

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
(повне найменування факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему Розробка інформаційного сайту культурних
(назва теми) заходів міста

Виконав: студент 4 курсу, групи КНТ 510 сп

спеціальності 123 «Комп'ютерна інженерія»
(код і найменування спеціальності)

Освітня програма (спеціалізація) _____
«Комп'ютерна інженерія»

ЧАРОВАТИЙ П. Г.

(прізвище та ініціали)

Керівник ЗЕЛЕНЬОВА І.Я.

(прізвище та ініціали)

Рецензент КУЛЯБА-ХАРИТОНОВА Т.І

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування вищого навчального закладу)

Факультет	Комп'ютерних наук і технологій
Кафедра	«Комп'ютерні системи та мережі»
Ступінь вищої освіти	бакалаврський
Спеціальність	123 Комп'ютерна інженерія
	(код і найменування)
Освітня програма (спеціалізація)	Комп'ютерна інженерія
	(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри КУДЕРМЕТОВ Р.К.
« » 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ЧАРОВАТОГО Павла Геннадійовича
(прізвище, ім'я, по батькові)

- Тема проекту (роботи) Розробка інформаційного сайту культурних заходів міста
керівник проекту (роботи) к.т.н., доцент, ЗЕЛЕНЬОВА Ірина Яківна,
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)
затверджені наказом вищого навчального закладу від «24» березня 2023 року № 71
- Строк подання студентом проекту (роботи) 23 травня 2023 року
- Вихідні дані до проекту (роботи) опис предметної області, технічне завдання
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Аналітичний огляд;
 - Проектування програмного забезпечення;
 - Тестування та реалізація сайту.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Плакат 1 - Аналоги сайтів;
Плакат 2 - Діаграма прецедентів і архітектура сайту культурних заходів міста;
Плакат 3 - Макети інтерфейсу сайту;
Плакат 4 - Інтерфейс користувача;
Плакат 5 - Тестування роботи сайту.

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ЗЕЛЕНЬОВА І.Я., доцент		
Нормоконтроль	ЩЕРБАК Н.В., ст. викладач		

7. Дата видачі завдання «01» березня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/П	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	01.03.2023 р.	
2	Аналіз предметної області.	15.03.2023 р.	
3	Вибір мови програмування та інших технологій розробки.	20.03.2023 р.	
4	Розробка архітектури програми.	01.04.2023 р.	
5	Розробка програми.	10.04.2023 р.	
6	Тестування та експериментальне дослідження програмного забезпечення.	01.05.2023 р.	
7	Оформлення пояснювальної записки та документів до неї.	10.05.2023 р.	
8	Оформлення графічного матеріалу	17.05.2023 р.	
9	Оформлення допоміжного матеріалу	19.05.2023 р.	

Студент(ка)

 Павло ЧАРОВАТИЙ
(підпис) (Ім'я прізвище)

Керівник проєкту (роботи)

 Ірина ЗЕЛЕНЬОВА
(підпис) (Ім'я прізвище)

РЕФЕРАТ

ПЗ: 55 с., 21 рис., 2 таблиць, 1 додаток, 12 джерел.

ВЕБСАЙТ, HTML, CSS, SASS, JAVASCRIPT, SSR, NEXT.JS, СТОРІНКА,
БРОНЮВАННЯ, ЕЛЕКТРОННА ПОШТА.

Об'єкт дослідження – програмні засоби для розробки вебсайту.

Предмет дослідження – вебсайт культурних заходів міста.

Мета дослідження – створення власного вебсайту для перегляду культурних заходів свого чи закордонного міста.

Клієнт може вибрати в якому місті та для якого заходу чи події він може зареєструватись.

Залучення нових клієнтів за рахунок оновлення та вдосконалення сайту, додавання нових функцій;

В результаті було створено інформаційний сайт культурних заходів міста. Цей вебсайт призначений для огляду інформації потрібної користувачу. Збереження адреси електронної пошти та реєстрації у заході, яку ввів користувач для реєстрації та подальшого підтримування зв'язку.

ЗМІСТ

Перелік скорочень та умовних познач	7
Вступ	8
1 Аналітичний огляд	9
1.1 Опис предметної області	9
1.2 Огляд аналогів	10
1.3 Алгоритм роботи	13
1.4 Вимоги до вебсайту	15
1.5 Обґрунтування вибору інструментів розробки	16
1.6 Огляд фреймворків	20
2 Проектування програмного забезпечення	27
2.1 Постановка задачі розробки	27
2.2 Моделювання прецедентів	28
2.3 Архітектура програмного забезпечення	31
2.4 Стек технологій розробки сайту	33
2.5 Реєстрація на обраний івент	34
2.6 Проектування інтерфейсу сайту	35
2.7 Інтерфейс користувача	38
3 Тестування та реалізація сайту	41
3.1 Тестування сайту	41
3.2 Рекомендації користувачу	44
Висновки	45
Перелік джерел посилання	46
Додаток А	48

Графічний матеріал:

Плакат 1 - Аналоги сайтів;

Плакат 2 - Діаграма прецедентів і архітектура сайту культурних заходів міста;

Плакат 3- Макети інтерфейсу сайту;

Плакат 4 - Інтерфейс користувача;

Плакат 5 – Тестування роботи сайту.

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

JSON	– JavaScript Object Notation;
SEO	– Search Engine Optimization
CSS	– Cascading Style Sheets
SAAS	– Syntactically Awesome Stylesheets
URL	– Uniform Resource Locator
AMD	– Advanced Micro Devices
HDD	– Hard Disk Drive
SSD	– Solid-State Drive
RAM	– Random Access Memory
VRAM	– Video Random Access Memory
HTML	– HyperText Markup Language
OS	– Operation System
DOM	– Document Object Model
JSX	– Javascript XML
XML	– eXtensible Markup Language
MVC	– Model View Controller
MVVM	– Model View ViewMode
SPA	– Single Page Application
CSR	– Client-Side Rendering
SSR	– Server-Side Rendering
SSG	– Static Site Generation
API	– Application Programming Interface
JS	– Javascript
HTTP	– HyperText Transfer Protocol

ВСТУП

В наш час інтернет вже став невід'ємною частиною життя людини, завдяки ньому ми взаємодіємо із всілякими видами інформації.

В даний час діяльність людини стала сильно залежати від інформаційних технологій і побудованих на їх основі інформаційних систем, що застосовуються для швидкого доступу до різноманітних джерел інформації та для обміну інформацією.

Інтернет надає можливість компаніям виходити на глобальний ринок і вести бізнес без фізичних обмежень території. Він відкриває доступ до нових аудиторій і дозволяє залучати клієнтів з різних країн світу. Цифрові канали комунікації: соціальні мережі, електронна реклама, контент-маркетинг дозволяють компаніям ефективно спілкуватися зі своєю аудиторією і залучати нових клієнтів.

Завдяки глобальному розвитку веб-ресурсів пошук культурних заходів і реєстрація на ці події та заходи у різних містах світу стала набагато легшою.

У наш час мабуть у кожного підприємства є власний сайт. Культурний відпочинок та відвідування культурних заходів в своєму місті чи за кордоном завжди цікавить та дає змогу розвiятися багатьом людям. Тому сайт зроблений з цією метою буде приносити прибуток місту і швидше знаходити клієнтів для підвищення привабливості міста.

Розробка вебсайту культурних заходів міста є доречною в наш час для людей які шукають місце для подальшого відпочинку, інформацію про цей або інші заходи. Також це додає простоти і зручності доступу до інформації про культурні заходи міста. Тому що, користувачам не потрібно шукати розсіяну інформацію в різних джерелах, оскільки це все можна подивитися в одному місті. Також це активно сприяє розвитку культурного життя міста, допомагаючи привернути більше уваги до культурних заходів та підтримувати місцевих артистів і виконавців. Відвідування культурних заходів, сприяє зростанню культурної свідомості та сприяє розвитку творчих індустрій міст.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Опис предметної області

Вебсайт – це може бути одна чи якась сукупність сторінок в інтернеті. Всі сторінки сайту об'єднуються доменним ім'ям, якоюсь спільною темою [1].

Один сайт – це об'єднання всіх сторінок під одним доменним ім'ям.

Сайти потрібні для різних цілей: продавати необхідні речі іншим людям, читати і здобувати безцінні знання, тримати зв'язок з іншими людьми, дивитись відео, розважати. Є багато причин для створення сайту.

Кожен сайт повинен мати якісний зовнішній вигляд і чітку структуру сайту. А також контент сайту, яку розробник або власник розміщує на сторінках. Це може бути текст, медіа, посилання.

Сайт культурних заходів міста є онлайн-платформою, яка надає інформацію про різноманітні культурні події та активності, що відбуваються в місті. Основною метою сайту є забезпечення користувачам доступу до повної та актуальної інформації про цікаві заходи, які вони можуть відвідати.

Одним із ключових функціональних елементів сайту є пошуковий інструмент, який дозволяє користувачам швидко знаходити заходи за різними критеріями. Також важливим аспектом сайту є також можливість взаємодії з організаторами культурних заходів.

На сайті можна знайти розклад або календар подій, де вказані дати, час та місце проведення різних культурних заходів. Це можуть бути концерти, музичні вистави.

Для кожного заходу надається детальна інформація, така як назва заходу, опис, інформація про виконавців або учасників, вартість квитків або умови відвідування. Користувачі можуть ознайомитись з категоріями заходів, такими як музика, театр, що дозволяє їм швидко знайти події, які їх цікавлять.

Додатковими ресурсами на сайті можуть бути фото- та відеоматеріали з минулих заходів, рецензії або огляди культурних подій, інтерв'ю з виконавцями або організаторами, що розширює інформаційну базу сайту.

Також, сайт може надавати можливість користувачам реєструватись для отримання сповіщень про майбутні заходи або можливість бронювання квитків онлайн.

Цей сайт створений з метою сприяти розвитку культурного життя в місті, зближення громади та знайомство з новими людьми, та надання інформації, що сприяє залученню людей до культурних подій та розвитку культурного туризму.

Загалом, сайт культурних заходів міста створений з метою зближення культурних подій та аудиторії та забезпечення доступу до різноманітних культурних заходів для всіх зацікавлених осіб.

1.2 Огляд аналогів

Аналогом вебсайту з тематики культурних заходів міста є «Songkick» [2], цей вебсайт спеціалізується на концертах та музичних заходах. Він надає інформацію про майбутні виступи артистів у різних містах, можливість підписки на сповіщення про концерти та купівлю квитків. Цей сайт зарубіжний і тому весь інтерфейс сайту там тільки на англійській мові, тому він не має великої зацікавленості серед більшості європейських і українських користувачів. Також до недоліків можна віднести відсутність маловідомих заходів, він залишає непомітні або менш популярні події непоміченими. Головна сторінка сайту «Songkick» приведена на рисунку 1.1.

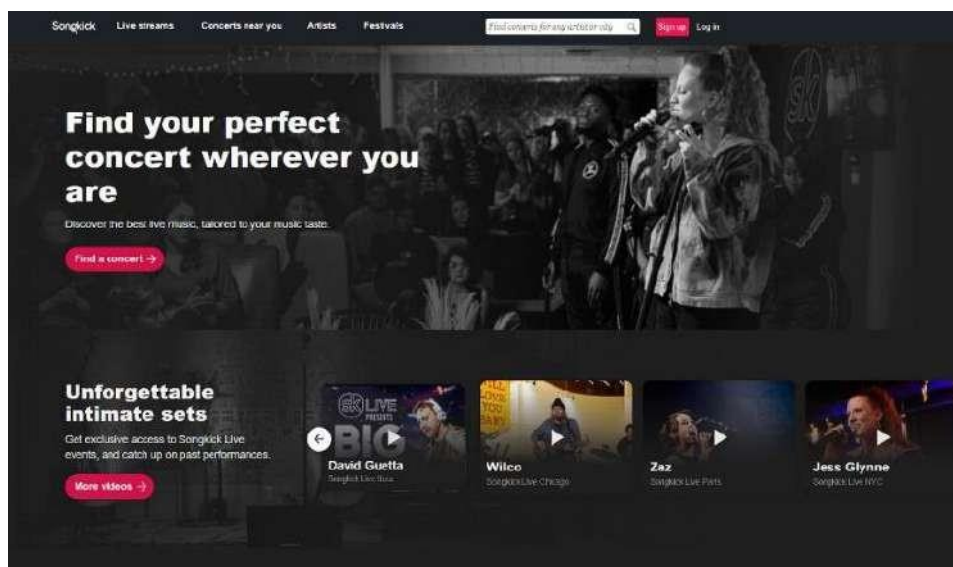


Рисунок 1.1 – Головна сторінка сайту «Songkick»

Ще одним прикладом сайту з івентами є сайт «Eventbrite» [3] є популярною платформою для організації та пошуку різних подій, включаючи культурні заходи.

Однією з багатьох особливостей сайту є те, що на цьому сайті можна створити власний івент і тому там можуть бути і маловідомі заходи.

До мінусів сайту можна віднести відсутність української мови. Не зручний і нагромаджений інтерфейс користувача. Головна сторінка сайту «Eventbrite» приведена на рисунку 1.2.

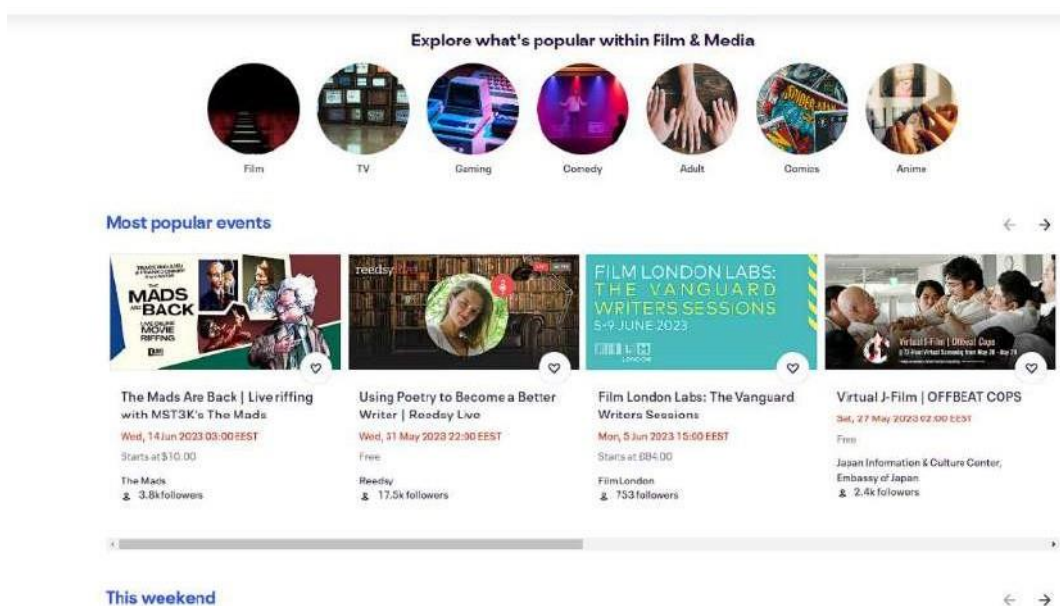


Рисунок 1.2 – Головна сторінка сайту «Eventbrite»

Ще одним аналогом є сайт «internet-bilet». Ця компанія існує більше 10 років, з моменту заснування, і цей сервіс дає людям можливість в режимі реального часу вибирати місця в залі, бронювати і оплачувати квитки через Інтернет. Цей сайт не має значних недоліків, в нього зрозумілий і зручний інтерфейс, як для комп'ютерів так і для мобільних телефонів і багатий функціонал.

До переваг цього сайту також можна віднести можна віднести:

- можливість змінювати мову інтерфейсу і контенту;
- вебсайт надає широкий вибір культурних заходів, включаючи концерти, виставки, фестивалі, театральні вистави та багато іншого;
- детальний опис заходів та івентів;
- є засоби зворотнього зв'язку;
- Зручний пошук та фільтрація, що дозволяють користувачам швидко знаходити події за категоріями, місцем, датою та іншими параметрами;
- реєстрація власного кабінету користувача, де можна подивитись відвідуванні заходи і отримувати сповіщення про майбутні концерти та події на пошту або sms;
- соціальна інтеграція, яка дозволяє користувачам ділитися подіями, запрошеннями та враженнями з іншими людьми через соціальні мережі.

Сайт «internet-bilet» приведен на рисунку 1.3.



Рисунок 1.3 – Головна сторінка сайту «internet-bilet»

Наведемо таблицю (табл 1.1.) для порівняння аналогів сайтів, критерії порівняння:

- засіб зворотнього зв'язку з клієнтом – користувач, відправляє електронну пошту для реєстрації на подію;
- можливість зареєструватися на обрану подію;
- зрозумілий і мінimalістичний інтерфейс;
- детальний опис заходів та івентів.

Таблиця 1.1 – Порівняння аналогів вебсайтів

Назва вебсайту	засіб зворотнього зв'язку з клієнтом	можливість зареєструватися на обрану подію	Зручний інтерфейс	детальний опис заходів
Розроблюваний застосунок	+	+	+	+
internet-bilet	+	+	+	+
Songkick	+	+	-	-
Eventbrite	+	+	-	+

Після детального аналізу порівняльної таблиці можна зробити висновок, що всі наведені вебсайти мають багатий функціонал та успішно справляються зі своєю головною функцією - надання користувачам можливості отримувати необхідну інформацію про культурні події міста.

1.3 Алгоритм роботи

Сучасні веб-сайти можуть використовувати різні типи зберігання даних, включаючи бази даних і формат JSON [4]. Обидва підходи мають свої переваги та

використовуються в різних ситуаціях, в залежності від потреб проекту. JSON є легким форматом обміну даними, що використовується для передачі структурованої інформації між системами.

Деякі аспекти використання JSON у порівнянні з базами даних:

Переваги використання JSON:

- простота, JSON є простим у використанні та розумінні форматом даних. Він використовує зрозумілу структуру об'єктів та масивів, що дозволяє легко зберігати та обробляти дані;
- легка інтеграція, JSON добре підходить для веб-сервісів та API, оскільки його можна легко передавати між клієнтом та сервером [5]. Він підтримується більшістю мов програмування та зручно використовується для передачі структурованих даних;
- гнучкість, JSON дозволяє зберігати різноманітні типи даних, включаючи об'єкти, масиви, рядки, числа, логічні значення. Це дозволяє створювати розгалужені структури даних та працювати з ними зручним способом;
- простота резервного копіювання та відновлення, JSON-файли можуть бути легко експортовані та імпортовані, що полегшує процес резервного копіювання та відновлення даних.

Хоча JSON і має деякі недоліки у порівнянні з базами даних, JSON не надає складних можливостей для виконання складних запитів та пошуку даних, як це робиться в базах даних і також відсутність механізмів безпеки та контролю доступу, JSON сам по собі не надає механізмів безпеки та контролю доступу до даних. Тому ці аспекти потрібно реалізовувати додатково, щоб забезпечити захист даних.

У моєму вебсайті відсутня потреба в складних запитах, тому я використовую формат JSON для зберігання та передачі даних. JSON є простим у використанні і дозволяє ефективно представляти структуровану інформацію без необхідності в складних запитів до бази даних.

Так як JSON не надає механізмів безпеки тому, у моєму коді, коли користувач натискає на кнопку, спочатку відбувається валідація адреси електронної пошти за

допомогою регулярного виразу. Якщо валідація успішна (адреса електронної пошти відповідає потрібному формату), тоді відбувається запит на сервер. Запит відправляється на встановлений шлях методом POST [6].

Отже, вибір між використанням бази даних та JSON залежить від конкретних потреб проекту.

1.4 Вимоги до вебсайту

Ефективна робота веб-сайту, пов'язаного з культурними заходами міста, можлива лише тоді, коли проектування та розробка сайту враховують широкий спектр критеріїв якості. Спочатку особливу увагу треба звернути на привабливий та естетично збалансований дизайн, який буде привертати увагу відвідувачів і викликати позитивні емоції. Крім того, системні критерії грають важливу роль у просуванні сайту. Важливо враховувати технічні аспекти, забезпечуючи безперебійну роботу, надійність та швидкість завантаження сторінок [8].

Для забезпечення успіху сайту культурних заходів міста також необхідно враховувати SEO-оптимізацію. Дотримання рекомендацій щодо структури сторінок, використання ключових слів у мета-тегах та заголовках, побудова дружніх URL-адрес, а також оптимізація зображень для швидкого завантаження - це лише кілька прикладів стратегій, що допоможуть забезпечити видимість сайту у пошукових системах та залучення більше відвідувачів.

Естетичні та психологічні критерії:

- привабливий та естетичний дизайн, що відображає характер культурних заходів і міста в цілому;
- використання візуальних елементів, які стимулюють інтерес відвідувачів і привертають їх увагу;
- створення позитивного емоційного досвіду для користувачів.

Системні критерії:

- надійність сайту, що забезпечує безперебійну роботу та запобігає виникненню помилок;
- доступність сайту, забезпечуючи його доступність для різних користувачів, включаючи людей з обмеженими можливостями;
- точне та зрозуміле викладення інформації про культурні заходи, дати, місце проведення, час та деталі подій;
- лаконічна структура сайту, що сприяє легкому знаходженню інформації та навігації по сайту;
- контроль над даними користувачів та їх безпечне зберігання;
- можливість безперебійної роботи сайту під час великих навантажень під час популярних культурних заходів.

Адаптивний дизайн:

- розробка сайту з урахуванням адаптивного дизайну, щоб забезпечити коректне відображення на будь-яких пристроях, включаючи мобільні пристрої;
- забезпечення чіткого відображення зображень та інших візуальних елементів на різних розмірах дисплея.

Соціальна інтеграція:

- можливість легкого поширення контенту сайту через кнопки соціальних мереж, що дозволить відвідувачам легко ділитися цікавими подіями зі своїми друзями та спільнотою;

Ці критерії можуть служити основою для розробки якісного та успішного сайту культурних заходів міста, привабливого відвідувачів та задовольняючого їх потреби.

1.5 Обґрунтування вибору інструментів розробки

CSS - мова опису стилів, яка використовується для оформлення і візуального оформлення веб-сторінок.

CSS дозволяє задавати кольори, шрифти, розміри, відступи, рамки та інші властивості елементів HTML.

Основні особливості CSS:

- CSS використовує селектори для вибору елементів, до яких будуть застосовуватися стилі. Селектори можуть базуватися на ідентифікаторах, класах, типах елементів;
- CSS надає широкий набір властивостей, які можна задавати для елементів HTML. Це включає розміри, кольори, шрифти, фонові зображення, позиціонування;
- каскадування та спадкування: CSS використовує концепцію каскадування, що означає, що стилі можуть бути задані на різних рівнях ієрархії. Крім того, стилі можуть спадкуватися від батьківських елементів до дочірніх.

SASS (Syntactically Awesome Style Sheets) - це метамова CSS, яка додає багато корисних функцій і можливостей до звичайного CSS. SASS дозволяє використовувати змінні, вкладені стилі, міксіни, умовні оператори та інші функції для полегшення написання і керування стилями.

Основні особливості SASS:

- перевикористання коду, SASS дозволяє використовувати змінні, міксіни та імпорти, що дозволяє перевикористовувати код і забезпечує більшу модульність і скорочення кількості дублюючого коду. Можна визначити змінні для кольорів, розмірів шрифтів, відступів і використовувати їх у багатьох місцях з одним змінним оголошенням;
- вкладеність SASS дозволяє вкладати стилі одного елемента всередину стилів іншого елемента. Це дозволяє структурувати код і зберігати стилі відносно їх контексту. Замість повторення селекторів у CSS, можна вкладати їх відповідно до HTML-структури, що полегшує читання та редагування стилів;
- міксіни SASS дозволяє створювати міксіни, які є шаблонами стилів, що можуть бути використані повторно. Міксіни можуть містити набір стилів, які застосовуються до елементів, і їх можна викликати в будь-якому місці, де потрібно

застосувати ці стилі. Це дозволяє легко створювати і використовувати набір стилів для різних елементів;

- умовні оператори SASS підтримує умовні оператори, такі як if-else, що дозволяють застосовувати різні стилі в залежності від певних умов. Це дає можливість динамічно змінювати стилі на основі певних умов або параметрів;
- імпорт SASS дозволяє імпортувати стилі з інших файлів, що дозволяє розділити стилі на кілька файлів для зручності управління. Можна розділити стилі на модулі, наприклад, окремий файл для кнопок, окремий для навігації, окремий для типографії і потім імпортувати їх у основний файл стилів.

CSS і SASS є потужними інструментами для оформлення та стилізації веб-сторінок [9].

JavaScript - це мова програмування, яка забезпечує можливість реалізації складних рішень у веб-документах. Вона дозволяє зробити сторінки веб-сайтів більш інтерактивними та обробляти дії користувачів. JavaScript є об'єктно-орієнтованою мовою, яка підтримується додатками, що працюють з дизайном веб-сайтів.

JavaScript має багато можливих застосувань, таких як:

- відображення періодично оновлюваного контенту, наприклад, інтерактивних карт;
- створення якісної анімації та графічних об'єктів у форматі 2D/3D;
- прокрутка відеозапису у медіапрогравачі;
- збереження даних у змінних;
- активація певного коду відповідно до сценаріїв, що відбуваються на сторінці веб-сайту;
- створення автоматично оновлюваного контенту;
- управління мультимедійними можливостями, такими як робота з відео та анімація зображень.

JavaScript, як і будь-яка інша мова програмування, має свої обмеження:

- обмежена функціональність для читання та завантаження файлів JavaScript, працюючи в браузері, має обмежені можливості щодо читання та запису файлів на локальному комп'ютері користувача;

- Javascript не містить деяких корисних інструментів має відсутність стандартної бібліотеки, відсутність стандартних інтерфейсів для роботи з серверами і базами даних, оригінальний JavaScript не має вбудованої системи керування пакетами, яка допомагала б встановлювати, оновлювати та керувати залежностями між різними компонентами проекту.

До сфери застосування JavaScript належить:

- розробка веб-додатків, JavaScript є основною мовою програмування для веб-розробки;

- мобільна розробка за допомогою фреймворків, таких як React Native і Ionic, JavaScript може бути використаний для створення мобільних додатків для iOS та Android;

- розробка серверних додатків: за допомогою платформи Node.js, JavaScript може бути використаний для розробки серверних додатків;

- розширення браузерів: JavaScript може бути використаний для створення розширень для різних браузерів, таких як Google Chrome, Mozilla, Firefox. Розширення дозволяють додавати нові функціональні можливості до браузерів;

- браузерні скрипти – програмні модулі Javascript дають дуже багато можливостей (автозаповнення форм, зміна формату сторінок, приховування небажаного змісту, додавання інтерактивних елементів на сторінках);

- закладки – JavaScript має широке застосування в роботі програми, що розміщуються в закладках браузера;

- розробка десктопних додатків за допомогою фреймворків, таких як Electron, JavaScript може бути використаний для створення десктопних додатків для Windows, macOS та Linux [10].

1.6 Огляд фреймворків

Існує багато різних браузерів, кожен з них мав свої особливості в реалізації JavaScript. Це створює проблеми з підтримкою і сумісністю коду для різних браузерів. Бібліотеки були корисними для розв'язання проблем з підтримкою різних браузерів та маніпулювання DOM. Однак, вони не забезпечують чітку структуру та рекомендації для повноцінної розробки фронтенду. Фреймворки з'явилися для цих цілей, надаючи повний стек інструментів та структурований підхід до фронтенд розробки.

Для розробки сайту культурних заходів міста було розглянуто три фреймворки, що спрощують цей процес: React, Angular та Vue.js [11].

React - це open-source JavaScript-бібліотека, створена компанією Facebook. Вона дозволяє побудувати користувацький інтерфейс програми у вигляді окремих компонентів та забезпечує їх синхронізацію зі змінами даних.

Vue.js - це прогресивний JavaScript-фреймворк, який також надає зручні засоби для створення інтерфейсів. Vue рекомендує використовувати шаблони як простішу альтернативу для опису інтерфейсу, але також підтримує render-функції та навіть підтримку JSX (синтаксис React). За допомогою шаблонів або render-функцій, розробники можуть описати, як має виглядати кожен компонент та як взаємодіяти з даними.

Angular - це фреймворк, розроблений спеціально для роботи з великими програмами. Він надає широкі можливості для структурування та організації складних додатків. Однією з важливих особливостей Angular є вбудована підтримка TypeScript, який є обов'язковим у роботі з цим фреймворком.

Результати порівняння розглянутих фреймворків розглянуті у таблиці 1.2.

Таблиця 1.2 — Порівняння фреймворків для створення сайту

Аспект	React	Angular	Vue.js
Рендерінг	Клієнтське та серверне відображення	Клієнтське відображення, серверна візуалізація	Клієнтське та серверне відображення
Обробка DOM	Створення копії DOM, порівняння та оновлення	Розділення на два потоки, браузерне відображення	Робота з шаблонами, генерація чистого HTML
Архітектура	MVC	Модульна архітектура, MVVM	Прогресивний фреймворк, модульна система
Обратна сумісність	Повна сумісність між версіями	Потрібне послідовне оновлення	Повна обратна сумісність
Технічна підтримка	Велике ком'юніті, але складно знайти документацію для нових бібліотек	Велика документація, активне ком'юніті	Докладна документація, скромне ком'юніті

Основною концепцією React є розбиття інтерфейсу на невеликі компоненти, які можна повторно використовувати і легко керувати.

Основні особливості React:

- віртуальний DOM, react використовує віртуальний DOM (Document Object Model), що дозволяє забезпечити ефективну та швидку роботу з інтерфейсом. Віртуальний DOM - це віртуальне представлення реального DOM, що забезпечує швидку перерисовку та оновлення компонентів;
- компонентний підхід, у React інтерфейс програми будується за допомогою компонентів. Кожен компонент представляє собою незалежну частину

інтерфейсу, яка може мати свою внутрішню логіку, стилізацію та стан. Компоненти можуть бути повторно використані і легко керовані;

- односторонній потік даних, у React дані передаються вниз по ієрархії компонентів. Це означає, що дані можуть бути оновлені тільки вгору через передачу пропсів (props). Цей підхід сприяє більш прогнозованому та керованому стану компонентів;

- JSX - це розширення синтаксису JavaScript, що дозволяє описувати інтерфейс за допомогою HTML-подібних тегів. Це дозволяє розробникам комбінувати JavaScript і HTML в одному файлі, що полегшує розробку та читабельність коду;

- велике співтовариство, React має велике та активне співтовариство розробників. Це означає, що є багато ресурсів, документації, пакетів і бібліотек, що полегшують розробку та вирішення проблем.

React має багато переваг, таких як ефективне використання ресурсів, простота розробки, розширюваність та широкі можливості.

Враховуючи, що фреймворк React дозволяє швидко розробляти інтерактивні веб-застосунки і має потужні можливості для роботи з інтерфейсом користувача, він був обраний в якості веб-фреймворку для розробки вебсайту для сайту культурних заходів міста.

Next.js є фреймворком, який розширює можливості React і дозволяє легко створювати швидкі веб-сайти. Використовуючи React, безкоштовну інтерфейсну бібліотеку JavaScript, Next.js дозволяє розробникам створювати інтерфейси користувача на основі компонентів.

React є одним з найпопулярніших фреймворків для створення інтерфейсів користувача. Він добре відомий серед розробників і має велику спільноту. Завдяки своїй компонентній структурі, React дозволяє побудувати ефективні та масштабовані інтерфейси.

З використанням Next.js можна створювати різноманітні типи веб-сайтів. Один з таких типів - це стекові веб-сайти, що складаються з кількох сторінок, подібних до традиційних веб-сайтів. Застосовуючи сучасний фреймворк, такий як

Next.js, розробники можуть отримати переваги швидкості та зручності в роботі зі сторінками.

Ще один популярний варіант використання React - це створення односторінкових програм (SPA). У таких додатках замість перезавантаження всієї сторінки відбувається динамічне оновлення лише поточної частини сторінки. З використанням Next.js і React, можна створити SPA з багатими інтерактивними функціями та відзивчивим користувацьким досвідом.

Таким чином, Next.js забезпечує потужний фреймворк для розробки веб-сайтів з використанням React, дозволяючи створювати швидкі та динамічні інтерфейси користувача зручним способом.

NextJS має три різні способи візуалізації React програм: Client-Side Rendering, Server-Side Rendering, Static-Site Generation.

Візуалізація на стороні сервера використовується, коли нам потрібно здійснювати рендеринг React компонентів на сервері, а результат передавати клієнту. Цей підхід може бути корисним для поліпшення продуктивності, покращення SEO та забезпечення кращої відмінності для користувачів з повільним Інтернетом або з обмеженими можливостями браузера.

Генерація статичного сайту використовується, коли нам потрібно зарано згенерувати статичні HTML файли для нашого React додатку. Це особливо корисно для статичних веб-сайтів, які не вимагають динамічного змісту. Завдяки генерації статичного сайту можна покращити продуктивність та забезпечити швидке завантаження сторінок.

Кожен з цих підходів має свої власні випадки використання та переваги, і їх можна використовувати в залежності від потреб та характеристик проекту.

Візуалізація на стороні клієнта (CSR) означає, що більша частина роботи вибір даних та створення HTML виконується браузером за потребою, коли це необхідно для користувача. Ось як виглядає процес:

- клієнт запитує сторінку за допомогою URL-адреси;
- сервер надсилає клієнту порожню сторінку разом з посиланнями на JavaScript-файли;

- браузер завантажує та виконує ці JavaScript-файли, що використовуються для створення сторінки;
- дані витягуються з API за допомогою JavaScript і заповнюються на сторінці.

Переваги візуалізації на стороні клієнта:

- сервер може швидко надати порожню сторінку, і візуалізація на стороні клієнта виконується швидше, оскільки браузер може швидко обробляти вміст та події. Хоча вимагає більше коду для обробки на клієнтській стороні, рендеринг відбувається швидко;
- візуалізація на стороні клієнта дозволяє показувати різні частини сторінки без перезавантаження або повторного відображення всієї сторінки. Це зменшує навантаження на пам'ять і обчислювальні ресурси, що приводить до швидшої відповіді, ніж рендеринг на стороні сервера;
- візуалізація на стороні клієнта ідеально підходить для SPA, оскільки дозволяє повторно використовувати компоненти інтерфейсу користувача без необхідності повторного запитування їх з сервера. Це сприяє швидкості та продуктивності SPA;

Недоліки візуалізації на стороні клієнта:

- при використанні візуалізації на стороні клієнта, спочатку клієнт отримує порожню сторінку. Це означає, що користувач також побачить порожню сторінку, коли відвідає вашу веб-сторінку вперше. Час, потрібний для завантаження програми JavaScript та створення сторінки, може бути довший, ніж у випадку візуалізації на стороні сервера;
- Метадані та контент завантажуються за допомогою JavaScript під час першого завантаження і пошукові системи можуть не побачити ці дані при скануванні сторінки. Це може негативно вплинути на пошукову видимість веб-сайту;
- в сучасних веб-додатках пакети JavaScript можуть бути досить великими. При використанні візуалізації на стороні клієнта, весь цей код повинен

бути завантажений на клієнтський браузер, що може зайняти час, особливо при повільному інтернет-з'єднанні.

Візуалізація на стороні сервера (SSR) дозволяє запускати нашу React програму на сервері, так що важка робота виконується до того, як сторінка досягає браузера. Ось як виглядає процес:

- клієнт запитує сторінку за допомогою URL-адреси;
- сервер отримує запит і запускає відповідний JavaScript, створюючи елементи DOM на стороні сервера;
- сервер запитує необхідні дані з API та передає їх компонентам React як пропси;
- сервер створює HTML, який містить готову сторінку, і надсилає його у відповідь клієнту.

Переваги візуалізації на стороні сервера:

- негайна відповідь, клієнт отримує першу сторінку, дані вже доступні. Це особливо корисно для повільних з'єднань, оскільки користувачі отримують зміст миттєво, без затримок;
- відсутність звернень клієнта, оскільки дані вже були отримані та оброблені сервером, клієнту не потрібно робити додаткові запити HTTP для отримання даних. Це зменшує навантаження на мережу та забезпечує швидшу відповідь;
- візуалізація на стороні сервера сприяє кращій видимості веб-сторінок для пошукових систем. Оскільки всі метадані та контент вже є в HTML, веб-сканери можуть легко індексувати сторінку і враховувати її в результатах пошуку;
- підтримка для користувачів без JavaScript, рендеринг на стороні сервера забезпечує роботу вашого додатку навіть для користувачів, які не мають JavaScript.

Мінуси рендеринга на стороні сервера:

- рендеринг на стороні сервера вимагає повторного відтворення програми як на сервері, так і на клієнті, що може призвести до збільшення часу завантаження між сторінками, особливо для великих програм;

- рендеринг відбувається на сервері, при одночасному доступі багатьох користувачів до програми можуть виникати затримки під час завантаження програми. Це може негативно вплинути на загальний досвід користувача;
- деякі компоненти інтерфейсу користувача можуть використовувати функціонал, що залежить від віконного об'єкта, який не існує при рендерингу на стороні сервера. Це може призвести до проблем сумісності з певними бібліотеками або компонентами.

Генерація статичного сайту (SSG) створює кілька статичних шляхів на основі даних, необхідних для сторінки. Під час збирання ці численні шляхи перетворюються на статичні сторінки та неймовірно швидко надаються клієнту:

- клієнт робить запит на сервер;
- сервер отримує запит і створює необхідні HTML-сторінки для статичних даних. Це може включати виконання розрахунків або отримання даних з бази даних або інших джерел;
- сервер відповідає клієнту готовими, наперед створеними статичними сторінками на основі запитуваної URL-адреси.

Переваги створення статичних сайтів:

- миттєва доступність, генерація статичних сайтів доступна одразу після створення, не потрібно очікувати обробки даних або виконання запитів до бази даних;
- оскільки HTML створюється перед надсиланням клієнту, статичні сторінки ідеально підходять для пошукової оптимізації. Їх зміст легко індексується пошуковими системами, що сприяє покращенню видимості в пошукових результатах;
- обробка статичних сторінок відбувається надзвичайно швидко, оскільки немає необхідності у виконанні обчислень або запитів до бази даних. Готові статичні сторінки можуть бути негайно надані клієнту;
- незалежність від сервера, бо для обслуговування статичних сторінок не потрібен сервер для моніторингу, що спрощує хостинг і зменшує вартість обслуги.

Мінуси генерації статичного сайту:

- якщо сайт є великим і складним, процес генерації статичного сайту може зайняти значну кількість часу;
- генерація статичного сайту передбачає завантаження даних лише один раз під час створення. Тобто не можна динамічно оновлювати дані на льоту без повторної генерації сайту;
- при генерації статичного сайту можуть виникати ті ж самі проблеми сумісності інтерфейсу користувача, як і при рендерингу на стороні сервера. Це пов'язано з тим, що під час генерації сайту об'єкт вікна недоступний, що може вплинути на певні функціональні можливості.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Постановка задачі розробки

Мета даного дипломного проєкту полягає у створенні інтуїтивно зрозумілого, зручного для користувача та естетично привабливого веб-сайту, на якому будуть опубліковані розписи різних подій. Ці події можуть включати в себе культурні заходи, конференції, фестивалі, спортивні події та багато іншого.

Функціональні вимоги:

Пошук і фільтрація: користувач повинен мати можливість шукати події за ключовими словами, датами, місцями та категоріями. Фільтри повинні бути легко доступні і зрозумілі.

Подробиці події: будь-яка подія може мати сторінку з детальною інформацією, включаючи опис, час і місце проведення, ціни на квитки (якщо це можливо), карту місця проведення та контактну інформацію або засіб зворотнього зв'язку.

Покупка квитків: система повинна надати можливість користувачам придбати квитки на культурні заходи, включаючи можливість вибору категорій квитків та безпечну оплату.

Інтеграція з соціальними медіа: система може надати можливість користувачам ділитися інформацією про культурні заходи через соціальні медіа, такі як Facebook, Twitter і Instagram.

Оцінки та відгуки: користувачі повинні мати можливість залишати оцінки та відгуки про культурні заходи, які вони відвідали, щоб іншим користувачам було зручно орієнтуватися в якості та рейтингу подій

Реєстрація на події: повинна надати можливість користувачам створювати облікові записи для доступу до додаткового функціоналу, такого як покупка квитків, реєстрація на події.

Нефункціональні вимоги:

Продуктивність: сайт повинен бути швидким і ефективним, забезпечуючи швидкий час завантаження і мінімальні затримки.

Безпека: дані користувачі мають бути захищені, усі транзакції мають бути безпечними.

Мобільна оптимізація: сайт повинен бути оптимізований для мобільних пристроїв, забезпечуючи однаково гарний досвід використання як на настільних, так і на мобільних пристроях.

Доступність: сайт повинен відповідати стандартам доступності для забезпечення рівного доступу для всіх користувачів.

Сайт на основі такої тематики має завдання мати:

- ненагромаджений та зручний інтерфейс користувача;
- можливість реєстрації або бронювання вільних місць для заданого заходу;
- наявність зворотного зв'язку на сайті;
- опис цих заходів або івентів.

2.2 Моделювання прецедентів

Діаграми прецедентів застосовуються для моделювання виду системи з точки зору прецедентів.

Графічне зображення прецедентів представляє собою структурований граф, що складається з набору різних елементів. У цьому графі виокремлюються актори - учасники системи, прецеденти (варіанти використання), які визначають дії та функціональні можливості системи, та взаємозв'язки між ними. Крім того, на діаграмі відображаються зв'язки між прецедентами та відношення узагальнення між акторами. Діаграми прецедентів допомагають візуалізувати модель варіантів використання системи. Діаграми прецедентів також використовуються для специфікації вимог до системи. Вони допомагають комунікувати зі зацікавленими сторонами та розуміти функціональність системи з їхньої перспективи. Діаграми прецедентів можна додатково розширити, використовуючи інші елементи, такі як системні адміністратори, додаткові взаємозв'язки або деталізовані описи дій та даних. [12].

Учасник (актор) - це група ролей, які логічно пов'язані і виконуються при взаємодії з прецедентами або сутностями.

Прецедент (варіант використання) - це опис конкретного аспекту поведінки системи з погляду користувача. Прецедент не вказує, як саме досягається певний результат, а лише описує, що саме виконується. Актор відвідувач сайту показаний на рисунку 2.1.

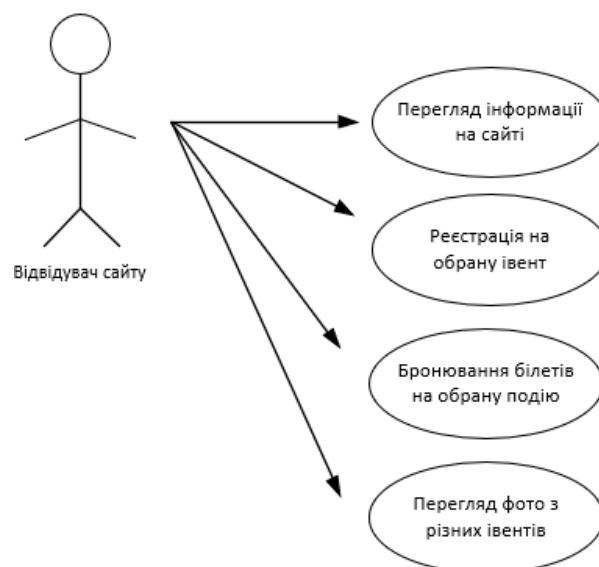


Рисунок 2.1 – Актор відвідувач сайту

Вказані прецеденти «Перегляд інформації на сайті» пошук необхідної або зацікавленої інформації для відвідувача. «Реєстрація на обраний івент» заповнення інформації відвідувача, а саме електронної пошти для подальшого зв'язку. «Бронювання білетів на обрану подію» можливість вибрати місця для відвідування івента. «Перегляд фото з різних івентів» додатковий перегляд інформації.

Актор адміністратор сайту показаний на рисунку 2.2. Прецеденти «Редагування інформації на сайті» зміна застарілої інформації для подальшого показу відвідувачам. «Редагування бронювання білетів» додаткові можливості адмінстратора для зміни інфрмації про зайнятість місць на культурний захід.

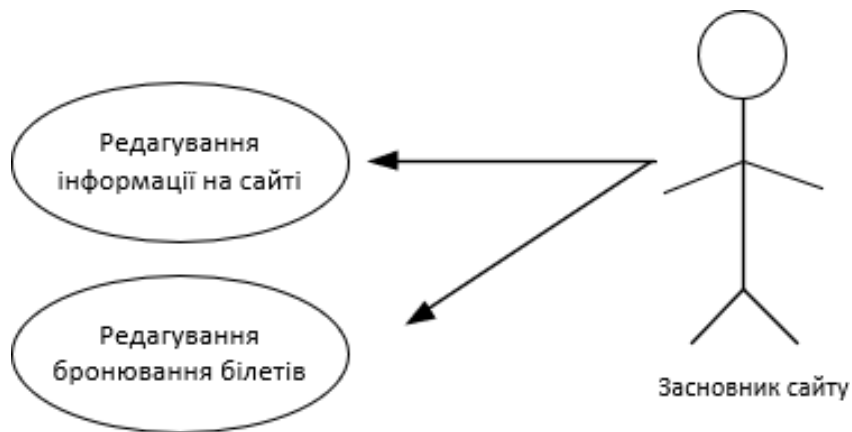


Рисунок 2.2 – Адміністратор сайту

Діаграма варіантів використання або діаграма прецедентів допомагає представити різні типи ролей та їх взаємодію з системою і представлена нижче на рисунку 2.3.

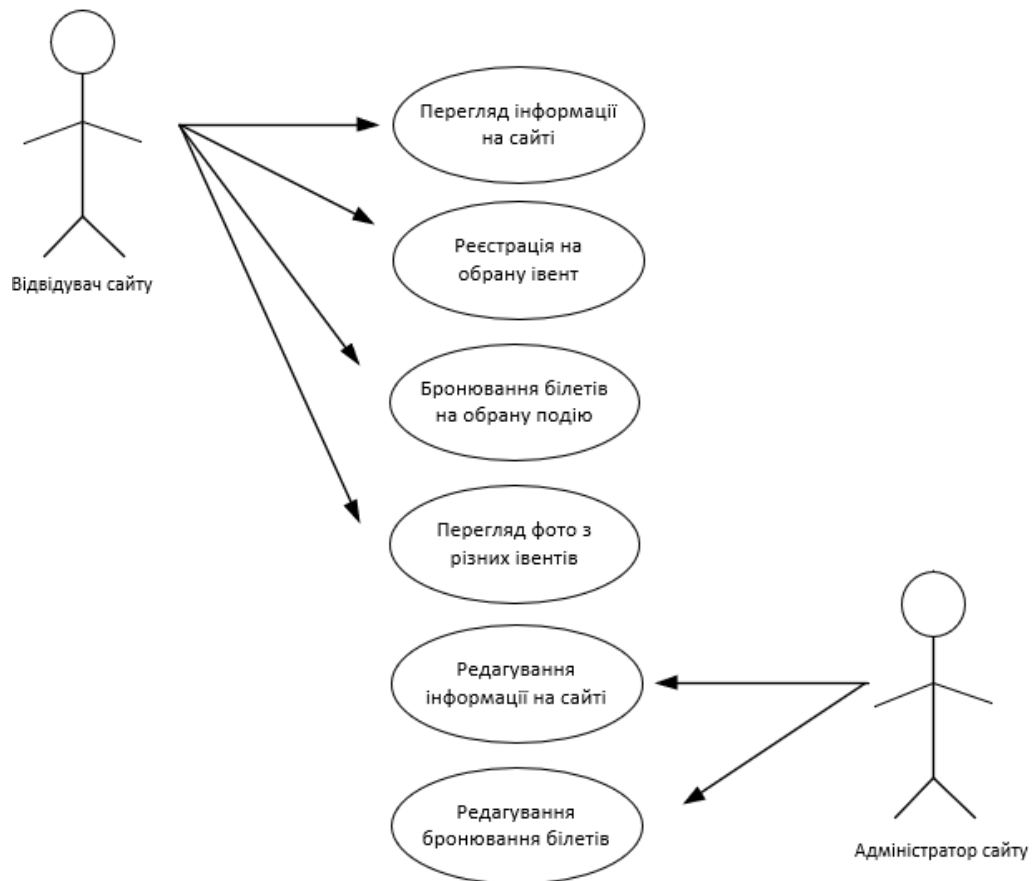


Рисунок 2.3 – Діаграма прецедентів

2.3 Архітектура програмного забезпечення

На вебсайті застосовується інноваційна архітектура клієнт-сервер. Ця унікальна архітектура забезпечує взаємодію та обмін даними між клієнтами та серверами, розподіляючи завдання і мережеві навантаження між постачальниками послуг (серверами) та замовниками послуг (клієнтами).

У моделі клієнт-сервер клієнти функціонують паралельно і незалежно один від одного, забезпечуючи ефективну роботу. Один сервер здатний обробляти запити від різних клієнтів одночасно, забезпечуючи швидку відповідь на їх запити.

Основним принципом клієнт-серверної архітектури є розподіл обов'язків між клієнтом та сервером, що дозволяє оптимізувати роботу мережесих додатків і забезпечити їх надійність, безпеку та продуктивність. Приклад клієнт-серверної архітектури представлений на рисунку 2.4.

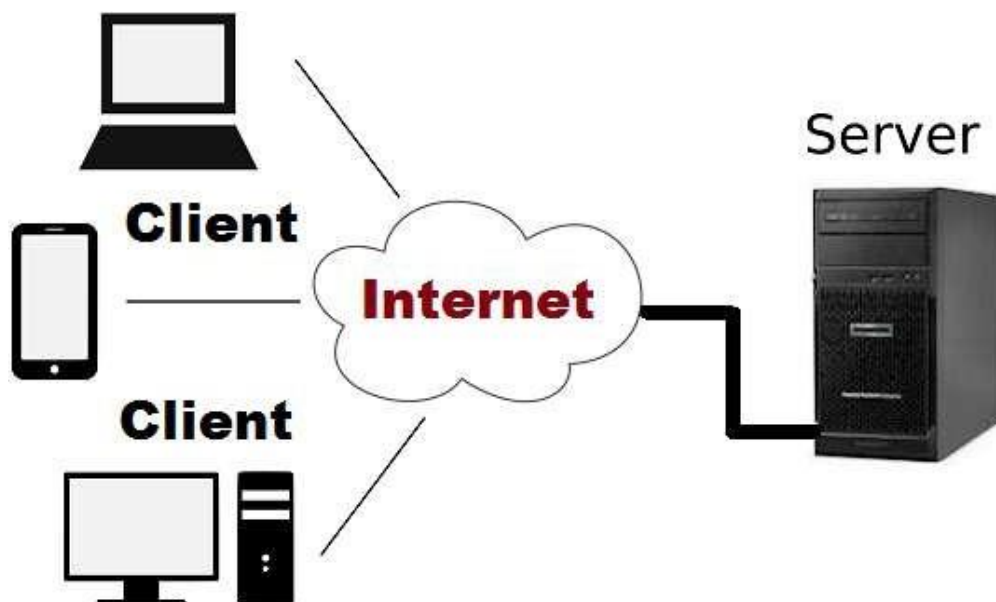


Рисунок 2.4 – Клієнт-серверна архітектура

Клієнт-сервер забезпечує доступ до трьох рівнів операцій:

- рівень представлення даних, який відповідає за інтерфейс користувача і представлення даних. Клієнтська програма забезпечує функції цього рівня, забезпечуючи зручну взаємодію користувача з ресурсом;
- прикладний рівень, на ньому реалізується основна логіка додатків і проводиться необхідна обробка інформації. Він забезпечує виконання складних операцій та обробку даних, що потребуються для ресурсу;
- рівень управління даними, що відповідає за зберігання даних та доступ до них. Вся логіка управління даними зосереджена на сервері, що забезпечує надійність і безпеку збереження і обробки інформації.

Архітектура клієнт-сервер є основою для розробки мережесих додатків і передбачає поділ їх на компоненти, які забезпечують різні сервіси. Кожен

компонент може виконуватися на окремих комп'ютерах, виконуючи ролі сервера або клієнта. Це дозволяє досягти покращення надійності, безпеки і продуктивності мережевих додатків і мережі в цілому.

2.4 Стек технологій розробки сайту

Сторона клієнта, відома також як front-end, включає розробку того, що користувач бачить на екрані і взаємодіє з ним. Це включає створення веб-сторінок, розміщення контенту, оформлення, валідацію форм та обробку подій.

Основні засоби, що використовуються у front-end.

Мова гіпертекстової розмітки HTML, де використовується розширення синтаксису JSX JavaScript. Це дозволяє описувати інтерфейс за допомогою HTML-подібних тегів. Каскадні таблиці стилів (CSS) використовуються для оформлення веб-сторінок, а також можна використовувати SASS, що надає багато корисних функцій і можливостей для розширення звичайного CSS.

JavaScript (JS) використовується для створення інтерактивних веб-сторінок та додавання динамічно оновлюваного контенту.

Бібліотека React використовується для розробки користувацького інтерфейсу додатків.

Сторона сервера включає додаток, який керує всією системою і відповідає за обробку логіки, безпеку та виконання бізнес-логіки.

Фреймворк Next.js надає додаткові можливості, такі як серверний рендеринг, статичне генерування та маршрутизація.

Шрифт Inter з пакета `@next/font/google` використовується для підключення шрифтів з Google Fonts.

Axios є бібліотекою, яка дозволяє здійснювати HTTP-запити з JavaScript. Вона використовується для звернення до сервера або отримання даних з зовнішніх джерел.

2.5 Реєстрація на обраний івент

Процедура реєстрації користувача призначена для занесення адреси електронної пошти користувача до бази даних сайту.

На стороні фронтенду, користувачу потрібно зробити наступне:

- ввести електронну адресу користувача (email address);
- натиснути кнопку “Submit” для завершення процедури реєстрації і відправки даних.

Після введення користувачем даних та їх відправки на сервер, виконується POST-запит на ендпоінт `"/api/email-registration"`. Цей ендпоінт виконує валідацію, обробку та збереження вхідних даних користувача. За допомогою асинхронного запиту POST, дані про електронну пошту та ідентифікатор події передаються на сервер для подальшої обробки. Валідація даних виконується на основі наступних регулярних виразів:

“ `/^[a-zA-Z0-9.!#$%&'+/=^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/`”

цей регулярний вираз використовується для валідації електронної пошти;

- `^[a-zA-Z0-9.!#$%&'+/=^_`{|}~-]+@`: початок регулярного виразу вказує, що адреса електронної пошти повинна починатися з будь-якої комбінації букв (латинського алфавіту в будь-якому регістрі), цифр або деяких спеціальних символів, вказаних у квадратних дужках;

- `[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)$`: Ця частина регулярного виразу визначає доменну частину електронної пошти. Вона включає комбінацію букв (латинського алфавіту в будь-якому регістрі), цифр і деяких спеціальних символів для першого рівня домену (наприклад, `.com`, `.org`, `.net`) та можливість додаткових рівнів домену, вказаних за допомогою `.(?:\.[a-zA-Z0-9-]+)`.

Отже, цей регулярний вираз перевіряє, чи відповідає введена адреса електронної пошти вказаному шаблону. Якщо адреса відповідає вимогам регулярного виразу, то вона вважається дійсною.

У разі виявлення помилок під час реєстрації передбачені спеціальні повідомлення, які надають детальну інформацію про причини виникнення помилок. Такі повідомлення обробляють наступні вказані випадки:

- дані про подію не знайдено;
- невірна адреса електронної пошти;
- ця електронна адреса вже зареєстрована у системі;
- інші помилки пов'язані з неможливістю прочитати або записати дані в файл.

2.6 Проектування інтерфейсу сайту

В даному розділі інтерфейс користувача є ключовою частиною системи і надає засоби взаємодії користувача з сайтом через веб-інтерфейс. Він визначає вигляд і оформлення сторінок проекту. Для показу сторінок, необхідно розробити сторінки відповідно до макетів.

Макети сторінок дозволяють показати, як будуть виглядати окремі елементи і компоненти на сторінках, а також взаємне розташування елементів.

Головна сторінка складається з хедеру в якому знаходиться зображення логотипу і навігаційне меню. У футері знаходиться додаткова інформація. У основному контексті сайту знаходяться зображення та невеликий опис міст у яких відбудуться івенти. Також є форма пошуку для швидкого орієнтування по сторінці.

Макет головної сторінки наведений на рисунку 2.5.



Рисунок 2.5 – Головний макет сторінки

Сторінка «про нас» загалом має таку ж саму будову як і головна сторінка сайту. Але у загальній частині сайту відображається написана інформація про сайт та розробників. І також є форма пошуку для швидкого орієнтування по сторінці. Сторінка «про нас» наведена на рисунку 2.6.



Рисунок 2.6 – Макет сторінки «Про нас»

Сторінка вибору культурних заходів має таку ж будову як і головна сторінка сайту. У основній частині сайту показані зображення самих культурних івентів та описи до них. І також є форма пошуку для швидкого орієнтування по сторінці. Сторінка «про нас» наведена на рисунку 2.7.



Рисунок 2.7 – Макет сторінки вибору івента

Сторінка реєстрації до культурних заходів теж має такий ж хедер і футер як і у інших. У основному контенті знаходиться зображення яке було на сторінці вибору культурних заходів, але трішки більше. І додається форма для заповнення реєстрації клієнтом і кнопка для відправки даних, які знаходяться під самим рисунком для зручної взаємодії з ними. І також ця сторінка має форму пошуку. Сторінка реєстрацій на культурний наведена на рисунку 2.8.



Рисунок 2.7 – Макет сторінки реєстрації на івент

2.7 Інтерфейс користувача

Згідно заданих макетів створено сторінки вебсайту, а саме, головна сторінка, сторінка «про нас», сторінка вибору культурних заходів і сторінка з формою реєстрації на заданий івент.

Головна сторінка містить загальну інформацію сайту, які інші сторінки будуть доповнювати. Головна мета цієї сторінки, щоб користувач розібрався зі структурою сайту.

Користувач може переходити на сторінки за допомогою навігаційного меню справа зверху в хедері, або натиснувши на відповідні зображення у основній частині сайту, щоб перейти на події які його цікавлять. І також може скористатися формою пошуку інформації. Головна сторінка наведена на рисунку 2.8.

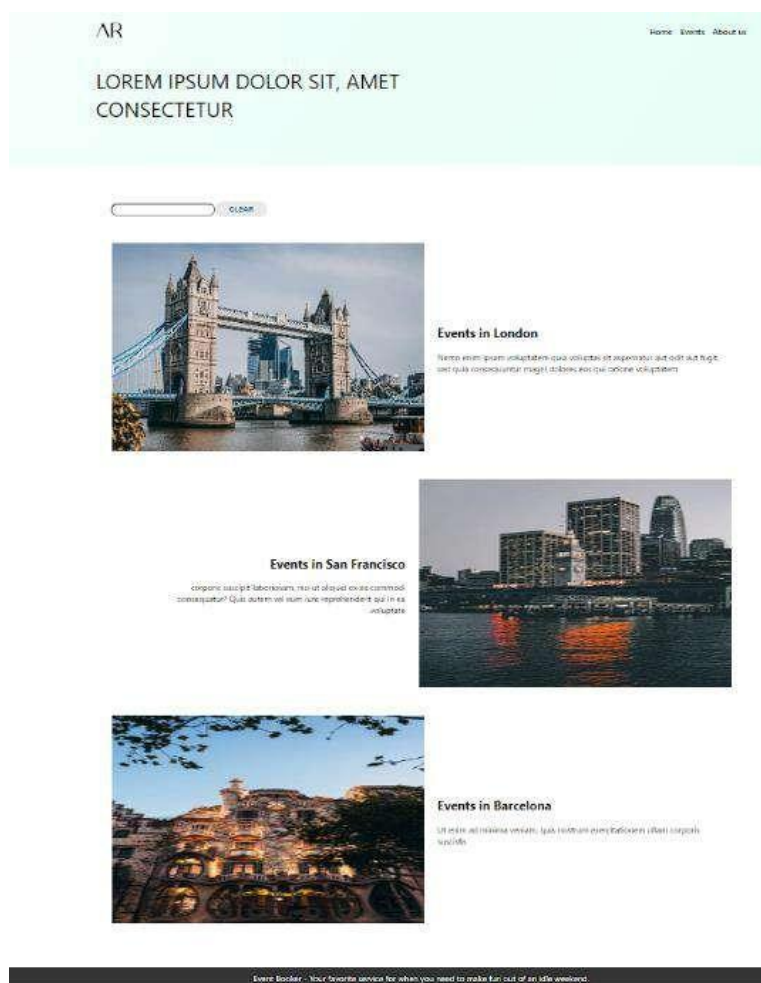


Рисунок 2.8 – Головна сторінка сайту

Сторінка «про нас» така сама тільки має основну секцію з текстовою інформацією про сайт. Сторінка «про нас» наведена на рисунку 2.9.

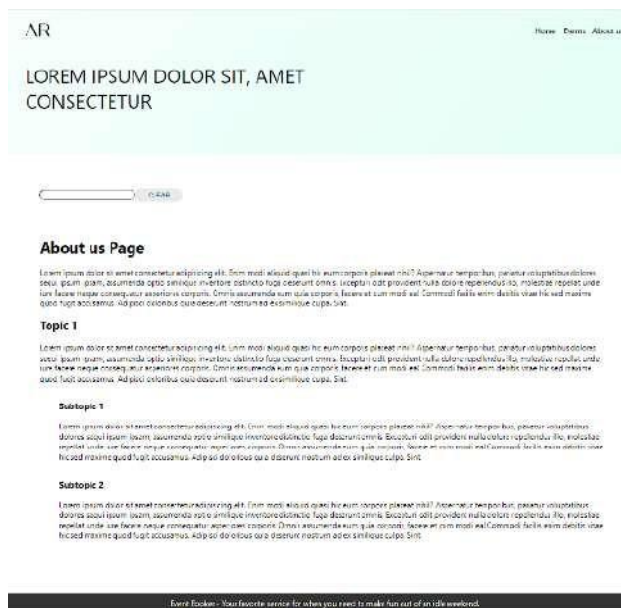


Рисунок 2.9 – Сторінка «про нас»

Сторінка вибору культурних заходів має таку ж будову як і головна сторінка сайту. У цій сторінці наведені зображення які дають змогу зрозуміти про що буде івент. І також налаштований слайдер для зручної зміни сторінок сайту. Сторінка вибору культурних заходів наведена на рисунку 2.10.

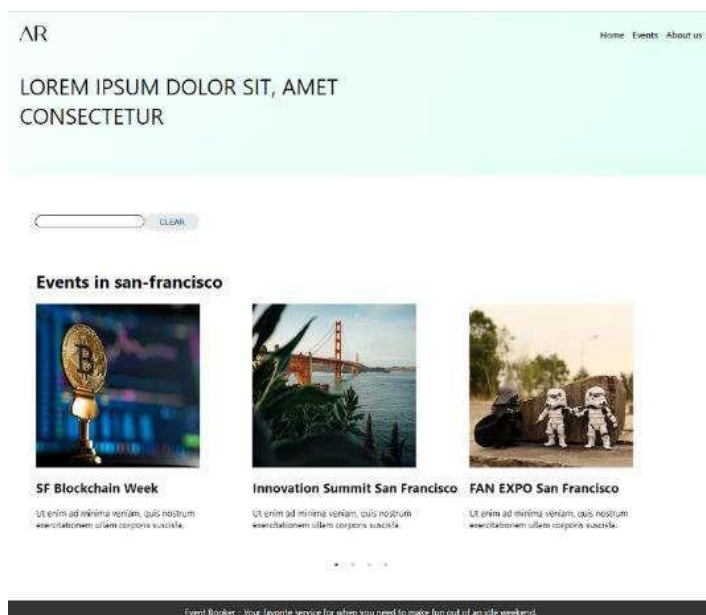
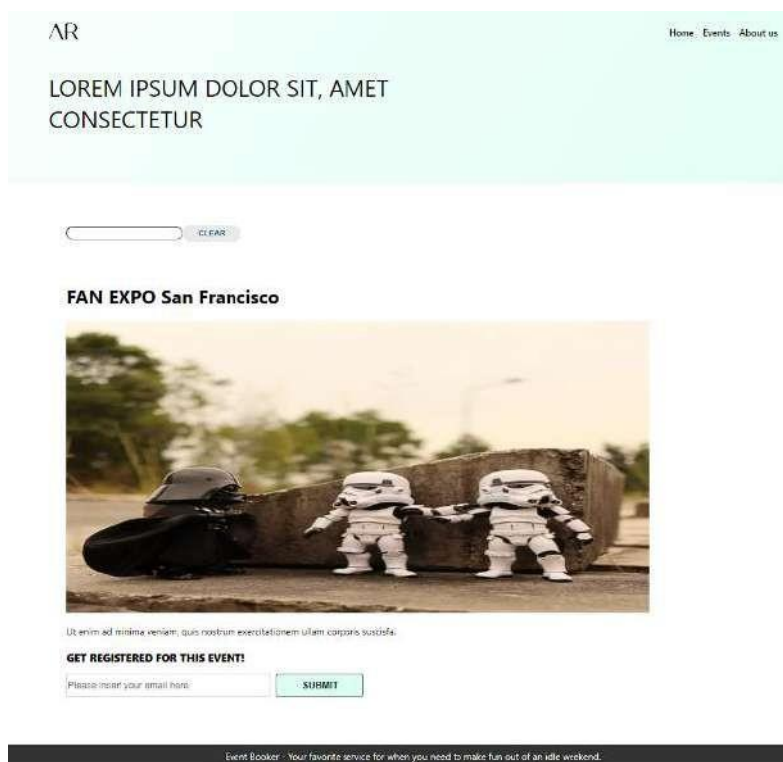


Рисунок 2.10 – Сторінка вибору культурного заходу

Сторінка реєстрації до культурних заходів має таку ж будову, як і головна сторінка. На цій сторінці надаються форма для реєстрації і кнопка для підтвердження, які загально загально не виділяються від стилю усього сайту. Сторінка реєстрації для культурних заходів, наведена на рисунку 2.11.



The screenshot shows a web page with a light green header. In the top left corner of the header is the text 'AR'. In the top right corner are the links 'Home', 'Events', and 'About us'. Below the header, on the left, is the text 'LOREM IPSUM DOLOR SIT, AMET CONSECTETUR'. Below this is a search bar with a 'CLEAR' button. The main content area features the title 'FAN EXPO San Francisco' above a photograph of two Star Wars Stormtroopers standing next to a black car. Below the photo is a line of placeholder text: 'Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit'. Underneath this is a section titled 'GET REGISTERED FOR THIS EVENT!' which contains an email input field with the placeholder text 'Please insert your email here' and a green 'SUBMIT' button. At the very bottom of the page is a dark grey footer with the text 'Event Booker - Your favorite service for when you need to make fun out of an idle weekend.'

Рисунок 2.11 – Сторінка реєстрації для культурних заходів

Сайт дуже простий і має візуальний стиль, який стимулює інтерес відвідувачів до перегляду сайту.

3 ТЕСТУВАННЯ ТА РЕАЛІЗАЦІЯ САЙТУ

3.1 Тестування сайту

Було проведено повноцінне тестування для розроблюваного вебсайту, а саме перевірка роботи навігаційного меню, перевірка відповідності заданому користувачем значенню пошуку, правильність валідації веденої інформації та перевірка цієї інформації в базі даних.

Перевірка відповідності заданому користувачем значенню пошуку, показала відповідність шуканої інформації, навігація на сайті працює і переходить за заданими адресами, також було тестування валідації заданої інформації і перевірка інформації в базі даних. За результатами валідації та перевірки інформації в базі даних не було знайдено помилок. Також було встановлено, що вебсайт має чітку структуру сайту, що сприяє легкому знаходженню інформації та навігації по сайту та зрозумілий і привабливий інтерфейс сайту, який допомагає новому користувачу швидко зрозуміти функціонал сайту. І ще перевірів роботу свого сайту в Lighthouse, який аналізує веб-сторінки і покращує їх якість. Він аналізує чотири показники: продуктивність, доступність, SEO та best practices.

Performance – продуктивність. Аналізує швидкість завантаження сайту. На цю оцінку впливає час блокування, відтворення стилів, завантаження інтерактивних елементів, шрифтів та контенту.

Best Practices – найкращі практики. Перевіряє безпеку сайту та використання сучасних стандартів веб-розробки. Оцінка залежить від того, чи використовується HTTPS, застарілі API, правильне кодування.

Accessibility – доступність. Перевіряє, чи всі користувачі можуть отримувати доступ до контенту і ефективно переміщатися по сайту. Ця оцінка залежить від зрозумілості та сприймання контенту, можливості керувати інтерфейсом і пересуватися вмістом без допомоги миші.

SEO - оцінює відповідність сторінки порадам Google з пошукової оптимізації. Тут перевіряється використання метатегів та доступ до індексації. Відповідні результати зображені на рисунках 3.1 – 3.6.

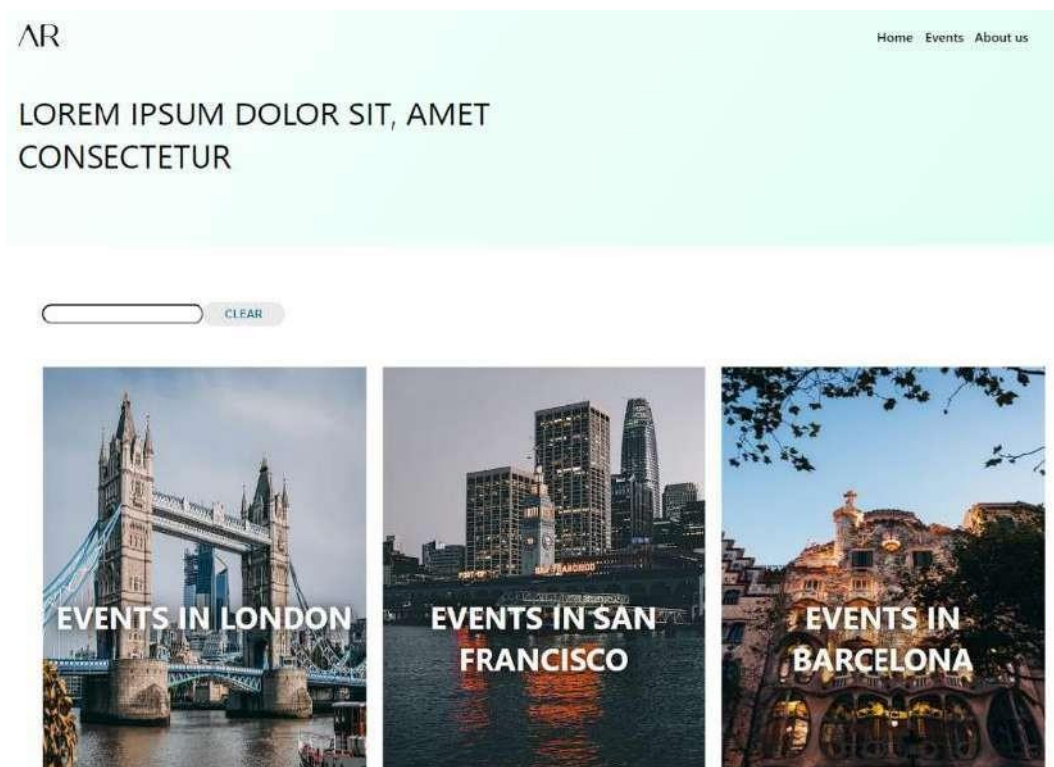


Рисунок 3.1 – Перевірка роботи навігаційного меню

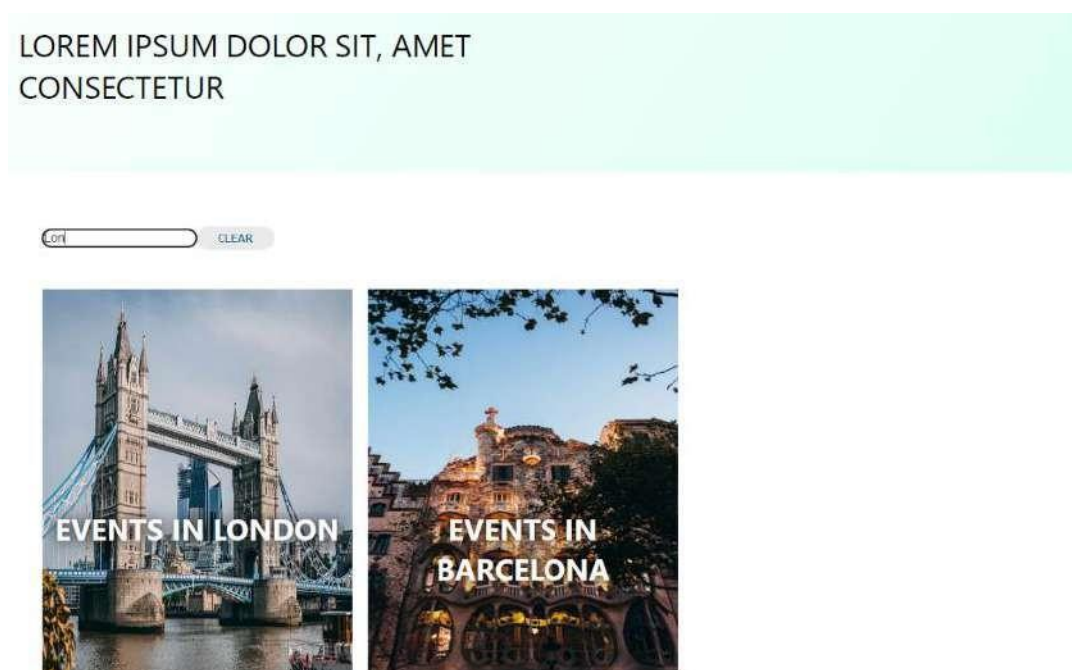


Рисунок 3.2 – Перевірка роботи значення пошуку

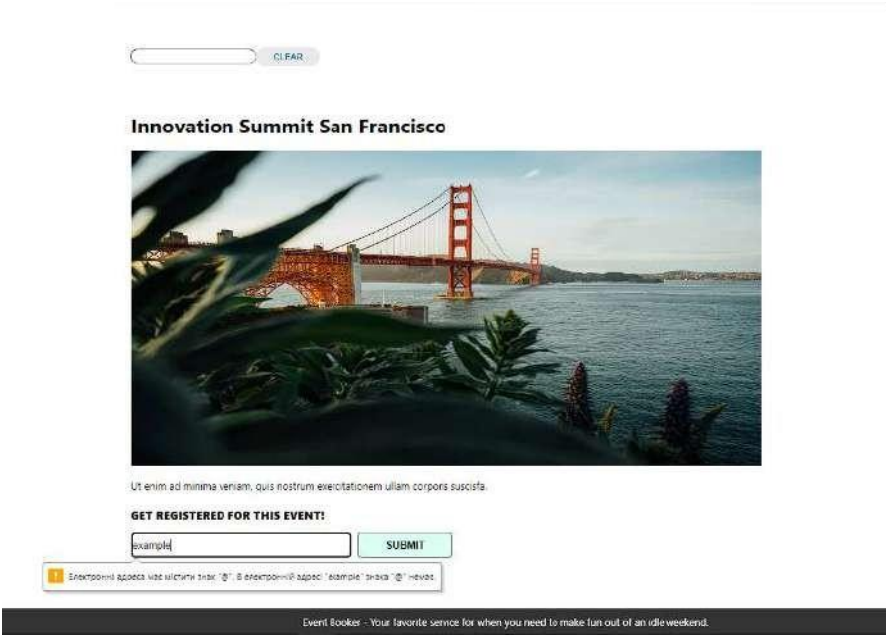


Рисунок 3.3 – Невдала реєстрація

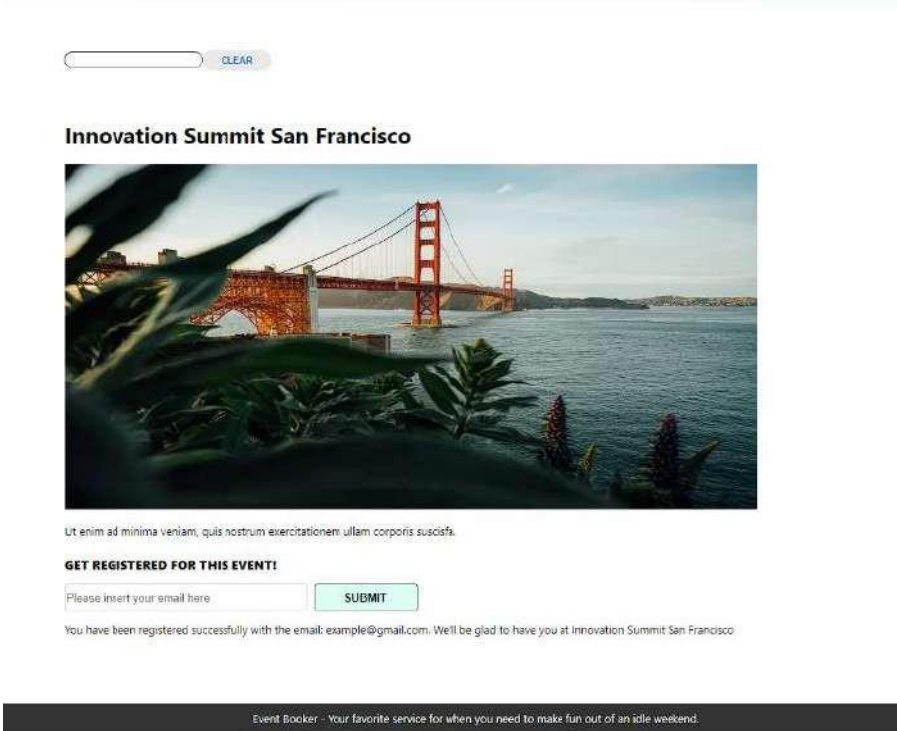


Рисунок 3.4 – Успішна реєстрація

```
{
  "id": "fan-expo-san-francisco",
  "title": "FAN EXPO San Francisco",
  "city": "san-francisco",
  "description": "Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit.",
  "image": "https://images.unsplash.com/photo-1608889825103-eb5ed786fc64?ixlib=rb-4.0.3&ixid=MjA3fDB8M4kwaG90by1uWd1fhx8FGVufDB8fHx8Sauto=format&fit=crop&w=888&q=80",
  "calendar": "Starts on Wed, 21 Jun 2023 21:30",
  "location": "Online",
  "emails_registered": ["example@gmail.com"]
},
{
```

Рисунок 3.5 – Додана інформація в базі даних

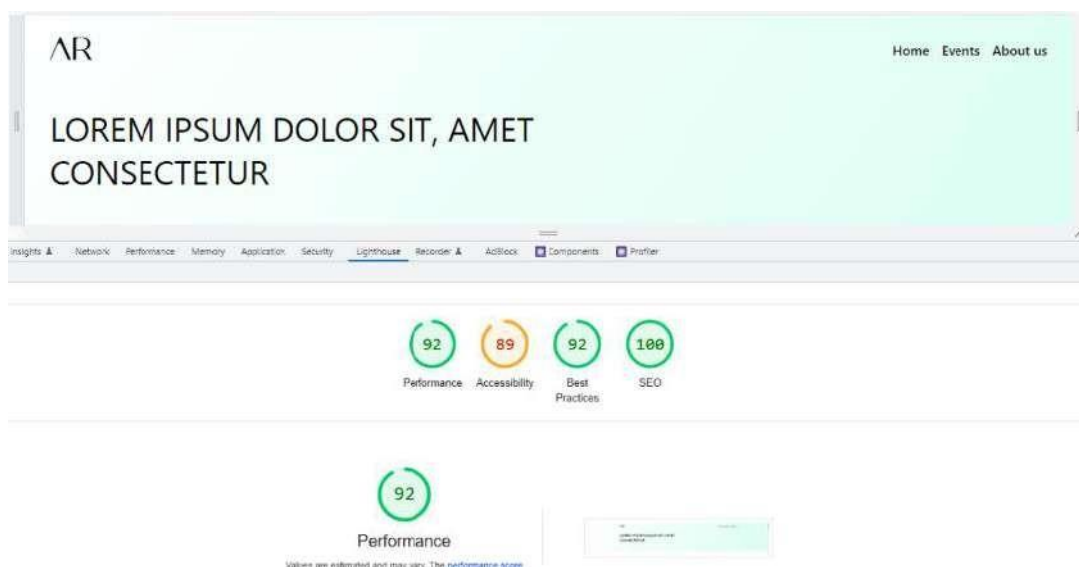


Рисунок 3.6 – Перевірка сайту у Lighthouse

3.2 Рекомендації користувачу

Для використання інформаційного сайту було виконано огляд апаратних і програмних характеристик робочого місця користувача.

Для користування даним сайтом користувач повинен мати достатні наступні мінімальні апаратні характеристики:

- центральний процесор: Intel або AMD з частотою від 1,8 GHz;
- RAM: 2 GB;
- відеоадаптер: 3D адаптер nVidia, Intel, ATI;
- відеопам'ять: 128 МБ;
- накопичувач HDD/SSD: 500 МБ.

Вимоги до програмного забезпечення для використання даного сайту наступні:

- будь-який сучасний браузер, краще останніх версій (Google Chrome, Opera, Firefox, Microsoft Edge, Safari).
- наявність операційної системи Windows 7/8/10/11 Linux, MacOS;
- стабільний та краще швидкий доступ до мережі Інтернет.

ВИСНОВКИ

В результаті виконання дипломної роботи був розроблений інформаційний вебсайт культурних заходів міста.

В процесі розробки вебсайту був здійснений огляд вже існуючих аналогів сайтів і розглянуті їх властивості і вимоги на основі яких я створив власний сайт. Функціонал мого розробленого сайту складається з:

- реєстрація користувача на вибраний культурний захід;
- пошуку інформації на сайті за допомогою ключових слів або фільтрів;
- навігація по сторінках сайту;
- відображення ключової інформації, яку хоче отримати користувач.

Було описано різницю між базою даних та збереження даних у форматі JSON.

Для даного вебсайту також була створена діаграма прецедентів що візуалізує варіанти використання системи і використовуються для специфікації вимог.

Мій вебсайт побудований на архітектурі «клієнт-сервер», яка забезпечує ефективну взаємодію між клієнтською та серверною частинами. Дозволяє зберігати та обробляти дані на сервері, а також надсилати відповідну інформацію клієнтам.

У роботі також описано різні мови програмування і фреймворки, визначено їхні переваги та надано порівняння з іншими фреймворками.

Також було визначені макети інтерфейсу сайту:

- головна сторінка;
- сторінка «про нас» з текстовою інформацією;
- сторінка вибору культурного заходу;
- сторінка реєстрації на обраний івент.

Також було проведено тестування вебсайту для перевірки роботи основного функціоналу сайту та відображення коректних повідомлень при появі помилок. В результаті можна зробити висновок, що поставлене завдання було успішно виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Стаття про “оцінку веб-сайта” URL: <https://resources.nu.edu/researchprocess/websitesevaluation> (дата звернення 02.03.2023).
2. Молчанов. В.П., Пандорін О. К., Технології розробки WEB-ресурсів. Навчальний посібник. Харків : ХНЕУ ім. С. Кузнеця, 2019. 130с (дата звернення 03.03.2023).
3. Eventbrite URL: <https://www.eventbrite.com> (дата звернення 05.03.2023).
4. Chiradeep BasuMallick. What Is JSON? Meaning, Types, Uses, and Examples URL: <https://www.spiceworks.com/tech/devops/articles/what-is-json/> (дата звернення 05.03.2023).
5. Shubham Gautam. Client Server Architecture URL: <https://www.enjoyalgorithms.com/blog/client-server-architecture> (дата звернення 07.03.2023).
6. Raivat Shah. Introduction to REST APIs URL: <https://towardsdatascience.com/introduction-to-rest-apis-90b5d9676004> (дата звернення 07.03.2023).
7. Yurii Kotula. Technology Stack for Web Application Development URL: Режим доступу до ресурсу: <https://intelvion.pro/blog/technology-stack-for-web-application-development/> (дата звернення 09.03.2023).
8. Яцишин В.В. Технологія оцінювання якості WEB-застосувань / В.В. Яцишин // Вісник ТДТУ.- Том 14.-№4.-2009.-С.132-140 (дата звернення 10.03.2023).
9. Офіційний ресурс продукту Saas URL: <https://sass-lang.com/> (дата звернення 13.03.2023).
10. Стаття про мову програмування Javascript URL: <https://info.kharkov.com/-ar/collaborator/mova-programuvannya-javascript-plyusi-ta-minusi/> (дата звернення 13.03.2023).

11. Стаття про порівняння JavaScript фреймворків Vue.js, React і Angular (2019) URL: <https://mkdev.me/ru/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019> (дата звернення 14.03.2023).

12. Олександр Марголін. UML для бізнес-моделювання URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення 14.03.2023).

ДОДАТОК А

Лістинг А.1 - Файл _app.jsx

```
import React from "react";
import { SearchProvider } from "../SearchContext";
import SearchComponent from "../SearchComponent";
import MainLayout from "@src/componentes/layout/main-layout";
import "@styles/globals.css";
import "@styles/general.sass";
export default function App({ Component, pageProps }) {
  return (
    <MainLayout>
    <SearchProvider>
    <div>
      <SearchComponent />
      <Component {...pageProps} />
    </div>
    </SearchProvider>
  </MainLayout>
  );
}
```

Лістинг А.2 - Файл _document.jsx

```
import { Html, Head, Main, NextScript } from 'next/document'
export default function Document() {
  return (
    <Html lang="en">
      <Head />
      <body>
        <Main />
        <NextScript />
      </body>
    </Html>
  )
}
```

Лістинг А.3 - Файл home-page.jsx

```
import Link from "next/link";
import Image from "next/image";
import { useSearch } from "/pages/SearchContext";
export const HomePage = ({ data }) => {
  const { searchQuery } = useSearch();
  const filteredData = data.filter((ev) => {
    const lowerCaseSearchQuery = searchQuery.toLowerCase();
```

```

    const isTitleMatch =
ev.title.toLowerCase().includes(lowerCaseSearchQuery);
    const isDescriptionMatch =
ev.description.toLowerCase().includes(lowerCaseSearchQuery);
    return isTitleMatch || isDescriptionMatch;
});
return (
  <div className="home_body">
    {filteredData.map((ev) => (
      <Link className="card" key={ev.id} href={` /events/${ev.id}`}>
        <div className="image">
          <Image alt={ev.title} src={ev.image} width={600}
height={400} />
        </div>

        <div className="content">
          <h2>{ev.title}</h2>
          <p>{ev.description}</p>
        </div>
      </Link>
    ))}
  </div>
);
};

```

ЛІСТИНГ А.4 - Файл header.jsx

```

import Link from "next/link";
import Image from "next/image";
export const Header = () => {
  return (
    <header>
      <div>
        <div className="topNav">
          <Image alt="logo" src={"/images/logo_black.png"} width={50}
height={50} />
          <nav>
            <ul>
              <li>
                <Link href="/">Home</Link>
              </li>
              <li>
                <Link href="/events">Events</Link>
              </li>
              <li>
                <Link href="/about-us">About us</Link>
              </li>
            </ul>
          </nav>
        </div>
        <p className="title">Lorem ipsum dolor sit, amet
consectetur</p>

```

```

        </div>
      </header>
    );
  };

```

Лістинг A.5 - Файл footer.jsx

```

export const Footer = () => {
  return <footer>Event Booker - Your favorite service for when you
need to make fun out of an idle weekend.</footer>;
};

```

A.6 Файл main-layout.jsx

```

import React from "react";
import { Footer } from "../footer/footer";
import { Header } from "../header/header";
const MainLayout = ({ children }) => {
  return (
    <div>
      <Header />
      <main> {children}</main>
      <Footer />
    </div>
  );
};
export default MainLayout;

```

Аналоги сайтів

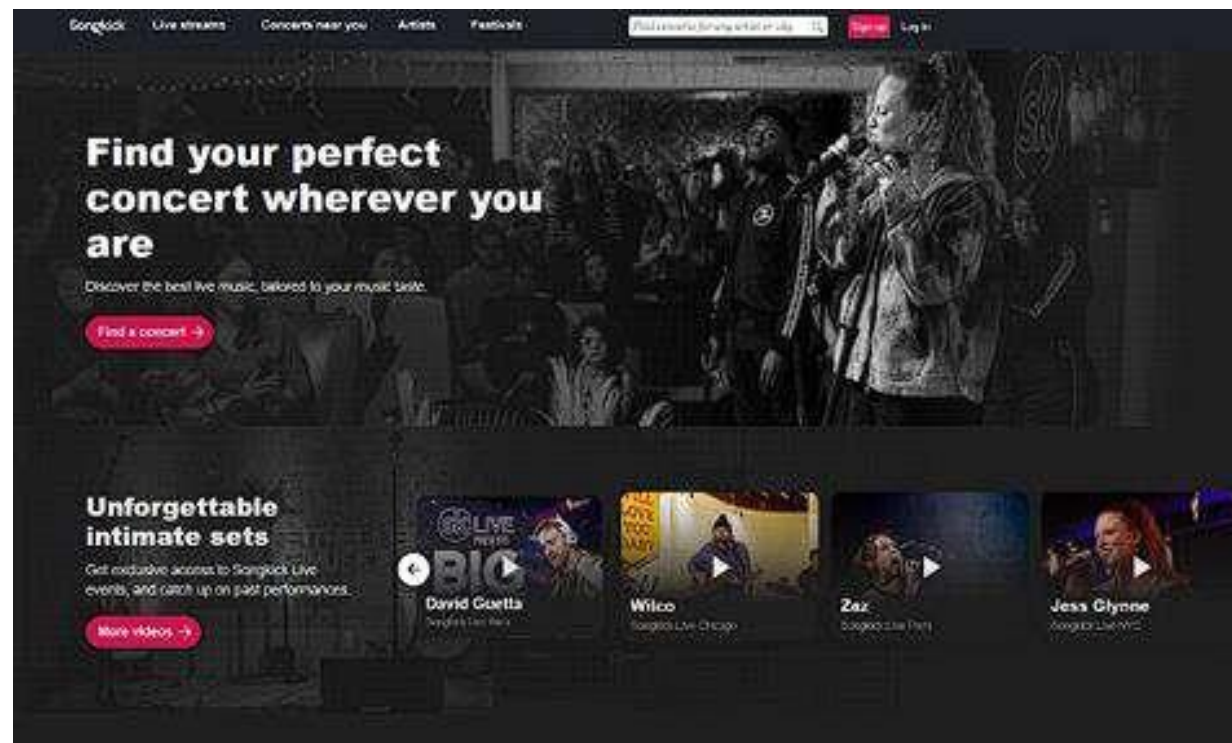


Рисунок 1 - сторінка сайту «Songkick»

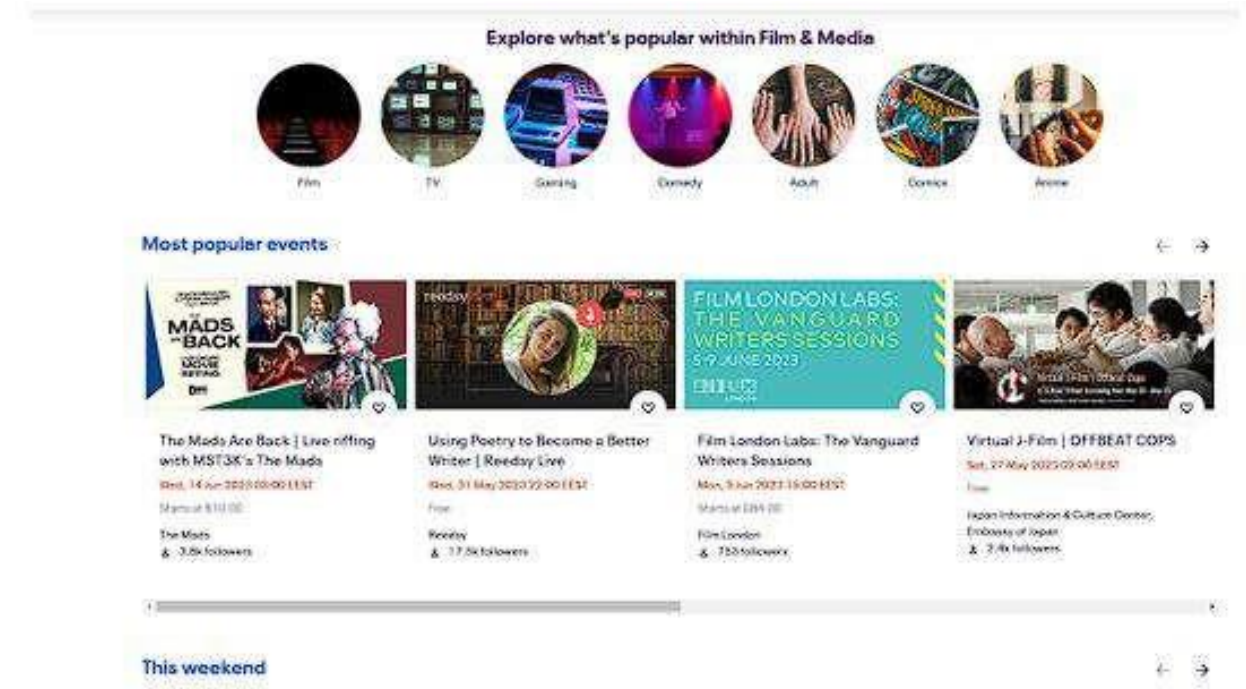


Рисунок 2 - сторінка сайту «Eventbrite»



Рисунок 2 - сторінка сайту «Eventbrite»

					13.02070849.00038 ПЛ1				
					Розробка інформаційного сайту культурних заходів міста Аналоги сайтів	Лист.		Маса	Масишт.
Зам.	Лист.	№ докум.	Підп.	Дата					
Розроб.	Чароватий П.Г.	Чар							
Перевір.	Зеленьова І.Я.								
Т.контр.									
						Лист 1		Листів 1	
Н.контр.	Щербак Н.В.					НУ "Запорізька політехніка"			
Затвердж.	Кулєрметов Р.К.					КНТ 510 - СП			

Діаграма прецедентів і архітектура сайту культурних заходів міста

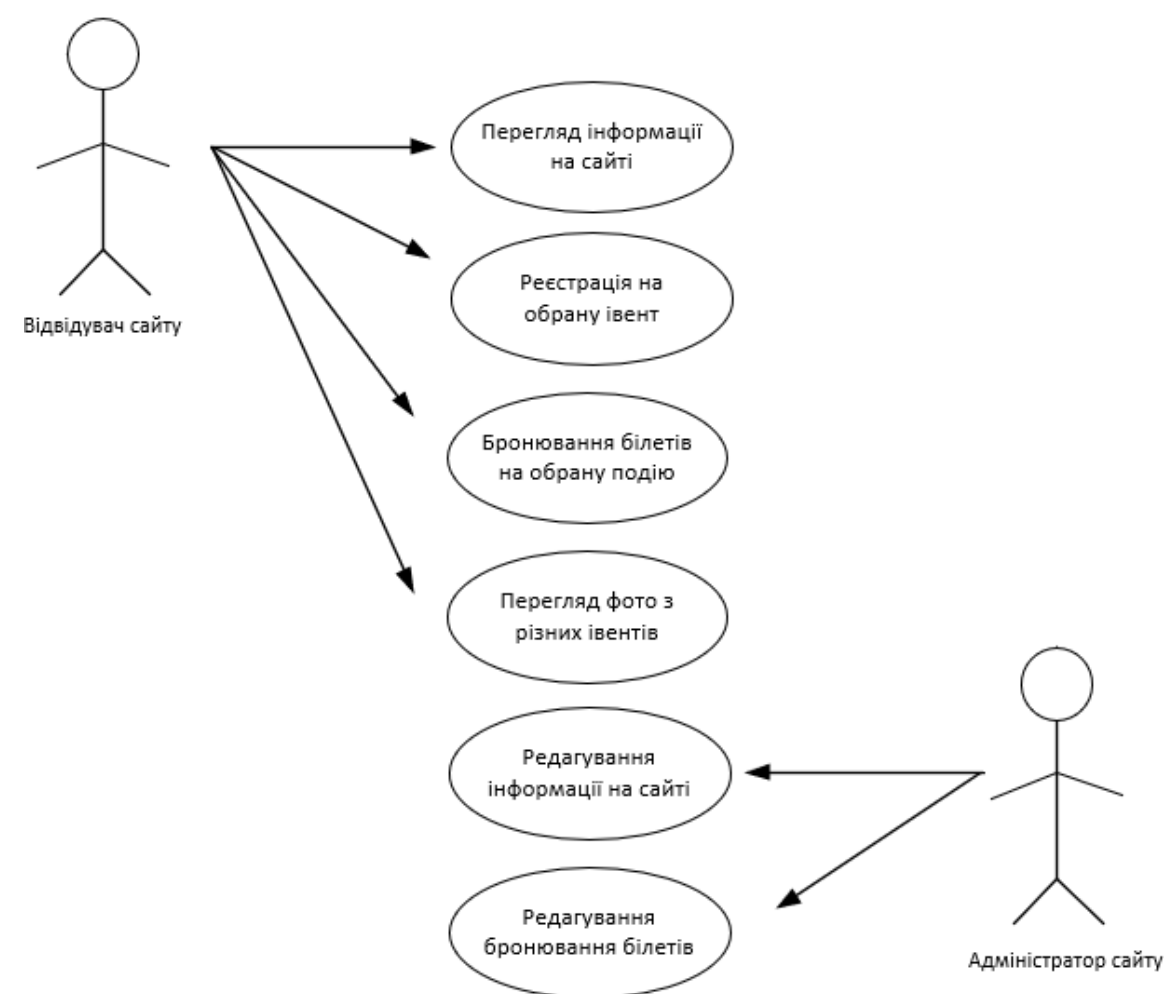


Рисунок 1 – діаграма прецедентів (діаграма варіантів використання)

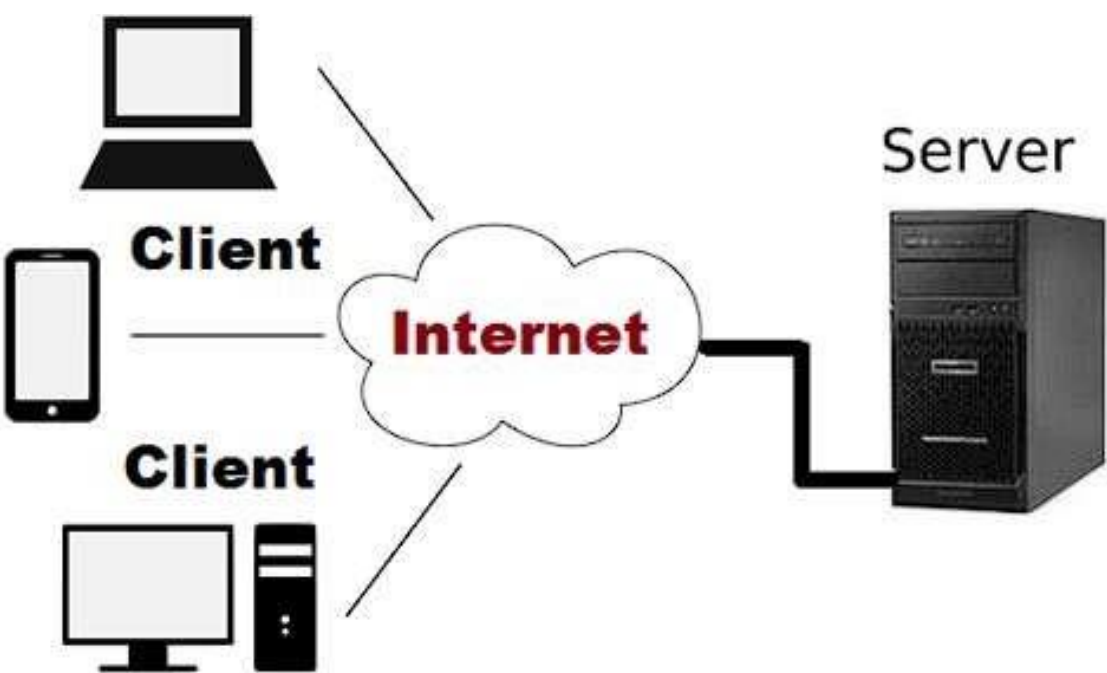


Рисунок 2 – архітектура сайту

						13.02070849.00038 ПЛ2				
					Розробка інформаційного сайту культурних заходів міста	Лит.		Маса	Масшт.	
Зам.	Лист.	№ докум.	Підп.	Дата						
Розроб.		Чароватий П.Г.	Чар							
Перевір.		Зеленьова І.Я.	Зел							
Т.контр.					Діаграма прецедентів і архітектура сайту культурних заходів міста	Лист 1		Листів 1		
						НУ "Запорізька політехніка" КНТ 510 - СП				
Н.контр.		Щербак Н.В.	Щерб							
Затвердж.		Кудерметов Р.К.	Куд							

Макети інтерфейсу сайту



Рисунок 1 – Головний макет сторінки



Рисунок 2 – Макет сторінки «про нас»



Рисунок 3 - Макет сторінки реєстрації на івент

					13.02070849.00038 ПЛЗ					
					Розробка інформаційного сайту культурних заходів міста Макети інтерфейсу сайту	Лит.		Маса	Масшт.	
Зам.	Лист.	№ докум.	Підп.	Дата		Лист 1		Листів 1		
Розроб.		Чароватий П.Г.	Чар							
Перевір.		Зеленьова І.Я.	Зел							
Т.контр.										
Н.контр.		Щербак Н.В.	Щерб					НУ "Запорізька політехніка"		
Затвердж.		Кудерметов Р.К.	Кудер					КНТ 510 - СП		

Інтерфейс користувача

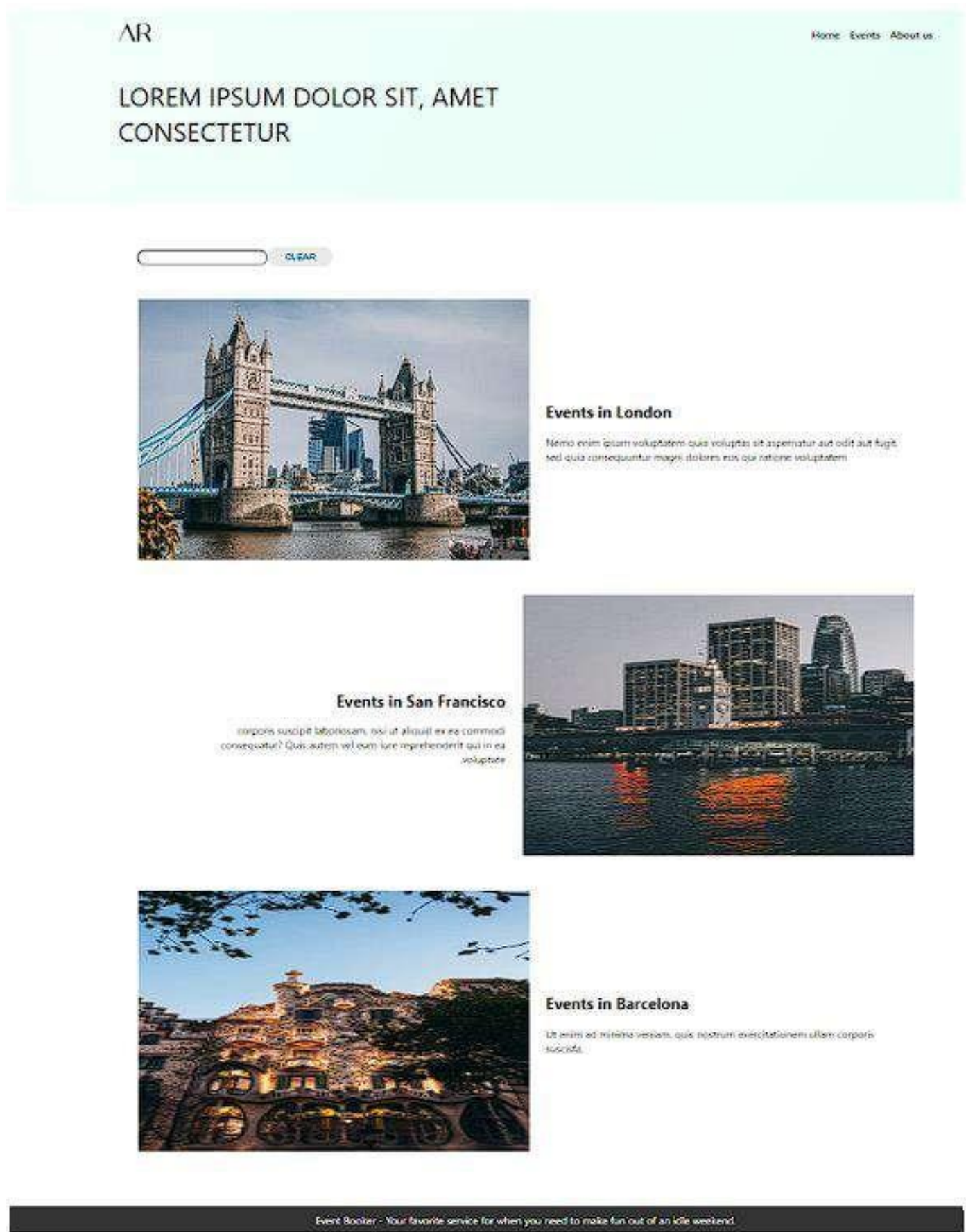


Рисунок 1 – Головна сторінка сайту

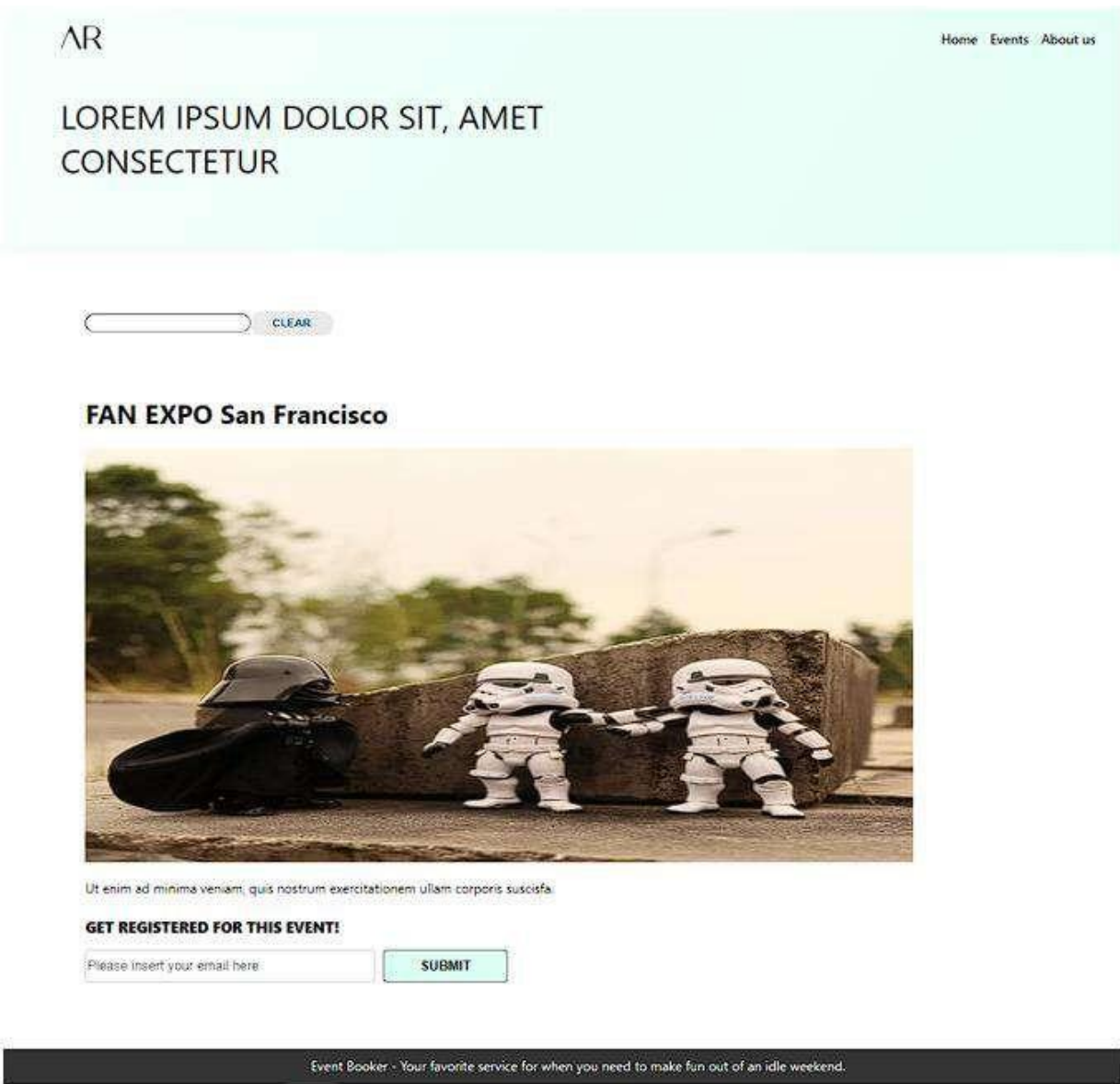


Рисунок 2 - Сторінка реєстрації

					13.02070849.00038 ПЛ1				
					Розробка інформаційного сайту культурних заходів міста Інтерфейс користувача	Лист.		Маса	Масшт.
Зам.	Лист.	№ докум.	Підп.	Дата		Лист 1		Листів 1	
Розроб.		Чароватий П.Г.	Чар					НУ "Запорізька політехніка"	
Перевір.		Зеленьова І.Я.	Зел					КНТ 510 - СП	
Т.контр.									
Н.контр.		Щербак Н.В.	Щерб						
Затвердж.		Кудерметов Р.К.	Куд						

Тестування роботи сайту

LOREM IPSUM DOLOR SIT, AMET
CONSECTETUR

0.00 CLEAR

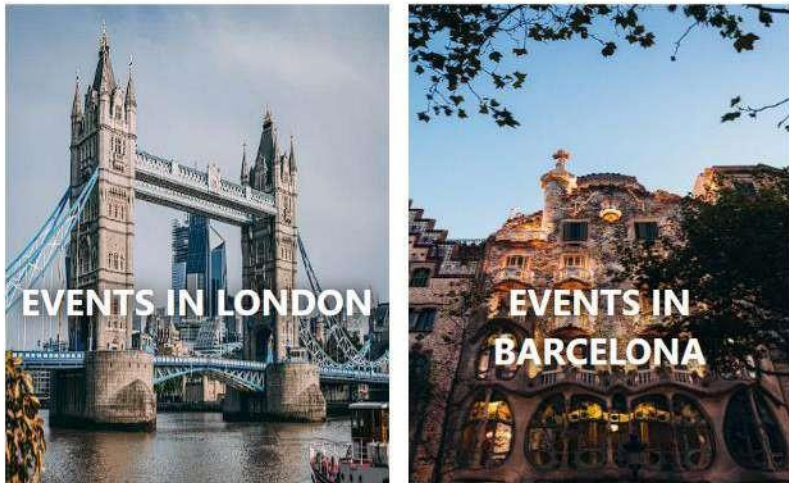


Рисунок 1 – Перевірка функції пошуку

Рисунок 3.3 – Невдала реєстрація

CLEAR

Innovation Summit San Francisco



GET REGISTERED FOR THIS EVENT!
Please enter your email here
SUBMIT
You have been registered successfully with the email example@gmail.com. We'll be glad to have you at Innovation Summit San Francisco.

Рисунок 2 – Приклад вдалої реєстрації

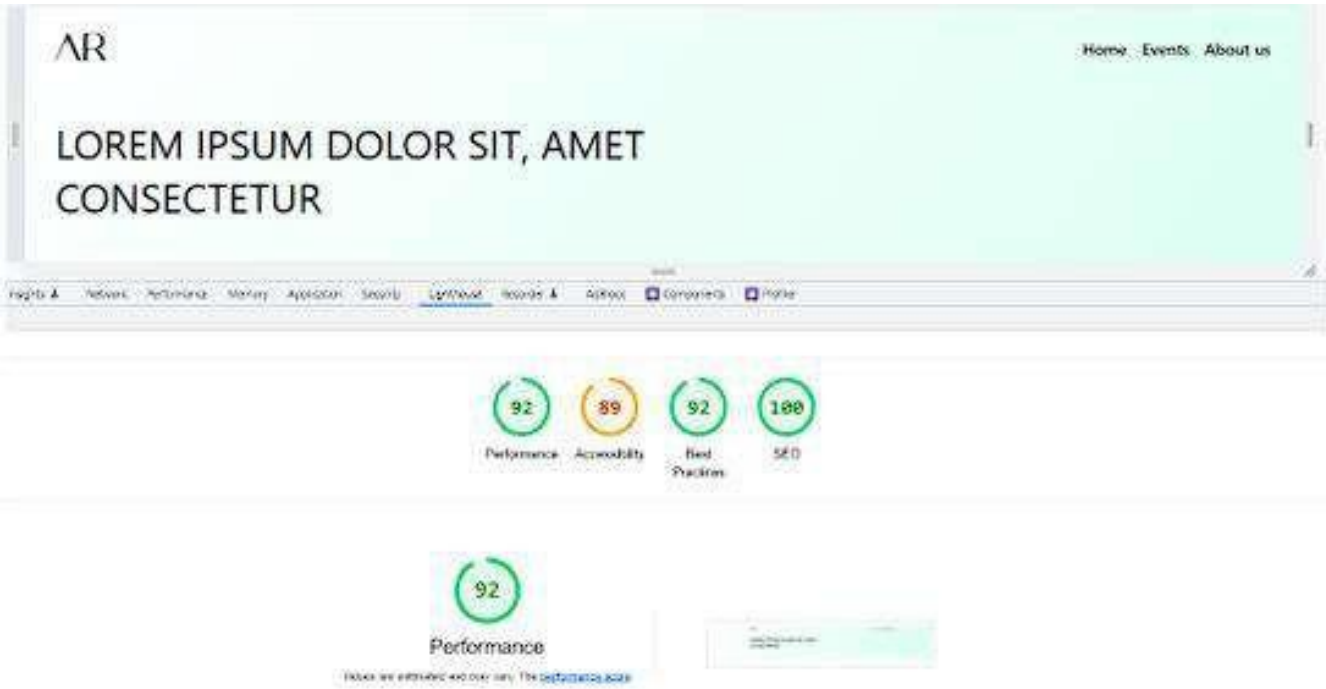


Рисунок 3 – Аналіз у інструменті Lighthouse

					13.02070849.00038 ПЛ2					
					Розробка інформаційного сайту культурних заходів міста	Лист.			Маса	Масшт.
Зам.	Лист.	№ докум.	Підп.	Дата	Тестування роботи сайту	Лист 1			Листів 1	
Розроб.		Чароватий П.Г.	Чар							
Перевір.		Зеленьова І.Я.	Зел							
Т.контр.										
Н.контр.		Щербак Н.В.	Щерб			НУ "Запорізька політехніка"				
Затвердж.		Кудерметов Р.К.	Куд			КНТ 510 - СП				

