

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Допущено до захисту»

Завідувач кафедри
комп'ютерних наук
доктор технічних наук,
професор

_____ Андрій БОНДАРЧУК
(підпис)

« ____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня «Магістр»
Спеціальність 122 Комп'ютерні науки
Освітня програма 122.00.02 Інформаційно-аналітичні системи

**Тема роботи: Персоналізована платформа для управління
кар'єрою спортсмена**

Виконав

студент групи
ІАСм-1-24-1.4д

_____ Писарев В.Д.

Науковий керівник

кандидат технічних наук,
доцент

_____ Машкіна І.В.

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри комп'ютерних наук
кандидат технічних наук, доцент

_____ І.В.Машкіна

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

**на тему: Персоналізована платформа для управління кар'єрою
спортсмена**

**Виконавець: студент групи ІАСм-1-24-1.4д Писарєв Владислав
Дмитрович**

1. Вихідні дані: функціональний прототип вебзастосунку для управління кар'єрою спортсмена
2. Основні завдання: дослідити потреб спортсменів, тренерів і спортивних організацій щодо управління кар'єрою, проаналізувати існуючі платформи та визначити ключові вимоги до функціоналу персоналізованої системи. Спроекувати архітектуру платформи, обрати технологічний стек, створити модель даних та розробити зручні користувацькі інтерфейси. Розробити платформу з модулями планування тренувань, відстеження прогресу, управління кар'єрою, аналітики та персоналізації.
3. Пояснювальна записка: *Обсяг – до 66 сторінок формату А4 комп'ютерного набору з дотриманням вимог стандарту та методичних рекомендацій кафедри.*
4. Графічні матеріали: *макети інтерфейсу.*
5. Додатки: *не передбачено.*
6. Строк подання роботи на кафедру: *«1» грудня 2025 р.*

Науковий керівник

Виконавець:

к.т.н., доцент

_____ Машкіна І.В.

_____ Писарєв В.Д.

РЕФЕРАТ магістерської роботи

Кваліфікаційна робота: 67 с., 21 рис., 7 табл., 37 літературних джерел.

Актуальність: Сучасний спорт активно інтегрує цифрові технології, що змінюють взаємодію між спортсменами, тренерами та іншими учасниками, створюючи потребу в ефективному управлінні спортивною кар'єрою.

Об'єкт дослідження: Цифрові платформи для підтримки управління профілем і діяльністю спортсменів.

Предмет дослідження: Методи та технології розробки персоналізованої платформи для обліку досягнень, аналітики та взаємодії користувачів із використанням інноваційних рішень (наприклад, блокчейн).

Мета: творити вебплатформу для ведення спортивних результатів, формування рейтингу, взаємодії спортсменів і тренерів та контролю цілісності даних.

Завдання дослідження: проаналізувати сучасні цифрові спортивні платформи, вивчити потреби спортсменів, тренерів і спортивних організацій, сформувати функціональні та нефункціональні вимоги до системи, розробити концептуальну модель платформи та реалізувати прототип з базовим функціоналом.

У результаті виконання роботи було (стислий зміст) було проведено аналіз сучасних спортивних цифрових платформ, визначено потреби спортсменів і тренерів, сформовано вимоги до системи, спроектовано архітектуру та модель даних, а також розроблено функціональний прототип вебплатформи з модулями обліку результатів, рейтингування та контролю цілісності даних.

Практичне значення отриманих результатів: розроблений функціональний прототип платформи може бути використаний спортсменами, тренерами та спортивними організаціями для ведення кар'єрних даних, автоматизації обліку результатів, формування рейтингів та забезпечення прозорості інформації завдяки механізму перевірки цілісності даних.

Наукова новизна: полягає у поєднанні традиційних інструментів спортивної аналітики з блокчейн-підходом до фіксації даних, що підвищує довіру до результатів і забезпечує неможливість їх несанкціонованої зміни; а також у створенні концептуальної моделі комплексної платформи управління спортивною кар'єрою.

Ключові слова: спортивна кар'єра, вебплатформа, рейтинг, результати змагань, блокчейн, хеш-ланцюг, база даних, аналітика.

ЗМІСТ

ЗМІСТ	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Порівняльний аналіз сучасних досліджень та розробок за темою дослідження	9
1.2 Дослідження потреб спортсменів, тренерів та спортивних організацій у сфері управління кар'єрою	12
1.3 Сформовані вимоги до функціоналу персоналізованої платформи	15
Висновок до розділу 1	16
РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ	17
2.1 Визначення функціональних та нефункціональних вимог	17
2.2 Вибір технологічного стеку	22
2.3 Архітектура системи	27
2.4 Проєктування моделі даних у Postgresql	32
2.5 Проєктування інтерфейсу користувача	38
2.6 Проєктування блокчейн-компонента	46
Висновок до розділу 2	50
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ	51
3.1 Загальні принципи побудови	51
3.2 Структура програмного застосунку	52
3.3 Інтерфейс користувача	53
3.4 Логіка серверної частини	57
3.5 Робота з базою даних	58
3.6 Модуль перевірки цілісності	59
3.7 Тестування та перевірка роботи	59
Висновок до розділу 3	60
РОЗДІЛ 4 ВПРОВАДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ПЛАТФОРМИ	61
4.1 План впровадження платформи	61
4.2 Оцінка ефективності за відгуками користувачів	61
4.3 Рекомендації для подальшого розвитку	62
Висновок до розділу 4	63
ВИСНОВОК	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:	65

ВСТУП

Сучасний спорт активно розвивається разом із цифровими технологіями, що змінюють підходи до взаємодії між спортсменами, тренерами, федераціями та іншими учасниками спортивної сфери. Спортсмени дедалі частіше стикаються з потребою ефективно керувати власною кар'єрою: вести профіль, систематизувати досягнення, планувати тренувальний і змагальний календар, аналізувати прогрес, взаємодіяти зі спонсорами й агентами та розвивати власний бренд. Існуючі цифрові рішення, такі як Strava, TrainingPeaks чи MySwimPro, зосереджуються переважно на фіксації тренувань або соціальній взаємодії, але не охоплюють комплексний підхід до управління кар'єрою спортсмена[4].

Об'єктом дослідження в роботі є сучасні цифрові платформи, що застосовуються у сфері спорту для підтримки процесів управління профілем і діяльністю спортсменів. Предметом дослідження є методи та технології розробки персоналізованої платформи, здатної інтегрувати облік спортивних досягнень, аналітику, взаємодію між користувачами, а також використання інноваційних рішень, таких як блокчейн, для підвищення довіри до даних.

Метою роботи є створення вебплатформи, що забезпечуватиме ведення спортивних результатів, формування рейтингу, підтримку процесів взаємодії між спортсменами і тренерами та контроль цілісності даних. Для досягнення цієї мети виконано аналіз наявних рішень, дослідження потреб користувачів, формування вимог до системи, проектування архітектури та практичну реалізацію прототипу.

У межах дослідження визначено такі ключові завдання: проаналізувати сучасні розробки у сфері цифрових спортивних платформ, дослідити потреби спортсменів, тренерів і спортивних організацій, сформувати функціональні й нефункціональні вимоги до системи,

розробити концептуальну модель та реалізувати прототип платформи з базовим функціоналом

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Порівняльний аналіз сучасних досліджень та розробок за темою дослідження

Упродовж останнього десятиліття розвиток інформаційних технологій значно вплинув на спортивну індустрію. З'явилася низка платформ і рішень, які дозволяють спортсменам, тренерам і спортивним організаціям ефективніше управляти різними аспектами спортивної діяльності.

У даному підрозділі проведено аналіз поточних досліджень і комерційних розробок, пов'язаних із темою створення персоналізованої платформи для управління спортивною кар'єрою.

На сьогоднішній день більшість платформ орієнтовані на один або кілька з таких напрямків:

- Трекінг фізичної активності (фітнес-додатки, платформи для моніторингу тренувань);
- Аналітика результатів (професійні платформи для тренерів і спортсменів);
- Управління змаганнями (системи реєстрації, турнірні платформи);
- Соціальні мережі для спортсменів (платформи з функціоналом взаємодії користувачів).

При цьому комплексні рішення, які б повністю охоплювали всі аспекти управління спортивною кар'єрою, включно з комерційними та медійними можливостями, на ринку практично відсутні.

Аналіз прикладів комерційних платформ

Strava є однією з найпопулярніших платформ для трекінгу спортивної активності, яка підтримує інтеграцію з великою кількістю пристроїв, таких як спортивні годинники та фітнес-браслети. Вона має потужну спільноту з понад 100 млн користувачів, пропонує базовий аналіз

тренувань і розвинені соціальні функції. Водночас платформа орієнтована переважно на аматорів і не містить інструментів для управління спортивною кар'єрою, фінансами, взаємодією зі спонсорами чи контрактною діяльністю, а також не використовує блокчейн-технології.

MySwimPro є спеціалізованим додатком для плавців, що пропонує персоналізовані плани тренувань, зручний інтерфейс та хорошу підтримку мобільних пристроїв. Він спрямований на покращення спортивних результатів, однак має дуже вузьку спеціалізацію та не включає функцій для управління кар'єрою, фінансової аналітики чи взаємодії зі спонсорами.

TrainingPeaks орієнтована на професійних спортсменів і тренерів, переважно у велоспорті та триатлоні. Платформа забезпечує глибоку аналітику тренувального процесу та підтримує інтеграцію з різними девайсами, що робить її популярною в професійному середовищі. Проте вона не містить модулів для комерційного управління спортивною кар'єрою, не має інструментів для співпраці зі спонсорами, побудови власного бренду спортсмена та не підтримує блокчейн-технології.

Критерії аналізу

Під час аналізу було використано такі критерії:

- Функціональне охоплення - наскільки платформа покриває аспекти управління кар'єрою (профіль спортсмена, тренування, фінанси, спонсори, календар подій тощо).
- Соціальна взаємодія - наявність механізмів для взаємодії між користувачами (спортсмени, тренери, агенти).
- Інтеграція із зовнішніми сервісами - підтримка API, зв'язок з носимими девайсами, інтеграція з соціальними мережами.
- Інноваційні технології - використання AI, блокчейн, NFT тощо.
- Зручність користування - UX/UI дизайн, мобільна доступність.
- Прозорість та безпека даних - наскільки платформа захищає персональні та фінансові дані користувачів.

Таблиця 1 - Порівняльний аналіз платформ

Платформа	Функціональне охоплення	Соціальна взаємодія	Інтеграції	Інновації (AI/Blockchain)	Зручність UX/UI	Прозорість даних
Strava	Трекінг тренувань, соціальні функції	Висока	Гарна	AI-аналітика, Blockchain - ні	Висока	Помірна
MySwim Pro	Персоналізовані плани тренувань	Обмежена	Обмежена	AI - частково	Висока	Добра
Training Peaks	Глибока аналітика тренувань, профілі	Обмежена	Відмінна	AI-аналітика	Висока	Добра

Висновки аналізу

- Існуючі фітнес-платформи (Strava, MySwimPro, TrainingPeaks) добре покривають тренувальний аспект, проте повністю ігнорують комерційне управління кар'єрою (спонсори, фінанси, контракти).
- В жодному із проаналізованих рішень не реалізовано єдиного інтегрованого підходу, що поєднує тренувальний процес, медіа-менеджмент, фінансову взаємодію, просування бренду і прозорість за допомогою блокчейн.

Таким чином, існує значна ринкова ніша для створення комплексної персоналізованої платформи, яка б дозволяла спортсмену управляти всіма аспектами власної кар'єри у єдиному середовищі.

Сучасні дослідження в області блокчейн у спорті

Блокчейн у спорті

- Ahmed-Sami Berkani et al. (2024) провели систематичний огляд застосування блокчейн-технологій у спортивній індустрії, виділивши ключові кейси: управління даними спортсменів, безпечне проведення подій, відслідковування спортивних колекцій, смарт-контракти для контрактних зобов'язань. Автори зазначають брак прототипних рішень для повноцінної реалізації потенціалу блокчейну [1, 11].
- Попередній доклад Preprints.org (2025) підтверджує потенціал блокчейну для прозорості, ефективності та стимулювання конкурентоздатності у спорті, одночасно виділяючи проблеми масштабованості, нормативних бар'єрів та спротиву стейкхолдерів [2].
- Стаття в Journal of Sport Industry & Blockchain Technology (2024) також підкреслює переваги блокчейну: прозорість записів, аудит, автентифікацію транзакцій, але звертає увагу на технічні обмеження, що потребують рахунку при проектуванні рішень [3].

Таким чином, розробка персоналізованої платформи для управління спортивною кар'єрою є перспективним напрямом, який дозволяє об'єднати наявні технології (AI, блокчейн, Big Data) у єдине інноваційне рішення.

1.2 Дослідження потреб спортсменів, тренерів та спортивних організацій у сфері управління кар'єрою

Розробка ефективної персоналізованої платформи для управління спортивною кар'єрою неможлива без глибокого розуміння потреб кінцевих користувачів - спортсменів, тренерів та представників спортивних організацій. Саме на ці потреби має орієнтуватися архітектура платформи, її функціонал та концептуальні підходи.

Для вивчення потреб було проаналізовано:

- результати опитувань спортсменів різного рівня підготовки (від юніорів до професіоналів);

- результати інтерв'ю з тренерами в індивідуальних та командних видах спорту;
- аналітичні дані щодо використання цифрових інструментів у спортивному менеджменті;
- рекомендації експертів зі спортивного маркетингу.

У результаті цього дослідження було сформульовано такі основні групи потреб.

Потреби спортсменів

Управління персональним профілем

- Ведення структурованої інформації про спортивні досягнення, нагороди, сертифікати.
- Завантаження відео- та фото матеріалів з тренувань та змагань.
- Створення портфоліо для представлення спонсорам та агентам.

Управління фінансовими аспектами

- Моніторинг контрактів та спонсорських угод.
- Прозорий облік виплат та бонусів.
- Підтримка взаємодії з агентами та бухгалтерськими службами.

Планування кар'єри

- Інтеграція з календарями змагань і тренувань.
- Створення персональних планів розвитку.
- Аналітика прогресу.

Підтримка персонального бренду

- Інтеграція з соціальними мережами.
- Монетизація контенту (фото, відео).
- Підтримка особистого маркетингу.

Прозорість і довіра

- Підтримка реєстрів досягнень на блокчейн.
- Незмінний облік контрактів і взаємодій.

Потреби тренерів

- Аналітика результатів спортсменів

- Детальний аналіз тренувального процесу.
- Порівняння динаміки результатів за періодами.
- Виявлення сильних і слабких сторін спортсменів.
- Управління тренувальними групами
 - Формування груп за рівнем підготовки.
 - Моніторинг участі в тренуваннях.
 - Планування навантажень.

Взаємодія з іншими учасниками платформи

- Комунікація з менеджерами і агентами.
- Обмін відеоматеріалами з аналітичними коментарями.
- Підтримка процесу підготовки до змагань.

Потреби спортивних організацій та федерацій

- Управління базами спортсменів
- Ведення єдиної бази спортсменів.
- Підтримка сертифікації та ліцензування.
- Моніторинг участі у змаганнях.

Підтримка прозорості

- Використання блокчейн для фіксації офіційних результатів.
- Автоматизація процесу реєстрації змагань.

Підвищення ефективності взаємодії

- Створення єдиної платформи для комунікації з тренерами та спортсменами.
- Інтеграція з системами електронного документообігу.

Загальні очікування користувачів

У процесі дослідження також були сформульовані загальні очікування щодо функціональності платформи:

- Простота і зручність інтерфейсу.
- Висока швидкодія.
- Забезпечення конфіденційності персональних даних.
- Доступність з мобільних пристроїв.

- Гнучкість персоналізації.

1.3 Сформовані вимоги до функціоналу персоналізованої платформи

На основі проведеного аналізу сучасних розробок, вивчення потреб користувачів та оцінки існуючих рішень було сформульовано перелік функціональних та нефункціональних вимог до майбутньої персоналізованої платформи для управління спортивною кар'єрою.

Загальні концептуальні вимоги

Платформа має забезпечувати:

- єдину точку входу для спортсменів, тренерів, агентів, спонсорів та спортивних організацій;
- прозорість процесів завдяки використанню блокчейн[15];
- широкі аналітичні можливості з використанням Big Data [13, 14];

Функціональні вимоги

Для спортсменів

- Профіль спортсмена
 - Розширене досьє (дані про спортсмена, досягнення).
 - Історія участі у змаганнях.
- Планування кар'єри
 - Інтеграція з календарями змагань та тренувань.
 - Інтерактивні планувальники.
 - Аналітика прогресу.
- Для тренерів
 - Аналітика тренувального процесу спортсменів.
 - Моніторинг прогресу груп спортсменів.
- Для спортивних організацій
 - Управління базами даних спортсменів.
 - Організація та моніторинг змагань.
 - Сертифікація та ліцензування.

- Ведення офіційних реєстрів (з використанням блокчейн).

Технічні вимоги

- Архітектура: мікросервісна, модульна.
- Безпека: шифрування даних, багатофакторна аутентифікація.
- Blockchain: використання смарт-контрактів для фіксації важливих подій та транзакцій.
- Мобільність: повнофункціональні клієнти для iOS та Android.

Нефункціональні вимоги

- Висока масштабованість.
- Висока доступність (99,9% uptime).
- Гнучкість інтерфейсу.
- Відповідність законодавчим вимогам (GDPR та ін.).

Висновок до розділу 1

У результаті проведеного аналізу було підтверджено високу актуальність розробки персоналізованої цифрової платформи для управління спортивною кар'єрою. Сучасні тенденції розвитку спорту демонструють зростаючу потребу спортсменів, тренерів та спортивних організацій у комплексних IT-рішеннях, що дозволяють ефективно управляти різними аспектами спортивної діяльності - від планування тренувань до моніторингу фінансових потоків і побудови персонального бренду.

Проведений порівняльний аналіз існуючих платформ показав, що ринок наразі представлений переважно фрагментованими рішеннями, які орієнтовані або на аналітику тренувального процесу, або на взаємодію з фанатами через Web3 технології. При цьому комплексної системи, яка б об'єднувала управління спортивними досягненнями, фінансами, контрактами, спонсорською діяльністю та персональним брендом спортсмена у єдиному цифровому середовищі, на даний момент фактично не існує.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ

2.1 Визначення функціональних та нефункціональних вимог

Перед безпосереднім проєктування програмної архітектури важливо чітко визначити, які саме задачі має вирішувати платформа та в яких умовах вона буде працювати. Вимоги є основою для всіх наступних етапів розробки: побудови моделі даних, вибору технологій, організації інтерфейсу та тестування. Якщо на цьому етапі вимоги будуть сформульовані нечітко або неповно, це призведе до постійних доробок, суперечностей та переробок на більш пізніх стадіях.

Функціональні вимоги описують, що саме повинен робити застосунок, а нефункціональні - як саме він має працювати з точки зору швидкодії, зручності, безпеки та інших характеристик якості.

Функціональні вимоги формувалися на основі аналізу предметної області, існуючих рішень та узагальнених потреб основних груп користувачів: спортсменів, тренерів та адміністраторів.

Платформа орієнтована на використання в індивідуальних та командних видах спорту, де важливе ведення історії виступів, прозорий облік результатів та формування рейтингу.

До основних функціональних вимог належать такі підгрупи.

1) Управління обліковими записами та ролями користувачів

Платформа має підтримувати кілька типів користувачів, зокрема:

- спортсмен;
- тренер;
- адміністратор.

Для цього передбачається реалізація таких можливостей:

- реєстрація нового користувача із зазначенням ролі;
- вхід до системи із перевіркою облікових даних;
- базові налаштування облікового запису;

2) Профіль спортсмена та облік кар'єрної інформації

Кожен спортсмен має мати власний структурований профіль, який включає:

- основні персональні дані (ПІБ, дата народження);
вид спорту;
- спортивний розряд або рівень підготовки;
- історію виступів у змаганнях;
- ключові спортивні досягнення (призові місця, рекорди, звання).

Функціонально це означає наявність:

- форми перегляду та редагування профілю спортсмена;
- прив'язки результатів виступів до конкретного профілю;
- можливості переглянути «кар'єрну стрічку» спортсмена у хронологічному порядку.

3) Облік змагань та їх структурування

Платформа має забезпечувати ведення довідника змагань. Для кожного змагання необхідно зберігати:

- назву;
- тип (місцеві, національні, міжнародні тощо);
- дату та місце проведення;
- вид спорту;
- список зареєстрованих учасників.

Функціональні можливості:

- створення, редагування та видалення записів про змагання (адміністратором);
- перегляд списків учасників;
- фільтрація змагань за датою, видом спорту, рівнем змагань.

4) Внесення та облік результатів виступів

Ключовим функціональним блоком платформи є облік результатів спортсменів на змаганнях.

Система повинна дозволяти:

- додавати результат спортсмена для конкретного змагання;
- вказувати основні параметри результату (зайняте місце, бали);
- переглядати результати окремого спортсмена або всього змагання у табличному вигляді.

Для зручності передбачається можливість:

- сортувати результати за місцем, часом, очками;
- фільтрувати виступи за періодом (наприклад, за сезон чи календарний рік).

5) Формування рейтингу спортсменів і тренерів

Ще однією важливою функцією є побудова рейтингу на основі результатів виступів.

Платформа має:

- автоматично перераховувати рейтинг спортсмена після додавання нового результату;
- враховувати вагу змагання (місцеві, національні, міжнародні), зайняте місце, розряд спортсмена;
- зберігати історію зміни рейтингу, щоб можна було проаналізувати динаміку.

Для тренерів можливий окремий рейтинг, що базується на виступах підопічних спортсменів.

Функціональні можливості у цьому модулі:

- перегляд рейтингових таблиць за видом спорту та періодом;
- перегляд особистого рейтингу спортсмена та його історії;
- побудова простих рейтингових «топів» (наприклад, 10 найкращих спортсменів у категорії).

6) Візуалізація результатів і статистики

Платформа має надавати користувачам наочні інструменти для аналізу:

- побудова графіків зміни результатів спортсмена у часі;
- відображення динаміки рейтингу;

Це дозволяє спортсмену та тренеру візуально оцінювати прогрес, а не лише дивитися на «сухі» таблиці.

7) Забезпечення прозорості даних за рахунок фіксації хешів

Для підвищення довіри до даних передбачається механізм, який фіксує стан важливих записів (результатів, рейтингових таблиць) у вигляді криптографічних хешів.

Система повинна:

- формувати хеш на основі ключових полів результатів та рейтингу;
- зберігати послідовність хешів у спеціальному реєстрі;
- дозволяти перевірити, чи не змінювалися дані заднім числом.

На рівні функціональних вимог це можна сформулювати як:

- можливість перевірки цілісності історичних даних;
- наявність службового журналу важливих змін.

Нефункціональні вимоги

Нефункціональні вимоги описують властивості платформи, які не пов'язані безпосередньо з конкретними функціями, але сильно впливають на якість роботи системи та сприйняття її користувачами.

Основні групи нефункціональних вимог такі.

1) Вимоги до продуктивності

- Платформа повинна забезпечувати прийнятний час відповіді на типові запити (відкриття списку змагань, перегляд результатів, побудова основних графіків).
- При одночасній роботі кількох десятків користувачів час відгуку не повинен суттєво збільшуватися.
- Операції, пов'язані з перерахунком рейтингу, можуть бути ресурсомісткішими, тому допускається їх виконання у фоновому режимі, але без помітних затримок для користувача.

2) Вимоги до надійності

- Застосунок повинен коректно обробляти помилки введення даних, некоректні запити та тимчасову недоступність окремих сервісів.

- Важливі операції (додавання результату, оновлення рейтингу, запис у реєстр хешів) мають виконуватися транзакційно, тобто або повністю успішно, або не виконуватися зовсім. Необхідно забезпечити періодичне резервне копіювання бази даних, щоб уникнути втрати інформації.

3) Вимоги до безпеки

- Дані користувачів (особливо облікові записи) мають захищатися за допомогою сучасних механізмів автентифікації.
- Доступ до окремих функцій повинен визначатися роллю користувача (рольова модель доступу).
- Дії, пов'язані з редагуванням важливих даних, мають бути доступні лише обмеженому колу користувачів (наприклад, адміністраторам).
- Бажано використовувати захищені протоколи передавання даних (HTTPS) при розгортанні платформи в реальному середовищі.

4) Вимоги до зручності використання (юзабіліті)

- Інтерфейс має бути інтуїтивно зрозумілим, без перевантаження зайвими елементами.
- Основні сценарії (перегляд своїх результатів, вибір змагання, перегляд рейтингу) повинні виконуватися за мінімальну кількість кроків.
- Важливо забезпечити коректне відображення інтерфейсу на різних розмірах екранів (ноутбуки, великі монітори, планшети).
- Повідомлення про помилки та успішні дії повинні бути зрозумілими та інформативними.

5) Вимоги до масштабованості

- Архітектура повинна дозволяти розширювати функціонал без повної переробки системи (наприклад, додати нові ролі, види звітів, додаткові модулі аналітики).

- Можливість вертикального та горизонтального масштабування (додавання ресурсів серверу, запуск додаткових екземплярів застосунку).

б) Вимоги до супроводу та розширюваності

- Код платформи має бути структурованим (розділення на модулі, логічні шари), щоб інші розробники могли легко орієнтуватися.
- Документація до API й основних модулів має бути достатньою для подальшого супроводу та інтеграції з іншими системами.
- Важливо передбачити можливість локалізації інтерфейсу (наприклад, підтримка кількох мов у майбутньому).

2.2 Вибір технологічного стеку

Проектування будь-якого сучасного веб застосунку передбачає не лише визначення його структури та логіки, а й вибір технологій, за допомогою яких буде реалізована система.

Правильно підібраний технологічний стек визначає не тільки швидкість розробки, але й подальшу стабільність роботи, зручність супроводу та можливість розширення функціоналу в майбутньому.

Вибір технологій для створення платформи управління спортивною кар'єрою здійснювався з урахуванням таких критеріїв:

- поширеність та стабільність технології на ринку;
- доступність документації та спільноти розробників;
- простота інтеграції між компонентами;
- підтримка сучасних архітектурних підходів (REST API, клієнт–серверна модель, контейнеризація);
- безпека та продуктивність у реальних умовах використання.

Загальна структура технологічного стеку

Платформа побудована за принципом клієнт–серверної архітектури, у якій є два основні рівні:

- Frontend (клієнтська частина) - відповідає за взаємодію користувача із системою через браузер;
- Backend (серверна частина) - забезпечує логіку роботи, обробку запитів, зберігання та обробку даних.

Окрім цих двох рівнів, виділяється ще рівень бази даних та службові сервіси, такі як контейнеризація (Docker), а також компонент для збереження хешів результатів (блокчейн-рівень).

Для кожного з цих рівнів було обрано конкретні технології, які наведені нижче.

Frontend - клієнтська частина

Для розробки інтерфейсу користувача було обрано JavaScript-бібліотеку React. React є однією з найпопулярніших технологій для створення веб інтерфейсів і має низку переваг:

- компонентна структура дозволяє розділяти код на незалежні частини, які можна повторно використовувати;
- висока швидкодія за рахунок віртуального DOM;
- підтримка адаптивного дизайну - інтерфейс автоматично підлаштовується під різні розміри екранів.

Крім того, React має велику спільноту розробників і детальну документацію, що значно полегшує навчання та усунення можливих проблем у процесі створення застосунку.

Альтернативами могли бути:

- Angular - повноцінний фреймворк із жорсткою структурою, який потребує більшої кількості часу для освоєння;
- Vue.js - легша бібліотека з подібними можливостями, але менше поширена у великих корпоративних проєктах.

Вибір React зумовлений його гнучкістю, швидкістю та простотою інтеграції з іншими сервісами.

Backend - серверна частина

Для реалізації серверної логіки обрано Node.js з використанням фреймворку Express.js.

Node.js - це середовище виконання JavaScript на стороні сервера, яке дозволяє працювати з великою кількістю запитів одночасно без значного навантаження на систему.

Основні переваги цього вибору:

- можливість використовувати єдину мову програмування (JavaScript) як на фронтенді, так і на бекенді;
- висока продуктивність завдяки неблокуючій моделі обробки запитів;
- велика кількість готових модулів (через npm);
- легкість у створенні REST API, що спрощує зв'язок між клієнтом і сервером.

Таким чином, Node.js + Express забезпечують баланс між швидкістю розробки, продуктивністю та простотою підтримки.

База даних

Для збереження даних використовується PostgreSQL - потужна реляційна система керування базами даних із відкритим вихідним кодом.

Основні причини вибору саме PostgreSQL:

- підтримка складних зв'язків між таблицями, зовнішніх ключів і транзакцій;
- можливість створювати користувацькі функції та тригери, що спрощує обробку даних на стороні бази;
- стабільність і масштабованість навіть при значному обсязі записів;
- активна спільнота та постійна підтримка нових стандартів SQL.

Також PostgreSQL чудово інтегрується з Node.js через бібліотеки pg або ORM-інструменти (наприклад, Sequelize або Prisma), що дозволяє швидко взаємодіяти з даними на рівні об'єктів.

Альтернативи:

- MySQL - більш проста у налаштуванні, але має обмеження при складних запитах і обробці транзакцій;
- MongoDB - документно орієнтована БД, зручна для зберігання неструктурованих даних, але не підходить для випадків із чіткими зв'язками між сутностями (як у спорті: спортсмен → змагання → результати → рейтинг).

Отже, PostgreSQL оптимально підходить для задач, де важлива цілісність і структурованість даних.

Контейнеризація та середовище розгортання

Для спрощення розгортання та підтримки середовища розробки використовується Docker. Це технологія контейнеризації, яка дозволяє запускати кожен компонент системи (клієнт, сервер, база даних) у власному ізольованому контейнері.

Переваги використання Docker:

- стабільність середовища - застосунок працює однаково на будь-якій машині;
- швидке розгортання - достатньо виконати одну команду для запуску всієї системи;
- легке оновлення або відкат версій;
- можливість організації окремих середовищ (розробка, тестування, продакшн).

Блокчейн-компонент

Окремим елементом архітектури є компонент перевірки цілісності даних на основі блокчейн-підходу. Його мета - гарантувати, що внесені результати спортсменів або рейтингів не можуть бути змінені безслідно.

Технологічно цей компонент реалізується як додатковий сервіс на Node.js, який:

- отримує дані про ключові події (новий результат, зміна рейтингу);
- формує криптографічний хеш (SHA256 або подібний алгоритм);
- зберігає цей хеш у таблиці разом із посиланням на попередній запис;

- дозволяє виконати перевірку ланцюга для підтвердження автентичності даних.

Такий підхід не потребує підключення до публічного блокчейну (Ethereum, Bitcoin тощо), а є внутрішнім рішенням, що імітує основний принцип - неможливість зміни минулих записів.

Підсумок вибору технологій

Узагальнимо вибраний стек технологій у таблиці:

Таблиця 2 – Обрані стек технологій

Категорія	Варіанти	Плюси	Мінуси	Вибір
Frontend	React	Гнучкий, популярний, швидка розробка	Потрібна структура	Так
	Angular	Строгий, корпоративний	Складніший	—
	Vue	Простий старт	Менша екосистема	—
Backend	Node.js + Express	Одна мова з фронтендом, швидкий API	Потребує контролю структури	Так
	Django	Безпека, ORM	Інша мова	—
	Spring	Надійний	Важкий	—
База даних	PostgreSQL	Сильні транзакції, складні запити	Складніша конфігурація	Так
	MySQL	Простий	Менше можливостей	—
	MongoDB	Гнучкий NoSQL	Слабкі зв'язки	—
Контейнеризація	Docker	Просте розгортання	—	Так
Цілісність даних	Hash-chain	Простий, безкоштовний	Не публічний	Так

2.3 Архітектура системи

Архітектура платформи - це логічна схема, що показує, як взаємодіють між собою всі частини застосунку: інтерфейс користувача, серверна логіка, база даних і допоміжні сервіси.

Грамотно побудована архітектура дозволяє системі бути стійкою до помилок, масштабованою та зрозумілою для подальшої підтримки.

Під час розробки застосунку для управління спортивною кар'єрою була обрана клієнт-серверна архітектура з чітким розділенням шарів. Такий підхід забезпечує незалежність частин системи: зміни в інтерфейсі не потребують переробки бекенду, а оновлення логіки не впливають на базу даних.

Загальна структура системи

Загальна архітектура платформи (рис. 1) складається з чотирьох основних рівнів:

1. Рівень представлення (Frontend) - відображає інформацію користувачам і забезпечує взаємодію з системою через вебінтерфейс.
2. Рівень бізнес-логіки (Backend) - обробляє запити користувачів, перевіряє дані, виконує розрахунки, керує доступом і взаємодіє з базою даних.
3. Рівень зберігання даних (Database) - відповідає за структуроване збереження інформації про користувачів, спортсменів, змагання, результати, рейтинги тощо.
4. Службові сервіси (Services) - допоміжні компоненти, такі як контейнеризація (Docker) та сервіс фіксації хешів даних (блокчейн-рівень).

Між цими рівнями відбувається постійний обмін інформацією за допомогою запитів REST API, що реалізовані через протокол HTTP.

Таким чином, усі дії користувача у веб інтерфейсі перетворюються на запити до серверної частини, яка у відповідь повертає необхідні дані або повідомлення про виконану операцію.

Загальна архітектура системи

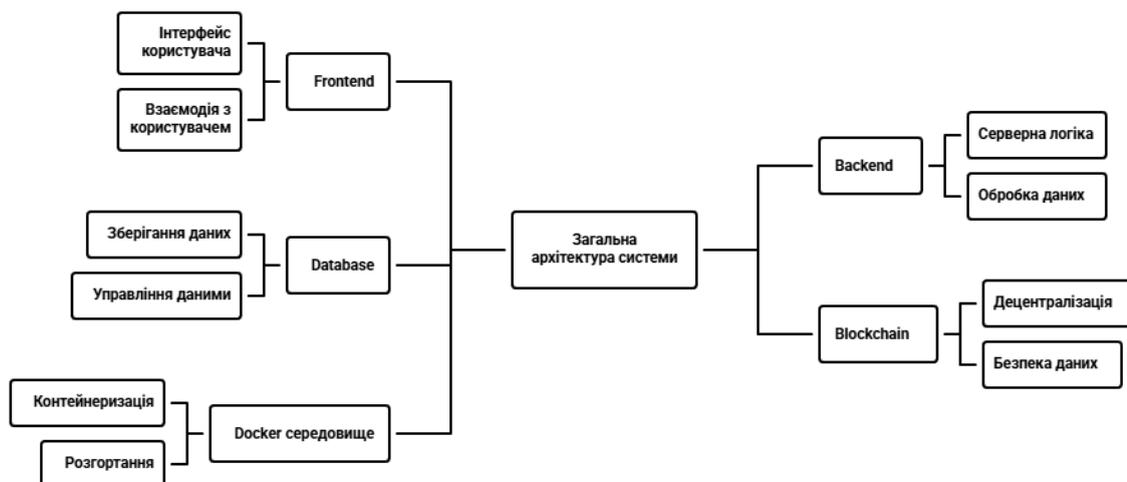


Рисунок 2.3.1. Загальна архітектура системи

Взаємодія компонентів

Послідовність взаємодії між компонентами платформи можна описати на прикладі типового сценарію:

1. Користувач відкриває сторінку платформи у браузері. React-компоненти завантажуються і відображають початковий інтерфейс.
2. Після авторизації користувач (наприклад, тренер) надсилає запит до серверу, щоб отримати список спортсменів або їх результати.
3. Серверна частина (Node.js/Express) приймає запит, перевіряє токен авторизації (JWT) і визначає, чи має користувач право на цю дію.
4. Якщо перевірка пройдена, сервер звертається до бази даних PostgreSQL, виконує запит і формує відповідь у форматі JSON.
5. Отримані дані повертаються клієнту, який динамічно оновлює інтерфейс без перезавантаження сторінки.

6. Якщо користувач додає новий результат змагання, сервер записує його до бази даних і паралельно передає узагальнені дані у блокчейн-сервіс, який формує новий хеш запису для контролю цілісності.

Цей сценарій повторюється для будь-якої операції - від перегляду списку змагань до формування рейтингу(рис. 2).

Послідовність запиту

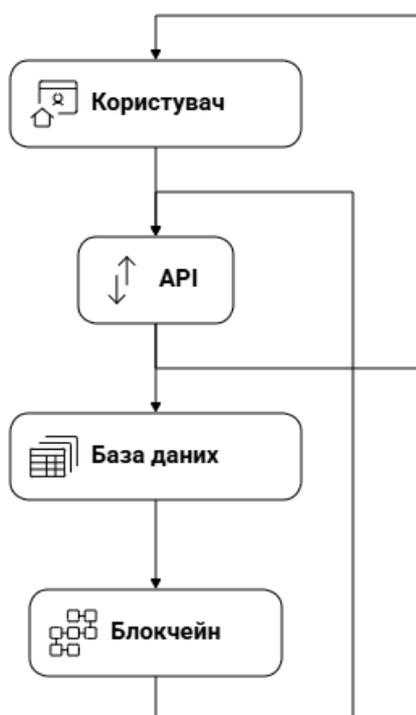


Рисунок 2.3.2. Послідовність запиту

Логічна структура архітектури

Щоб зберегти чистоту коду і спростити підтримку, архітектура поділена на логічні шари:

- Presentation Layer (UI) - веб інтерфейс, створений на React. Відповідає за відображення таблиць, форм, графіків і керування взаємодією користувача.

- Application Layer (API) - набір маршрутів Express, що приймають запити від користувача, перевіряють дані, викликають потрібні сервіси й повертають результат.
- Data Access Layer (DAL) - рівень доступу до бази даних. Реалізований через ORM, що дозволяє працювати з таблицями як з об'єктами.
- Blockchain Layer - допоміжний сервіс, який відповідає за створення хешів та контроль незмінності даних.

Таке розділення дозволяє легко масштабувати систему. Наприклад, якщо потрібно додати новий модуль (наприклад, “Планування тренувань”), він підключається на рівні бізнес-логіки і не впливає на вже існуючі частини.

Цикл логічних шарів



Рисунок 2.3.3. Цикл логічних шарів системи

Особливості клієнт–серверної взаємодії

У процесі взаємодії між клієнтською та серверною частинами платформи використовується REST API, який базується на стандартних HTTP-методах і забезпечує структурований та передбачуваний обмін даними. Клієнт надсилає запити на сервер у форматі JSON, що дозволяє легко передавати інформацію та інтегрувати її з фронтенд-інтерфейсом. Отримання даних здійснюється через метод GET, який використовується, зокрема, для отримання списку спортсменів, результатів або змагань. Створення нових записів відбувається за допомогою методу POST, наприклад під час додавання нового користувача, спортивної події чи результату. Для редагування наявної інформації застосовуються методи PUT або PATCH, тоді як видалення записів виконується через метод DELETE.

API має чітко організовану структуру маршрутів, що забезпечує логічний доступ до основних ресурсів системи, таких як авторизація, спортсмени, змагання, результати та рейтинги (наприклад: `/api/auth/login`, `/api/athletes`, `/api/competitions`, `/api/marks`, `/api/ratings`). Кожен запит проходить обов'язкову перевірку безпеки через `middleware`, де здійснюється розшифрування JWT-токена та визначення ролі користувача. У разі відсутності відповідних прав доступу сервер повертає код помилки 403 (Forbidden), що захищає систему від несанкціонованих дій і забезпечує коректний розподіл доступу між різними категоріями користувачів.

Контейнеризація та середовище виконання

Для уникнення проблем, пов'язаних із відмінностями робочих середовищ, у проєкті застосовується технологія Docker. Усі основні компоненти системи — сервер, клієнтська частина, база даних та блокчейн-сервіс — розгортаються у вигляді окремих контейнерів, що забезпечує їхню ізоляцію та незалежність. Завдяки цьому вся система може бути запущена однією командою `docker-compose up`, що значно спрощує розгортання як у середовищі розробки, так і на реальному

сервері. Використання Docker гарантує однакові умови роботи для всіх учасників команди, мінімізує ризик конфліктів між компонентами та дозволяє швидко оновлювати окремі частини застосунку без порушення роботи інших контейнерів.

Розширюваність архітектури

Під час проєктування архітектури системи враховувалася її подальша масштабованість та можливість розширення функціональності без необхідності істотної перебудови існуючих компонентів. Завдяки модульній структурі застосунку та використанню єдиного REST API у майбутньому можна безперешкодно додавати нові модулі, наприклад компонент аналітики тренувань із власними таблицями та бізнес-логікою, не змінюючи при цьому основних структур. Передбачена також можливість інтеграції з публічними блокчейн-мережами для фіксації хешів не лише локально, а й у глобальному децентралізованому реєстрі. Крім того, архітектура дозволяє реалізувати мобільний застосунок, який використовуватиме ті самі API-ендпоїнти, що й вебверсія. Такий підхід забезпечує гнучкість системи та дає змогу розширювати її функціонал без повного перепроєктування.

2.4 Проєктування моделі даних у PostgreSQL

В цьому розділі визначається, як зберігається інформація про спортсменів, користувачів, змагання, результати та інші сутності.

Основна мета - забезпечити цілісність, узгодженість та можливість швидкого доступу до даних без дублювання.

Для реалізації бази даних було використано PostgreSQL - реляційну систему, яка поєднує класичний підхід до структурованих даних і підтримку сучасних типів, таких як jsonb для зберігання метаданих.

Загальні принципи побудови структури

Під час проєктування бази даних було враховано кілька базових принципів:

- Нормалізація даних - уникнення дублювання інформації та розбиття на логічно незалежні таблиці (наприклад, спортсмени, користувачі, змагання, результати).
- Зв'язність - між таблицями встановлено зовнішні ключі (FOREIGN KEY), що забезпечують логічну цілісність (наприклад, marks.athlete_id пов'язаний із athletes.id).
- Незмінність ключових записів - для запобігання спотворенню історичних даних додано тригери, які блокують оновлення чи видалення певних записів.
- Гнучкість - де потрібно, використано тип jsonb, щоб зберігати додаткові метадані (наприклад, деталі змагань або додаткові характеристики спортсмена).

Таблиця athletes

Ця таблиця зберігає основну інформацію про спортсменів.

Таблиця 3 - Таблиця спортсменів

Поле	Тип	Призначення
id	uuid	Унікальний ідентифікатор спортсмена
name	text	Ім'я або ПІБ спортсмена
meta	jsonb	Додаткові дані (наприклад, біографічна інформація, параметри)
sport	text	Вид спорту
rank	text	Розряд або категорія спортсмена
is_active	boolean	Статус активності
created_at	timestampz	Дата створення запису

Для швидкого пошуку створено індекси за полями name, sport, rank.

Таким чином, при запитах на відбір спортсменів за видом спорту або розрядом система працює ефективно навіть при великому обсязі даних.

Таблиця users

Таблиця користувачів є базовою для всієї системи, оскільки об'єднує спортсменів, тренерів і адміністраторів.

Ролі користувачів визначені за допомогою обмеження:

CHECK (role IN ('admin', 'coach', 'athlete'))

Таблиця 4 - Таблиця users

Поле	Тип	Призначення
id	uuid	Унікальний ідентифікатор користувача
email	text	Унікальна адреса електронної пошти
display_name	text	Відображуване ім'я
password_hash	text	Захешований пароль
role	text	Роль користувача (адмін, тренер, спортсмен)
athlete_id	uuid	Посилання на відповідного спортсмена (для спортсменів)
dob	date	Дата народження
created_at	timestampz	Дата створення

Таке поєднання дозволяє мати єдиний список користувачів, у якому кожен запис пов'язаний або з конкретним спортсменом, або з тренером, або є адміністратором.

Таблиця competitions зберігає інформацію про змагання, у яких беруть участь спортсмени.

Таблиця 5 - Таблиця змагань

Поле	Тип	Призначення
id	uuid	Ідентифікатор змагання
owner_id	uuid	Посилання на користувача, який створив запис
name	text	Назва змагання
date	date	Дата проведення
meta	jsonb	Додаткова інформація (тип змагання, місце, рівень тощо)
created_at	timestampz	Дата створення запису

Ця таблиця може зберігати як офіційні чемпіонати, так і внутрішні турніри. Для гнучкості використано поле meta, у якому можна зберігати, наприклад, категорію змагання чи рівень (національний, міжнародний).

Таблиця marks

Marks - це центральна таблиця системи, яка містить результати виступів спортсменів.

Таблиця 6 - Таблиця Оцінок

Поле	Тип	Призначення
id	uuid	Ідентифікатор результату
competition_id	uuid	Посилання на таблицю змагань
user_id	uuid	Користувач, який вніс результат
athlete_id	uuid	Спортсмен, до якого належить результат
participant	text	Ім'я учасника (може дублювати ім'я спортсмена, якщо потрібно зберегти історичну версію)
score	numeric(10,2)	Основний результат (бали, час тощо)

sport_rank	text	Розряд спортсмена на момент змагання
placement	int	Місце, зайняте у змаганні
payload	jsonb	Додаткові метадані
created_at	timestamptz	Дата внесення результату

Для збереження цілісності результатів встановлено обмеження ON DELETE CASCADE, тобто якщо змагання видаляється, усі результати автоматично видаляються разом із ним.

Індекси створені для полів:

```
CREATE INDEX idx_marks_competition ON marks(competition_id);
```

```
CREATE INDEX idx_marks_athlete ON marks(athlete_id);
```

```
CREATE INDEX idx_marks_created_at ON marks(created_at);
```

Таблиці batches і batch_items - блокчейн-рівень

Ці таблиці відповідають за фіксацію незмінності даних. Вони створюють своєрідний внутрішній блокчейн, де кожен запис має хеш попереднього блоку.

- batches - містить агреговану інформацію про групу результатів (пакет даних): часовий діапазон, меркл-корінь, хеш, час анкерування.
- batch_items - таблиця, яка пов'язує окремі результати (marks) з конкретним пакетом (batches) і зберігає шлях у меркл-дереві (merkle_path).

Таке рішення гарантує, що дані про результати неможливо змінити без порушення цілісності ланцюга.

Таблиця coach_athletes

Ця таблиця реалізує зв'язок багато-до-багатьох між тренерами (users.role='coach') і спортсменами (athletes).

Таблиця 7 - Таблиця зв'язку тренера зі спортсменом

Поле	Тип	Призначення
coach_id	uuid	Ідентифікатор тренера
athlete_id	uuid	Ідентифікатор спортсмена

Первинний ключ складається з обох полів, що гарантує унікальність зв'язку між конкретним тренером і спортсменом.

При видаленні користувача або спортсмена запис автоматично видаляється завдяки ON DELETE CASCADE.

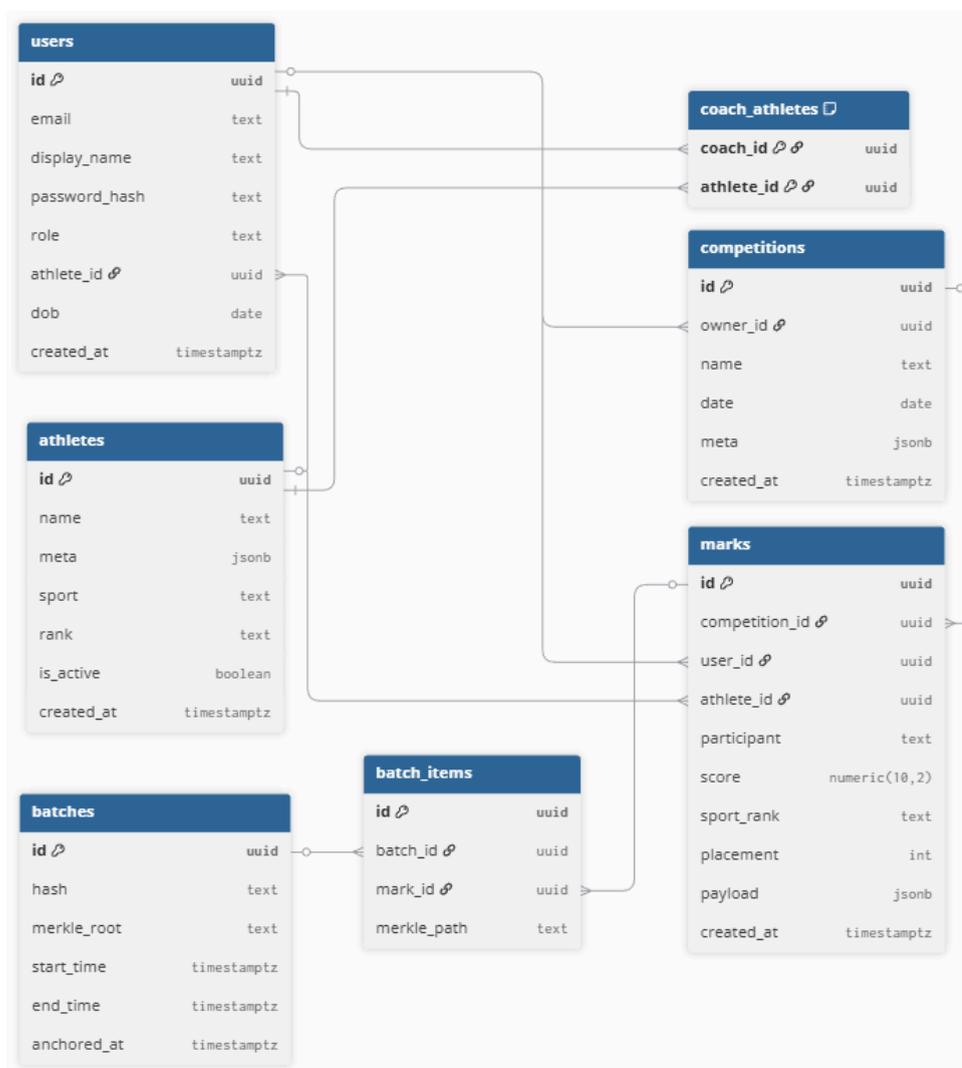


Рисунок. 2.4.1. Схема бази даних

Тригери та обмеження незмінності

Для деяких таблиць (marks, batches, batch_items) створені тригери, які забороняють змінювати або видаляти записи після внесення.

Функція prevent_modifications() генерує виняток при спробі змінити такі дані. Це зроблено, щоб гарантувати прозорість та довіру до інформації про результати.

Приклад тригера:

```
CREATE TRIGGER marks_immutable BEFORE UPDATE OR DELETE  
ON marks
```

```
FOR EACH ROW EXECUTE FUNCTION prevent_modifications();
```

Таким чином, навіть адміністратор не може «підправити» історію результатів - можна лише додати новий запис із позначкою, що він коригує попередній.

2.5 Проєктування інтерфейсу користувача

На етапі проєктування інтерфейсу користувача основна увага приділяється тому, щоб створити зручну, логічну та зрозумілу систему навігації для різних категорій користувачів - спортсменів, тренерів і адміністраторів.

Метою є розробка інтерфейсу, який забезпечуватиме швидкий доступ до основних функцій платформи, мінімальну кількість кліків для виконання дій і просте візуальне сприйняття.

Загальні підходи до проєктування

Під час планування інтерфейсу було визначено кілька ключових принципів, яким він має відповідати. По-перше, інтерфейс повинен бути простим і мінімалістичним: усі елементи розташовуються логічно, без надлишкових кнопок чи графічних деталей. По-друге, важлива послідовність — усі сторінки системи оформлюються в єдиному стилі, щоб користувач швидко орієнтувався в інтерфейсі. Крім того, дотримується єдність кольорів і шрифтів, використовується нейтральна

палітра з білим і сірим кольорами та синім акцентом, а також чіткий шрифт для зручності читання. Навігація робиться інтуїтивно зрозумілою, щоб меню не потребувало додаткових пояснень, а інтерфейс адаптується під різні екрани пристроїв. Важливим принципом є також зворотний зв'язок: користувач отримує повідомлення після будь-якої дії, наприклад, про збереження даних, помилку або успішне виконання операції.

Концепція головної сторінки

Головна сторінка платформи планується як інформаційна панель (Dashboard), що відображатиме основні показники системи - кількість спортсменів, змагань, останні оновлення, короткі статистичні дані. Вона слугуватиме стартовою точкою, з якої користувач зможе швидко перейти до потрібного розділу.

Передбачається, що на сторінці буде кілька інформаційних блоків:

- коротка статистика (наприклад, кількість активних змагань);
- останні додані результати;
- таблиця поточного рейтингу;
- повідомлення про нові події або оновлення системи.

Користувач одразу зможе побачити основну активність без необхідності переходів між сторінками.



Рисунок 2.5.1. Макет головної сторінки

Навігація та структура сторінок

Загальна навігація проєктується у вигляді верхньої панелі з меню, де розміщуватимуться основні розділи: Головна, Змагання, Рейтинг, Моя сторінка (профіль користувача), Реєстрація/Вхід.

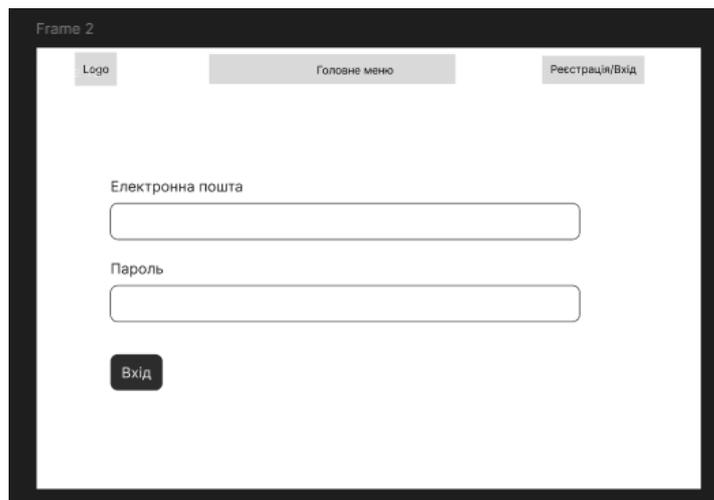
Меню має бути постійно доступним на кожній сторінці, щоб користувач міг легко повернутися або змінити розділ без додаткових дій. Для майбутньої локалізації також передбачено перемикач мови у верхньому правому куті.

Проєктування сторінки “Вхід”

Сторінка входу (авторизації) планується у максимально простому вигляді - дві текстові форми (електронна адреса та пароль) і кнопка «Увійти». У випадку некоректного введення даних система повинна показувати коротке, але зрозуміле повідомлення про помилку.

Також на цій сторінці передбачено посилання «Немає акаунту? Зареєструйтесь», яке відкриватиме сторінку реєстрації.

Основна ідея - не перевантажувати користувача, а дати можливість увійти до системи буквально в два кліки.



The image shows a wireframe of a login page. At the top, there is a header with three items: 'Logo', 'Головне меню', and 'Реєстрація/Вхід'. Below the header, there are two input fields. The first is labeled 'Електронна пошта' and the second is labeled 'Пароль'. Below the password field, there is a button labeled 'Вхід'.

Рисунок 2.5.2. Макет сторінки Вхід

Проектування сторінки “Реєстрація”

Під час розробки інтерфейсу реєстрації заплановано використання послідовного заповнення полів із логічним поділом на секції:

1. Основна інформація: ім’я, електронна адреса, пароль.
2. Додаткова інформація: дата народження, роль користувача (спортсмен або тренер).
3. Спортивна інформація: вид спорту, спортивний розряд.

Для спрощення заповнення будуть реалізовані випадаючі списки з варіантами спорту та розрядів.

Після натискання кнопки “Зареєструватися” користувач має отримати коротке повідомлення про успіх або помилку.

The image shows a wireframe of a registration page. At the top, there is a navigation bar with three items: 'Logo', 'Головне меню', and 'Реєстрація/Вхід'. Below the navigation bar, the registration form consists of the following elements:

- A text input field labeled 'Ім'я'.
- A text input field labeled 'Електронна пошта'.
- A text input field labeled 'Пароль'.
- Two radio button options: 'Атлет' and 'Тренер'.
- A dropdown menu labeled 'Дата народження'.
- A dropdown menu labeled 'Вид спорту'.
- A dropdown menu labeled 'Розряд'.
- A dark button labeled 'Створити акаунт'.

Рисунок 2.5.3. Макет сторінки Реєстрація

Проектування сторінки «Змагання»

Розділ «Змагання» є одним із центральних.

Планується, що він міститиме таблицю зі списком змагань, де відобразатимуться: назва, дата, вид спорту, рівень або категорія,

Також передбачається панель фільтрації (пошук за датою, видом спорту, ключовими словами) та кнопка для додавання нового змагання (для користувачів із правами адміністратора або тренера). При натисканні на рядок відкриватиметься детальна інформація - список учасників, їх результати та посилання на рейтинг.

The image shows a web interface for registration. At the top, there are three navigation buttons: "Logo", "Головне меню", and "Реєстрація/Вхід". Below these are search filters: "Пошук по імені" (text input with "Value"), "Вид спорту" (dropdown), "Тип змагань" (dropdown), "Дата від" (dropdown), and "Дата до" (dropdown). A "Застосувати фільтр" button is positioned to the right of the date filters. Below the filters are two empty tables. The first table is titled "Список змагань" and has 6 columns and 2 rows. The second table is titled "Список учасників та інформація про них" and has 5 columns and 2 rows.

Рисунок 2.5.4. Макет сторінки Реєстрація

Проектування сторінки “Рейтинг”

Сторінка рейтингу проектується у двох вкладках: Рейтинг спортсменів, Рейтинг тренерів.

У таблиці рейтингу передбачено такі колонки: місце у рейтингу, ім’я учасника, вид спорту, кількість виступів, загальний рейтинг.

Також планується реалізувати графік зміни рейтингу у часі для вибраного спортсмена або групи спортсменів.

У верхній частині сторінки будуть фільтри за періодом, видом спорту та іменем користувача.

Рисунок 2.5.5. Макет сторінки рейтингу

Проектування сторінки “Моя сторінка”

Планується створити особисту сторінку користувача, де буде зосереджено всю індивідуальну інформацію: ім'я та роль, вид спорту і розряд, історія виступів у змаганнях, рейтинг спортсмена або групи (для тренера).

Тут же користувач зможе змінювати свої дані або переглядати власні результати у вигляді таблиці чи графіка.

Для наочності передбачено невеликий блок із ключовими показниками - кількість змагань, середній результат, останні досягнення.



Рисунок 2.5.6. Макет сторінки рейтингу

Дизайн, стилі та доступність

Під час проєктування стилів інтерфейсу планується використання єдиного CSS-файлу, у якому будуть описані всі кольори, шрифти, відступи та елементи управління. Інтерфейс має бути легким для сприйняття: білє тло, сірі відтінки для блоків, сині кнопки для основних дій. Усі кнопки й таблиці матимуть заокруглені кути, а повідомлення - легкі тіні, щоб створювати ефект “плаваючих карток”. Передбачено адаптивну верстку, щоб інтерфейс однаково добре працював як на комп’ютерах, так і на мобільних пристроях. Також буде реалізовано багатомовну підтримку (українська та англійська мови).

2.6 Проєктування блокчейн-компонента

Для підвищення рівня довіри до результатів змагань і забезпечення прозорості їх зберігання у платформі передбачається впровадження блокчейн-компонента. Основна ідея полягає у створенні внутрішнього механізму перевірки цілісності даних, який дозволить виявляти будь-які несанкціоновані зміни в історичних записах [7].

Блокчейн-підхід у цьому випадку не передбачає використання публічних мереж, таких як Ethereum чи Bitcoin, а реалізується як внутрішній реєстр хешів усередині системи.

Така концепція поєднує переваги технології блокчейну (незмінність, послідовність, підтвердження даних) із гнучкістю локальної бази даних PostgreSQL.

Мета впровадження блокчейн-компонента

Під час проєктування було визначено кілька завдань, які має вирішувати цей компонент:

1. Підвищення довіри до результатів.
2. Усі ключові записи про результати змагань повинні бути зафіксовані в реєстрі, щоб жоден користувач (навіть адміністратор) не міг їх змінити заднім числом без сліду.
3. Контроль цілісності даних.

Система повинна мати змогу перевірити, чи відповідає поточна версія записів оригінальним даним.

4. Формування історії подій.

Кожна зміна або додавання результату створює новий блок у внутрішньому ланцюгу.

5. Підвищення репутаційної надійності платформи.

Застосування технології блокчейну навіть у локальному варіанті підкреслює надійність і сучасність системи.

Концептуальна модель

Блокчейн-компонент планується реалізувати як послідовність блоків, де кожен блок міститиме:

- набір результатів або статистичних записів за певний період;
- хеш попереднього блоку;
- власний хеш, сформований на основі вмісту поточного блоку;
- часову позначку створення;
- службову інформацію (ідентифікатор, автор, тип події).

Кожен новий блок створюється тільки після завершення певного циклу - наприклад, після додавання результатів одного змагання або після закінчення дня змагань.

Структура ланцюга блоків

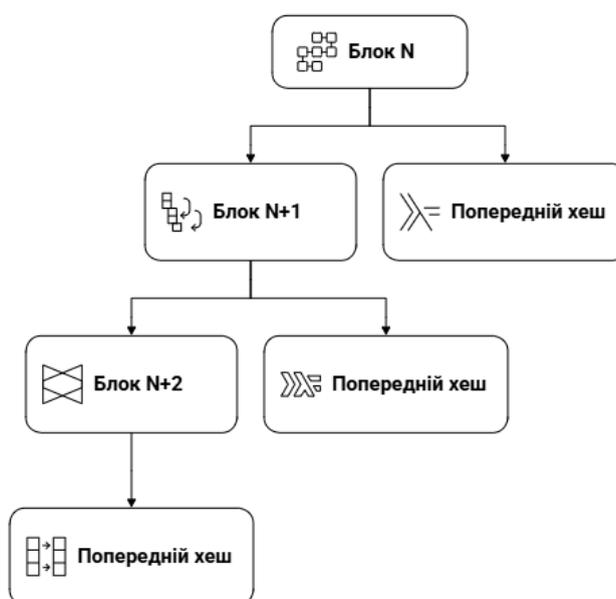


Рисунок 2.6.1. Структура ланцюга блоків

Формування хешів

Під час формування кожного блоку планується використовувати механізм хешування - тобто створення унікального цифрового відбитка даних.

Хеш генерується на основі таких полів: ідентифікатор змагання, ідентифікатор спортсмена, результат або рейтинг, дата внесення запису, хеш попереднього блоку.

Якщо навіть одне з цих полів буде змінене, хеш також зміниться, і система одразу зможе виявити спробу несанкціонованої модифікації.

Для підвищення надійності планується використовувати стандартні криптографічні алгоритми хешування, що гарантують стійкість до підбору даних.

Взаємодія з базою даних

З технічного погляду блокчейн-компонент буде тісно інтегрований із базою даних PostgreSQL.

Для цього планується створити дві таблиці:

- `batches` - для збереження заголовків блоків (час створення, хеш, попередній хеш, підпис користувача);
- `batch_items` - для записів усередині блоку (ідентифікатори результатів, посилання на змагання, проміжні хеші).

Така структура забезпечує зручність пошуку та перевірки, а також дозволяє переглядати історію будь-якого результату від моменту створення до останньої перевірки.

Перевірка цілісності даних

Одним із ключових завдань компоненту є перевірка цілісності ланцюга. Для цього розробляється процедура, яка порівнюватиме обчислені хеші з тими, що зберігаються у базі даних. Якщо система виявить невідповідність, вона повідомить адміністратора про можливе порушення. Передбачається також створення інтерфейсу перевірки, де користувач зможе обрати певний період і запустити перевірку ланцюга. Результат перевірки відобразатиметься у вигляді таблиці або графічної позначки (наприклад, зелений індикатор - дані цілі, червоний - виявлено розбіжності).

Журнал подій і доступ до перевірених даних

Після впровадження блокчейн-компонента система зможе автоматично фіксувати всі ключові події в журналі - додавання змагання, внесення результату, створення блоку, перевірку цілісності.

Для користувача ці події відобразатимуться у вигляді коротких записів у хронологічному порядку.

Такий підхід дозволить не лише бачити поточний стан, але й відслідковувати повну історію взаємодії з даними.

У майбутньому планується додати можливість публічного перегляду хешів у вигляді окремої сторінки «Блокчейн-реєстр», де будуть вказані: номер блоку, дата створення, хеш блоку, попередній хеш, кількість записів усередині блоку.

езпека та обмеження доступу

Під час проєктування компоненту особлива увага приділяється безпеці.

Доступ до функцій створення блоків і перевірки їх цілісності матимуть лише користувачі з роллю адміністратора. Звичайні користувачі зможуть переглядати лише результати перевірки, але не змінювати структуру ланцюга.

Крім того, кожен блок матиме цифровий підпис, який прив'язуватиметься до облікового запису адміністратора, що створив його. Це дозволить чітко визначити, хто саме зафіксував зміни в системі.

Висновок до розділу 2

У другому розділі розроблено проєкт платформи для управління кар'єрою спортсмена.

Визначено функціональні й нефункціональні вимоги, обґрунтовано вибір технологічного стеку, спроектовано архітектуру системи, модель бази даних, інтерфейс користувача та концепцію блокчейн-компонента.

Запропонована структура поєднує сучасні технології (React, Node.js, PostgreSQL, Docker) з гнучкою клієнт–серверною архітектурою та внутрішнім механізмом контролю цілісності даних.

Результатом етапу проєктування є створення логічної основи для подальшої реалізації й тестування системи, що забезпечує прозорість, зручність і надійність роботи майбутньої платформи.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ

Розділ присвячений практичній реалізації веб платформи, яка забезпечує ведення спортивних результатів, розрахунок рейтингу спортсменів і перевірку достовірності даних через механізм хешування.

Система реалізована за клієнт–серверною моделлю, що складається з фронтенд-застосунку, серверної частини, бази даних та внутрішнього модуля контролю цілісності.

3.1 Загальні принципи побудови

Програмна архітектура базується на розподілі обов’язків між трьома рівнями:

- інтерфейсом користувача, який відповідає за взаємодію з користувачем та відображення даних;
- сервером додатку, що реалізує логіку обробки запитів, обчислень і перевірок;
- базою даних, у якій зберігаються всі сутності системи - користувачі, спортсмени, змагання, результати та рейтинги.

Обмін даними відбувається через REST API, а формат передавання - JSON. Система має розмежування доступу за ролями: спортсмени, тренери, адміністратори. Кожен користувач взаємодіє лише з тими розділами, що відповідають його правам.

Ієрархія архітектури системи



Рисунок 3.1.1 Ієрархія архітектури системи

3.2 Структура програмного застосунку

Система складається з кількох взаємопов'язаних модулів.

Її структура має чітке розділення на клієнтську частину, серверну частину та допоміжні модулі.

- Клієнтська частина - реалізована у вигляді односторінкового застосунку, який забезпечує навігацію між сторінками без перезавантаження. Тут реалізовано авторизацію, перегляд змагань, рейтингів, особистих результатів, а також багатомовну підтримку.
- Серверна частина - відповідає за логіку роботи API, перевірку користувачів, створення та редагування записів, розрахунок рейтингів спортсменів і тренерів.
- Модуль бази даних - оперує таблицями користувачів, спортсменів, змагань, результатів, рейтингів і зв'язками між ними.

- Модуль перевірки цілісності (хешування) - створює хеш-ланцюг результатів, що унеможлиблює їх непомітну зміну.

3.3 Інтерфейс користувача

Користувацький інтерфейс реалізовано у вигляді веб застосунку з сучасним дизайном і зрозумілою навігацією. Основні сторінки - це головна панель, авторизація, реєстрація, змагання, рейтинг і персональний профіль. Навігаційна система забезпечує швидкий перехід між розділами, а всі дані завантажуються асинхронно.

Панель користувача відображає основну інформацію: коротку статистику, останні змагання та рейтингові показники.

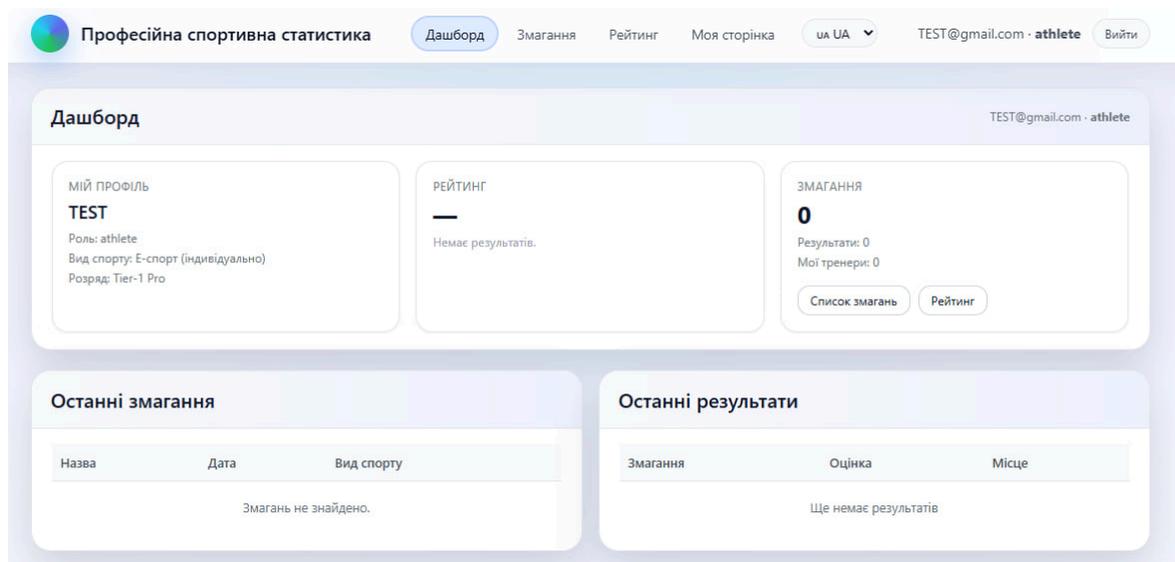


Рисунок 3.3.1 Головна панель

Сторінка змагань надає змогу переглядати всі події, фільтрувати їх за видом спорту та датою, а адміністраторам - створювати нові.

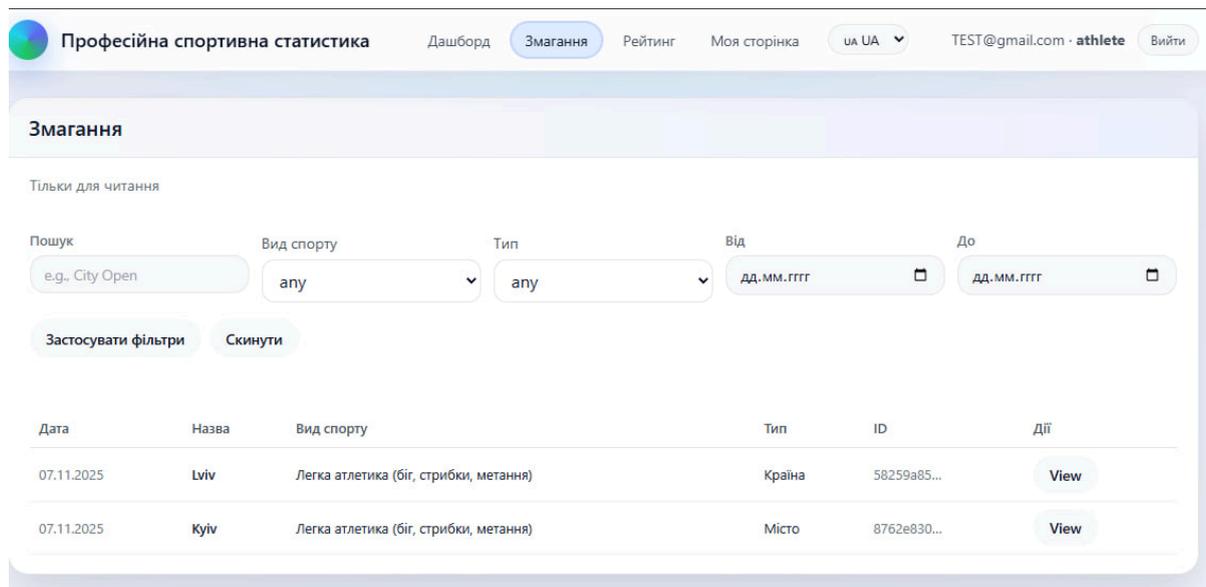


Рисунок 3.3.2 Сторінка змагань

Сторінка рейтингу має дві вкладки - для спортсменів і тренерів. Вона підтримує фільтри за видом спорту, розрядом, діапазоном дат і прізвищем.

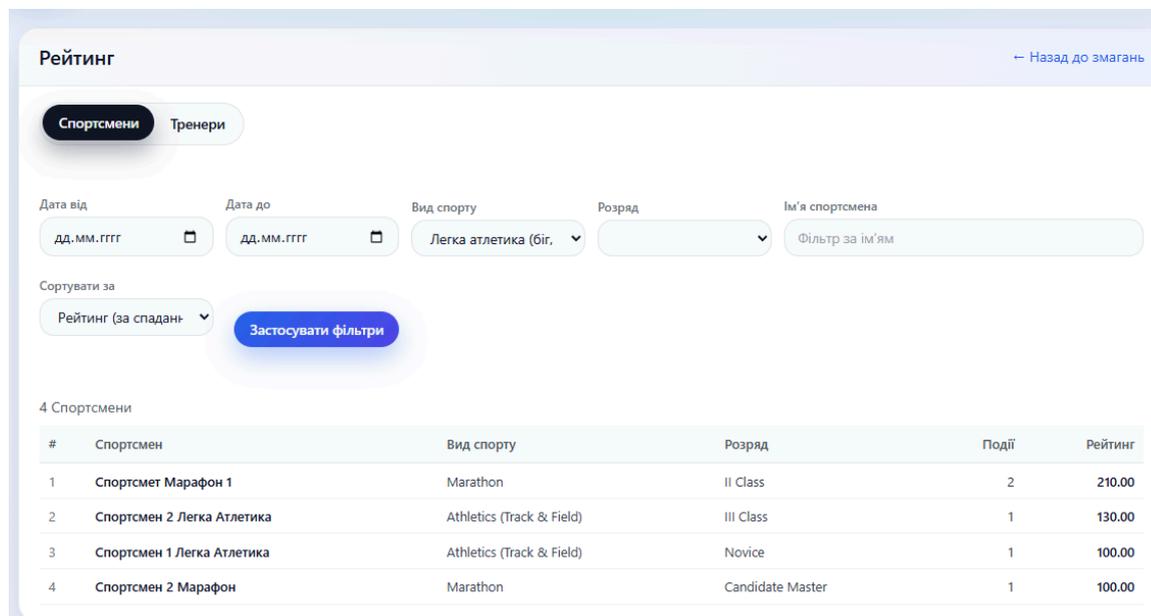


Рисунок. 3.3.3 Сторінка Рейтингу Спортсменів

Спортсмени **Тренери**

Дата від:

Дата до:

Вид спорту:

Розряд:

Ім'я тренера:

Сортувати за:

Застосувати фільтри

2 Тренери

#	Тренер	Спортсмени	Результати	Рейтинг
1	Тренер Марафон	2	3	310.00
2	Тренер Атлетика	2	2	230.00

Рисунок 3.3.4 Сторінка Рейтингу Тренерів

Особистий профіль показує персональні дані користувача, пов'язані з результатами та рейтингом.

Мій профіль

Ім'я в профілі: Спортсмен Марафон 1

Роль: athlete

Рейтинг: 280.00 (2 Тренер)

Дата народження: 17.10.2025

Вид спорту: Марафон

Розряд: II розряд

Міні тренера

Закріплено: Тренер Марафон

Обрати тренера:

Змагання (закріплені спортсмени)

Дата	Змагання	Оцінка	Спортсмени
2025.11.10	Київ	1	1
2025.11.10	Львів	2	2

Місця спортсменів за змаганнями (вид тренера)

Інформація: 3 змагання (без нечислових оцінок) | 0 + показати на графіку 3

Рисунок 3.3.5 Сторінка Користувача

Сторінки входу та реєстрації містять адаптивні форми, повідомлення про помилки, вибір ролі користувача й спортивної спеціалізації.

Професійна спортивна статистика

Увійти Реєстрація ua UA

Увійти

Електронна пошта
vladpysarev3@gmail.com

Пароль
.....

Увійти

Немає акаунта? Реєстрація

Професійна спортивна статистика

Увійти Реєстрація ua UA

Реєстрація

Ім'я в профілі

Електронна пошта
vladpysarev3@gmail.com

Пароль
.....

Role
 Athlete Coach

Дата народження
дд.мм.гггг

Вид спорту
Оберіть вид спорту

Розряд
Оберіть свій розряд

Створити акаунт

Вже маєте акаунт? Увійти

Рисунок. 3.3.6 Сторінки Входу/Реєстрації

Візуальне оформлення інтерфейсу виконано у світлих тонах з акцентними кольорами, закругленими елементами та адаптивними відступами.

Є підтримка двох мов - української та англійської, між якими користувач може перемикається в шапці сайту.

3.4 Логіка серверної частини

Серверна частина розроблена на базі Node.js із використанням фреймворку Express.

Вона реалізує низку REST API маршрутів, які обробляють запити клієнта - від авторизації до побудови рейтингів. Дані передаються у форматі JSON, а доступ захищено за допомогою токенів автентифікації (JWT).

Основні напрямки роботи сервера:

- керування користувачами: реєстрація, авторизація, зміна профілю, перевірка ролей;
- змагання: створення, редагування, фільтрація за видом спорту, типом і датами;
- результати: внесення оцінок, обчислення місць, зв'язок із таблицями спортсменів;
- рейтинги: автоматичний розрахунок рейтингу спортсменів і тренерів за формулою, що враховує місце, ранг і кількість участей;
- інтеграція блокчейн-рівня: збереження хешів записів для перевірки незмінності.

Розрахунок рейтингу виконується на основі результатів виступів спортсменів, де за перше місце нараховується найбільша кількість балів, а також враховується коефіцієнт складності залежно від спортивного розряду.

Сервер підтримує різні ролі доступу: спортсмени бачать лише власні результати, тренери - своїх підопічних, а адміністратор - усі записи. Реалізація розмежування прав доступу базується на моделі RBAC (Role-Based Access Control) у багаторівневій архітектурі [10].

3.5 Робота з базою даних

База даних реалізована на основі PostgreSQL. Вона містить таблиці для користувачів, спортсменів, змагань, результатів, рейтингів, тренерських зв'язків і хешів блоків. Кожна таблиця має зв'язки “один до багатьох” або “багато до багатьох”, що забезпечує логічну цілісність даних.

Основні сутності:

- користувачі - містять персональні дані, роль і посилання на профіль спортсмена;
- спортсмени - мають характеристики (вид спорту, ранг, активність);
- змагання - містять опис, дату, тип (міські, національні, міжнародні);
- результати - зберігають оцінки, місця та зв'язки зі спортсменами й змаганнями;
- рейтинги - розраховуються автоматично на основі формули;
- записи хешів - забезпечують перевірку цілісності кожного результату.

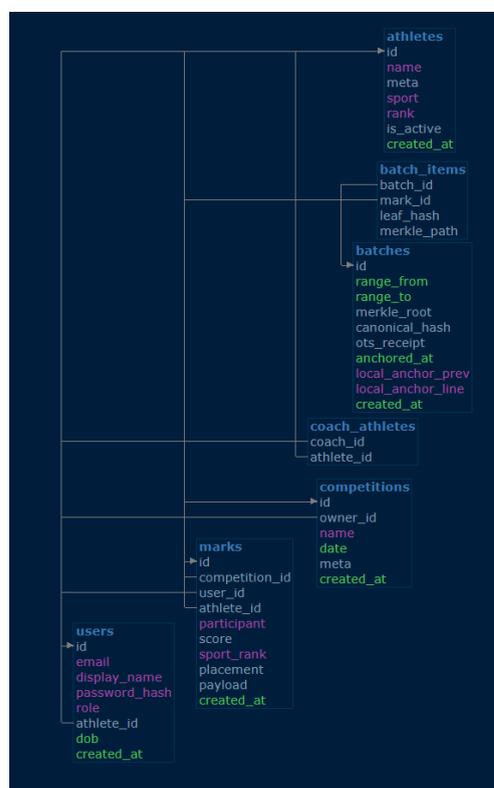
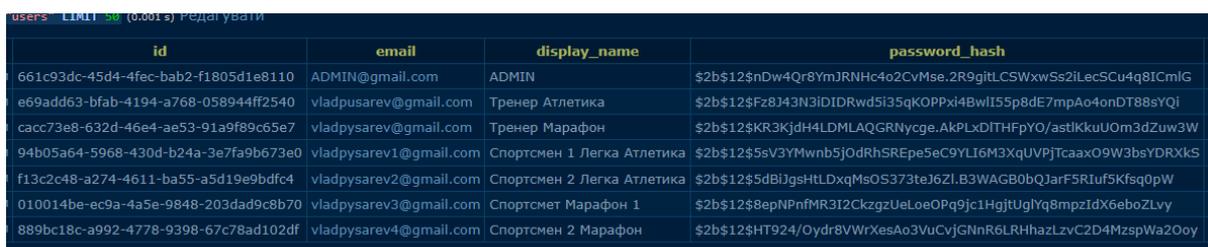


Рисунок 3.5.1. Схема бази даних в PostgreSQL

3.6 Модуль перевірки цілісності

Для забезпечення довіри до даних у системі створено модуль, який формує ланцюг хешів усіх результатів. Кожен результат після внесення в базу перетворюється у стандартизований рядок, який хешується за алгоритмом SHA-256. Коли накопичується певна кількість записів, вони об'єднуються в блок, для якого обчислюється кореневий хеш (Merkle root). Цей блок додається до ланцюга, де кожен наступний містить хеш попереднього. Таким чином, будь-яка спроба змінити попередній результат призведе до порушення хеш-ланцюга, і система зможе це виявити. Функція перевірки дозволяє повторно обчислити хеш і порівняти його з тим, що збережено в базі.



id	email	display_name	password_hash
661c93dc-45d4-4fec-bab2-f1805d1e8110	ADMIN@gmail.com	ADMIN	\$2b\$12\$nDw4Qr8YmJRNhc4o2CvMse.2R9gitLCSWxwSs2iLecSCu4q8ICmlG
e69add63-bfab-4194-a768-058944ff2540	vladpusarev@gmail.com	Тренер Атлетика	\$2b\$12\$Fz8J43N3iDIDRwd5i35qKOPpxi4BwI55p8dE7mpA04onDT88sYQi
cacc73e8-632d-46e4-ae53-91a9f89c65e7	vladpysarev@gmail.com	Тренер Марафон	\$2b\$12\$KR3KjdH4LDMLAQGRNycge.AkPLxDITHFpYO/astlKkuUOm3dZuw3W
94b05a64-5968-430d-b24a-3e7fa9b673e0	vladpysarev1@gmail.com	Спортсмен 1 Легка Атлетика	\$2b\$12\$5sV3YMwnb5J0dRhsREpe5eC9YLI6M3XqUVPJTcaaxO9W3bsYDRXkS
f13c2c48-a274-4611-ba55-a5d19e9bdfc4	vladpysarev2@gmail.com	Спортсмен 2 Легка Атлетика	\$2b\$12\$5dBiJgsHTLDxqMs0S373teJ6ZL.B3WAGB0bQJarF5RIuf5Kfsq0pW
010014be-ec9a-4a5e-9848-203dad9c8b70	vladpysarev3@gmail.com	Спортсмет Марафон 1	\$2b\$12\$8epNpnmR3I2CkzgzUeLoeOPq9jc1HgjtUglYq8mpzIdX6eboZLvy
889bc18c-a992-4778-9398-67c78ad102df	vladpysarev4@gmail.com	Спортсмен 2 Марафон	\$2b\$12\$HT924/Oydr8VWrXesAo3VuCvjGNr6LRHhazLzvC2D4MzspWa2Ooy

Рисунок 3.6.1 Таблиця юзерів в базі даних

3.7 Тестування та перевірка роботи

Після реалізації всіх компонентів проведено комплексне тестування.

Тестування охоплювало:

- авторизацію та реєстрацію користувачів;
- створення та редагування змагань;
- внесення результатів і автоматичне оновлення рейтингу;
- перевірку збереження хешів після кожного оновлення;
- відображення статистики та роботу фільтрів.

Для тестування використовувалися інструменти Postman (перевірка API-запитів) і pgAdmin (контроль структури бази). Роботу інтерфейсу перевіряли у браузері, а стабільність - під час одночасних запитів від кількох користувачів.

Висновок до розділу 3

У результаті реалізації створено робочу систему, яка поєднує веб інтерфейс, серверну логіку, базу даних і механізм контролю достовірності. Система підтримує розмежування доступу, багатомовність, фільтри й автоматичний розрахунок рейтингів. Розроблена архітектура є масштабованою, безпечною та зручною для подальшого розширення функцій, зокрема інтеграції з мобільними застосунками чи офіційними базами спортивних федерацій.

РОЗДІЛ 4 ВПРОВАДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ПЛАТФОРМИ

4.1 План впровадження платформи

Щоб впровадити платформу в реальних умовах, найкраще почати з пілотного тестування.

Для цього можна відібрати невелику групу користувачів: спортсменів-аматорів, тренерів та по можливості представників спортивних клубів.

Процес впровадження можна розділити на кілька етапів:

1. Підготовка тестового середовища.

Створюється окремий тестовий сервер і база даних. Додаються тестові користувачі й кілька змагань для прикладу.

2. Залучення користувачів.

Користувачам надається доступ до платформи та коротка інструкція щодо використання основних функцій: перегляду змагань, введення результатів, роботи з рейтингами.

3. Проведення тестових змагань.

Дані вводяться в реальному часі, щоб оцінити швидкодію, стабільність та зручність інтерфейсу.

4. Збір відгуків.

Після кількох тестових сесій проводиться опитування користувачів щодо зручності роботи, можливих помилок та пропозицій.

4.2 Оцінка ефективності за відгуками користувачів

Після тестування важливо отримати зворотний зв'язок.

Для цього можна використовувати невелику анкету з такими запитаннями:

1. Наскільки зручним був інтерфейс?
2. Чи було легко знайти потрібні розділи (змагання, рейтинг, профіль)?
3. Чи швидко працювала система?
4. Чи були помилки під час введення даних або авторизації?
5. Чи достатньо інформації на сторінках рейтингу та результатів?
6. Які функції хотілося б додати?

На основі таких відповідей можна зробити висновки щодо того, що працює добре, а що треба покращити.

Часті побажання, які зазвичай виникають:

- покращити пошук та фільтри;
- додати більше статистики;
- розширити можливості профілю;
- зробити мобільну версію.

4.3 Рекомендації для подальшого розвитку

Після пілотного впровадження можна визначити основні напрямки вдосконалення платформи:

1. Масштабування для більших організацій.

Можна додати підтримку клубів, федерацій або регіональних відділень.

2. Мобільна версія або додаток.

Користувачам буде зручніше працювати зі смартфона, особливо на змаганнях.

3. Більше аналітики.

Додати графіки прогресу спортсменів, порівняння між сезонами, аналіз середніх результатів.

4. Сповіщення та повідомлення.

Інформування про нові змагання, зміни в рейтингу або нові результати.

5. Інтеграція з іншими системами.

Наприклад, офіційними сайтами федерацій або системами електронної реєстрації спортсменів.

Розвиток системи



Рисунок 4.3.1 Цикл розвитку системи

Висновок до розділу 4

Впровадження платформи доцільно починати з пілотного тестування серед спортсменів і тренерів, оскільки саме вони зможуть надати найбільш корисний зворотний зв'язок.

Отримані відгуки допоможуть визначити слабкі місця та вдосконалити функціонал перед повним запуском.

Подальший розвиток системи можливий у напрямках масштабування, розширення аналітики, мобільності та інтеграції з зовнішніми спортивними сервісами.

ВИСНОВОК

У ході виконання роботи було спроектовано та реалізовано веб платформу, призначену для збору, збереження та аналізу результатів спортсменів. Під час розробки вдалося поєднати сучасні веб технології, надійний серверний підхід і логічно організовану базу даних. Це дозволило створити систему, яка може стати корисним інструментом для спортсменів, тренерів і спортивних організацій.

У першій частині роботи було проведено аналіз вимог та визначено, які саме функції мають бути присутні в системі. Було сформовано цілі, задачі та загальну концепцію майбутнього застосунку. На основі цього виконано детальне проектування: вибрано стек технологій, описано архітектуру, логіку роботи серверної частини та структуру бази даних, а також розроблено інтерфейс користувача.

На етапі реалізації було створено повноцінний клієнт–серверний застосунок. У ньому реалізовано сторінки для входу, реєстрації, перегляду змагань, введення результатів, перегляду рейтингів і роботи з персональним профілем. Серверна частина обробляє всі запити, взаємодіє з базою даних та забезпечує доступ відповідно до ролі користувача. Після впровадження основних модулів була проведена оцінка роботи системи. Тестування показало, що платформа працює стабільно, коректно відображає дані та забезпечує зручну взаємодію для користувача. На основі відгуків сформовано пропозиції щодо покращення, зокрема - потребу в розширенні аналітики, мобільній версії та інтеграції з офіційними спортивними організаціями.

Загалом виконана робота дозволила створити сучасну, зрозумілу та функціональну платформу для обліку та аналізу спортивних результатів. Вона може стати основою для реального впровадження у спортивних клубах, школах та федераціях, а також має потенціал для подальшого розвитку та масштабування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Berkani, A.-S., Khelifi, A., Benslimane, A. Blockchain Applications in Sports Management: A Systematic Review. *Innovative Sports Technology*. 2024. Vol. 2, No. 3, pp. 45–60. DOI: link.springer.com
2. Unlocking Blockchain’s Potential in Sports: A Multisport Comparative Study of Adoption, Performance, and Innovation. Bucea-Manea-Țoniș R. Constanța M., Antonescu A preprints.org+1society-labs.elifesciences.org+1
3. Rahman, S., Uddin, M., Chowdhury, F. Sports Innovation and Blockchain Technologies: Trends, Challenges and Applications. *SIBT – Sports Innovation and Blockchain Technologies*. NDP Academic Publishing, 2024. URL: <https://ndpapublishing.com/index.php/sibt>
4. Nedelko, Z., Meško, M. Digital Transformation in the Sports Industry. *Sporticopedia – SMB*. 2023. Vol. 1, No. 1, pp. 309–318. DOI: 10.58984/smbic2301309d URL: https://www.researchgate.net/publication/377648261_DIGITAL_TRANSFORMATION_IN_THE_SPORTS_INDUSTRY
5. Abramov, V., Astafieva, M., Boiko, M., Bodnenko, D., Bushma, A., Vember, V., Hlushak, O., Zhylytsov, O., Ilich, L., Kobets, N., Kovaliuk, T., Kuchakovska, H., Lytvyn, O., Lytvyn, P., Mashkina, I., Morze, N., Nosenko, T., Proshkin, V., Radchenko, S., ... Yaskevych, V. (2021). Theoretical and practical aspects of the use of mathematical methods and information technology in education and science. <https://doi.org/10.28925/9720213284km>.
6. Бушма, Олександр Володимирович та Машкіна, Ірина Вікторівна та Носенко, Тетяна Іванівна та Яскевич, Владислав Олександрович (2024) Кваліфікаційна робота магістра: Навчально-методичний посібник для спеціальності «Комп’ютерні науки» Київський

столичний університет імені Бориса Грінченка,
Україна <https://elibrary.kubg.edu.ua/id/eprint/50205/>

7. Замрій, І.В. та Шахматов, І.О. та Яскевич, Владислав Олександрович (2024) BlockchainSQLSecure: інтеграція блокчейн-технології для зміцнення захисту від SQL ін'єкцій Вісник Київського національного університету імені Тараса Шевченка. Фізико-математичні науки (1(78)). с. 160-168. ISSN 2218-2055 <https://doi.org/10.17721/1812-5409.2024/1.29>
8. Писарєв , В. (2025). ІНТЕГРОВАНА ІНФОРМАЦІЙНА ПЛАТФОРМА ДЛЯ СПОРТСМЕНІВ І ТРЕНЕРІВ. Збірник матеріалів Всеукраїнської конференції молодих учених "Інформаційні технології" (ISSN: 2664-2638) <https://zcit.kubg.edu.ua/index.php/journal> (с. 77)
9. Писарєв , В. (2025). ІНФОРМАЦІЙНА ПЛАТФОРМА ДЛЯ ПРОФЕСІЙНИХ СПОРТСМЕНІВ І ТРЕНЕРІВ. Збірник матеріалів Всеукраїнської конференції молодих учених "Інформаційні технології" (ISSN: 2664-2638), 9(9), 39–40. <https://zcit.kubg.edu.ua/index.php/journal/article/view/27>
- 10.Рзаєва, Світлана Леонідівна та Складанний, Павло Миколайович та Машкіна, Ірина Вікторівна та Костюк, Юлія Володимирівна (2025) Модель реалізації керування доступом на основі ролей (RBAC) у багаторівневій архітектурі сховища даних Сучасний захист інформації, 63 (3). с. 137-149. ISSN 2409-7292 <https://doi.org/10.31673/2409-7292.2025.031671>
- 11.BERKANI, Ahmed-Sami, et al. Blockchain use cases in the sports industry: a systematic review. International Journal of Networked and Distributed Computing, 2024, 12.1: 17-40.
- 12.YUAN, Xiaochun. Improving Data Security and Privacy in Sports Health Monitoring through Blockchain. Systems and Soft Computing, 2025, 200308.

13. OBI, Ogugua Chimezie, et al. Data science in sports analytics: A review of performance optimization and fan engagement. *World Journal of Advanced Research and Reviews*, 2024, 21.1: 2663-2670.
14. BAI, Zhongbo; BAI, Xiaomei. Sports big data: management, analysis, applications, and challenges. *Complexity*, 2021, 2021.1: 6676297.
15. Bondarchuk, A., Dibrivniy, O., Grebenyuk, V., & Onyshchenko, V. (2021, October). Motion Vector Search Algorithm for Motion Compensation in Video Encoding. In *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)* (pp. 345-348). IEEE.
16. Бондарчук, А. П. Основи інфокомунікаційних технологій / А. П. Бондарчук, Г. С. Срочинська, М. Г. Твердохліб.— К., 2015.— 76 с.
17. Тушич, А. М., К. П. Сторчак "Вимоги до інтелектуальних систем аналізу даних та їх класифікацій." *Телекомунікаційні та інформаційні технології 1* (2019): 31-36.
18. Бондарчук, А., Жебка, В., Корецька, В., Шилкіна, А. (2024). Порівняльна характеристика web-орієнтованих інструментів автоматизації освітнього процесу в умовах цифрової трансформації. *Публічно-управлінські та цифрові практики*, (1), 13-21.
19. Дібрівний, О. А., et al. "Використання принципів SOLID при розробці відео ігор на основі ігрового двигуна UNITY." *Телекомунікаційні та інформаційні технології 1* (2020): 79-87.
20. Співак, Світлана, Андрій Бондарчук, and Олексій Черевик. "AI-система для професійної орієнтації у сфері 3D-графіки." *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка» 3.31* (2025): 698-709.
21. Співак, Світлана Михайлівна, et al. "Adapting education to the 3D graphics market using AI." *Телекомунікаційні та інформаційні технології 89.4* (2025): 215-221.

22. Співак, С. М., & Горбатовський, Д. В. (2025). Тривимірний комп'ютерна графіка. Основи моделювання Blender.
23. Онищенко, В. В., and А. П. Бондарчук. "Програмна інженерія: проблеми та перспективи." Зв'язок 1 (2015): 10-13.
24. Перепелиця, А. О., and О. А. Дібрівний. "Аналіз підходів до адаптивного компонування та ідентифікування елементів в Immediate Mode GUI для їх використання в Unity." Зв'язок 2 (2025): 35-42.
25. Гребенюк, В. В., Дібрівний, О. А., & Негоденко, О. В. (2020). Порівняльний аналіз нерелативних методів оцінювання якості відеоматеріалу. Зв'язок, (6), 61-65.
26. Рзаєва, Світлана Леонідівна та Складанний, Павло Миколайович та Машкіна, Ірина Вікторівна та Костюк, Юлія Володимирівна (2025) Модель реалізації керування доступом на основі ролей (RBAC) у багаторівневій архітектурі сховища даних Сучасний захист інформації, 63 (3). с. 137-149. ISSN 2409-7292 <https://doi.org/10.31673/2409-7292.2025.031671>
27. Співак, Світлана Михайлівна та Машкіна, Ірина Вікторівна та Носенко, Тетяна Іванівна та Білоус, Владислав Володимирович та Глушак, Оксана Михайлівна (2025) Оптимізація customer support за допомогою AI чат-ботів: практичний кейс Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 4 (28). с. 727-739. ISSN 2663-4023 <https://doi.org/10.28925/2663-4023.2025.28.838>
28. Skladannyi, Pavlo та Mashkina, Iryna та Rzaeva, Svitlana та Kostiuk, Yuliia (2025) Methods of GDPR for Ensuring Data Storage Security Against Leaks and Threats Телекомунікаційні та інформаційні технології, 87 (2). с. 59-76. ISSN 2412-4338 <https://doi.org/10.31673/2412-4338.2025.027860>
29. Marisyk, S. та Matselyuk, Y. та Charny, D. та Zabulonov, Y. та Nosenko, Tetiana та Pugach, O. та Rudoman, M. (2024) Application of the Latest

- Design of Combined Adsorber-Settler Structure in the Purification (Deactivation) of Liquid Radioactive Wastes (LRW) Книжкова серія. с. 137-145. ISSN 2366-2557
30. Бушма, Олександр Володимирович та Турукало, Андрій Валерійович (2022) Оцінка параметрів програмної реалізації шкального відображення даних Cybersecurity: Education, Science, Technique, 4 (16). с. 142-158. ISSN 2663-4023 <https://doi.org/10.28925/2663-4023>
 31. Бушма, Олександр Володимирович та Турукало, Андрій Валерійович (2021) Багатоелементні шкальні індикаторні пристрої у вбудованих системах Кібербезпека: освіта, наука, техніка, 3 (11). с. 43-60. ISSN 2663-4023 <https://doi.org/10.28925/2663-4023>
 32. Бушма, Олександр Володимирович та Абрамов, Вадим Олексійович (2022) Підвищення достовірності визначення концентрації газів в середовищі моніторингу In: III Міжнародна науково-практична конференція: "Інформаційні технології та цифрова економіка" III International scientific and practical conference: "Information technologies and digital economy", 19-20 april 2022, Kyiv. <https://elibrary.kubg.edu.ua/id/eprint/41648/>
 33. Білоус, Владислав та Бодненко, Дмитро та Локазюк, Олександра та Складаний, Павло та Абрамов, Вадим (2025) Програмне забезпечення для кібердоказів як інструмент цифрової криміналістики у розслідуванні кіберзлочинів. Забезпечення кібербезпеки в інформаційно-телекомунікаційних системах. 2025 (3991). С. 26-37. ISSN 1613-0073 <https://ceur-ws.org/Vol-3991/>
 34. Білоус, Владислав Володимирович та Бодненко, Дмитро Миколайович та Хохлов, Олексій Костянтинівич та Локазюк, Олександра Вікторівна та Стаднік, Ірина Петрівна (2024) Open Source Intelligence for War Crime Documentation Workshop Cybersecurity Providing in Information and Telecommunication Systems

(CPITS 2024) (3654). с. 368-375. ISSN 1613-0073 :
<https://ceur-ws.org/Vol-3654/short3.pdf>

35. Андрій Петрович Бондарчук, Ірина Юріївна Мельник, Євгеній Іванович Суханевич, Вадим Олексійович Абрамов Підвищення ефективності систем відеоспостереження за рахунок гібридного методу відбору ключових кадрів і інтерпретації рішень Телекомунікаційні та інформаційні технології 2025 , №3 с101-105
<https://doi.org/10.31673/2412-4338.2025.038710>
36. PRINCIPE, Vitor; RIBEIRO, Tiago; LÓPEZ-CARRIL, Samuel. Transformative blockchain technological approaches to sports events. *Frontiers in Sports and Active Living*, 2025, 7: 1547137.
37. Кучаковська, Г., & Хорольська, К. (2025). Інтеграція Lean Canvas у підготовку IT-фахівців як механізму когнітивного спрощення та раціоналізації складних рішень у життєвому циклі IT-продукту. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2(30), 305-315.