

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко

підпис

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
ВЕБЗАСТОСУНОК БРОНЮВАННЯ КВИТКІВ НА ЗАХОДИ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.22010921

Здобувач

_____ А. В. Тебенко

підпис

«__» _____ 2024 р.

Керівник PhD, ст. викл. кафедри ІПЗ

_____ К. О. Антіпова

підпис

«__» _____ 2024 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва

підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко

«_____» _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано здобувачу групи 409 факультету комп'ютерних наук

_____ Тебенко Андрію Васильовичу _____

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Вебзастосунок бронювання квитків на заходи

Затверджена наказом по ЧНУ від «22» _____ грудня _____ 2023 р. № _____ 269 _____

2. Строк представлення кваліфікаційної роботи «_____» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є вебзастосунок бронювання квитків на заходи

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до програмного забезпечення;
- визначення архітектури для проектування програмного забезпечення;
- моделювання та проектування програмного забезпечення;
- розробка програмного забезпечення;
- здійснення тестування роботи програмного забезпечення

5. Перелік графічних матеріалів

Презентація

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
А. О. Алексеєва	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи PhD, ст. викл. кафедри ІПЗ Антіпова Катерина Олександрівна

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Тебенко Андрій Васильович

(прізвище, ім'я, по батькові)

(підпис)

Дата видачі завдання « ____ » _____ 2024р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

Тема: Вебзастосунок бронювання квитків на заходи

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	22.12.2023	24.12.2023	виконано
2.	Огляд літератури за темою роботи	26.12.2023	02.01.2024	виконано
3.	Складання календарного плану КРБ	14.02.2024	15.02.2024	виконано
4.	Аналіз предметної області	20.02.2024	25.02.2024	виконано
5.	Розробка проєктних рішень	05.03.2024	12.03.2024	виконано
6.	Моделювання та конструювання ПЗ	15.03.2024	20.03.2024	виконано
7.	Кодування проєкту	23.03.2024	20.04.2024	виконано
8.	Розробка частини з охорони праці	22.04.2024	28.04.2024	виконано
9.	Відгук керівника КРБ	03.05.2024	04.05.2024	виконано
10.	Оформлення КРБ та презентації	05.05.2024	24.05.2024	виконано
11.	Попередній захист	03.06.2024	03.06.2024	виконано
12.	Рецензування	04.06.2024	20.06.2024	виконано
13.	Завершення оформлення КРБ та презентації	20.06.2024	24.06.2024	виконано
14.	Захист кваліфікаційної роботи	26.06.2024	26.06.2024	виконано

Розробив здобувач Тебенко Андрій Васильович _____
(прізвище, ім'я, по батькові) (підпис)
«__» _____ 2024 р.

Керівник роботи PhD, ст. викл. кафедри ПЗ Антіпова Катерина Олександрівна _____
(прізвище, ім'я, по батькові) (підпис)
«__» _____ 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок бронювання квитків на заходи»

Здобувач 409 групи: Тебенко Андрій Васильович

Керівник: PhD, ст. викл. кафедри ІІЗ Антіпова К. О.

Дана робота присвячена розробці вебзастосунку для зручного бронювання квитків на заходи. Розробка вебзастосунку для бронювання квитків на заходи є актуальною задачею в умовах сучасного цифрового світу. Вебзастосунок дозволяє користувачам швидко та зручно обирати бажані заходи, переглядати доступні місця, а також здійснювати бронювання. Це не лише покращує користувацький досвід, але й сприяє збільшенню кількості заброньованих квитків. Вебзастосунок може сприяти підвищенню ефективності організації заходів, забезпечуючи легкість доступу та кращий контроль над процесом управління бронюванням квитків.

Об'єкт кваліфікаційної роботи – процес бронювання квитків на розважальні заходи.

Предмет кваліфікаційної роботи – інструментарій для розробки вебзастосунку.

Метою кваліфікаційної роботи є розробка вебзастосунку для бронювання квитків на заходи з метою забезпечення швидкого та зручного процесу бронювання.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз предметної сфери та аналогічних застосунків;
- проєктування та моделювання системи;
- програмна реалізація вебзастосунку;
- тестування та оптимізація вебзастосунку.

У вступі визначається актуальність теми, мета, предмет, об'єкт дослідження та завдання.

У першому розділі розглянуто аналіз предметної області та проаналізовано існуючі застосунки-аналоги. Було розглянуто актуальність теми кваліфікаційної роботи, об'єкт, предмет та мету. Розроблено специфікацію вимог, у які зазначено призначення та межі проєкту, загальний опис програмного забезпечення, функції системи, вимоги до інформаційного забезпечення, вимоги до технічного забезпечення, вимоги до програмного забезпечення, вимоги до зовнішніх інтерфейсів, властивості програмного забезпечення.

У другому розділі виконано проєктування та моделювання вебзастосунка. Описано короткий, середній та повний сценарій вебзастосунка. Створено діаграму варіантів використання, станів вебзастосунка та діаграму класів. Ознайомлено з технологіями, які використовувались при створенні вебзастосунка бронювання квитків на захід.

У третьому розділі розглянуто процес розробки вебзастосунка з використанням WordPress плагіну для ефективного бронювання квитків на різноманітні заходи, зокрема реалізовано функціонал для забезпечення зручного вибору кількості квитків, отримання необхідної інформації про ціни та доступні місця, а також створено відповідні сторінки, які включають обліковий запис, теми заходів, перелік подій, інформацію про організаторів, а також контактну форму.

У четвертому розділі було детально розглянуто процес тестування та оптимізації вебзастосунка. Продемонстровано три онлайн-інструмента для перевірки швидкості роботи веб-сайту. Розглянуто вебзастосунок на різних розширеннях екрану. Оптимізовано вебзастосунок завдяки плагіну WP-Optimize, та розглянуто переваги використання даного плагіну.

У висновках проводиться аналіз виконаних робіт та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 61 сторінку, містить 4 розділи, 53 ілюстрацій, 1 таблицю, 20 джерел в переліку посилань.

Ключові слова: *створення вебзастосунку, база даних, адаптивний інтерфейс, розробка на CMS WordPress, бронювання квитків.*

ABSTRACT

of the Bachelor's Thesis

«Web application for booking event tickets»

Student: Andrii Tebenko

Supervisor: PhD, senior Lecturer Katerina Antipova

This work is devoted to the development of a web application for convenient reservation of tickets for events. Development of a web application for booking event tickets is an urgent task in today's digital world. The web application allows users to quickly and conveniently choose the desired activities, view available places, and make reservations. This not only improves the user experience, but also helps increase the number of tickets booked. A web application can help increase the efficiency of event management by providing ease of access and better control over the ticket booking management process.

The object of qualification work is the process of booking tickets for entertainment events.

The subject of the qualification work is a toolkit for developing a web application.

The purpose of the qualification work is to develop a web application for booking event tickets in order to ensure a quick and convenient booking process.

The introduction defines the relevance of the topic, the purpose, subject, and object of the research, and the tasks are defined.

To achieve the goal, the following tasks must be completed:

- analysis of the subject area and similar applications;
- system design and modeling;
- software implementation of the web application;
- testing and optimization of the web application.

In the first section, the analysis of the subject area was considered and the existing analog applications were analyzed. The relevance of the topic of the qualification work, object, subject and purpose were considered. A requirements specification has been

developed, which specifies the purpose and boundaries of the project, a general description of the software, system functions, requirements for information support, requirements for technical support, requirements for software, requirements for external interfaces, and software properties.

In the second section, the design and modeling of the web application is performed. Short, medium and full web application script is described. Created use case diagram, web application states and class diagram. The technologies used in the creation of the web application for booking tickets for the event were introduced.

In the third section, the process of developing a web application using a WordPress plugin for the efficient booking of tickets for various events is considered, in particular, the functionality is implemented to ensure convenient selection of the number of tickets, obtaining the necessary information about prices and available seats, as well as the corresponding pages are created, which include the account, topics events, a list of events, information about the organizers, as well as a contact form.

In the fourth chapter, the process of testing and optimizing the web application was considered in detail. Three online tools for testing website speed are demonstrated. The web application is considered on different screen extensions. The web application was optimized thanks to the WP-Optimize plugin, and the advantages of using this plugin were considered.

The conclusions analyze the work performed and the results obtained.

The bachelor's qualification work contains 61 pages, 4 chapters, 53 illustrations, 1 table, and 20 sources in the list of references.

Keywords: creation of a web application, database, adaptive interface, development on CMS WordPress, ticket booking.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Бронювання квитків на заходи.....	6
1.2 Огляд застосунків-аналогів	6
1.3 Специфікація вимог до програмного забезпечення.....	11
Висновки до розділу 1	15
2 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ	16
2.1 Створення сценаріїв.....	16
2.2 Створення діаграми варіантів використання	18
2.3 Створення діаграми станів вебзастосунку	20
2.4 Створення діаграми класів	21
2.5 Огляд технологій	22
2.5.1 Використання бібліотеки JQuery та фреймворку Bootstrap.....	23
2.5.2 Опис роботи з CMS WordPress.....	24
2.5.3 Проєктування та опис бази даних	27
Висновки до розділу 2	31
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКА.....	32
3.1. Створення діаграми розгортання	32
3.2. Підключення бази даних	33
3.3. Створення кастомних WordPress полів	34

3.4.	Використання плагіну Events Manager	37
3.5.	Створення сторінок	39
3.6.	Програмна реалізація	44
	Висновки до розділу 3	49
4	ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ВЕБЗАСТОСУНКА.....	50
4.1.	Тестування ПЗ.....	50
4.2.	Оптимізація вебзастосунок	55
	Висновки до розділу 4	58
	ВИСНОВКИ	59
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60

ПЕРЕЛІК СКОРОЧЕНЬ

БД	– База даних
ПЗ	– Програмне забезпечення
AJAX	– Asynchronous JavaScript And XML
API	– Application Programming Interface
CMS	– Content Management System
CSS	– Cascading Style Sheets
CVS	– Comma-Separated Values
DOM	– Document Object Model
HTML	– HyperText Markup Language
JPEG	– Joint Photographic Experts Group
PHP	– Hypertext Preprocessor
PNG	– Portable Network Graphics
SQL	– Structured Query Language
SSL	– Secure Sockets Layer
UML	– Unified Modeling Language
URL	– Uniform Resource Locator
WebP	– WEB Pictures
WYSIWYG	– What You See Is What You Get

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій та помітний зріст популярності розважальних та культурних подій створюють потребу у вдосконаленні процесу бронювання квитків на заходи через вебзастосунки. Актуальність даної теми визначається необхідністю забезпечення швидкості, зручності та ефективності у взаємодії організаторів подій, агентств з продажу квитків та кінцевих користувачів. Вебзастосунок бронювання може сприяти підвищенню ефективності організації заходів, забезпечуючи легкість доступу та кращий контроль над процесом управління бронюванням квитків.

Розроблений вебзастосунок повинен забезпечувати швидкий та легкий доступ до квитків для організаторів подій та кінцевих користувачів.

Об'єктом кваліфікаційної роботи є процес бронювання квитків на розважальні заходи.

Предметом кваліфікаційної роботи є інструментарій для розробки вебзастосунку.

Метою кваліфікаційної роботи є розробка вебзастосунка для бронювання квитків на заходи з метою забезпечення швидкого та зручного процесу бронювання.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз предметної сфери та аналогічних застосунків;
- проектування та моделювання системи;
- програмна реалізація вебзастосунка;
- тестування та оптимізація вебзастосунка.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Бронювання квитків на заходи

Бронювання квитків через Інтернет стало необхідністю у сучасному світі зі швидким розвитком технологій та цифровою трансформацією у всіх сферах життя. Цей процес забезпечує низку переваг, які роблять його важливою складовою частиною планування розважальних заходів або інших подій. Великий вибір доступних опцій є ще однією перевагою онлайн-бронювання. Застосунок повинен мати інтуїтивно зрозумілий і простий інтерфейс, який дозволяє користувачам легко знаходити потрібні події або бронювати потрібну кількість місць. Застосунок має надавати докладну інформацію про кожну подію, включаючи час, місце проведення, вартість квитків та опис. Користувачі повинні мати можливість переглядати та управляти своїми бронюваннями, включаючи скасування або зміну деталей, а також мати доступну підтримку для вирішення будь-яких питань або проблем з бронюванням. Застосунок повинен бути мобільною додатком або мати адаптивний веб-дизайн для зручного доступу з мобільних пристроїв. Платформи для бронювання квитків зазвичай пропонують широкий спектр варіантів заходів, цін, що дозволяє користувачам знайти оптимальний варіант, враховуючи їхні захоплення та бюджет.

Крім того, багато сервісів надають різноманітні знижки та пропозиції, які можуть значно зберегти кошти. Це стимулює користувачів шукати найвигідніші умови та зробити свій вибір відповідно до цього. Користувачі можуть легко користуватися бронюванням на захід за допомогою онлайн-платформ, що робить процес управління заходом більш зручним та ефективним.

1.2 Огляд застосунків-аналогів

Огляд застосунків-аналогів є ключовим етапом у розгляді програмного забезпечення. Це дозволяє визначити наявність на ринку програмних рішень, що

можуть задовольнити потреби користувачів, або відіграти роль основи для створення нового програмного продукту. Крім того, такий аналіз допомагає у визначенні критеріїв функціональності, інтерфейсу користувача, швидкодії, масштабованості та інших параметрів. Було розглянуто вебзастосунки Concert.ua, Karabas.com, Kontramarka.ua для виявлення їх основних функцій та недоліків, що необхідно врахувати для покращення створеного веб-застосунку.

Назва: Concert ua (рис. 1.1).

Архітектура: client-server.

Функції:

1. бронювання квитків на захід;
2. пошук та фільтрація;
3. візуалізація місць;
4. система онлайн платежів;
5. електронні та мобільні білети.

Переваги:

1. система знижок та промокодів;
2. підтримка клієнтів;
3. візуалізація місць;
4. опції цін (вір місця, стоячі місця, тощо);
5. мультимовність.

Недоліки:

1. труднощі з поверненням та обміном;
2. відсутність попереднього бронювання квитків;
3. неефективна підтримка.

<https://concert.ua/>

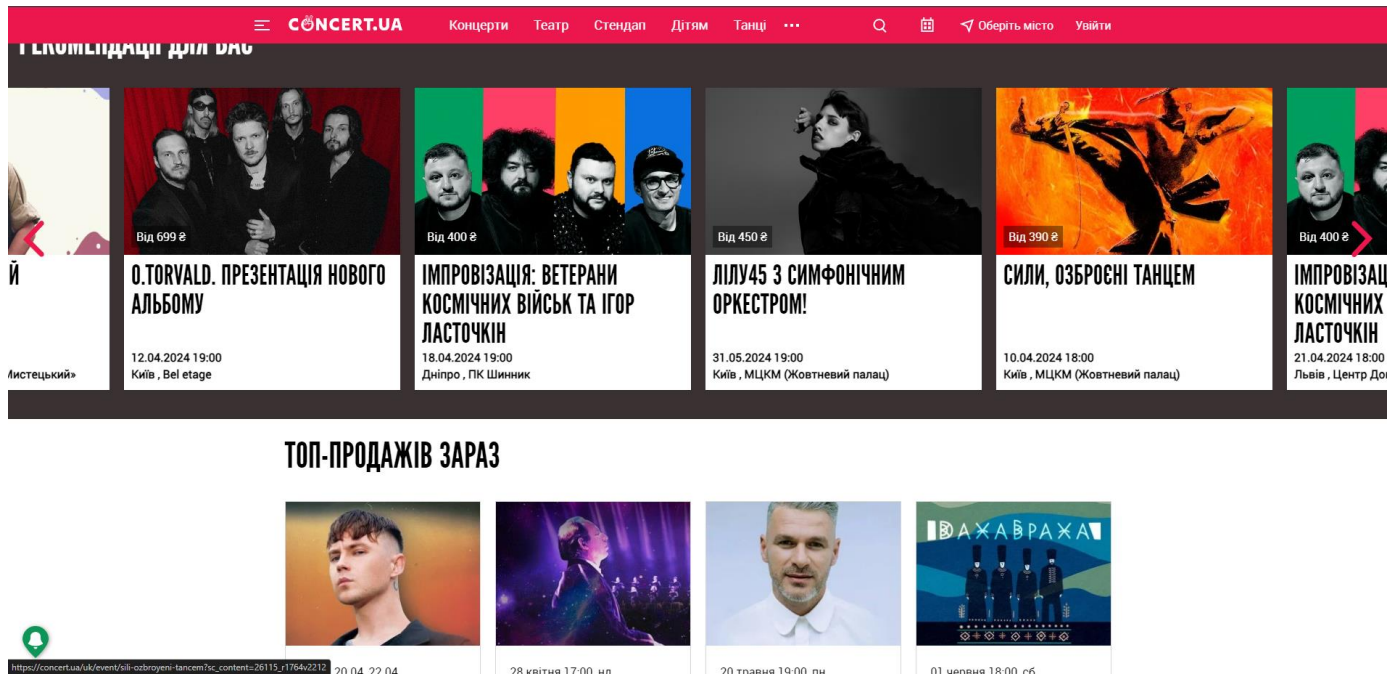


Рисунок 1.1 – Веб-застосунок Concert.ua

Назва: Karabas (рис. 1.2).

Архітектура: client-server.

Функції:

1. бронювання квитків на захід;
2. пошук та фільтрація;
3. візуалізація місць;
4. система онлайн платежів;
5. електронні та мобільні білети.

Переваги:

1. головна сторінка, з найпопулярнішими новинами;
2. стрічка останніх подій;
3. наявність карти сайту;
4. мультимовність.

Недоліки:

1. труднощі з поверненням та обміном;
2. затримка у відправленні квитків на пошту.

<https://karabas.com/>

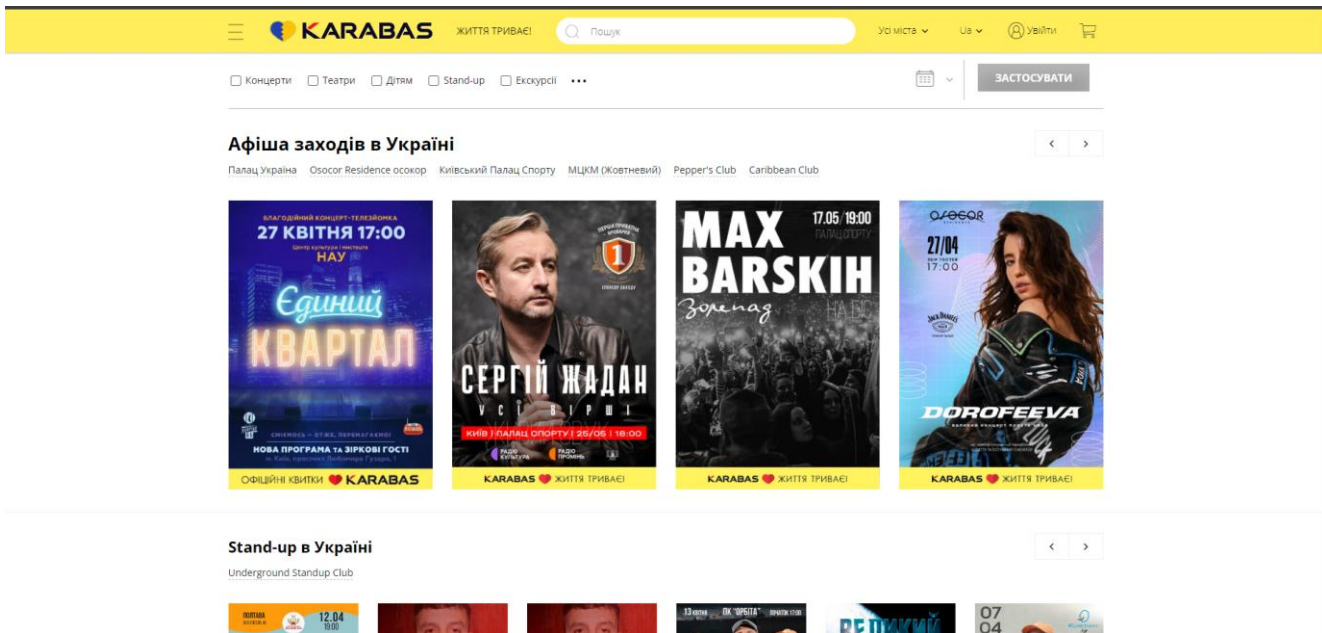


Рисунок 1.2 – Веб-застосунок Karabas.com

Назва: Kontramarka (рис. 1.3).

Архітектура: client-server.

Функції:

1. бронювання квитків;
2. електронні та мобільні білети;
3. пошук та фільтрація;
4. система онлайн платежів;
5. актуальна інформація про події.

Переваги:

1. мультимовність;
2. календар з подіями;
3. поле пошуку інформації.

Недоліки:

1. повільне завантаження сторінок вебзастосунку;
2. помилка при створенні облікового запису;
3. конфлікт кольорів у вебзастосунку (поєднання синього та червоного кольору).

<https://kontramarka.ua>

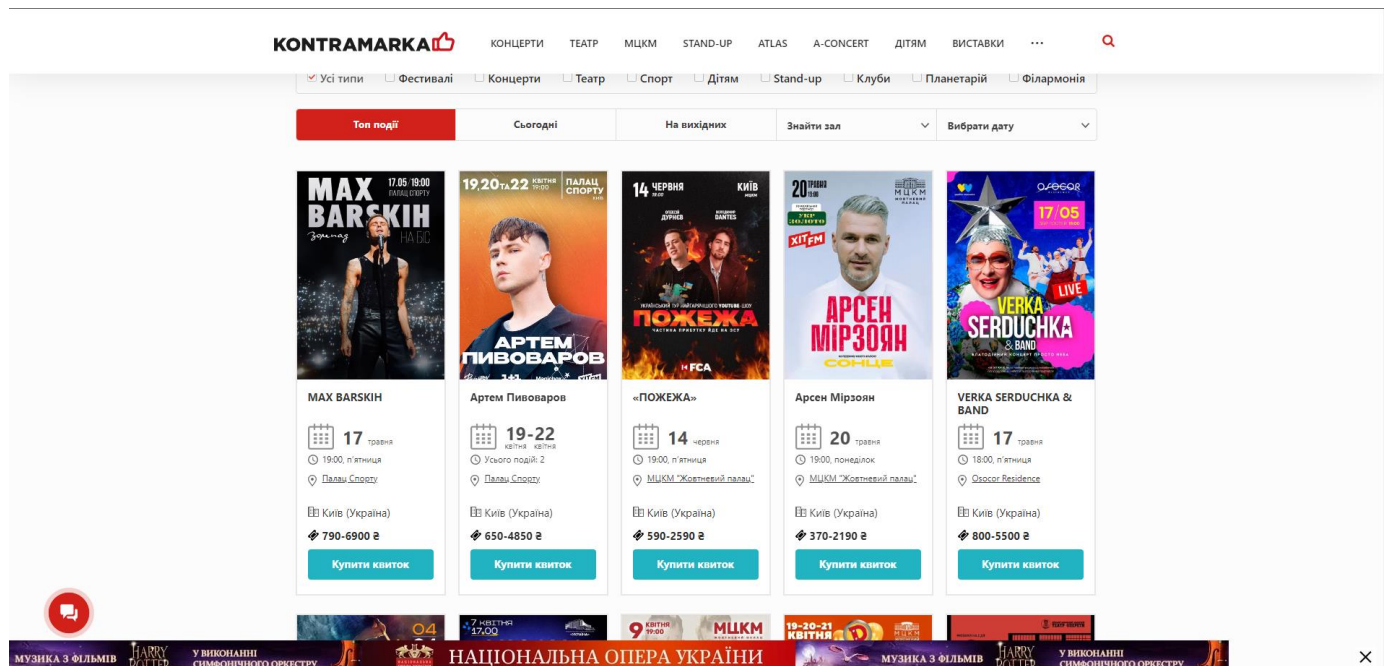


Рисунок 1.3 – Веб-застосунок Kontramarka.ua

Фактори, на які звернено особливу увагу при бронюванні квитків на заходи:

1. простота використання веб-застосунку;
2. швидке бронювання квитка;
3. актуальність інформації про заходи;
4. доступність використання на різних платформах;
5. якість обслуговування клієнтів і підтримка користувачів;
6. вартість квитків і доступність знижок.

1.3 Специфікація вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи (застосунку), для якого розробляється ПЗ

Призначенням застосунку є швидкий та легкий доступ для бронювання квитків на заходи шляхом розробленого вебзастосунка.

Погодження, що ухвалені в програмній документації

Було погоджено, що для створення загального ПЗ буде використовуватись CMS WordPress.

Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 15.05.2024р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування:

Вебзастосунок призначений для бронювання квитків на заходи.

Характеристики користувачів

Основні характеристики користувачів: наявність смартфона, планшету або ПК та доступу до мережі Інтернет.

Загальна структура та склад системи

Система складається з наступних частин:

1. Клієнтська Front-end частина:
 - дизайн та інтерфейс вебзастосунку;
 - програмна логіка (JavaScript, JQuery).
2. Серверна Back-end частина:
 - база даних;
 - серверна логіка (PHP);
 - інтерфейс взаємодії з базою даних MySQL.
3. Адміністративна панель:

- використання CMS WordPress.
4. База даних:
- взаємодія з базою данихMySQL.

Загальні обмеження

Єдине обмеження для роботи з ПЗ – наявність доступу до мережі Інтернет.

ФУНКЦІЇ СИСТЕМИ

Бронювання квитків на захід

Опис функції

Функція дозволяє адміністратору надавати актуальну інформацію користувачеві про наявність квитків на заходи.

Вхідна і вихідна інформація

Вхідна інформація – запит користувача про наявність квитків на заходи.

Вихідна інформація – інформація про доступні квитки на вказані заходи, включаючи кількість доступних квитків, ціни та інші відомості.

Функціональні вимоги

1. Реєстрація та авторизація

Ця функція дозволяє користувачам створювати обліковий запис на платформі та увійти в систему для доступу до персонального кабінету.

Вхідні дані:

- для реєстрації: ім'я, електронна пошта, пароль;
- для авторизації: електронна пошта, пароль.

Вихідні дані:

- успішна реєстрація: підтвердження успішного створення облікового запису;
- успішна авторизація: доступ до особистого кабінету користувача.

2. Бронювання квитків з обранням кількості місць

Ця функція дозволяє користувачам обирати подію, кількість та тип місць, які вони хочуть забронювати.

Вхідні дані:

- вибір події: ідентифікатор або назва події;
- кількість та тип місць, які користувач хоче забронювати (наприклад, стандартний, VIP).

Вихідні дані:

- підтвердження бронювання: підтвердження успішного бронювання квитків;
- запис бронювання в базу даних: інформація про заброньовані квитки зберігається в системі.

3. Перегляд статистики продажу квитків в адмін-панелі

Ця функція дозволяє адміністраторам переглядати статистику продажів квитків для аналізу популярності та ефективності продажів.

Вхідні дані:

- діапазон дат для аналізу (наприклад, за певний період часу);
- вибір конкретної події або всіх подій.

Вихідні дані:

- відображення статистики продажів: інформація про кількість проданих квитків, загальний дохід за обраний період.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації

В цьому вебзастосунку вхідні дані отримуються від адміністратора, що заповнює дані використовуючи CMS WordPress.

Нормативно-довідникова інформація (класифікатори, довідники тощо)

Вимоги до даного пункту відсутні.

Вимоги до способів організації, збереження та ведення інформації

Обмін даними відбувається через обробку запитів, використовуючи БД MySQL.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Для розробки програмного забезпечення немає значних технічних обмежень.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Система складається з: клієнтської частини, серверної частини та БД.

Системне програмне забезпечення

Використання операційної системи Windows 7,10,11 чи Linux.

Програмне забезпечення ведення інформаційної бази

Використано БД MySQL.

Мова і технологія розробки ПЗ

Програмне забезпечення має розроблюватися з використанням CMS WordPress.

Мови розробки – PHP та JavaScript.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс повинен бути легко зрозумілим для користувача завдяки застосуванню стандартних рішень при верстці.

Апаратний інтерфейс

Програмний застосунок має бути адаптивним для всіх пристроїв.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Можливість бронювання квитків доступна для будь-якого користувача, що має бажання придбати квиток на захід чи подію, за умови наявності пристрою з Інтернетом та веб-браузером.

Супроводжуваність

Забезпечення функціональності веб-сайту для бронювання квитків, усунення технічних помилок та відповідь на запитання користувачів. Надання допомоги

користувачам щодо процесу бронювання, відповіді на запитання через електронну пошту.

Переносимість

Застосунок доступний на будь-якому пристрої.

Продуктивність

Залежить від Інтернет з'єднання.

Надійність

Хешований пароль в базі даних.

Безпека

Лише адміністратор має доступ до редагування.

Висновки до розділу 1

В першому розділі кваліфікаційної роботи бакалавра було розглянуто аналіз предметної області та проаналізовано існуючі застосунки-аналоги. Було ознайомлено з актуальністю теми кваліфікаційної роботи, об'єктом, предметом та метою. Розроблено специфікацію вимог, у які зазначено призначення та межі проєкту, загальний опис ПЗ, функції системи, вимоги до інформаційного забезпечення, вимоги до технічного забезпечення, вимоги до програмного забезпечення, вимоги до зовнішніх інтерфейсів, властивості програмного забезпечення.

2 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ

2.1 Створення сценаріїв

Use Case сценарій – це спосіб опису функціональності системи, через опис окремих сценаріїв використання або взаємодії користувачів з системою [6]. Це важлива частина процесу розробки програмного забезпечення, яка дозволяє розібратися в тому, як система має працювати з різними типами користувачів або в різних ситуаціях. При створенні вебзастосунка для бронювання квитків на захід, використання сценаріїв допомогло розібратися у функціональних вимогах і взаємодії з користувачами.

Короткий сценарій

Користувач відкриває вебзастосунок, обирає подію, вказує тип та кількість квитків. Система підтверджує бронювання, і користувач отримує підтвердження по електронній пошті або на сторінці з деталями події.

Середній сценарій

Головний сценарій (успішний):

Користувач відкриває вебзастосунок для бронювання квитків на захід. На головній сторінці він обирає бажану подію і переходить до її деталей, де вказує тип квитка (наприклад, VIP, стандарт) та вказує кількість квитків для бронювання. Користувач обирає всі необхідні параметри квитків. Система підтверджує вибір користувача і резервує обрані квитки. Користувач отримує підтвердження бронювання на екрані та через електронну пошту з деталями події та обраних квитків.

Альтернативний сценарій (недостатня кількість квитків):

Коли користувач обирає бажану подію і вказує кількість квитків для бронювання, система перевіряє наявність цієї кількості квитків. Якщо квитків недостатньо (наприклад, менше, ніж обрана кількість), система показує повідомлення

про те, що обраної кількості квитків немає в наявності. Користувач може повернутися до вибору кількості квитків або обрати інший захід для бронювання.

Повний сценарій

Таблиця 2.1 – Повний сценарій

Use Case Name	Бронювання квитків на захід.
Scope	Вебзастосунок для бронювання квитків на захід.
Primary Actor	Адміністратор.
Stakeholders and interests	<ul style="list-style-type: none"> – Користувач: зацікавлений у доступності і виборі квитків. – Адміністратор: зацікавлений у перегляді статистики продажів квитків. – Керівник проєкту: зацікавлений у продажі квитків на захід.
Preconditions	Доступність квитків, доступність місць, час та дата.
Main Success Scenario	<ul style="list-style-type: none"> – Після входу адміністратор опинився у адмін-панелі. – Перейшов до керування заходами. – Переходить на сторінку додавання заходу. – Вводить необхідну інформацію, та редагує вміст заходу. – Зберігає захід. – Переходить у режим додавання зображень. – Додає зображення та вводить опис заходу. – Додає зображення та вводить його опис. – Зберігає введену інформацію. – Виходить з адмін панелі.
Extensions	Не коректний формат зображення для WYSIWYG-редактору, тому захід не доданий.
Special Requirements	<ul style="list-style-type: none"> – Система має виконувати запити не довше 1 секунди. – Доступна підтримка у вебзастосунку.
Technology and Data Variations List	Для успішного виконання усіх кроків, необхідно мати стабільне інтернет-з'єднання.

2.2 Створення діаграми варіантів використання

Діаграма варіантів використання – це графічне зображення можливої взаємодії користувача з системою [1]. Діаграма варіантів використання показує різні варіанти використання та різні типи користувачів, які має система, і часто супроводжуватиметься діаграмами інших типів. Діаграма варіантів використання є потужним інструментом для візуалізації можливих способів взаємодії користувачів з системою. Вона дозволяє ідентифікувати різні сценарії використання, а також ролі та типи користувачів, які мають взаємодіяти з системою. Головними компонентами діаграми є актори (які зображуються у вигляді фігур, таких як люди або інші системи) та варіанти використання (представлені у вигляді еліпсів), які показують конкретні дії або функції, доступні для користувачів.

Діаграма варіантів використання дозволяє уточнити вимоги до системи та допомагає визначити, як система буде використовуватися на практиці. Дані діаграми часто використовуються на етапі аналізу вимог для визначення функціональності системи з погляду користувача. Вони допомагають команді розробників та зацікавленим сторонам отримати уявлення про те, як система повинна працювати в реальному світі і які задачі вона повинна вирішувати.

Крім того, діаграма варіантів використання може стати основою для подальшої розробки системи та визначення її архітектури. Загалом, використання діаграм варіантів використання сприяє кращому розумінню функціональних вимог до системи та сприяє успішному розробленню програмного забезпечення, яке відповідає потребам користувачів [16].

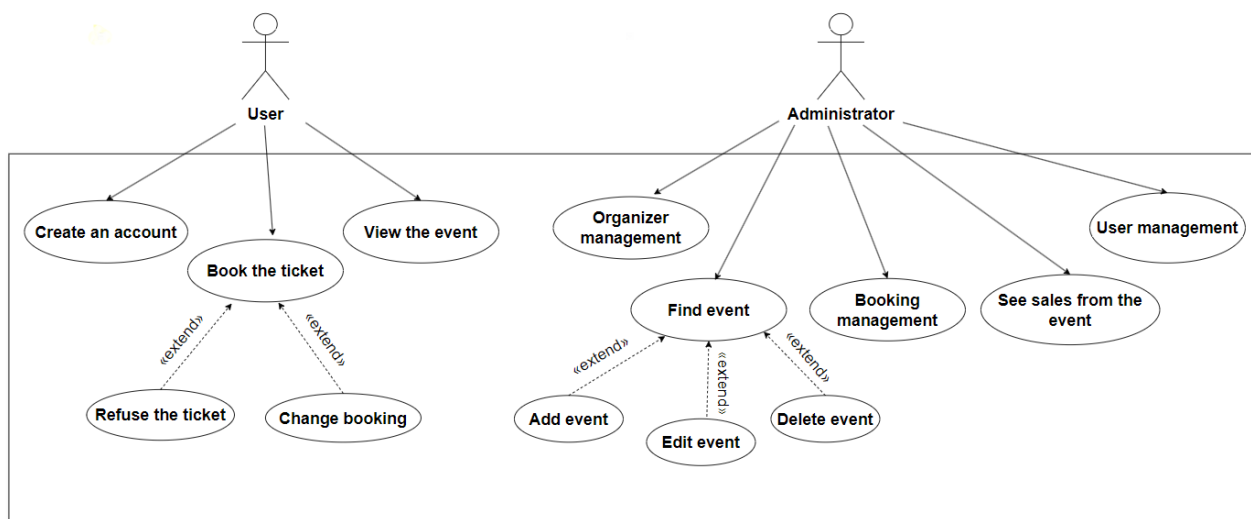


Рисунок 2.1 – Діаграма варіантів використання

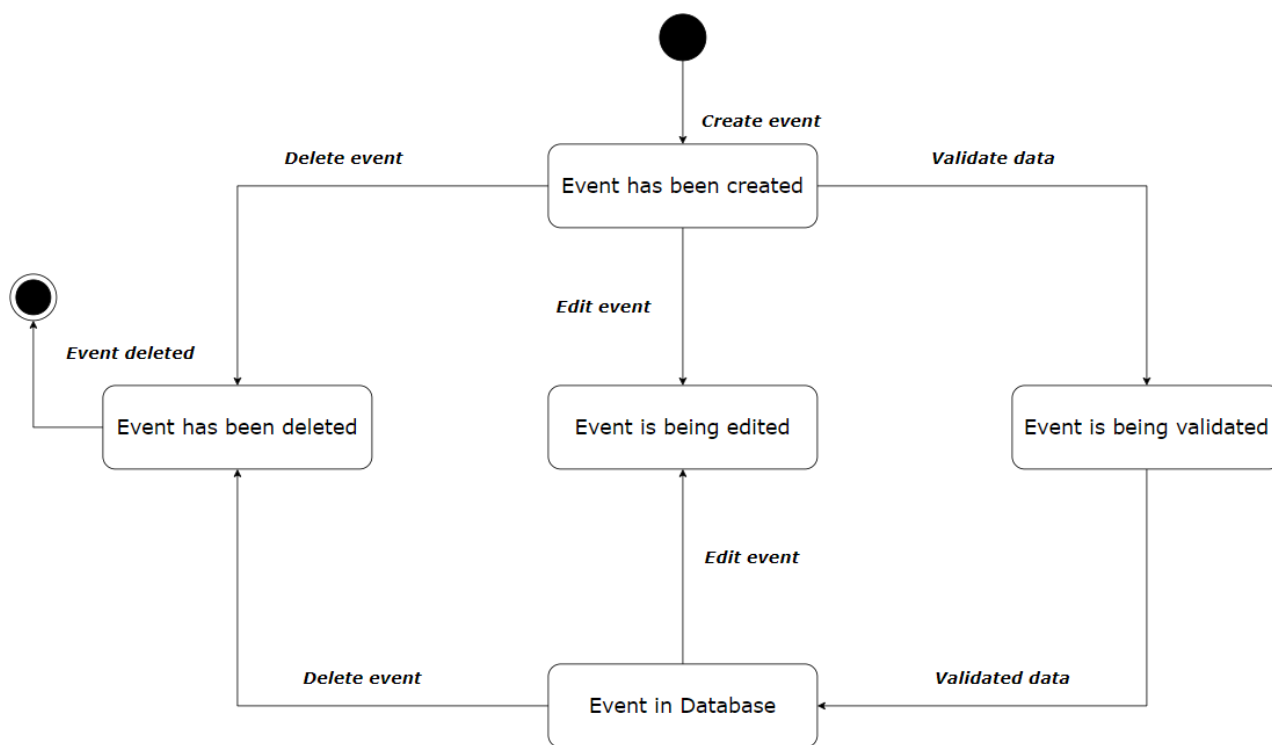


Рисунок 2.2 – Діаграма станів заходу

На рис. 2.1 можна переглянути діаграму варіантів використання проєкту та побачити усі функції користувача та адміністратора. Адміністратор маніпулює контентом на сторінці, додає або оновлює заходи, має доступ до облікового запису

користувача та доступ до бронювання квитків. Також адміністратор має доступ до перегляду та аналізу продажів квитків.

Відвідувач має змогу створити обліковий запис, переглянути доступні заходи, та забронювати квиток або відмовитись від бронювання у разі потреби.

2.3 Створення діаграми станів вебзастосунку

Діаграма станів і переходів демонструє, як об'єкти переходять з одного стану в інший. Ця діаграма може бути корисною для моделювання роботи програмного забезпечення, де станами можуть бути різні фази роботи програми (ініціалізація, очікування введення, обробка даних, вивід результатів тощо), а переходами - події, які спричиняють зміну цих станів (наприклад, натискання кнопки, отримання сигналу від зовнішнього джерела тощо). Використання діаграм станів і переходів допомагає виявити можливі недоліки в дизайні системи, спростити її розуміння та взаємодію з нею, а також полегшити налагодження та вдосконалення [5].

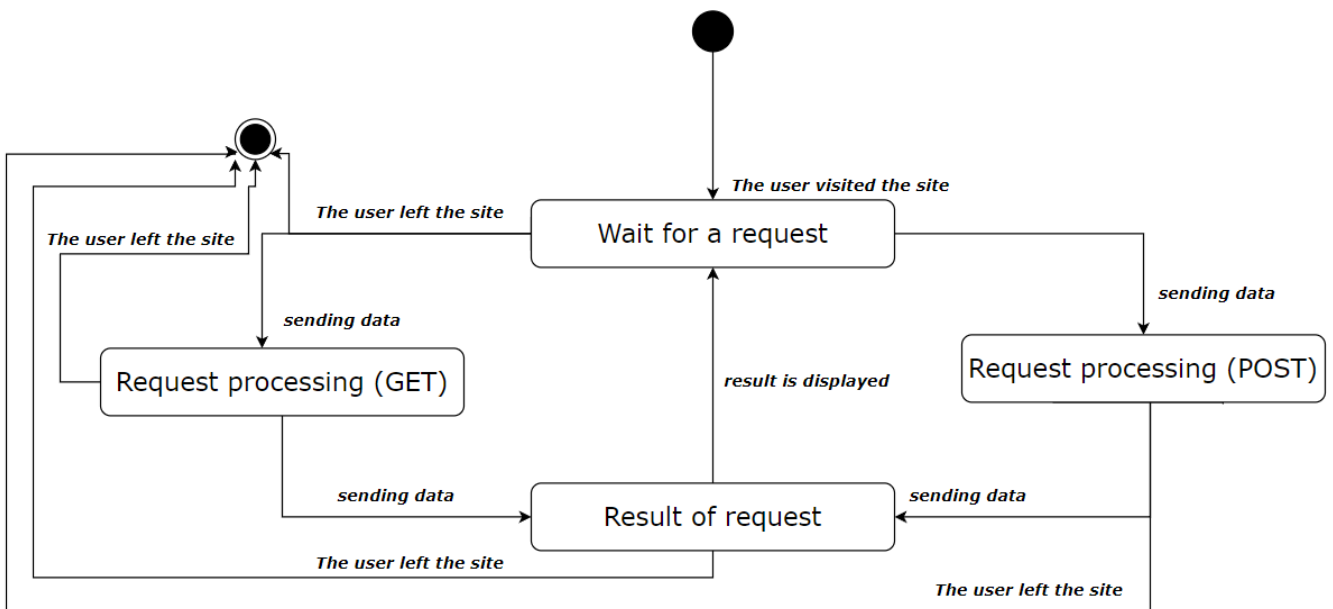


Рисунок 2.3 – Діаграма станів вебзастосунку

На рис. 2.3 було створено діаграму стану програми, об'єктом якої є вебзастосунок. Точкою входу є користувач поза веб-сайтом. Коли користувач відкриває сайт, вебзастосунок переходить в стан очікування запитів користувача. Користувач може використовувати інтерфейс для відправки GET запиту на сервер, таким чином вебзастосунок залишиться в стані обробки запиту. Під час надсилання оброблених даних, веб-сайт перебуває в стані генерації результатів, що відповідають обробленим даним, і згенеровані результати відображаються користувачеві. Користувач також може надіслати запит POST, і він буде оброблений, результати згенеровані та повернуті користувачеві. Точка виходу – це час, коли користувач залишає веб-сторінку. Це може статися в будь-якому стані вебзастосунку.

2.4 Створення діаграми класів

Діаграми класів – це тип діаграм UML, які використовуються в розробці програмного забезпечення для візуального представлення структури та зв'язків класів у системі. UML – це стандартизована мова моделювання, яка допомагає проектувати та документувати програмні системи. Вони є невід'ємною частиною процесу розробки програмного забезпечення, допомагаючи як на етапі проектування, так і на етапі документації [7].

Діаграми класів надають загальний огляд дизайну системи, допомагаючи взаємодіяти та документувати структуру програмного забезпечення [9]. Вони показують класи системи, а також взаємодії між ними через асоціації, агрегації, композиції, спадковість тощо. Ці діаграми допомагають розробникам та архітекторам легше розуміти та аналізувати структуру програми або системи [15]. Вони є фундаментальним інструментом об'єктно-орієнтованого проектування та відіграють вирішальну роль у життєвому циклі розробки програмного забезпечення.

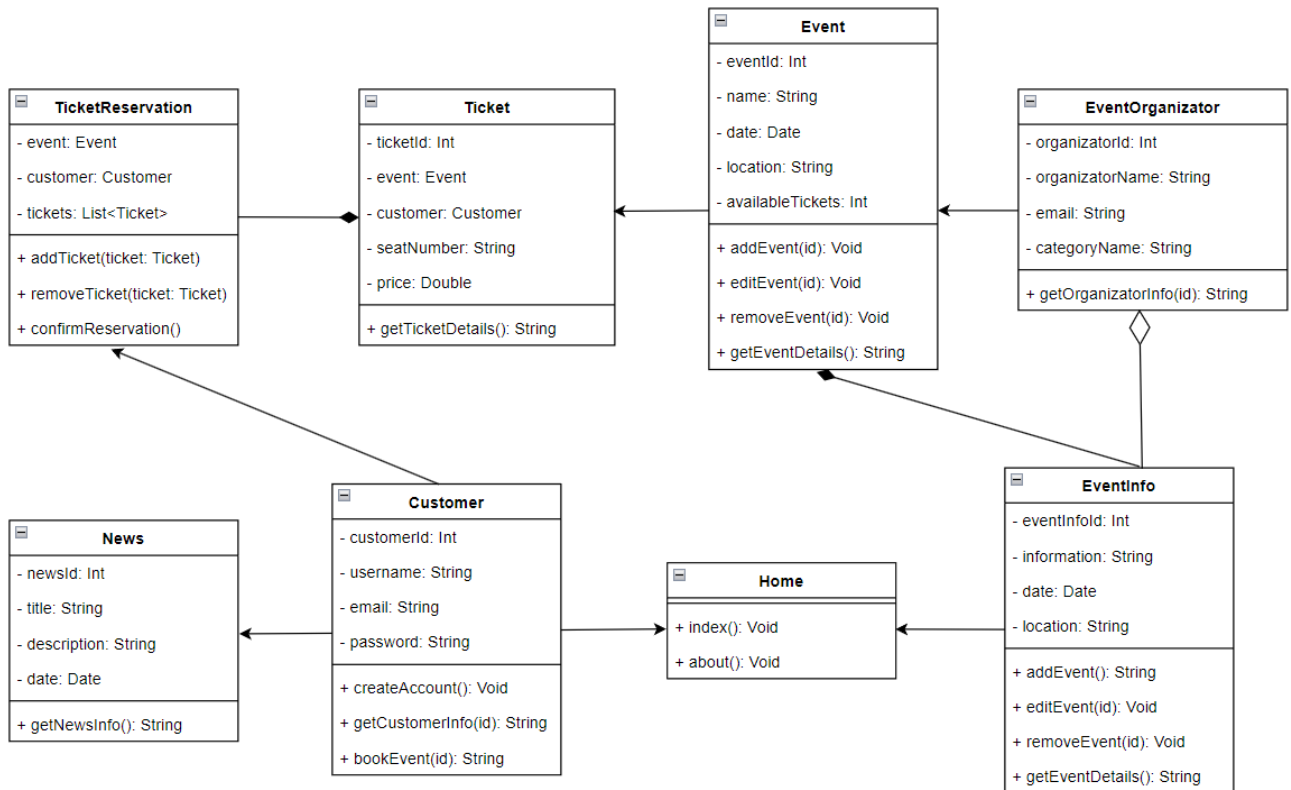


Рисунок 2.4 – Діаграма класів

На рис. 2.4 створена діаграма класів вебзастосунку бронювання квитків на заходи. На даній діаграмі зазначено взаємодію між класами, які використовуються у вебзастосунку: ticketReservation, ticket, event, eventOrganizator, news, customer, home, eventInfo.

2.5 Огляд технологій

При розробці вебзастосунку для бронювання квитків на заходи, були використані наступні технології:

1. Мови програмування: JavaScript, PHP.
2. База даних: MySQL.
3. CMS: WordPress.
4. Веб-інтерфейс: PhpMyAdmin.
5. Бібліотеки: Jquery, Bootstrap.

Обрані технології мають важливе значення для розробки вебзастосунка для бронювання квитків на заходи. JavaScript використовується для реалізації веб-інтерфейсу та взаємодії користувача з додатком. Він дозволяє створювати динамічні елементи на сторінці, перевіряти форми та дані і забезпечувати інтерактивність. PHP використовується для розробки серверної частини додатку, обробки даних від користувача, взаємодії з базою даних і генерації веб-сторінок [14]. MySQL є базою даних, яка дозволяє зберігати дані про заходи, квитки, користувачів. Вона забезпечує ефективний доступ до даних і дозволяє проводити складні запити для взаємодії з інформацією у вебзастосунку [2]. jQuery – це бібліотека JavaScript, яка спрощує роботу з маніпулюванням DOM та взаємодією з елементами сторінки. Вона допомагає реалізувати складні функції і анімації. Bootstrap — це CSS-фреймворк, який забезпечує готові компоненти та стилі для швидкого створення адаптивного веб-інтерфейсу.

2.5.1 Використання бібліотеки JQuery та фреймворку Bootstrap

JQuery – це одна з найпопулярніших JavaScript бібліотек, яка значно спрощує взаємодію з DOM, анімацією, обробкою подій та використання AJAX. Завдяки jQuery можна легко створювати обробники подій до елементів та реагувати на їх взаємодію з користувачем [13]. Крім того, вона надає методи для вибору елементів на сторінці та зміни їх властивостей. Особливість jQuery полягає в тому, що вона спрощує виконання AJAX-запитів, дозволяючи асинхронно отримувати чи відправляти дані на сервер без перезавантаження сторінки. Бібліотека має велику спільноту розробників, яка постійно створює різноманітні плагіни для розширення її функціональності. Серед таких плагінів можна знайти рішення для створення каруселей, валідації форм, роботи з датами та багато іншого. Однією з головних переваг jQuery є те, що вона робить код кросбраузерним, тобто забезпечує його працездатність в різних веб-браузерах.

Bootstrap є однією з найпопулярніших фронтенд-бібліотек, яка дозволяє створювати стильні та адаптивні веб-інтерфейси з великою швидкістю та легкістю. Вона базується на HTML, CSS та JavaScript, та надає рішення для швидкої розробки проєктів, включаючи готові компоненти, сітку для розміщення контенту та стилізацію елементів. Bootstrap забезпечує широкий спектр готових компонентів, таких як навігаційні панелі, кнопки, форми, таблиці, модальні вікна та багато інших, що значно прискорює розробку інтерфейсу. Крім того, вона має вбудовану підтримку для адаптивного дизайну, що дозволяє створювати інтерфейси, які оптимально виглядають на різних пристроях та екранних розмірах. Bootstrap також має активну спільноту розробників, яка постійно розширює його функціональність за допомогою різноманітних плагінів та розширень. Ці розширення можуть включати такі функції, як анімація, підтримка різних мов, додаткові компоненти та інструменти для роботи зі стилями. Однією з головних переваг Bootstrap є його простота використання та здатність швидко створювати стильний та сучасний веб-інтерфейс з мінімальними зусиллями.

2.5.2 Опис роботи з CMS WordPress

WordPress – це система керування контентом (CMS) з відкритим вихідним кодом, яка дозволяє користувачам створювати та публікувати контент, а також керувати веб-сайтами [18]. WordPress став однією з найпопулярніших платформ для створення сайтів завдяки своїй гнучкості, зручності використання та широкому вибору плагінів та тем.

WordPress почав свою історію як платформа для створення блогів, але з часом перетворився на потужний інструмент для будь-яких типів веб-сайтів, від особистих блогів до корпоративних порталів [19]. Його успіх ґрунтується на кількох ключових перевагах:

1. Гнучкість. WordPress дозволяє легко налаштовувати веб-сайт під потреби замовника. Завдяки безкоштовним і платним темам і плагінів, розробник може створити унікальний дизайн і функціональність веб-сайту.

2. Зручність використання. інтерфейс WordPress дуже інтуїтивний, що дозволяє навіть початківцям швидко орієнтуватись у створенні та редагуванні контенту. Додавання нових статей, зображень, відео або зміна дизайну стає простим завданням.

3. Відкритий вихідний код. це означає, що WordPress безкоштовний для використання, і його можна змінювати та пристосовувати до власних потреб. Багато розробників по всьому світу активно співпрацюють над покращенням платформи.

4. Онлайн-спільнота. однією великою перевагою є активна спільнота користувачів WordPress. Є безліч форумів, блогів та ресурсів, де можна отримати поради, допомогу або навіть найняти фахівців для вирішення проблем.

CMS WordPress підходить для таких типів веб-сайтів, як:

1. Блоги. WordPress починався як інструмент для ведення блогів і досі залишається однією з найкращих платформ для ведення блогів.

2. Сайти електронної комерції. За допомогою таких плагінів, як WooCommerce, ви можете легко створити онлайн-магазин.

3. Веб-сайти для бізнесу. Малі та великі підприємства можуть отримати швидкий та зручний доступ до редагування веб-сайту з використанням WordPress.

4. Освітні платформи. За допомогою плагінів розробник має змогу налаштувати платформу онлайн-курсів за допомогою WordPress.

Загалом, WordPress – це потужний і універсальний інструмент для створення веб-сайтів будь-якого рівня складності. Він підходить як для початківців, так і для досвідчених веб-розробників, завдяки своїй простоті використання та безлічі можливостей для розширення функціоналу.

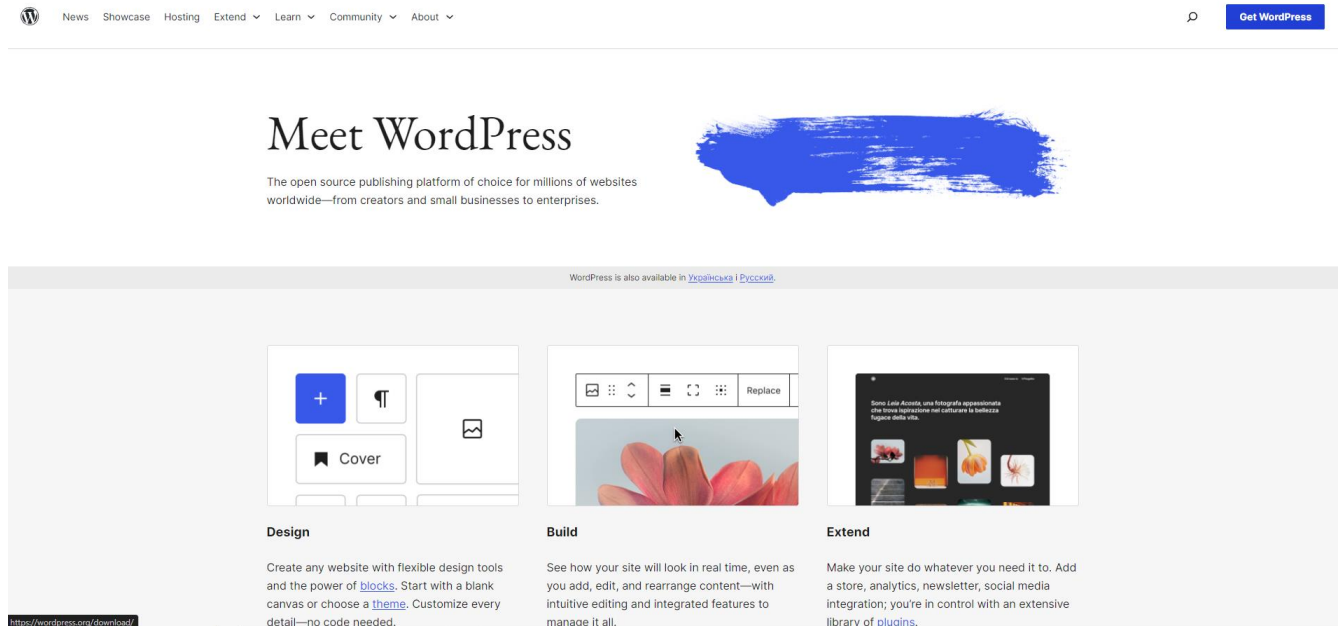


Рисунок 2.5 – Документація WordPress

Розробка веб-додатків на платформі WordPress має чимало переваг, зокрема, щодо мінімізації взаємодії з базою даних під час роботи з сайтом. Однією з основних переваг є автоматичне створення та керування базою даних WordPress. Під час розробки вебзастосунків на WordPress значна частина процесів відбувається автоматично завдяки вбудованим можливостям і плагінами цієї платформи.

WordPress надає зручний інтерфейс для збереження та отримання даних з бази даних, що спрощує розробку сайтів. Коли ви створюєте новий вебзастосунок на WordPress, ви можете користуватися вбудованими функціями для роботи з даними, не займаючись напряду SQL запитамі або складним підключенням до бази даних.

Ще однією перевагою є готові функції WordPress для зберігання, оновлення та видалення даних, що робить розробку вебзастосунків ефективнішою і менш витратною з точки зору програмування. Структура бази даних WordPress досить стандартна і добре документована, що спрощує розширення та адаптацію бази даних під потреби конкретного проекту.

Взаємодія з базою даних через WordPress зазвичай здійснюється за допомогою вбудованих функцій WordPress API, таких як WP_Query для отримання даних з бази, або функцій для додавання, оновлення або видалення записів. Отже, розробка вебзастосунків на WordPress має велику перевагу у спрощенні взаємодії з базою даних, що дозволяє розробникам більше уваги приділяти функціональності та дизайну сайту, замість витрачання часу на базові операції з базою даних.

2.5.3 Проектування та опис бази даних

Паралельно з розробкою сайту, розробляється також і база даних. Для підключення та роботи з базою даних було використано застосунок phpMyAdmin. phpMyAdmin – це веб-інтерфейс, який дозволяє керувати базами даних за допомогою браузера без необхідності знати команди SQL. PhpMyAdmin дозволяє виконувати різноманітні операції з базами даних, такі як створення та видалення таблиць, оптимізація, виправлення та перевірка, внесення змін до вмісту таблиць. Крім того, є можливість імпорту та експорту баз даних. Це програмне забезпечення працює через веб-інтерфейс і може бути встановлене на будь-якому сервері.

Для авторизації треба використати комбінацію логіна та пароля, яка була використана при створенні бази даних (рис. 2.6).

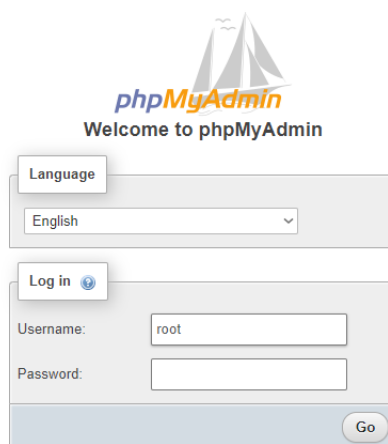


Рисунок 2.6 – Вхід в обліковий запис phpMyAdmin

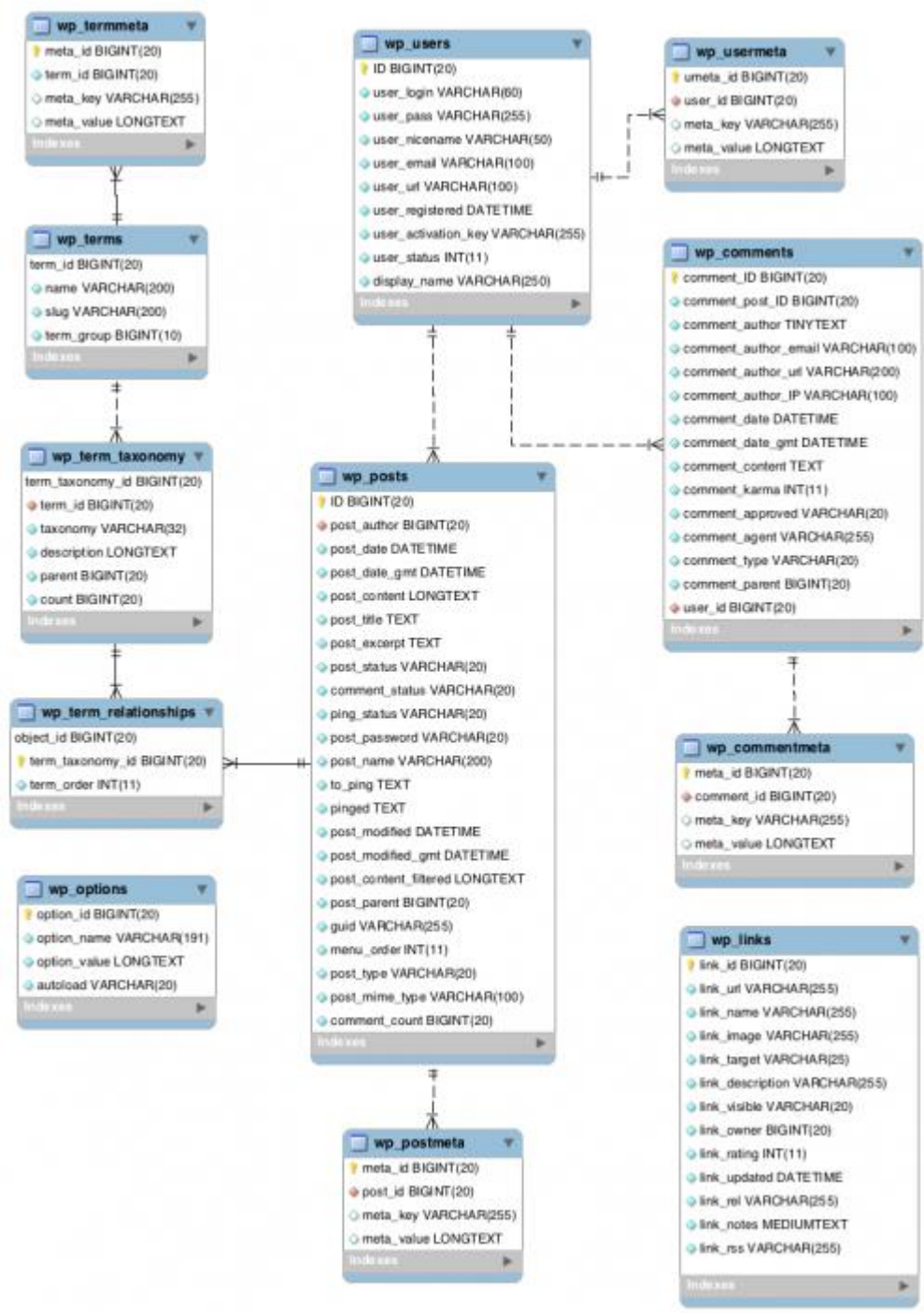


Рисунок 2.7 – Модель бази даних WordPress

В інтерфейсі phpMyAdmin є панель попереднього перегляду баз даних, де відображається список доступних баз даних. Користувач може вибрати базу даних з

цього списку та розпочати роботу з її таблицями та даними. Праворуч нагорі знаходиться ряд навігаційних кнопок, які забезпечують доступ до різних функцій та інструментів (рис. 2.8).

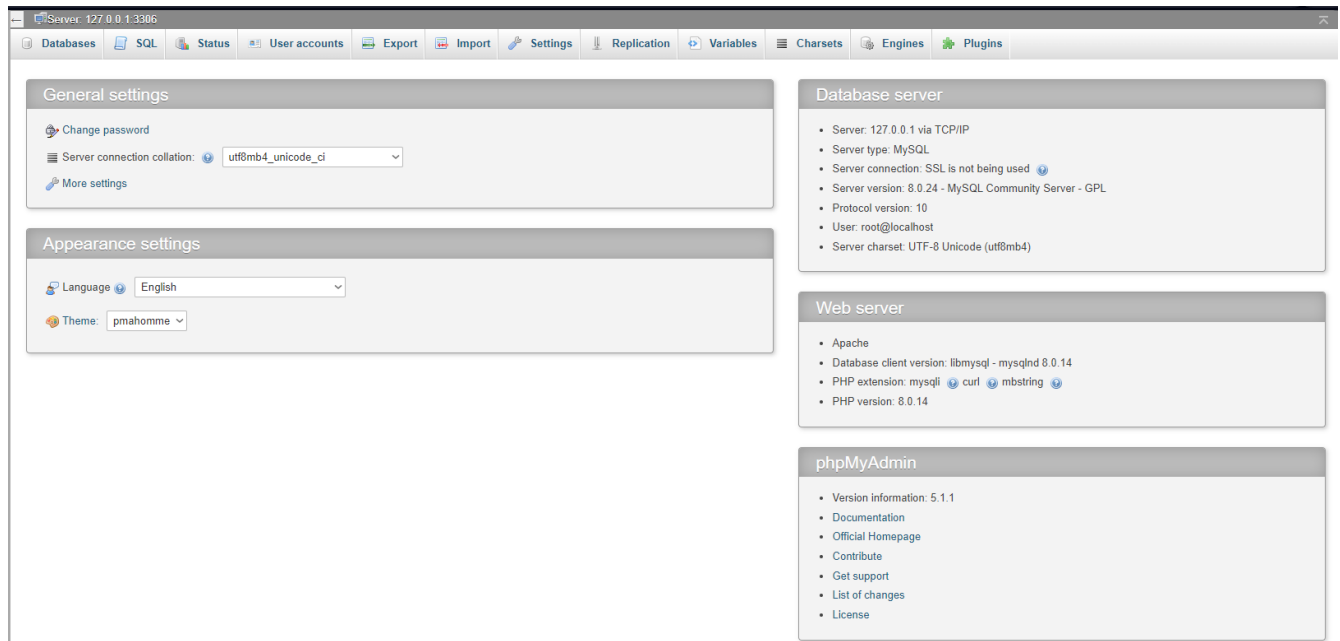


Рисунок 2.8 – Інтерфейс phpMyAdmin

У колонці зліва можна побачити всі доступні бази даних MySQL (рис. 2.9 – рис. 2.10). Перейшовши на обрану базу даних, розробник може побачити всі доступні таблиці. Доступні стандартні таблиці WordPress, включаючи:

1. `wp_commentmeta`. Ця таблиця зберігає додаткові метадані (наприклад, ключ-значення) для коментарів, які зберігаються в таблиці `wp_comments`;
2. `wp_comments`. Зберігаються всі коментарі, які залишили відвідувачі на сайті. Це включає дані про автора, текст коментаря, дату, статус;
3. `wp_options`. У цій таблиці зберігаються налаштування та параметри WordPress, такі як налаштування теми, активні плагіни, налаштування сайту тощо;
4. `wp_postmeta`. Зберігає додаткові метадані для записів (постів) WordPress, такі як дати, автори, тощо;

5. `wp_posts`. Основна таблиця для зберігання всіх записів (постів) на сайті WordPress. Кожен запис має відповідний тип, наприклад, пости, сторінки, прикріплення, тощо;
6. `wp_termmeta`. Аналогічно до `wp_postmeta`, ця таблиця зберігає додаткові метадані для термінів (категорій, ярликів тощо);
7. `wp_terms`. Зберігаються всі терміни (категорії, ярлики, мітки), які використовуються на сайті;
8. `wp_terms_relationships`. Ця таблиця встановлює зв'язок між записами (постами) і термінами (категоріями, ярликами тощо), визначаючи, який термін призначений якому запису;
9. `wp_term_taxonomy`. Відповідає за зберігання таксономічної інформації про терміни (категорії, ярлики, мітки) і визначає їхні взаємозв'язки з записами;
10. `wp_users`. Таблиця зберігає додаткові метадані для користувачів WordPress, такі як імена, адреси електронної пошти, ролі тощо.

Table	Rows	Engine	Collation	Charset	Comment
vaiedkzyx3_mailpoet_user_agents	535	InnoDB	utf8mb4_unicode_ci	utf8mb4	144.0 KiB
vaiedkzyx3_mailpoet_user_flags	3	InnoDB	utf8mb4_unicode_ci	utf8mb4	32.0 KiB
vaiedkzyx3_options	1,013	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	2.4 MiB
vaiedkzyx3_postmeta	29,459	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	7.5 MiB
vaiedkzyx3_posts	4,659	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	62.4 MiB
vaiedkzyx3_pum_subscribers	0	InnoDB	utf8mb4_unicode_ci	utf8mb4	80.0 KiB
vaiedkzyx3_termmeta	6	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	48.0 KiB
vaiedkzyx3_terms	21	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	48.0 KiB
vaiedkzyx3_term_relationships	501	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	64.0 KiB
vaiedkzyx3_term_taxonomy	21	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	48.0 KiB
vaiedkzyx3_usermeta	19,511	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	4.5 MiB
vaiedkzyx3_users	1,030	InnoDB	utf8mb4_unicode_520_ci	utf8mb4	384.0 KiB
vaiedkzyx3_wfblockedplog	0	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wfblocks7	0	InnoDB	utf8_general_ci	utf8	64.0 KiB
vaiedkzyx3_wfconfig	283	InnoDB	utf8_general_ci	utf8	2.3 MiB
vaiedkzyx3_wfcrawlers	0	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wffilechanges	0	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wffilemods	43,369	InnoDB	utf8_general_ci	utf8	21.5 MiB
vaiedkzyx3_wfhits	6	InnoDB	utf8_general_ci	utf8	64.0 KiB
vaiedkzyx3_wfhover	0	InnoDB	utf8_general_ci	utf8	32.0 KiB
vaiedkzyx3_wfissues	22	InnoDB	utf8_general_ci	utf8	128.0 KiB
vaiedkzyx3_wfknownfilelist	43,369	InnoDB	utf8_general_ci	utf8	10.5 MiB
vaiedkzyx3_wfvivetrachhuman	0	InnoDB	utf8_general_ci	utf8	32.0 KiB
vaiedkzyx3_wflocs	1	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wflogins	2,001	InnoDB	utf8_general_ci	utf8	576.0 KiB
vaiedkzyx3_wflogs_secrets	0	InnoDB	utf8_general_ci	utf8	32.0 KiB
vaiedkzyx3_wflogs_role_counts	0	MEMORY	utf8_general_ci	utf8	0 B
vaiedkzyx3_wflogs_settings	30	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wfnotifications	9	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wfpendingissues	0	InnoDB	utf8_general_ci	utf8	80.0 KiB
vaiedkzyx3_wfreversecache	1	InnoDB	utf8_general_ci	utf8	16.0 KiB
vaiedkzyx3_wfsecurityevents	0	InnoDB	utf8_general_ci	utf8	16.0 KiB

Рисунок 2.9 – Таблиці бази даних

	option_id	option_name	option_value	autoload
<input type="checkbox"/>	1	siteurl	http://events-site	yes
<input type="checkbox"/>	2	home	http://events-site	yes
<input type="checkbox"/>	3	blogname	EventAT Site	yes
<input type="checkbox"/>	4	blogdescription	It's the site for events	yes
<input type="checkbox"/>	5	users_can_register	1	yes
<input type="checkbox"/>	6	admin_email	andrewtebenko@gmail.com	yes
<input type="checkbox"/>	7	start_of_week	1	yes
<input type="checkbox"/>	8	use_balanceTags	0	yes
<input type="checkbox"/>	9	use_smilies	1	yes
<input type="checkbox"/>	10	require_name_email	1	yes
<input type="checkbox"/>	11	comments_notify	1	yes
<input type="checkbox"/>	12	posts_per_rss	10	yes
<input type="checkbox"/>	13	rss_use_excerpt	0	yes
<input type="checkbox"/>	14	mailserver_url	mail.example.com	yes
<input type="checkbox"/>	15	mailserver_login	login@example.com	yes
<input type="checkbox"/>	16	mailserver_pass	password	yes
<input type="checkbox"/>	17	mailserver_port	110	yes
<input type="checkbox"/>	18	default_category	1	yes

Рисунок 2.10 – Налаштування бази даних

Таким чином, було ознайомлено з базою даних та таблицями веб-застосунків, використовуючи WordPress.

Висновки до розділу 2

В другому розділі кваліфікаційної роботи бакалавра було ознайомлено з проектуванням та моделюванням вебзастосунка. Описано короткий, середній та повний сценарій вебзастосунка. Створено діаграму варіантів використання, станів вебзастосунка та діаграму класів.

Ознайомлено з технологіями, які використовувались при створенні вебзастосунка бронювання квитків на захід. Описано використання мов програмування JavaScript та PHP. Ознайомлено з використанням бібліотеки JQuery та фреймворку Bootstrap. Описано використання CMS WordPress та бази даних MySQL. Розглянуто можливості використання веб-інтерфейсу phpMyAdmin. Було розглянуто базу даних та таблиці у веб-інтерфейсі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКА

3.1. Створення діаграми розгортання

Діаграма розгортання – це тип діаграми UML, яка показує архітектуру виконання системи, включаючи такі вузли, як апаратне або програмне середовище виконання, і проміжне програмне забезпечення, що їх з'єднує. Діаграми розгортання зазвичай використовуються для візуалізації фізичного обладнання та програмного забезпечення системи. Діаграми розгортання допомагають змоделювати апаратну топологію системи порівняно з іншими типами діаграм UML, які здебільшого описують логічні компоненти системи.

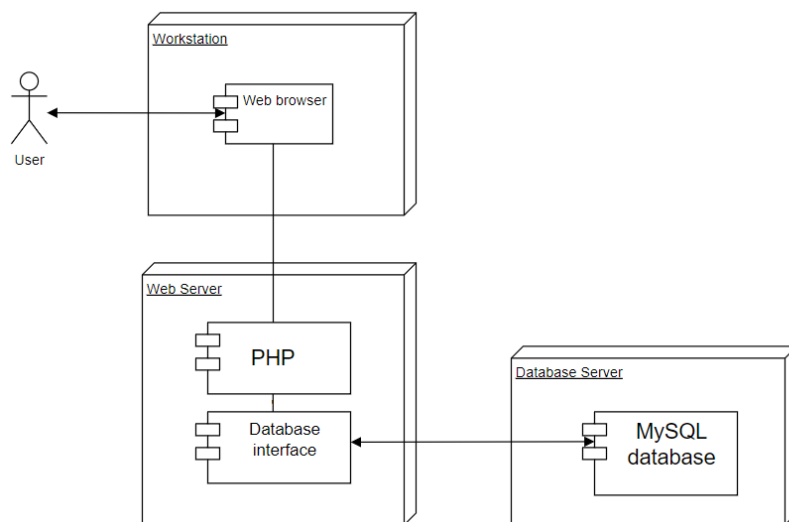


Рисунок 3.1 – Діаграма розгортання

На рис. 3.1 зображена діаграма розгортання, на якій зображено зв'язок між користувачем та базою даних. Вебзастосунок відправляє запити та отримує відповіді від web-сервера. Сервер надає користувацький інтерфейс, а також використовує інтерфейс бази даних для доступу до даних, їх зміни, видалення, модифікації. В даній роботі було використано мову програмування PHP та базу даних MySQL. PHP виконує SQL-запити до бази даних, отримує результати від бази даних (такі як вибірка

даних), обробляє їх та передає для відображення веб-сторінки користувачеві через web-сервер.

3.2. Підключення бази даних

Щоб забезпечити підключення бази даних до проєкту WordPress, необхідно створити файл `wp-config.php` у кореневій директорії проєкту. Цей файл є основою інсталяції WordPress, що містить ключові параметри конфігурації веб-сайту [4]. Він забезпечує з'єднання з базою даних, де зберігається важлива інформація про веб-застосунок, включаючи налаштування, контент публікацій, теми та плагіни, а також дані користувачів.

Крім цього, `wp-config.php` виконує інші важливі функції, такі як:

1. включення ключів безпеки для захисту веб-сайту;
2. зміну префіксів таблиць для збільшення безпеки;
3. переміщення основних директорій файлів WordPress, таких як `wp-plugins`, `wp-uploads` і `wp-content`;
4. виконання інших додаткових налаштувань, що дозволяють додаткову кількість контролю над вебзастосунком.

Отже, `wp-config.php` не лише створює зв'язок між файлами сайту та базою даних, а й забезпечує важливі можливості налаштування та захисту веб-сайту.

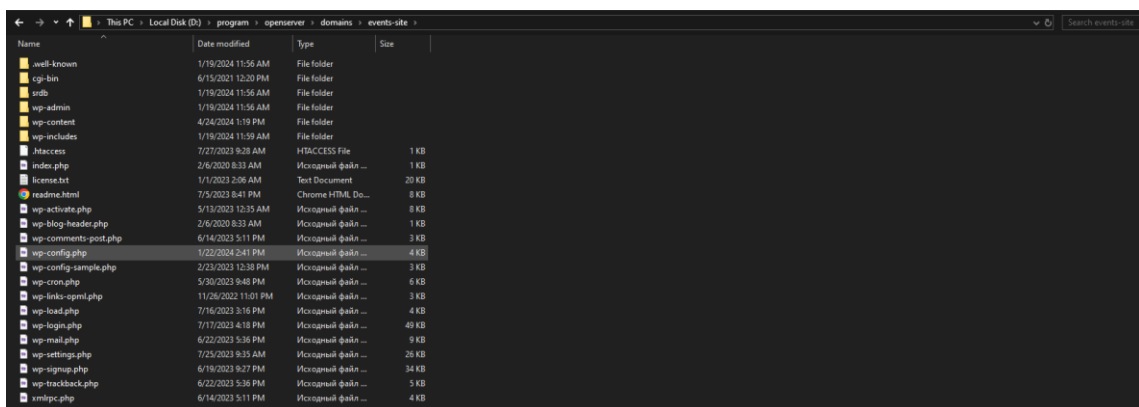


Рисунок 3.2 – Розташування файлу `wp-config.php`


```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'eventsites' );  
  
/** MySQL database username */  
define( 'DB_USER', 'root' );  
  
/** MySQL database password */  
define( 'DB_PASSWORD', '' );  
  
/** MySQL hostname */  
define( 'DB_HOST', 'localhost' );  
  
/** Database Charset to use in creating database tables. */  
define( 'DB_CHARSET', 'utf8mb4' );  
  
/** The Database Collate type. Don't change this if in doubt. */  
define( 'DB_COLLATE', '' );
```

Рисунок 3.3 – Підключення проєкту до бази даних

Таким чином, було підключено базу даних до проєкту.

3.3. Створення кастомних WordPress полів

Advanced Custom Fields – це плагін WordPress, який дозволяє додавати додаткові поля вмісту на екрани редагування WordPress. Ці додаткові поля вмісту частіше називають користувацькими полями, і вони дозволяють зручніше та швидше створювати різні блоки вебзастосунка.

Кожне поле додається до групи, яка дає змогу як упорядковувати поля, так і вказувати екрани редагування, де вони відображаються. Кожне поле містить параметри, за допомогою яких можна налаштувати вигляд поля (його тип), де зберігається його значення (його назва) і як воно функціонує (його налаштування). Ці параметри полів можна налаштувати під час редагування групи полів. Advanced Custom Fields підтримує умовну логіку, яка дозволяє показувати або приховувати поля залежно від значень інших полів. Це робить процес введення даних більш динамічним та адаптивним до потреб користувача. Використовуючи групи полів, розробники можуть легко виводити значення користувацьких полів у шаблони тем WordPress. Це дозволяє створювати індивідуальні макети сторінок та відображати специфічний вміст у потрібному форматі [8].



Рисунок 3.4 – Advanced Custom Fields

Для створення кастомних полів у WordPress за допомогою плагіну Advanced Custom Fields потрібно перейти до розділу ‘Поля додаткових метаданих’, створити нове поле, налаштувати його параметри, додати на сторінку і зберегти зміни (рис. 3.5 – рис. 3.6).

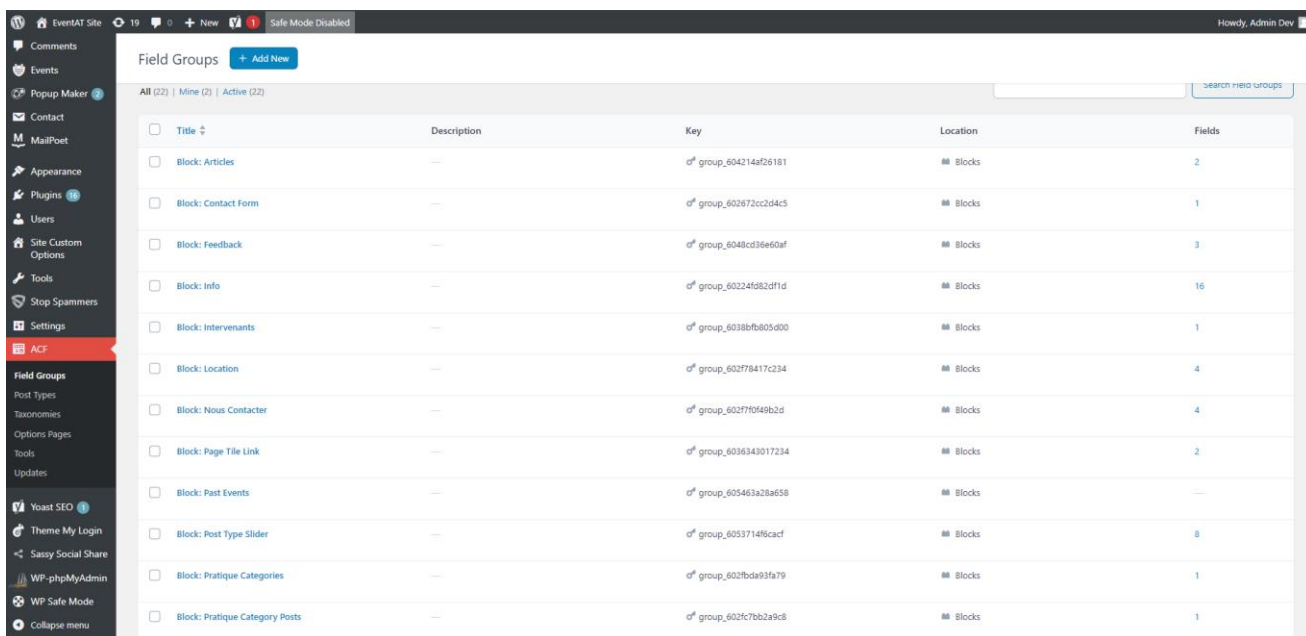


Рисунок 3.5 – Створення кастомних полів

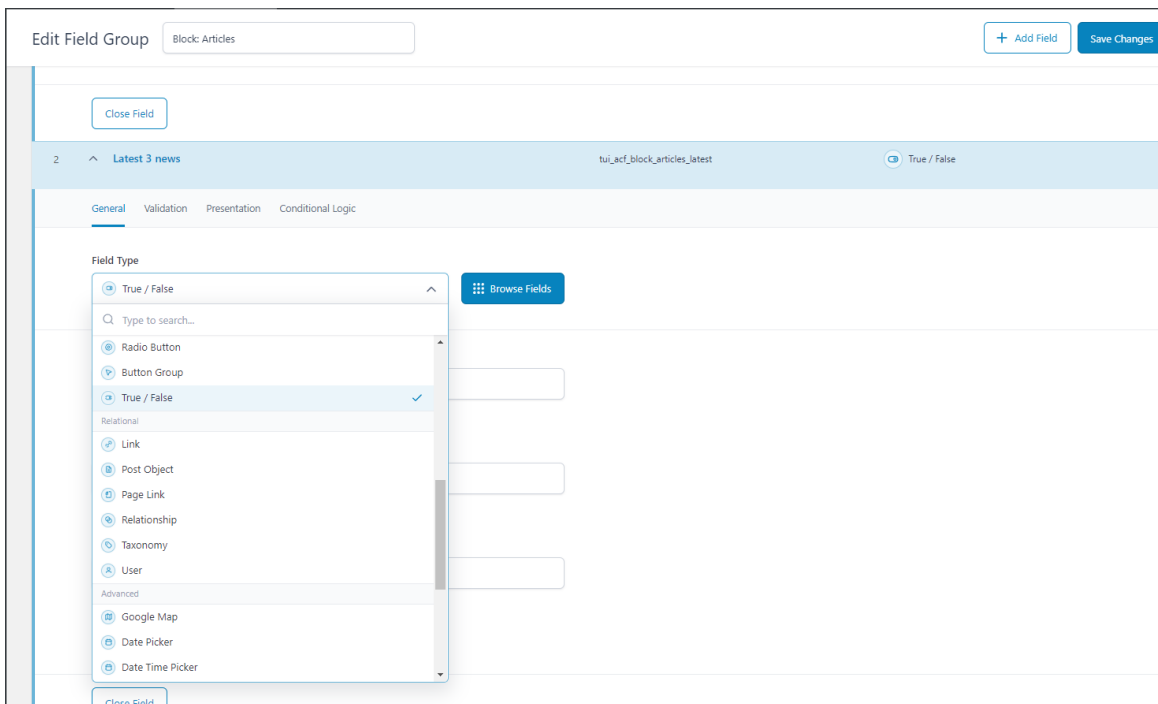


Рисунок 3.6 – Редагування кастомного блоку

Для роботи з кастомними полями у веб-сайті потрібно створити PHP-файл, в якому будуть описані поля. Після цього, даний файл потрібно підключити до проекту. Якщо для коректної роботи з кастомними полями необхідна взаємодія з DOM-елементами, також можливо підключити JavaScript-файл (рис. 3.7 – рис. 3.8).

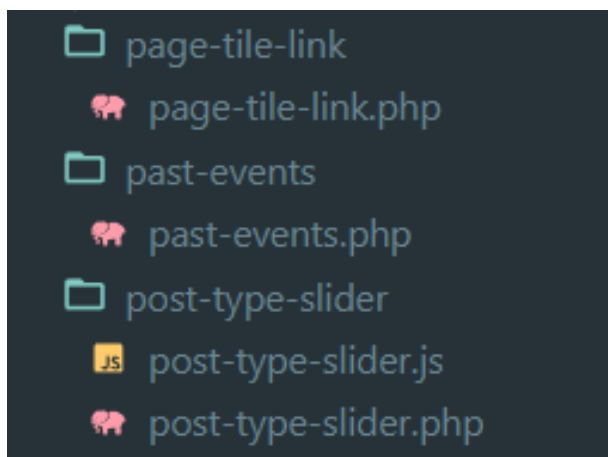


Рисунок 3.7 – Створення файлу для кастомного блоку

```
acf_register_block_type( [
    'name'           => 'tui-acf-nous-contacter',
    'title'          => __( 'Nous Contacter' ),
    'description'    => __( 'Nous Contacter block' ),
    'render_template' => ( __DIR__ ) . '/blocks/nous-contacter/nous-contacter.php',
    'category'       => 'tui-acf-blocks',
    'icon'           => 'email-alt2',
    'supports'       => [
        'align'       => false,
        'anchor'      => false,
        'customClassName' => true,
        'jsx'         => false,
    ]
    1 );
```

Рисунок 3.8 – Підключення кастомного блоку в проєкт

Таким чином, був підключений кастомний блок до WordPress проєкту.

3.4. Використання плагіну Events Manager

Event Manager – це інструмент для системи управління вмістом WordPress, який дозволяє користувачам створювати, редагувати та керувати подіями та календарями подій безпосередньо на веб-сайті [10].

Основні функції, які має плагін:

1. календарі, сторінки пошуку, списки, таблиці та форми бронювання;
2. реєстрація події (один день із часом початку/закінчення);
3. періодична та багатоденна реєстрація заходів;
4. управління бронюваннями (включаючи затвердження/відмови, експорт CVS);
5. кілька типів місцезнаходжень. Фізичне розташування, онлайн-події (URL-адреси);
6. категорії подій;
7. додавання заходу до Google Calendar;
8. безліч тегів, шаблонів для використання у постах та сторінках.

Кафедра інженерії програмного забезпечення
Вебзастосунок бронювання квитків на заходи

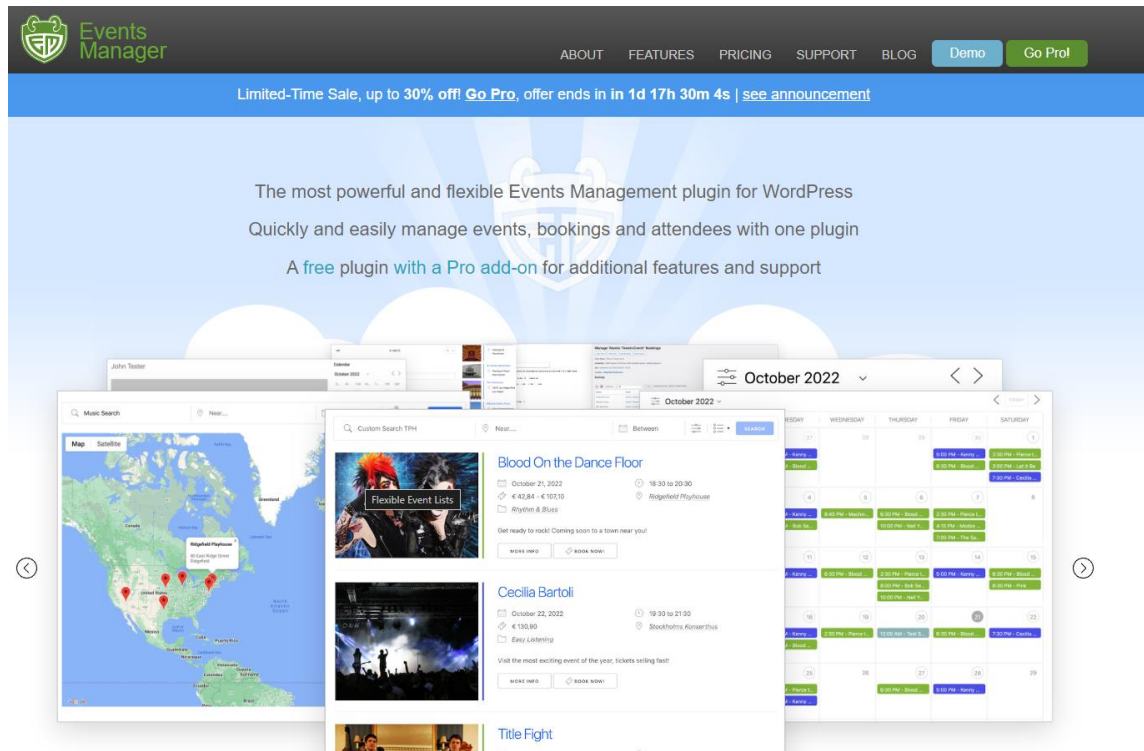


Рисунок 3.9 – Офіційна сторінка Events Manager plugin

Для створення нового заходу потрібно перейти до плагіну та натиснути відповідну кнопку створення. Після цього необхідно обрати назву події, опис події, дату, час, організатора події та місце проведення події. Також слід вказати ціну на захід та тип білету (рис. 3.10 – рис. 3.11).

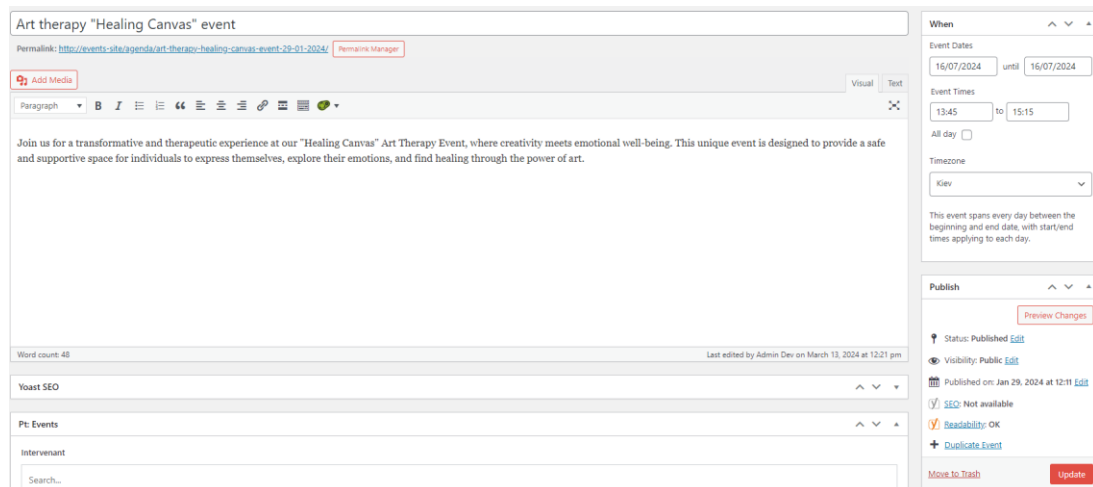


Рисунок 3.10 – Створення заходу

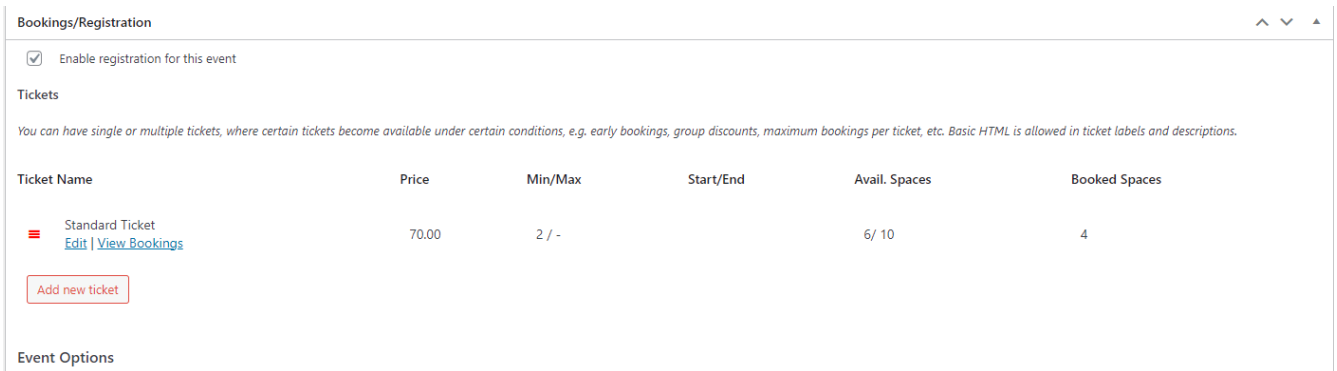


Рисунок 3.11 – Створення білету

Таким чином, було ознайомлено з використанням плагіну Events Manager у WordPress проєкті. Ознайомлено зі створенням заходу та білету.

3.5. Створення сторінок

Щоб створити сторінку у WordPress, треба увійти до панелі керування, перейти до розділу ‘Сторінки’ та натиснути ‘Додати нову’. Далі потрібно ввести заголовок та контент сторінки, використовуючи редактор для форматування тексту та додавання медіа-вмісту. Після завершення треба опублікувати сторінку та перезавантажити її (рис. 3.12).

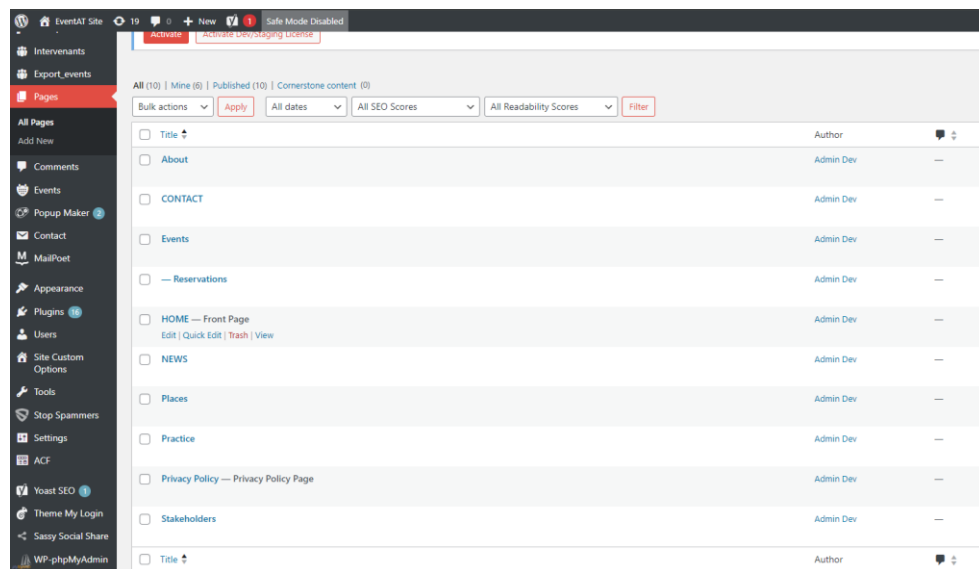


Рисунок 3.12 – Сторінки вебзастосунку бронювання квитків на захід

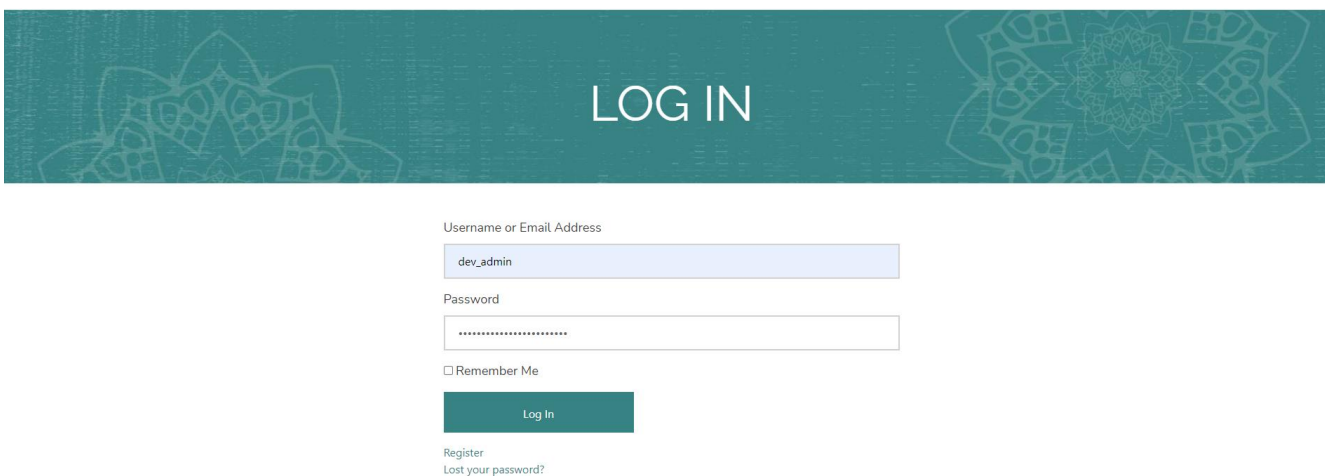


Рисунок 3.14 – Створення входу в обліковий запис

На головній сторінці вебзастосунку створено новий блок із варіантами заходів, які можуть зацікавити користувачів при бронюванні квитків (рис. 3.15). Користувач може знайти широкий спектр заходів. Для кращого ознайомлення з кожною темою додано детальний опис категорії заходів.

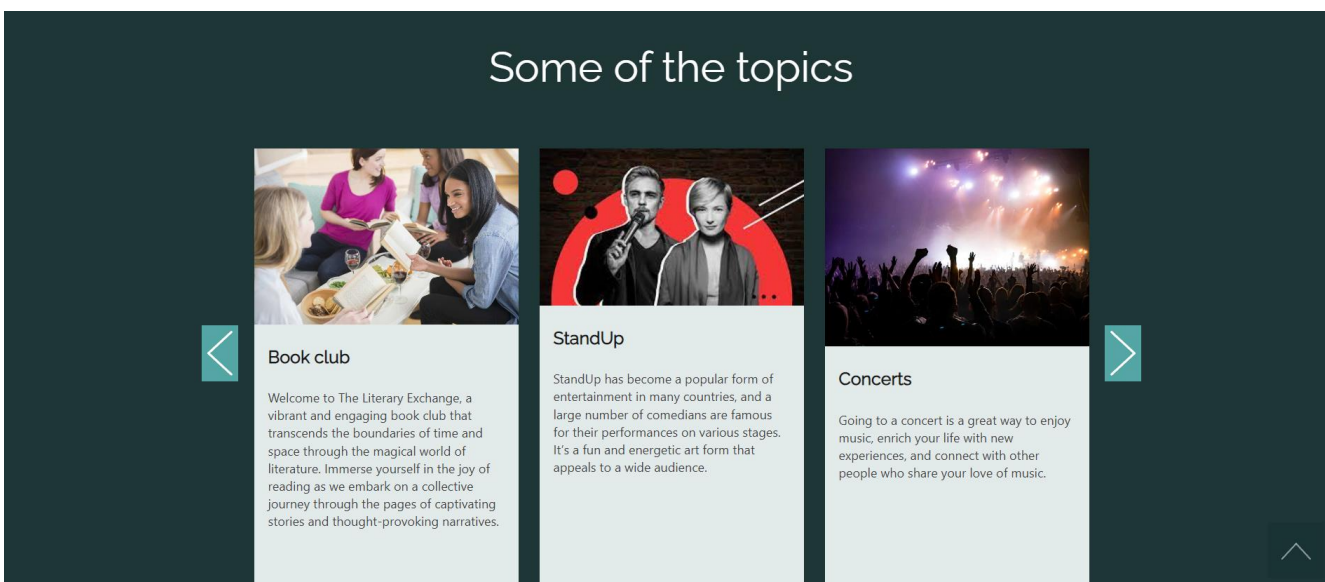


Рисунок 3.15 – Створення сторінки з темами заходів

Створено головну сторінку для бронювання квитків (рис. 3.16 – рис. 3.17). На цій сторінці користувачі мають змогу бронювати квитки на різноманітні події та

заходи, включаючи концерти, театральні вистави, спортивні матчі та інші заходи. Користувачі також зможуть обирати тип квитка залежно від вподобань та вимог. Наприклад, користувачі зможуть обрати звичайні місця або VIP-місця.

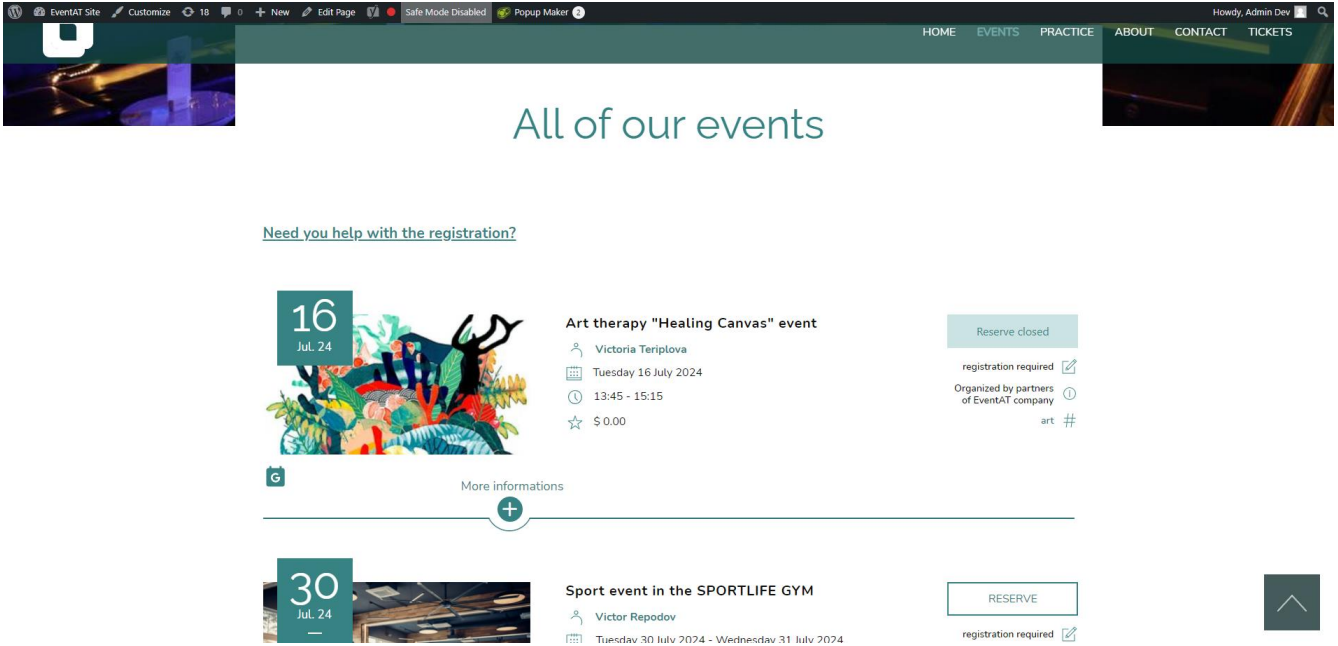


Рисунок 3.16 – Створення сторінки заходів

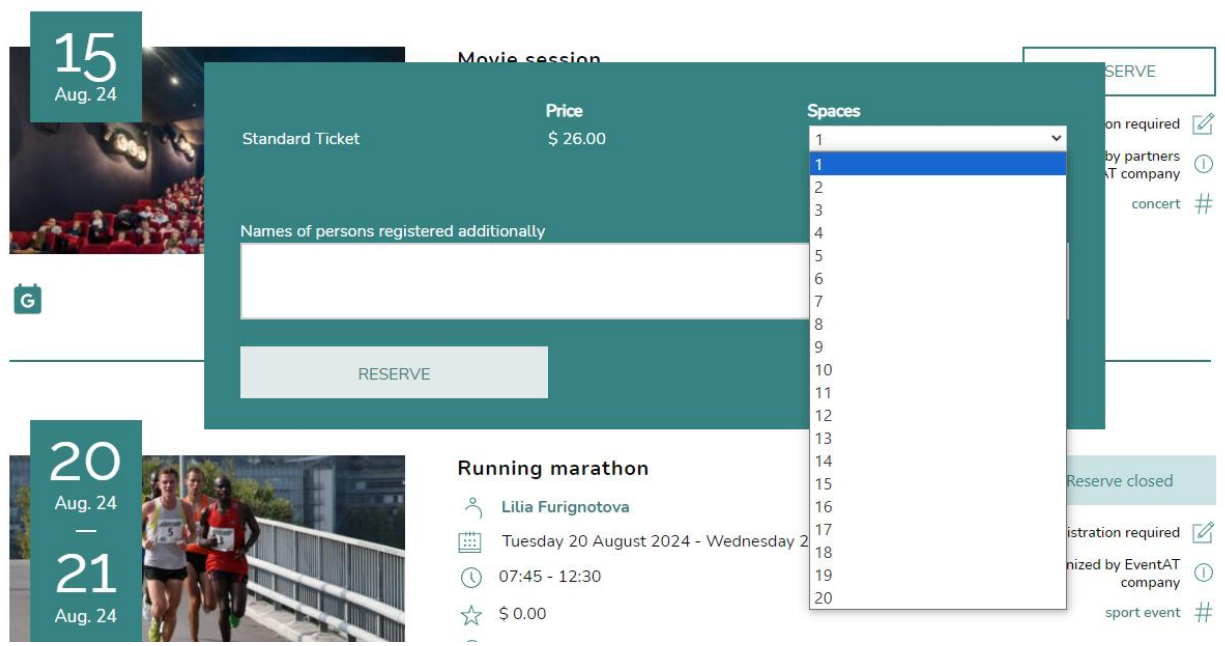


Рисунок 3.17 – Обрання кількості білетів

Створено сторінку для бронювання квитків (рис. 3.18). На цій сторінці користувач може переглянути інформацію про квитки, які він раніше придбав або доступні для придбання на заходи. У таблиці наведено заголовок події, дату проведення, кількість придбаних квитків та статус їх бронювання.



Event	Date	Spaces	Status
Running marathon	20 August 2024	1	Approved
Art therapy "Healing Canvas" event	16 July 2024	4	Approved
Concert By Vanessa Florey	12 March 2024	1	Approved Past event
MOVEMENT MEDICINE	04 February 2024	1	Cancelled Past event
DANCE MEDITATION	29 January 2024	1	Awaiting Payment Past event
DANCE MEDITATION	24 January 2024	1	Awaiting Payment Past event
DANCE MEDITATION	15 January 2024	1	Awaiting Payment Past event

Рисунок 3.18 – Створення сторінки бронювання

Створено сторінку з переліком організаторів подій. Тепер користувачі можуть використовувати фільтри за іменем організатора або за категорією заходу. На цій сторінці доступна докладна інформація про кожного організатора, що дозволяє краще зрозуміти їхню діяльність та внести відповідні вибори щодо участі в подіях.

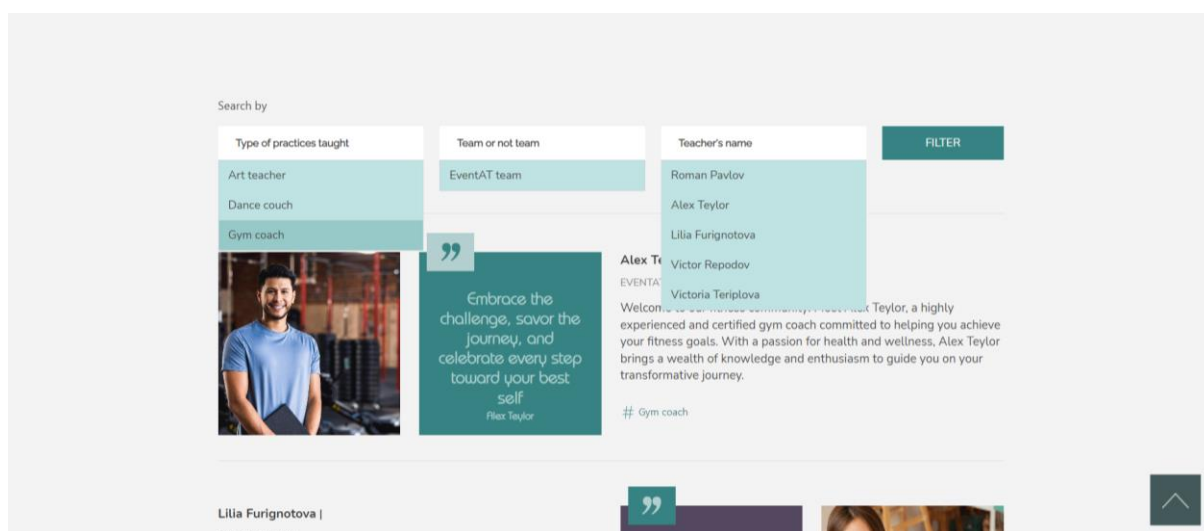


Рисунок 3.19 – Створення сторінки організаторів

Створено сторінку із контактною формою. Тепер користувачі можуть залишити своє запитання для підтримки та отримати відповідь на електронну пошту.

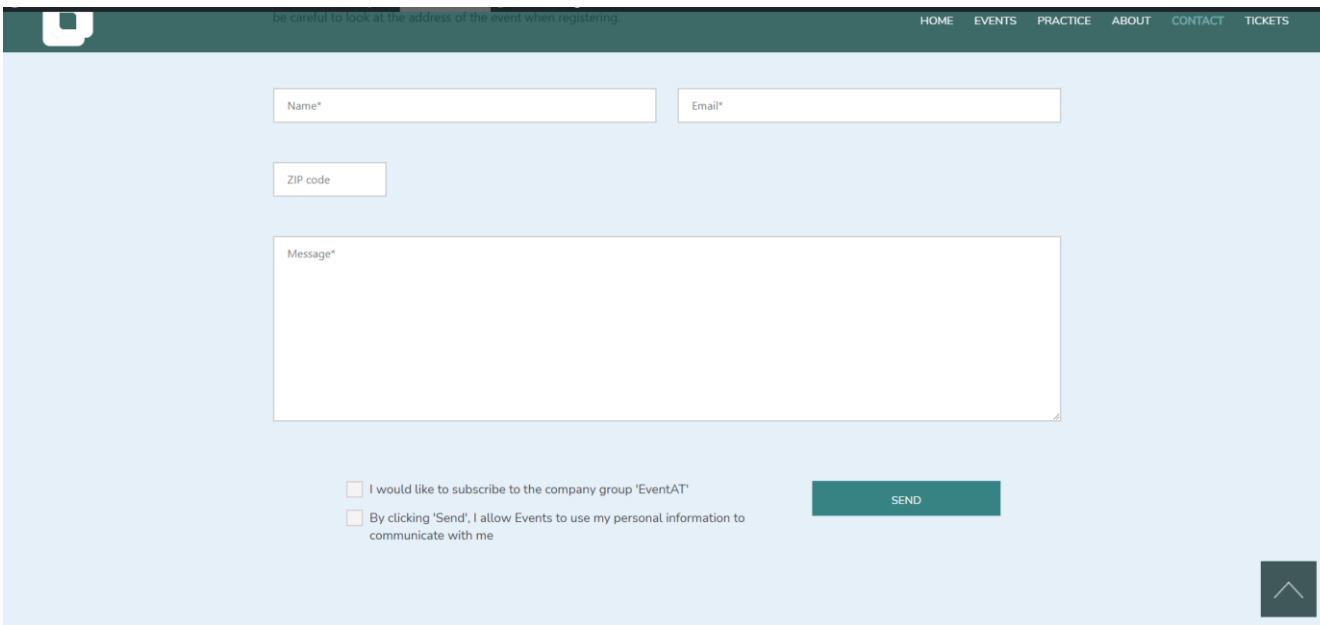


Рисунок 3.20 – Створення контактної форми

Було створено сторінки у веб-додатку для бронювання квитків на події. Нові сторінки включають обліковий запис, теми подій, перелік подій, вибір кількості квитків, сторінку бронювання, інформацію про організаторів та контактну форму.

3.6. Програмна реалізація

Створено категорії для заходів (рис. 3.21 – рис. 3.22). Клас `EM_Categories` є підкласом `EM_Taxonomy_Terms` та має властивості, що визначають характеристики категорій, такі як глобальність, назва таксономії, мета-ключ, назва термінів і клас термінів. Він має методи конструктора, перевизначення методів `__get()` і `__isset()` для зручного доступу до категорій, а також статичні методи для отримання, виведення, пагінації та пошуку категорій [11].

```

public static function get( $args = array() ){
    self::$instance = new EM_Categories();
    return parent::get($args);
}

public static function output( $args = array() ){
    self::$instance = new EM_Categories();
    return parent::output($args);
}

public static function get_pagination_links($args, $count, $search_action = 'search_cats', $default_args = array()){
    self::$instance = new EM_Categories();
    return parent::get_pagination_links($args, $count, $search_action, $default_args);
}

public static function get_post_search($args = array(), $filter = false, $request = array(), $accepted_args = array()){
    self::$instance = new EM_Categories();
    return parent::get_post_search($args, $filter, $request, $accepted_args);
}

public static function get_default_search( $array_or_defaults = array(), $array = array() ){
    self::$instance = new EM_Categories();
    return parent::get_default_search($defaults,$array);
}
}

```

Рисунок 3.21 – Створення категорій заходів

The screenshot shows the WordPress admin interface for managing event categories. On the left is the 'Add New Event Category' form with fields for Name, Slug, Parent Category, and Description. On the right is a table listing existing categories with columns for Name, Description, Slug, Count, and ID.

Name	Description	Slug	Count	ID
Art	—	art	1	33
Concert	—	concert	2	30
Dance	—	dance	0	29
Performance	—	performance	0	32
Sport event	—	sport-event	2	31

Рисунок 3.22 – Відображення категорій в адмін-панелі

Створено функцію для бронювання квитків на захід (рис. 3.23). Функція `em_get_booking()` призначена для отримання об'єкту бронювання за ідентифікатором. Вона перевіряє, чи вже існує глобальна змінна `$EM_Booking`, щоб не створювати її знову, і якщо так, повертає об'єкт бронювання. Якщо аргумент `id` є об'єктом ідентифікатора, функція перевіряє його ідентифікатор бронювання. Якщо аргумент `id` є числом, функція перевіряє його відповідність ідентифікатору бронювання. Так само відбувається перевірка для асоціативного масиву та рядка. Якщо бронювання вже

існує, воно повертається. У випадку, якщо `id` не є об'єктом бронювання, функція створює новий об'єкт `EM_Booking` за допомогою отриманого `id` і повертає його.

```
function em_get_booking($id = false) {
    global $EM_Booking;
    //check if it's not already global so we don't instantiate again
    if( $EM_Booking instanceof EM_Booking ){
        if( is_object($id) && $EM_Booking->booking_id == $id->booking_id ){
            return apply_filters('em_get_booking', $EM_Booking);
        }else{
            if( is_numeric($id) && $EM_Booking->booking_id == $id ){
                return apply_filters('em_get_booking', $EM_Booking);
            }elseif( is_array($id) && !empty($id['booking_id']) && $EM_Booking->booking_id == $id['booking_id'] ){
                return apply_filters('em_get_booking', $EM_Booking);
            }elseif( is_string($id) && strlen($id) == 32 && $EM_Booking->booking_uuid == $id ){
                return apply_filters('em_get_booking', $EM_Booking);
            }
        }
    }
    if( $id instanceof EM_Booking ){
        return apply_filters('em_get_booking', $id);
    }else{
        return apply_filters('em_get_booking', new EM_Booking($id));
    }
}
```

Рисунок 3.23 – Створення функції для бронювання квитка

Створено функцію для отримання даних з бронювання (рис. 3.24). Функція `get_post()` виконує операції, пов'язані з отриманням даних про бронювання. Вона перевіряє, чи потрібно перевизначити доступність, і якщо змінна `EM_Bookings::$disable_restrictions` встановлена, то встановлює `override_availability` в `true`. Функція створює об'єкт `EM_Tickets_Bookings` та викликає дію `em_booking_get_post_pre`. Після цього функція обробляє дані з запиту, такі як `event_id`, `em_tickets` та `booking_comment`, перераховує доступні місця та ціну, отримує інформацію про особу.

```
function get_post( $override_availability = false ){
    if( EM_Bookings::$disable_restrictions ) $override_availability = true;
    $this->tickets_bookings = new EM_Tickets_Bookings($this);
    do_action('em_booking_get_post_pre', $this);
    $result = array();
    $this->event_id = absint($REQUEST['event_id']);
    if( isset($REQUEST['em_tickets']) && is_array($REQUEST['em_tickets']) && ($REQUEST['em_tickets'] || $override_availability) ){
        if( !$this->get_tickets_bookings()->get_post( $override_availability ) ){
            $this->add_error($this->tickets_bookings->get_errors());
        }
        $this->booking_comment = ( !empty($REQUEST['booking_comment']) ) ? wp_kses_data(wp_unslash($REQUEST['booking_comment'])) : '';
        if( !empty($this->booking_id) && $this->can_manage() ){
            $this->booking_tax_rate = ( !empty($REQUEST['booking_tax_rate']) && is_numeric($REQUEST['booking_tax_rate']) ) ? $REQUEST['booking_tax_rate'] : 0;
        }
        $this->get_spaces(true);
        $this->calculate_price();
        $this->get_person();
        $this->compat_keys();
    }
    return apply_filters('em_booking_get_post', empty($this->errors), $this);
}
```

Рисунок 3.24 – Створення функції для отримання даних з бронювання

`get_price` функція призначена для отримання ціни бронювання. Вона приймає два параметри: `$format` та `$format_deprecated`. Перевіряється, чи вже була обчислена ціна бронювання (`$this->booking_price`). Якщо ні, вона обчислюється за допомогою методу `calculate_price()` і застосовується фільтр `em_booking_get_price` до отриманої ціни. На останньому етапі, якщо вказано `$format`, ціна форматується за допомогою методу `format_price()`, інакше повертається округлена ціна до двох знаків після коми.

```
function get_price( $format = false, $format_deprecated = null ){
    if( $format_deprecated !== null ) $format = $format_deprecated;
    if( $this->booking_price === null ){
        $this->calculate_price();
        $this->booking_price = apply_filters('em_booking_get_price', $this->booking_price, $this);
    }
    if($format){
        return $this->format_price($this->booking_price);
    }
    return round($this->booking_price,2);
}
```

Рисунок 3.25 – Створення функції для отримання ціни квитка

Функція `em_get_event` призначена для отримання заходу (рис. 3.26). Вона приймає два параметри: `$id`, який є ідентифікатором заходу, і `$search_by`, який вказує, чи шукати захід за ідентифікатором заходу чи поста. Перш ніж шукати захід, функція перевіряє, чи існує об'єкт події у глобальному контексті і чи він належить до класу `EM_Event`. Якщо такий об'єкт існує, вона порівнює його з параметром `$id` та повертає об'єкт заходу, якщо вони співпадають. Якщо об'єкт `$id` не є об'єктом заходу, функція перевіряє, чи заданий `$search_by` є «event_id» чи «post_id», і відповідно порівнює ці значення з властивостями об'єкта заходу. Якщо знайдено відповідність, повертається об'єкт заходу. У разі використання кешу функція спробує знайти захід у кеші за ідентифікатором, згідно з вказаним `$search_by`. Якщо захід знайдено у кеші, повертається відповідний об'єкт.

```

function em_get_event($id = false, $search_by = 'event_id') {
    global $EM_Event;
    if( is_object($EM_Event) && get_class($EM_Event) == 'EM_Event' ){
        if( is_object($id) && $EM_Event->post_id == $id->ID ){
            return apply_filters('em_get_event', $EM_Event);
        }elseif( !is_object($id) ){
            if( $search_by == 'event_id' && $EM_Event->event_id == $id ){
                return apply_filters('em_get_event', $EM_Event);
            }elseif( $search_by == 'post_id' && $EM_Event->post_id == $id ){
                return apply_filters('em_get_event', $EM_Event);
            }
        }
    }
    if( is_object($id) && get_class($id) == 'EM_Event' ){
        return apply_filters('em_get_event', $id);
    }elseif( !defined('EM_CACHE') || EM_CACHE ){
        $event_id = false;
        if( is_numeric($id) ){
            if( $search_by == 'event_id' ){
                $event_id = absint($id);
            }elseif( $search_by == 'post_id' ){
                $event_id = wp_cache_get($id, 'em_events_ids');
            }
        }elseif( !empty($id->ID) && !empty($id->post_type) && ($id->post_type == EM_POST_TYPE_EVENT || $id->post_type == 'event-recurring') ){
            $event_id = wp_cache_get($id->ID, 'em_events_ids');
        }
        if( $event_id ){
            $event = wp_cache_get($event_id, 'em_events');
            if( is_object($event) && !empty($event->event_id) && $event->event_id ){
                return apply_filters('em_get_event', $event);
            }
        }
    }
    return apply_filters('em_get_event', new EM_Event($id,$search_by));
}

```

Рисунок 3.26 – Створення функції для отримання заходу

Функція `em_get_location` використовується для отримання об'єкта місця події за його ідентифікатором або ідентифікатором запису. Спочатку функція перевіряє, чи існує глобальна змінна `EM_Location` та чи є вона об'єктом класу `EM_Location`. Якщо так, відбувається перевірка, чи збігається ідентифікатор місця події з параметром `$id`, якщо так, то повертає цей об'єкт. Якщо об'єкт не знайдено, функція перевіряє кеш для швидкого доступу до даних. Якщо кешування ввімкнено, вона перевіряє кеш наявних ідентифікаторів місць подій. Якщо знайдено в кеші, повертається об'єкт місця події. Якщо ж об'єкт не знайдено ні в кеші, ні в глобальній змінній, функція створює новий об'єкт `EM_Location` з параметрами `$id` і `$search_by` і повертає його.

```
function em_get_location($id = false, $search_by = 'location_id') {
    global $EM_Location;
    if( is_object($EM_Location) && get_class($EM_Location) == 'EM_Location' ){
        if( is_object($id) && $EM_Location->post_id == $id->ID ){
            return apply_filters('em_get_location', $EM_Location);
        }elseif( !is_object($id) ){
            if( $search_by == 'location_id' && $EM_Location->location_id == $id ){
                return apply_filters('em_get_location', $EM_Location);
            }elseif( $search_by == 'post_id' && $EM_Location->post_id == $id ){
                return apply_filters('em_get_location', $EM_Location);
            }
        }
    }
    if( is_object($id) && get_class($id) == 'EM_Location' ){
        return apply_filters('em_get_location', $id);
    }elseif( !defined('EM_CACHE') || EM_CACHE ){
        //check the cache first
        $location_id = false;
        if( is_numeric($id) ){
            if( $search_by == 'location_id' ){
                $location_id = $id;
            }elseif( $search_by == 'post_id' ){
                $location_id = wp_cache_get($id, 'em_locations_ids');
            }
        }elseif( !empty($id->ID) && !empty($id->post_type) && $id->post_type == EM_POST_TYPE_LOCATION ){
            $location_id = wp_cache_get($id->ID, 'em_locations_ids');
        }
        if( $location_id ){
            $location = wp_cache_get($location_id, 'em_locations');
            if( is_object($location) && !empty($location->location_id) && $location->location_id ){
                return apply_filters('em_get_location', $location);
            }
        }
    }
    return apply_filters('em_get_location', new EM_Location($id,$search_by));
}
```

Рисунок 3.27 – Створення функції для отримання місцезнаходження заходу

Було створено функцію для визначення місця проведення заходу, яка надає користувачам необхідну інформацію для пошуку місця на карті.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи бакалавра було детально розглянуто процес розробки вебзастосунка з використанням плагіну Events Manager для ефективного бронювання квитків на різноманітні заходи, зокрема реалізовано функціонал для забезпечення зручного вибору кількості квитків, отримання необхідної інформації про ціни та доступні місця, а також створено відповідні сторінки, які включають обліковий запис, теми заходів, перелік подій, інформацію про організаторів, а також контактну форму.

4 ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ ВЕБЗАСТОСУНКА

4.1. Тестування ПЗ

Тестування веб-сайту передбачає комплексне вивчення його різних компонентів і функцій, щоб переконатися, що він працює належним чином. Одним з ключових аспектів тестування є функціональне тестування, яке перевіряє, чи всі функції сайту працюють відповідно до вимог. Це включає тестування користувацьких інтерфейсів, форм, процесів реєстрації, входу та інтерактивних елементів. Також оцінюється зручність використання, функціональність, сумісність, продуктивність, безпека та доступність вебзастосунку на різних пристроях і браузерах. Завдяки ретельному тестуванню, розробники можуть визначати та усувати проблеми, тим самим підвищивши загальну якість і надійність веб-сайту до його запуску (рис. 4.1 – рис. 4.3).

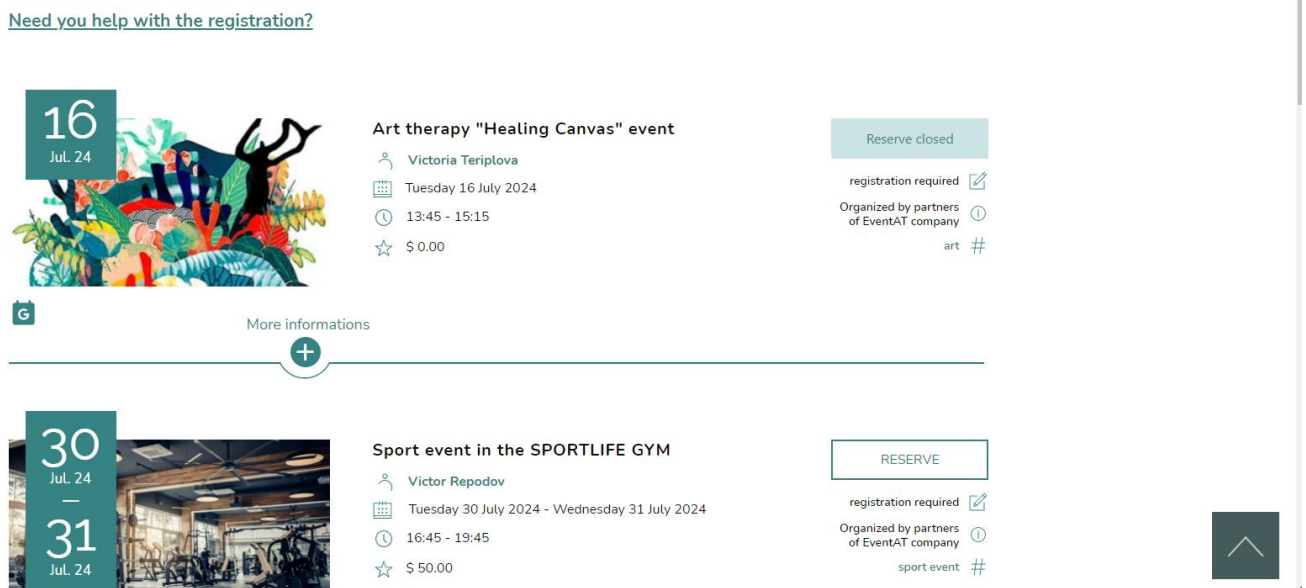


Рисунок 4.1 – Сторінка з бронюванням на десктопній версії

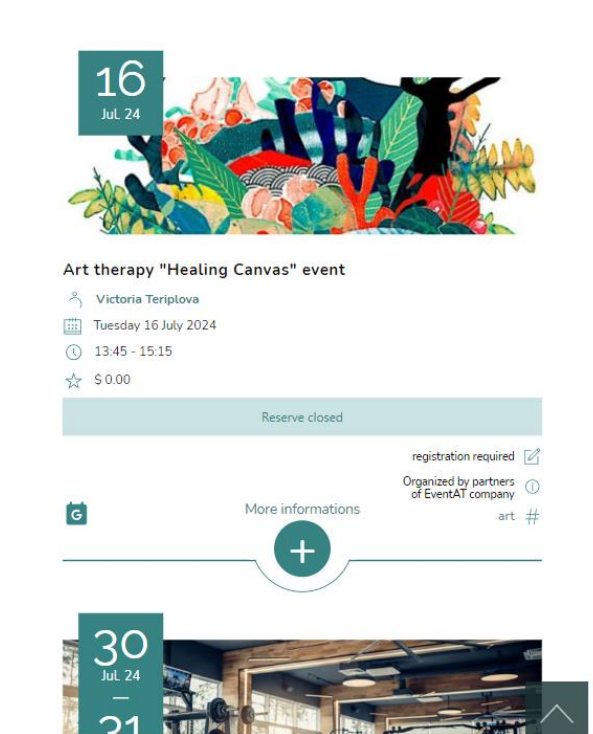


Рисунок 4.2 – Сторінка з бронюванням на планшетній версії



Рисунок 4.3 – Сторінка з бронюванням на мобільній версії

Швидкість загрузки веб-сайту грає важливу роль у забезпеченні зручності користувачів та впливає на його позиції в пошукових системах. Щоб перевірити швидкість загрузки вебзастосунку, можна використати спеціальні інструменти для аналізу швидкості сторінок. Проведемо аналіз деяких застосунків, та перевіримо швидкість роботи вебзастосунку для бронювання квитків на захід:

1. Google PageSpeed Insights – це надзвичайно корисний інструмент для веб-розробників і власників веб-сайтів. Цей онлайн сервіс від Google, для аналізу швидкості завантаження сторінок та надання докладної звітності з продуктивності веб-сайту [3]. Він враховує різні фактори, такі як час завантаження сторінки, розмір файлів, компресія зображень, ресурси CSS та JavaScript, інтеграцію кешування та багато іншого. Одним з ключових переваг PageSpeed Insights є те, що він не просто надає звіт про проблеми, але й рекомендує конкретні кроки для їх вирішення. Це може включати деякі підказки з оптимізації зображень, мінімізації CSS та JavaScript, використання кешування браузера і багато іншого. Такі рекомендації допомагають вдосконалити продуктивність сайту і забезпечити кращий користувацький досвід. Крім того, PageSpeed Insights враховує мобільність, що стає все більш важливим аспектом веб-розробки. Він аналізує швидкість завантаження на мобільних пристроях і надає рекомендації для поліпшення цієї частини веб-сайту (рис. 4.4 – рис. 4.6).



Рисунок 4.4 – Google PageSpeed Insights

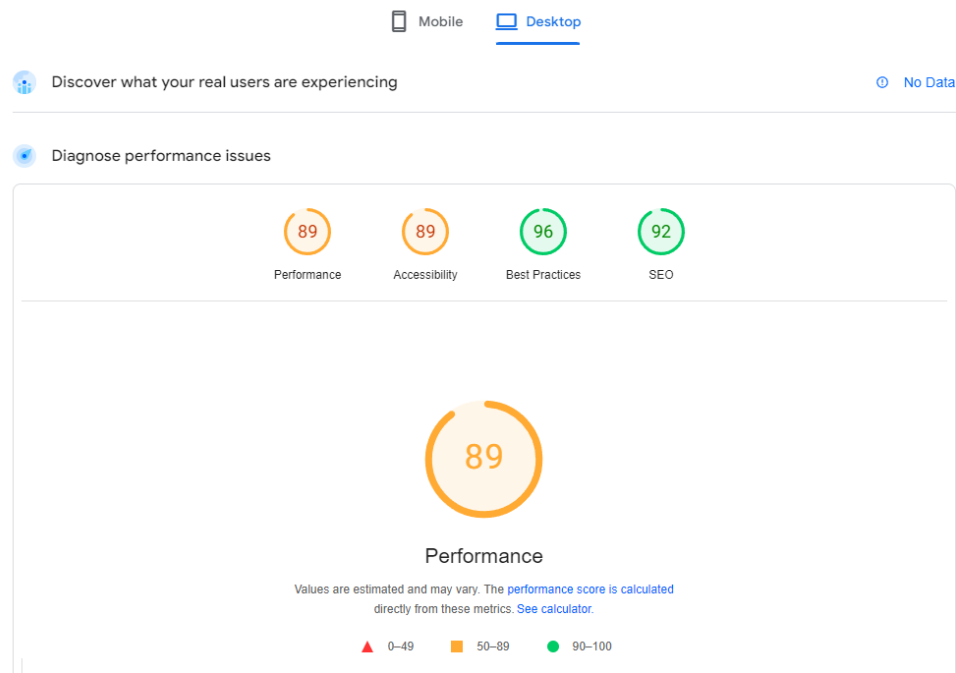


Рисунок 4.5 – Застосування PageSpeed Insights для перевірки швидкості веб-сайту

DIAGNOSTICS	
▲ Reduce initial server response time	— Root document took 1,600 ms
▲ Eliminate render-blocking resources	— Potential savings of 590 ms
▲ Serve images in next-gen formats	— Potential savings of 412 KiB
▲ Reduce unused CSS	— Potential savings of 90 KiB
▲ Properly size images	— Potential savings of 49 KiB
▲ Reduce unused JavaScript	— Potential savings of 457 KiB
▲ Avoid an excessive DOM size	— 2,506 elements
▲ Largest Contentful Paint element	— 1,240 ms
■ Minify CSS	— Potential savings of 2 KiB
■ Minify JavaScript	— Potential savings of 4 KiB
■ Serve static assets with an efficient cache policy	— 66 resources found
■ Ensure text remains visible during webfont load	
■ Does not use passive listeners to improve scrolling performance	
■ Defer offscreen images	— Potential savings of 738 KiB
■ Avoid serving legacy JavaScript to modern browsers	— Potential savings of 6 KiB
■ Avoid enormous network payloads	— Total size was 2,838 KiB

Рисунок 4.6 – Показ детального аналізу вебзастосунку

2. GTmetrix – це онлайн-інструмент для аналізу швидкості та оптимізації веб-сторінок. Він дозволяє перевіряти різні параметри продуктивності веб-сайту, такі як час завантаження, оптимізація зображень, кешування та інші. GTmetrix надає

детальні звіти з рекомендаціями щодо покращення швидкості та продуктивності веб-сайту, щоб забезпечити кращий користувацький досвід. Користуючись GTmetrix, можна вдосконалити роботу веб-сайту, забезпечити швидше завантаження сторінок і підвищити його ефективність (рис. 4.7 – рис. 4.8).

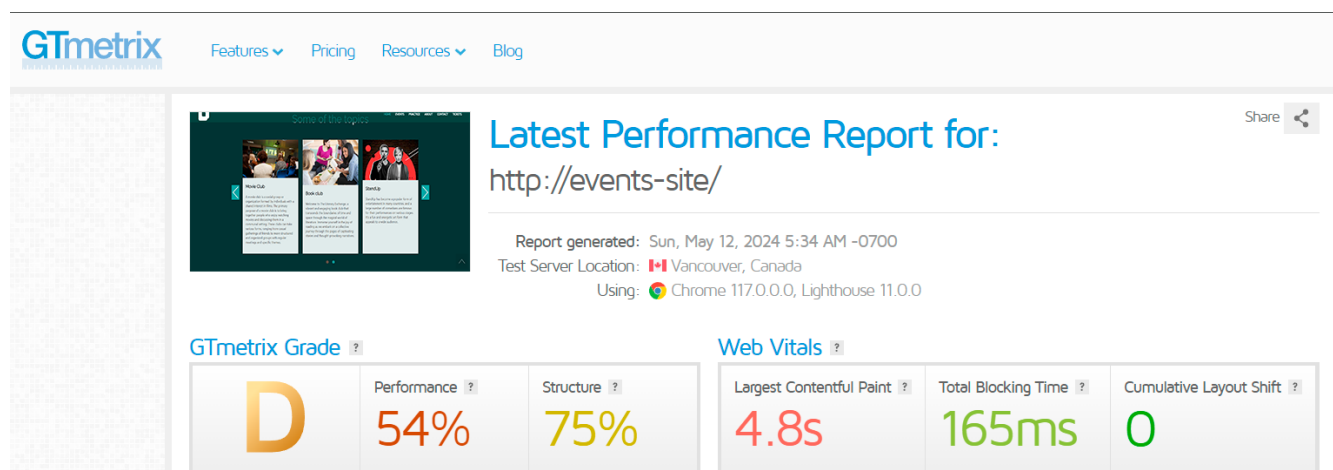


Рисунок 4.7 – Використання GTmetrix

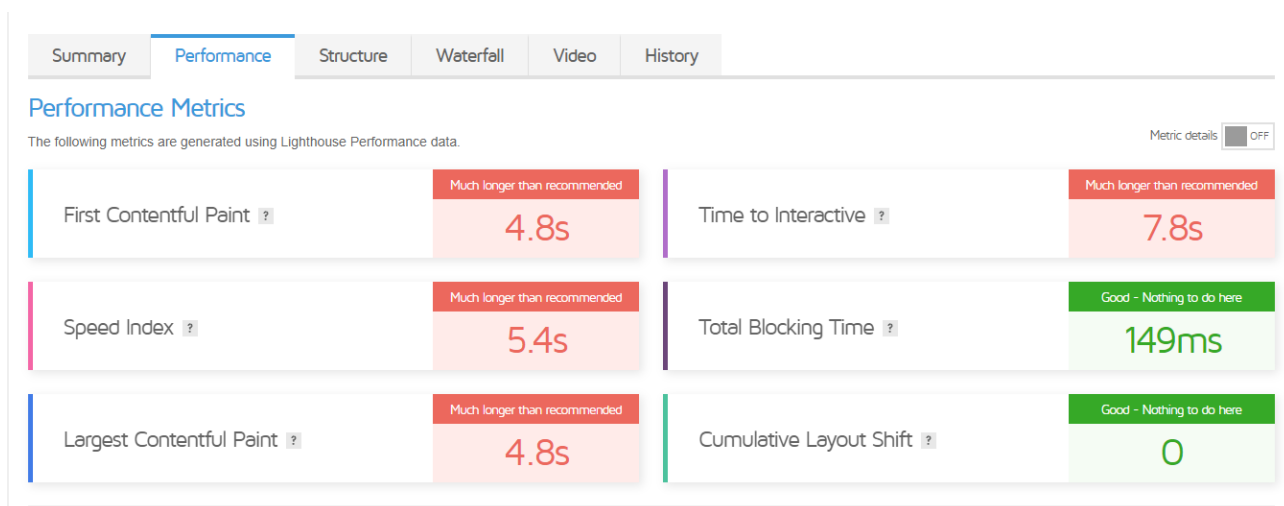


Рисунок 4.8 – Основні проблемні місця сайту

3. Web Page test – це онлайн-інструмент, який генерує ефективність на основі набору правил, розроблених у контексті продуктивності сторінки та пріоритетів користувачів (рис. 4.9). Принципи включають:

- кешування ресурсів;
- час прийому-передачі «клієнт-сервер»;
- розмір завантаження та вивантаження даних;

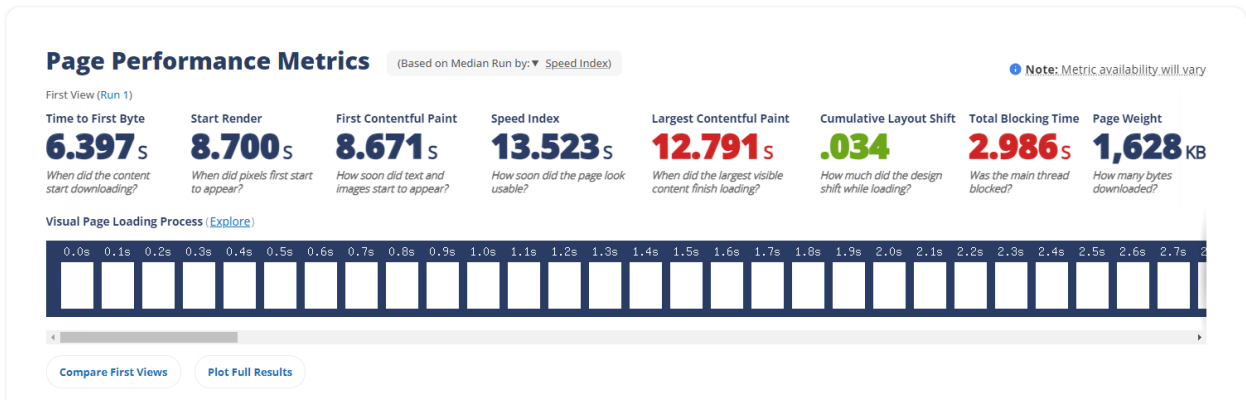


Рисунок 4.9 – Застосування вебзастосунку Web Page test

Таким чином, було проаналізовано швидкість роботи веб-застосунку, зокрема, час завантаження сторінок, швидкість відгуку сервера та інші показники продуктивності. Результати аналізу дозволили виявити проблемні аспекти, що можуть впливати на користувацький досвід.

4.2. Оптимізація вебзастосунку

Оптимізація вебзастосунку є складним і багатогранним процесом, що охоплює багато аспектів від продуктивності до безпеки і зручності користувачів. Вона спрямована на покращення швидкості завантаження, зменшення використання ресурсів і забезпечення найкращого досвіду користувачів. Одним із ключових аспектів оптимізації є зменшення розміру файлів шляхом мінімізації та стиснення CSS, JavaScript файлів та зображень. Це допомагає скоротити час завантаження сторінки і зменшити використання пропускну здатності. Кешування на стороні клієнта і сервера також відіграє важливу роль у цьому процесі, дозволяючи зберігати часто використовувані ресурси локально і зменшувати навантаження на сервер.

Асиметричне завантаження критичних ресурсів спочатку, а другорядних поступово, допомагає прискорити відображення сторінки для користувача. Серверна оптимізація включає вибір ефективного серверного програмного забезпечення, з оптимальними налаштуваннями, та балансування навантаження для розподілу трафіку між кількома серверами, що підвищує стабільність і швидкодію.

Оптимізація бази даних є ще одним важливим аспектом, що включає індексацію для прискорення запитів та написання ефективних SQL-запитів для уникнення надлишкових запитів [12]. SEO-оптимізація спрямована на покращення видимості у пошукових системах за допомогою правильних мета-тегів та унікального і релевантного контенту. Забезпечення безпеки вебзастосунка включає захист від DDoS-атак, а також використання SSL для шифрування даних і безпечної передачі інформації. Кросбраузерна сумісність забезпечує коректне відображення і функціонування вебзастосунка на різних браузерах і пристроях, що досягається шляхом ретельного тестування і адаптації дизайну та функціоналу [17]. Оптимізація вебзастосунка є безперервним процесом, що вимагає регулярного моніторингу та оновлення для підтримки високої продуктивності, надійності та зручності для користувачів.

Для оптимізації вебзастосунка використано WordPress плагін WP-Optimize. Це потужний плагін для оптимізації веб-сайту, який допомагає зберігати WordPress-сайт без змін в користувацькому інтерфейсі, зменшуючи розмір бази даних, швидшаючи завантаження сторінок і поліпшуючи загальну продуктивність (рис. 4.10). Ось кілька кроків, які ви зазвичай виконується за допомогою плагіну:

1. очищення бази даних. Плагін дозволяє легко видаляти непотрібні дані з бази даних, такі як застарілі записи, спамові коментарі, непотрібні таблиці тощо. Це допомагає зменшити розмір бази даних і поліпшує швидкість веб-сайту;

2. конвертація WebP. За рекомендацією Google WebP може зменшити розмір файлу на 34% порівняно з JPEG і на 26% порівняно з PNG.

3. Зменшення розміру файлів лише в один клік. Мінімізує HTML, JavaScript і CSS. Треба увімкнути або вимкнути налаштування для оптимізації файлів.
4. Відкладене завантаження. Є опція завантажувати лише зображення та частину веб-сторінки, коли вона видима для користувача. Таким чином, це допоможе загрузити сторінку швидше.

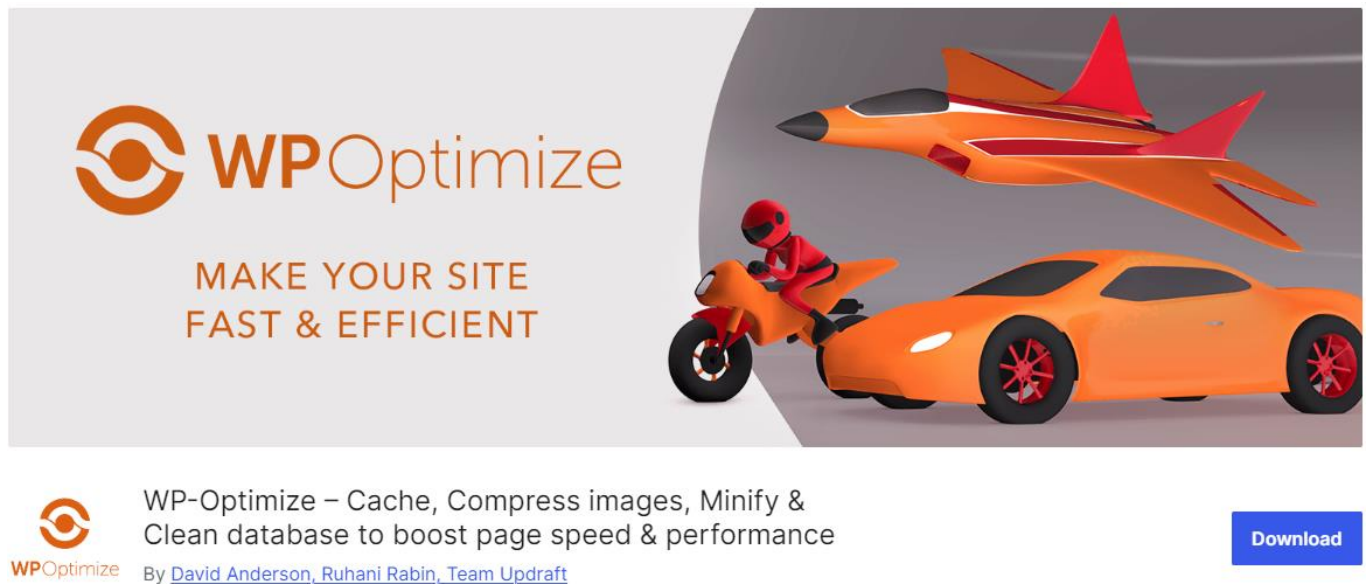


Рисунок 4.10 – Плагін WP-Optimize

Оптимізовано базу даних, очищено всі редакції постів, видалено спам і вимкнено коментарі, видалено непідтверджені коментарі (рис. 4.11 – рис. 4.13).

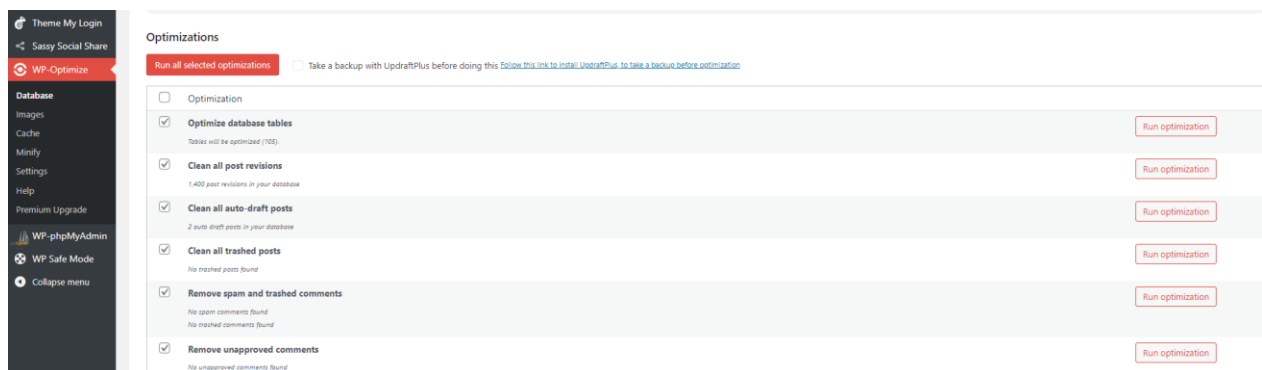


Рисунок 4.11 – Огляд плагіну WP-Optimize

Кафедра інженерії програмного забезпечення
Вебзастосунок бронювання квитків на заходи

vaiedkzyx3_wp_phpmyadmin_extension_errors_log	0	16.00 KB	16.00 KB	InnoDB	0 bytes
vaiedkzyx3_wpmailsmtp_debug_events	0	16.00 KB	0 bytes	InnoDB	0 bytes
vaiedkzyx3_wpmailsmtp_tasks_meta	0	16.00 KB	0 bytes	InnoDB	0 bytes
vaiedkzyx3_yoast_indexable	2,609	1.52 MB	656.00 KB	InnoDB	4.00 MB
vaiedkzyx3_yoast_indexable_hierarchy	2,635	128.00 KB	208.00 KB	InnoDB	0 bytes
vaiedkzyx3_yoast_migrations	24	16.00 KB	16.00 KB	InnoDB	0 bytes
vaiedkzyx3_yoast_primary_term	59	16.00 KB	32.00 KB	InnoDB	0 bytes
vaiedkzyx3_yoast_seo_links	1,214	240.00 KB	128.00 KB	InnoDB	0 bytes
106 Tables	365,335	152.86 MB	16.47 MB	-	31.00 MB

Total size of database: 169.33 MB

Рисунок 4.12 – Огляд таблиць бази даних

<input type="checkbox"/>	Optimization	
<input checked="" type="checkbox"/>	Optimize database tables	Run optimization
<input type="checkbox"/>	Clean all post revisions	Run optimization
	<small>1,400 post revisions in your database</small>	
<input type="checkbox"/>	Clean all auto-draft posts	Run optimization
	<small>2 auto draft posts in your database</small>	
<input checked="" type="checkbox"/>	Clean all trashed posts	Run optimization
<input checked="" type="checkbox"/>	Remove spam and trashed comments	Run optimization
<input checked="" type="checkbox"/>	Remove unapproved comments	Run optimization

Рисунок 4.13 – Процес оптимізації файлів

Загалом, WP-Optimize надає інструменти для моніторингу стану сайту та створення звітів про його продуктивність, що допомагає вчасно виявляти та вирішувати проблеми [20]. Використання даного плагіну дозволяє підтримувати WordPress-сайт в оптимальному стані, забезпечуючи його швидкість, продуктивність і безпеку.

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи бакалавра було детально розглянуто процес тестування та оптимізації вебзастосунка. Було розглянуто три онлайн-інструмента для перевірки швидкості роботи веб-сайту. Розглянуто вебзастосунок на різних розширеннях екрану, а саме на десктопній, планшетній та мобільній версії. Оптимізовано вебзастосунок завдяки плагіну WP-Optimize, та розглянуто переваги використання даного плагіну.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було створено вебзастосунок для бронювання квитків на заходи. Цей вебзастосунок призначений для спрощення процесу бронювання квитків на різноманітні заходи, надаючи користувачам можливість отримати всю необхідну інформацію про події та зручно забронювати квиток на захід. Вебзастосунок пропонує інтерактивний інтерфейс, який забезпечує простоту та ефективність користування. Користувачі можуть бронювати квитки та переглядати деталі заходів, такі як дата, час, місце проведення та опис заходу.

Зазначену мету досягнуто завдяки виконанню наступних завдань:

- аналіз предметної сфери та аналогічних застосунків;
- проєктування та моделювання системи;
- програмна реалізація вебзастосунка;
- тестування та оптимізація вебзастосунка.

Для подальшого вдосконалення вебзастосунку з бронювання квитків на заходи, можна розглянути розробку додаткових модулів доступу. Це можуть бути модулі для інтеграції з соціальними мережами, системами онлайн-платежів, а також модулі для корпоративного використання, що дозволять організаторам заходів ефективніше керувати бронюванням. Крім того, можливе видозмінення інтерфейсу вебзастосунку з метою підвищення його інформативності, привабливості та зручності для користувачів. Це включає в себе оновлення дизайну для покращення візуальної привабливості, впровадження нових елементів навігації для зручнішого користування, а також оптимізацію відображення інформації про події для забезпечення максимальної доступності і зрозумілості.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Варіанти використання та сценарії. URL: <https://www.maxzosim.com/use-cases-and-scenarios/> (дата звернення: 08.02.2024).
2. Корнієнко С. К., Качан О. І. База даних. Робота з СКБД MySQL. Запоріжжя: ЗНТУ, 2016. 34с.
3. Оптимізація сайту на WordPress для Google PageSpeed. URL: <https://redstone.media/wordpress-google-page-speed> (дата звернення: 11.02.2024).
4. Посібник з налаштування WordPress. URL: <https://hostiq.ua/wiki/ukr/wordpress-settings/> (дата звернення: 15.02.2024).
5. Створення схеми діаграми станів. URL: <http://surl.li/hznew> (дата звернення: 21.02.2024).
6. Створення сценаріїв використання. URL: <http://surl.li/tiduf> (дата звернення: 28.02.2024).
7. Що таке діаграма класів UML. URL: <https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/> (дата звернення: 05.03.2024).
8. Advanced Custom Fields Documentation. URL: <https://www.advancedcustomfields.com/resources/> (дата звернення: 12.03.2024).
9. Deployment Diagram Tutorial. URL: <https://www.lucidchart.com/pages/uml-deployment-diagram> (дата звернення: 20.03.2024).
10. Events Manager Classes. URL: <https://github.com/bippo/events-manager-wordpress/blob/master/classes/em-event.php> (дата звернення: 25.03.2024).
11. Events Manager Documentation. URL: <https://wp-events-plugin.com/documentation/> (дата звернення: 30.03.2024).

12. Eight Ways to Optimize Your WordPress Site's Performance. URL: <https://wordpress.com/go/tutorials/eight-ways-to-optimize-your-wordpress-sites-performance/> (дата звернення: 02.04.2024).
13. JQuery documentation. URL: <https://jquery.com/> (дата звернення: 03.04.2024).
14. PHP documentation. URL: <https://www.php.net/manual/en/intro-what-is.php> (дата звернення: 06.04.2024).
15. UML Class Diagram Tutorial. URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/> (дата звернення: 12.04.2024).
16. UML Use Case Diagram Tutorial. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram> (дата звернення: 18.04.2024).
17. Web Performance Optimization: Techniques and Tools to Improve Website Speed. URL: <https://stfalconcom.medium.com/web-performance-optimization-techniques-and-tools-to-improve-website-speed-458754219aa0> (дата звернення: 26.04.2024).
18. WordPress documentation. URL: <https://wordpress.org/documentation/> (дата звернення: 02.05.2024).
19. WordPress SEO. URL: <https://hostiq.ua/blog/ukr/wordpress-seo/> (дата звернення 08.05.2024).
20. WP-Optimize – Cache, Compress images, Minify & Clean database to boost page speed & performance. URL: <https://wordpress.org/plugins/wp-optimize/> (дата звернення: 14.05.2024).