  <h2>Звіт за 30 днів</h2>
  <p>Користувачів: {report30.users_count}</p>
  <p>Записів: {report30.enroll_count}</p>
</div>

{/* Тренери */}
<div className="card">
  <h2>Тренери</h2>
  <ul className="list">
    {trainers.map(t=>(
      <li key={t.user_id}>
        {editTrainer?.user_id===t.user_id ? (
          <input
            value={editTrainer.first_name}
            onChange={e=>setEditTrainer(f=>({...f,first_name:e.target.value}))}>

```

```

        <input                                value={editTrainer.last_name}
onChange={e=>setEditTrainer(f=>({...f,last_name:e.target.value}))}/>
        <input                                value={editTrainer.email}
onChange={e=>setEditTrainer(f=>({...f,email:e.target.value}))}/>
        <button onClick={saveEditTrainer}>Зберегти</button>
        <button onClick={()=>setEditTrainer(null)}>Скасувати</button>
    </>
  ): (
    <
      {t.first_name} {t.last_name} ({t.email})
      <button onClick={()=>startEditTrainer(t)}>Редагувати</button>
      <button onClick={()=>deleteTrainer(t.user_id)}>Видалити</button>
    </>
  )}
</li>
)}}
</ul>
<form onSubmit={addTrainer} className="form-inline">
  <h3>Додати тренера</h3>
  <input          placeholder="Ім'я"          value={newTrainer.first_name}
onChange={e=>setNewTrainer(f=>({...f,first_name:e.target.value}))} required/>
  <input          placeholder="Прізвище"      value={newTrainer.last_name}
onChange={e=>setNewTrainer(f=>({...f,last_name:e.target.value}))} required/>
  <input type="email" placeholder="Email" value={newTrainer.email}
onChange={e=>setNewTrainer(f=>({...f,email:e.target.value}))} required/>
  <input type="password" placeholder="Пароль" value={newTrainer.password}
onChange={e=>setNewTrainer(f=>({...f,password:e.target.value}))} required/>
  <button type="submit">Додати</button>
</form>
</div>

```

```

{ /* Секції */ }
<div className="card">
  <h2>Секції</h2>
  {sections.map(s=>(
    <div key={s.section_id} className="section-item">
      {editSection?.section_id===s.section_id ? (
        ◇
        <input                                value={editSection.title}
onChange={e=>setEditSection(z=>({...z,title:e.target.value}))}/>
        <input                                value={editSection.description}
onChange={e=>setEditSection(z=>({...z,description:e.target.value}))}/>
        <input type="number" min="1" value={editSection.capacity}
onChange={e=>setEditSection(z=>({...z,capacity:e.target.value}))}/>
        <label>
          Тренер:
          <select                                value={editSection.trainer_id||""}
onChange={e=>setEditSection(z=>({...z,trainer_id:e.target.value}))}>
            <option value="">– без тренера –</option>
            {trainers.map(t=><option                                key={t.user_id}
value={t.user_id}>{t.first_name} {t.last_name}</option>)}
          </select>
        </label>
        <button onClick={saveEditSection}>Зберегти</button>
        <button onClick={()=>setEditSection(null)}>Скасувати</button>
      </>
    ) : (
      ◇
      <p><strong>{s.title}</strong> ({s.capacity} місць)</p>
      <p>{s.description}</p>

```

```

    <p><strong>Тренер:</strong> {s.trainer_name||'-'}</p>
    <button onClick={()=>startEditSection(s)}>Редагувати</button>
    <button onClick={()=>deleteSection(s.section_id)}>Видалити</button>
  </>
)}
<hr/>
</div>
)}}
{!editSection && (
  <form onSubmit={addSection} className="form-inline">
    <h3>Додати секцію</h3>
    <input
      placeholder="Назва"
      value={newSection.title}
      onChange={e=>setNewSection(z=>({...z,title:e.target.value}))}
      required/>
    <input
      placeholder="Опис"
      value={newSection.description}
      onChange={e=>setNewSection(z=>({...z,description:e.target.value}))}
      required/>
    <input
      type="number"
      min="1"
      placeholder="Вмістимість"
      value={newSection.capacity}
      onChange={e=>setNewSection(z=>({...z,capacity:e.target.value}))}
      required/>
    <label>
      Тренер:
      <select
        value={newSection.trainer_id}
        onChange={e=>setNewSection(z=>({...z,trainer_id:e.target.value}))}>
        <option value="">- без тренера -</option>
        {trainers.map(t=><option key={t.user_id} value={t.user_id}>{t.first_name}
        {t.last_name}</option>)}
      </select>
    </label>
    <button type="submit">Додати</button>
  </form>
)}}

```

```

    </div>
  </div>
</div>
);
} // client/src/components/TrainerDashboard.js
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import dayjs from 'dayjs';
import './TrainerDashboard.css';

export default function TrainerDashboard() {
  const [enrollments, setEnrollments] = useState([]);
  const [sessions, setSessions] = useState([]);
  const [error, setError] = useState(null);
  const [form, setForm] = useState({
    enrollment_id: "",
    session_time: "",
    comment: ""
  });

  useEffect(() => {
    fetchEnrollments();
    fetchSessions();
  }, []);

  async function fetchEnrollments() {
    try {
      const { data } = await axios.get('/api/trainer/enrollments', {
        headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }
      }

```

```

    });
    setEnrollments(data);
  } catch (err) {
    console.error('fetchEnrollments error', err);
    setError('Не вдалося завантажити записи');
  }
}

```

```

async function fetchSessions() {
  try {
    const { data } = await axios.get('/api/trainer/sessions', {
      headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }
    });
    setSessions(data);
  } catch (err) {
    console.error('fetchSessions error', err);
    setError('Не вдалося завантажити сесії');
  }
}

```

```

const handleChange = e => {
  setForm(prev => ({ ...prev, [e.target.name]: e.target.value }));
};

```

```

const handleAdd = async e => {
  e.preventDefault();
  try {
    await axios.post('/api/trainer/sessions', form, {
      headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }
    }
  }
}

```

```

});
setForm({ enrollment_id: "", session_time: "", comment: "" });
fetchSessions();
} catch (err) {
  console.error('addSession error', err);
  setError('Не вдалося додати сесію');
}
};

```

```

const handleUpdate = async session => {
  if (!session.session_id) return;
  try {
    await axios.put(
      `/api/trainer/sessions/${session.session_id}`,
      { session_time: session.session_time, comment: session.trainer_comment },
      { headers: { Authorization: `Bearer ${localStorage.getItem('token')}` } }
    );
    fetchSessions();
  } catch (err) {
    console.error('updateSession error', err);
    setError('Не вдалося оновити сесію');
  }
};

```

```

const handleDelete = async id => {
  if (!id || !window.confirm('Скасувати це заняття?')) return;
  try {
    await axios.delete(`/api/trainer/sessions/${id}`, {
      headers: { Authorization: `Bearer ${localStorage.getItem('token')}` }
    }

```

```

});
fetchSessions();
} catch (err) {
  console.error('deleteSession error', err);
  setError('Не вдалося скасувати сесію');
}
};

return (
  <div className="trainer-dashboard">
    <h1>Панель тренера</h1>
    {error && <div className="alert error">{error}</div>}

    <form className="session-form" onSubmit={handleAdd}>
      <h2>Додати заняття</h2>
      <select
        name="enrollment_id"
        value={form.enrollment_id}
        onChange={handleChange}
        required
      >
        <option value="">Оберіть запис студента</option>
        {enrollments.map(e => (
          <option key={e.enrollment_id} value={e.enrollment_id}>
            {e.student_name} — {e.section_title}
          </option>
        ))}
      </select>
      <input

```

```

type="datetime-local"
name="session_time"
value={form.session_time}
onChange={handleChange}
required
/>
<textarea
name="comment"
placeholder="Коментар тренера"
value={form.comment}
onChange={handleChange}
/>
<button type="submit" className="btn btn-primary">Додати</button>
</form>

<table className="sessions-table">
<thead>
<tr>
<th>Секція</th>
<th>Студент</th>
<th>Час</th>
<th>Коментар</th>
<th>Дії</th>
</tr>
</thead>
<tbody>
{sessions.length === 0 ? (
<tr><td colspan="5" className="no-data">Немає призначених
занять</td></tr>

```

```

):(
  sessions.map(sess => (
    <tr key={sess.session_id}>
      <td>{sess.section_title}</td>
      <td>{sess.student_name}</td>
      <td>
        <input
          type="datetime-local"
          value={dayjs(sess.session_time).format('YYYY-MM-DDTHH:mm')}
          onChange={e => {
            sess.session_time = e.target.value;
            setSessions([...sessions]);
          }}
        />
      </td>
      <td>
        <textarea
          value={sess.trainer_comment || ""}
          onChange={e => {
            sess.trainer_comment = e.target.value;
            setSessions([...sessions]);
          }}
        />
      </td>
      <td>
        <button className="btn btn-success" onClick={() =>
handleUpdate(sess)}>ЗМІНИТИ</button>
        <button className="btn btn-danger" onClick={() =>
handleDelete(sess.session_id)}>Скасувати</button>

```

```

        </td>
    </tr>
    ))
    })
</tbody>
</table>
</div>
);
} /* Повідомлення успіху та помилки */
.alert {
    padding: 0.75rem 1rem;
    margin-bottom: 1rem;
    border-radius: var(--border-radius);
    font-size: 0.95rem;
}
.alert.success {
    background-color: #e6f4ea;
    color: #216e37;
    border: 1px solid #a3d9a5;
}
.alert.error {
    background-color: #fdecea;
    color: #941d20;
    border: 1px solid #f5c2c0;
}
/* =====
Змінні теми
===== */
:root {

```

```
--primary: #1e1e2f;
--secondary: #4e4e6a;
--accent: #ff6f61;
--bg: #f4f4f8;
--text: #333333;
--white: #ffffff;
--border-radius: 8px;
}

/* =====
   Скидання та базові стилі
   ===== */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
body {
  font-family: 'Roboto', sans-serif;
  background-color: var(--bg);
  color: var(--text);
  line-height: 1.6;
}

/* =====
   Хедер та навігація
   ===== */
.header {
  display: flex;
```

```
justify-content: space-between;
align-items: center;
padding: 1rem 2rem;
background-color: var(--primary);
color: var(--white);
position: sticky;
top: 0;
z-index: 1000;
}
.header a {
  color: var(--white);
  text-decoration: none;
  font-weight: 500;
  transition: opacity .3s;
}
.header a:hover {
  opacity: .8;
}
.header nav {
  display: flex;
  gap: 1rem;
}

/* =====
  Головна сторінка
  ===== */
.homepage {
  padding: 2rem 1rem;
  text-align: center;
```

```
}  
.homepage h1 {  
  font-size: 2.25rem;  
  margin-bottom: 1.5rem;  
  color: var(--primary);  
}  
  
/* =====  
  Карусель із фото  
===== */  
.gallery {  
  overflow: hidden;  
  margin: 0 auto 2rem;  
  max-width: 800px;  
  border-radius: var(--border-radius);  
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
}  
.carousel-track {  
  display: flex;  
  width: 200%;  
  animation: scroll 20s linear infinite;  
}  
.carousel-track img {  
  flex: 0 0 auto;  
  width: 33.333%;  
  height: auto;  
  object-fit: cover;  
}  
@keyframes scroll {
```

```

0% { transform: translateX(0); }
100% { transform: translateX(-50%); }
}

/* =====
   Форми (логія, реєстрація, звернення)
   ===== */

.form,
.form.support-form {
  background-color: var(--white);
  padding: 2rem;
  max-width: 450px;
  margin: 0 auto 2rem;
  border-radius: var(--border-radius);
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}
.form h2,
.form.support-form h3 {
  margin-bottom: 1rem;
  color: var(--primary);
}
.form input,
.form select,
.form textarea,
.form button,
.form.support-form input,
.form.support-form textarea,
.form.support-form button {
  width: 100%;

```

```

padding: .75rem 1rem;
margin-bottom: 1rem;
border: 1px solid var(--secondary);
border-radius: var(--border-radius);
font-size: 1rem;
}
.form button,
.form.support-form button {
background-color: var(--accent);
border: none;
color: var(--white);
cursor: pointer;
font-weight: 600;
transition: background .3s;
}
.form button:hover,
.form.support-form button:hover {
background-color: #e65a52;
}

/* =====
Кабінети (Dashboard)
===== */
.dashboard {
padding: 2rem 1rem;
display: grid;
gap: 1.5rem;
}
.dashboard h2 {

```

```

color: var(--primary);
margin-bottom: 1rem;
}
.dashboard > div {
background-color: var(--white);
padding: 1.5rem;
border-radius: var(--border-radius);
box-shadow: 0 2px 4px rgba(0,0,0,0.05);
}
@media (min-width: 768px) {
.dashboard {
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
}
}

/* =====
Адаптивні налаштування
===== */
@media (max-width: 600px) {
.header {
flex-direction: column;
gap: .5rem;
}
.carousel-track img {
width: 100%;
}
.gallery {
max-width: 100%;
}
}

```

```
}  
/* _____ Загальні стилі для NavBar _____ */  
.navbar {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  background: #2c3e50;  
  color: #ecf0f1;  
  padding: 0.75rem 1.5rem;  
}  
.navbar a {  
  color: #ecf0f1;  
  text-decoration: none;  
  margin-right: 1rem;  
  font-weight: 500;  
}  
.navbar-right a:last-child {  
  margin-right: 0;  
}  
.greeting {  
  margin-right: 0.75rem;  
  font-size: 0.95rem;  
}  
.btn-logout {  
  background: transparent;  
  border: 1px solid #ecf0f1;  
  color: #ecf0f1;  
  padding: 0.4rem 0.8rem;  
  border-radius: 4px;
```

```
    cursor: pointer;
    font-size: 0.9rem;
}
.btn-logout:hover {
    background: rgba(236, 240, 241, 0.1);
}

/* ————— Стилі для головної сторінки і каруселі ————— */
.home {
    text-align: center;
    padding: 2rem 1rem;
}

.carousel {
    position: relative;
    overflow: hidden;
    margin: 2rem auto;
    max-width: 800px;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}

.carousel-track {
    display: flex;
    width: calc(6 * 100%);
    animation: scroll 20s linear infinite;
}

.carousel-slide {
```

```
flex: 0 0 calc(100% / 3);
}

.carousel-slide img {
width: 100%;
display: block;
object-fit: cover;
}

/* Анімація нескінченного скролу */
@keyframes scroll {
0% { transform: translateX(0%); }
100% { transform: translateX(-50%); }
}

/* ===== NavBar ===== */
.navbar {
display: flex;
justify-content: space-between;
align-items: center;
background: #2c3e50;
padding: 0.75rem 1.5rem;
color: #ecf0f1;
}

.navbar a {
color: #ecf0f1;
text-decoration: none;
margin-right: 1rem;
font-weight: 500;
}
```

```
.navbar-right a:last-child {
  margin-right: 0;
}
.greeting {
  margin-right: 1rem;
  font-size: 0.95rem;
}
.btn-logout {
  background: transparent;
  border: 1px solid #ecf0f1;
  color: #ecf0f1;
  padding: 0.4rem 0.8rem;
  border-radius: 4px;
  cursor: pointer;
}
.btn-logout:hover {
  background: rgba(236,240,241,0.1);
}

/* ===== Homepage & Carousel ===== */
.home {
  text-align: center;
  padding: 2rem 1rem;
}
.carousel {
  overflow: hidden;
  margin: 2rem auto;
  max-width: 800px;
  border-radius: 8px;
```

```
    box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}
.carousel-track {
    display: flex;
    width: calc(6 * 100%);
    animation: scroll 20s linear infinite;
}
.carousel-slide {
    flex: 0 0 calc(100%/3);
}
.carousel-slide img {
    width: 100%;
    display: block;
    object-fit: cover;
}
@keyframes scroll {
    0% { transform: translateX(0%); }
    100% { transform: translateX(-50%); }
}

/* ===== Support Form ===== */
.support-form {
    background: #fff;
    max-width: 500px;
    margin: 2rem auto;
    padding: 1.5rem;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0,0,0,0.1);
    display: flex;
```

```
flex-direction: column;
gap: 0.75rem;
}
.support-form h2 {
margin: 0 0 0.5rem 0;
}
.support-form input,
.support-form textarea {
padding: 0.5rem;
border: 1px solid #ccc;
border-radius: 4px;
font-size: 1rem;
width: 100%;
box-sizing: border-box;
}
.support-form button {
background: #3498db;
color: #fff;
border: none;
padding: 0.75rem;
border-radius: 4px;
cursor: pointer;
font-size: 1rem;
}
.support-form button:hover {
background: #217dbb;
}
```