

# Додаток А

## Результат перевірки роботи на плагіат



Дата звіту 6/1/2025  
Дата редагування ---



Звіт не був оцінений

### Звіт подібності

#### метадані

Назва організації

**Kharkiv National University of Radio Electronics**

Заголовок

**2025\_Б\_ПІ\_ПЗПІ-21-4\_Галушка\_М\_В\_скорочений**

Автор

Науковий керівник / Експерт

**Галушка Максим Віталійович Євген Кардаш**

підрозділ

**каф. ПІ**

#### Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

2.60%

2.60%

КП 1

1.20%

1.20%

КЦ

25

Довжина фрази для коефіцієнта подібності 2

5261

Кількість слів

42211

Кількість символів

#### Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.


Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		2
Парафрази (SmartMarks)		11

#### Подібності за списком джерел


Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

## Додаток Б

## Слайди презентації



МІНІСТЕРСТВО  
ОСВІТИ І НАУКИ  
УКРАЇНИ




ХАРКІВСЬКИЙ  
НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ  
РАДІОЕЛЕКТРОНИКИ

## Програмна система для управління заходами

Виконав:  
ст. гр. ПЗП-21-4  
Галушка М.В.

Керівник:  
доцент кафедри ПП  
Ворочек О.Г.



10 червня 2025

## Мета роботи

Метою роботи є створення програмної системи для зручного управління заходами через веб-інтерфейс з підтримкою ролей користувачів. Система забезпечує швидкий доступ до інформації про події, автоматизує рутинні процеси: реєстрація, повідомлення та використовує сучасний стек технологій (React, Node.js, Express, PostgreSQL). Актуальність теми обумовлена зростанням потреби в онлайн-інструментах для організації подій, конференцій і семінарів.

## Аналіз проблеми

Платформа	Переваги	Недоліки
<b>Eventbrite</b>	<ul style="list-style-type: none"> <li>- Легке створення подій</li> <li>- Продаж квитків</li> <li>- Email-маркетинг</li> </ul>	<ul style="list-style-type: none"> <li>- Обмежена кастомізація</li> <li>- Складність інтеграції з іншими системами</li> </ul>
<b>Ticketmaster</b>	<ul style="list-style-type: none"> <li>- Надійна система продажу квитків</li> <li>- Підтримка масових заходів</li> <li>- Розширені платіжні опції</li> </ul>	<ul style="list-style-type: none"> <li>- Обмежене управління подіями</li> <li>- Не підходить для невеликих чи внутрішніх подій</li> </ul>
<b>Bizzabo</b>	<ul style="list-style-type: none"> <li>- Потужна аналітика</li> <li>- Підтримка корпоративних мережеских заходів</li> <li>- Багатофункціональність</li> </ul>	<ul style="list-style-type: none"> <li>- Висока вартість</li> <li>- Складність використання для малого/середнього бізнесу</li> </ul>

## Постановка задачі та опис системи

Розробити веб-програму для ефективного управління подіями, яка дозволяє створювати, редагувати, видаляти та переглядати заходи з підтримкою різних ролей користувачів з відповідними правами доступу.

Система має забезпечувати централізоване зберігання даних про події та користувачів, надавати зручний та інтуїтивно зрозумілий інтерфейс, автоматизувати ключові процеси та сприяти ефективній організації як онлайн, так і офлайн-заходів.

## Вибір технологій розробки

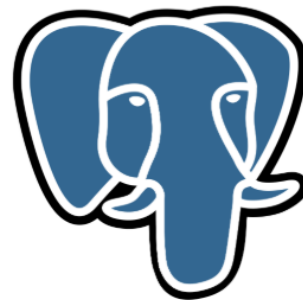
**NestJS**



**Next.js**

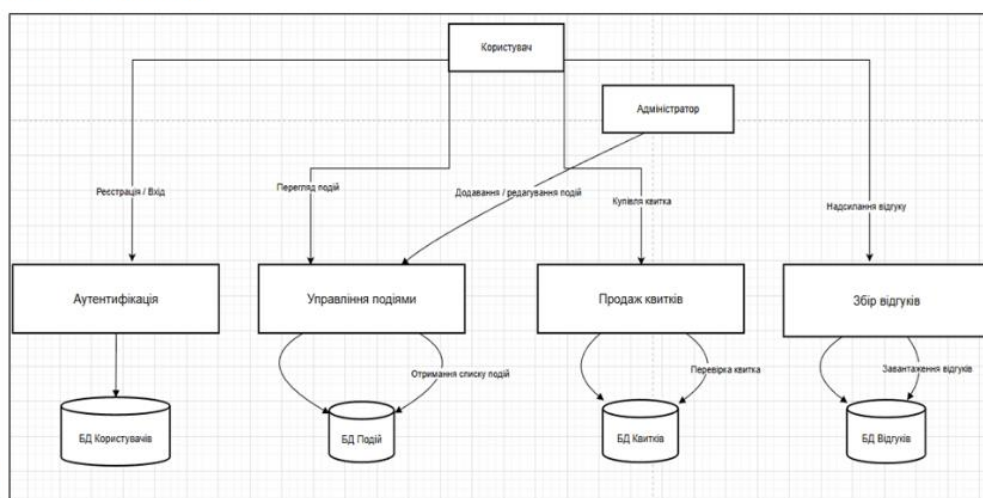


**PostgreSQL**



5

## Архітектура створеного програмного забезпечення



6

## Опис програмного забезпечення, що було використано у дослідженні

Розробка велася за клієнт-серверною архітектурою: спочатку створено структуру БД, потім реалізовано серверну логіку (авторизація, обробка запитів, керування подіями), і на завершення – адаптивний фронтенд. Використані технології: TypeScript (фронтенд/сервер), NestJS (REST API), TypeORM (PostgreSQL), Next.js (фронтенд), PostgreSQL (БД). Інструменти: GitHub (контроль версій), Visual Studio Code (IDE).



## Дизайн системи

### Методи:

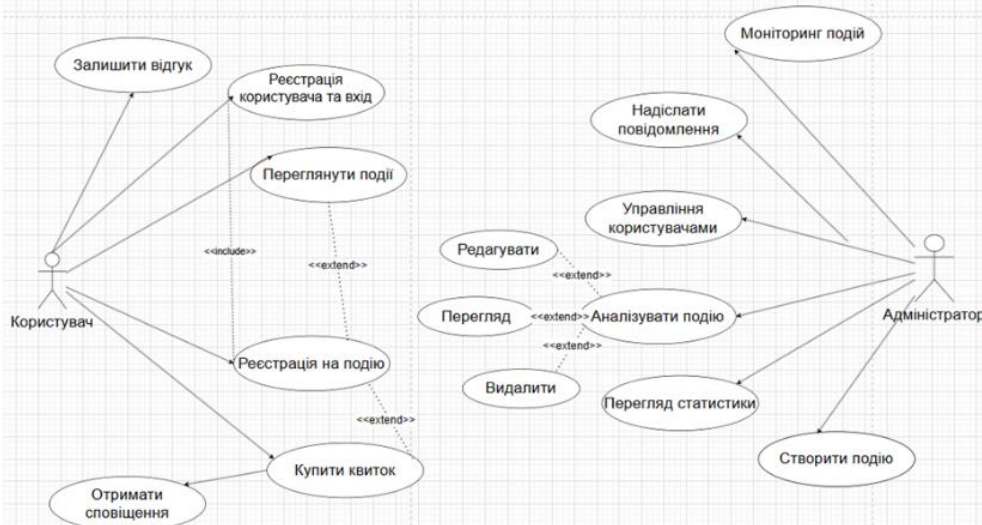
- компонентно-орієнтований підхід до створення інтерфейсу;
- REST-архітектура для взаємодії між фронтендом і бекендом;
- реляційна модель для зберігання даних;

### Послідовність розробки:

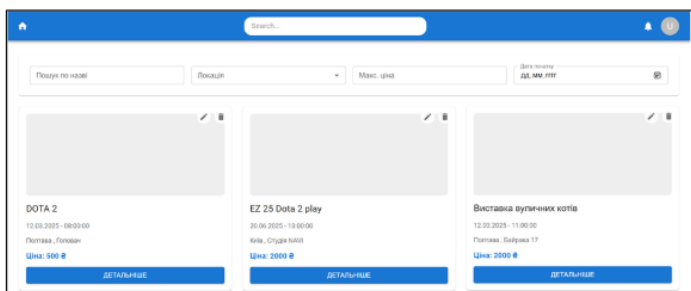
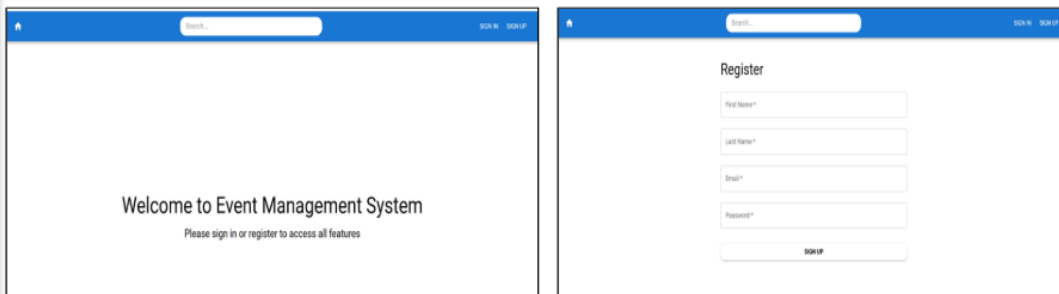
- проєктування бази даних та визначення ролей користувачів;
- реалізація API на NestJS (бекенд);
- розробка клієнтської частини на Next.js (фронтенд);
- інтеграція системи авторизації та сповіщень;
- тестування та налаштування адаптивного дизайну;



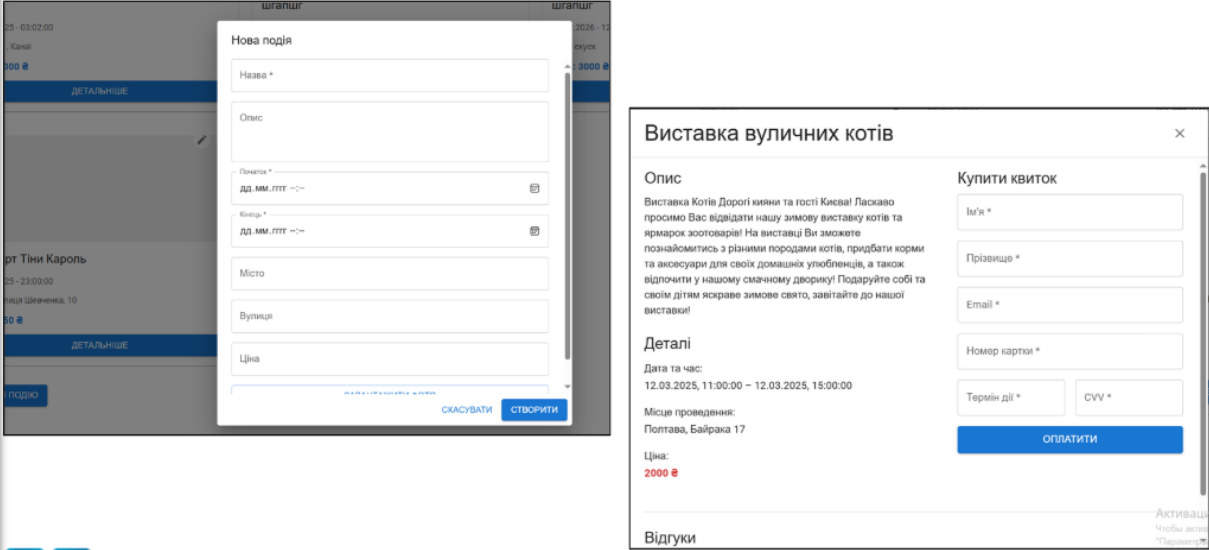
# Приклад реалізації



# Інтерфейс користувача



## Інтерфейс користувача



The image displays two screenshots of a web application interface. The left screenshot shows a 'Нова подія' (New Event) form with fields for Name, Description, Start/End dates, Location, and Price. The right screenshot shows the details for an event titled 'Виставка вуличних котів' (Street Cat Exhibition), including a description, dates, location, and a 'Купити квиток' (Buy Ticket) form with fields for Name, Surname, Email, Card Number, and CVV. A price of 2000 ₴ is displayed. The SE software logo is visible in the bottom left corner of the interface.

11

## Тестування

**Тест-кейс №1: Створення нового заходу авторизованим користувачем**

**Мета:** Перевірити коректність функції створення заходу.

**Передумови:**

1. Відкрити веб-застосунок. (Пройдено)
2. Авторизуватися (логін/пароль) та бути перенаправленим на головну сторінку. (Пройдено)

**Кроки тестування:**

1. Натиснути «Створити захід» → відкривається форма. (Пройдено)
2. Ввести назву, опис, дату/час, вартість, місто та адресу. (Всі поля коректно заповнюються та перевіряються). (Пройдено)
3. Натиснути «Зберегти» → подія успішно створена, з'являється повідомлення. (Пройдено)
4. Перейти до перегляду події через «Докладніше» → сторінка події відображає введені дані. (Пройдено)

**Результат тестування:** ПРОЙДЕНО (Р) **Тестувальник:** Галушка М.В. **Дата:** 28.05.2025



12

## Тестування

**Тест-кейс №2: Придбання квитка на захід**

**Мета:** Перевірити успішне придбання квитка авторизованим користувачем.

**Передумови:**

1. Користувач має обліковий запис. (Пройдено)

**Кроки тестування:**

1. Увійти в систему. (Пройдено)
2. Перейти на головну сторінку заходів. (Пройдено)
3. Вибрати захід зі списку та натиснути «Детальніше». (Пройдено)
4. Відкривається форма покупки квитка. Заповнити поля: ім'я, прізвище, email, номер картки. (Пройдено)
5. Підтвердити оплату → з'являється повідомлення про успішну оплату. (Пройдено)
6. Придбаний квиток присутній у списку. (Пройдено)

**Результат тестування:** ПРОЙДЕНО (P) **Тестувальник:** Галушка М.В. **Дата:** 29.05.2025



## Підсумки

Кваліфікаційна робота дозволила отримати практичний досвід розробки повнофункціонального клієнт-серверного веб-застосунку для управління подіями та продажу квитків. Було реалізовано зручний інтерфейс користувача та надійна серверна логіка з використанням сучасних технологій. Це закріпило теоретичні знання та покращило розуміння повного циклу розробки веб-систем.

