

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

122 Комп'ютерні науки
Освітня програма «Прикладне програмування»

Кваліфікаційна робота бакалавра

на тему: «Веб-сервіс для перегляду культурних заходів»

Виконав студент 4 курсу групи ПП-43
(МС)

(Підпис)

Гармаш П. П.

(прізвище, ім'я, по батькові)

Керівник доцент, к.т.н. Силантьєв С.О.

(прізвище, ім'я, по батькові)

Попередній захист:

(Висновок: "До захисту в екзаменаційній комісії")

Завідувач кафедри
(Підпис)

20.05.2021 Плескач В.Л.
(Прізвище, ініціали)

(Дата)

Київ – 2021 року

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

Назва теми: «Веб-сервіс для перегляду культурних заходів»

Освітня програма: Прикладне програмування
Спеціальність: Комп'ютерні науки

ПІБ

Підпис

Гармаш Петро Петрович



Назва роботи українською та англійською мовами

Веб-сервіс для перегляду культурних заходів

Web-service for the viewing cultural events

Мета бакалаврської роботи, завдання

Мета бакалаврської роботи: Удосконалення системи обліку у ветеринарній клініці

План роботи:

1. Сучасні підходи до розроблення та впровадження веб-сервісів для перегляду культурних заходів
2. Аналіз архітектурних рішень і вибір програмних засобів для реалізації веб-сервісів
3. Програмна реалізація веб-сервісу для перегляду культурних заходів

ПІБ, ступінь, звання наукового керівника роботи:

к.т.н. Силантьєв Сергій Олексійович 

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Но мер	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	26.10.2020	виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	23.11.2020	виконано
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	01.12.2020	виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	18.02.2021	виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	25.02.2021	виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	05.03.2021	виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	09.04.2021	виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	07.05.2021	виконано
9.	Подання роботи у першому варіанті	11.05.2021	виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	12.05.2021	виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	24.05.2021	виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	28.05.2021	виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	11.06.2021	виконано
14.	Захист кваліфікаційної роботи бакалавра	25.06.2021	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1
Завдання до дипломної роботи	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	1
Вступ	1
Розділ 1	15
Розділ 2	15
Розділ 3	14
Висновки	1
Перелік використаних джерел	3
Додатки	3

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата			
Розробн.				Відомість дипломної роботи	Лист	Листів
Керівн.					1	59
Н/контр.						
Завідувач кафедри						

АНОТАЦІЯ

Дипломна робота: 59 с., 25 рис., 7 таб., 6 дод., 28 літературних джерел.

Ця дипломна робота присвячена розробленню веб-сервісу для перегляду культурних заходів.

Метою дипломної роботи є побудова веб-сервісу для перегляду культурних заходів, для цього необхідно вирішити такі завдання: Для досягнення поставленої мети треба вирішити такі **завдання**:

- Дослідити теоретичні основи побудови веб-сервісів у режимі онлайн
- Проаналізувати програмно-технічні рішення, різні види забезпечення веб-сервісу
- Спроекувати, реалізувати і впровадити веб-сервіс з урахуванням інженерних вимог

Об'єкт дослідження.

Програмно-технічні, організаційні принципи, концептуальні підходи до побудови веб-сервісу з визначеним функціоналом.

Предмет дослідження.

Програмно-технічні, організаційні засади, принципи, підходи щодо побудови веб-сервісу для перегляду культурних заходів.

Методи дослідження.

Системний аналіз, теорія проектування інтерфейсів, методи реалізації інтерфейсів, аналіз основних функцій, створення програмного застосунку веб-сервісу.

Ключові слова: веб-сервіс, перегляд культурних заходів, CMS-система, WordPress, Figma, бронювання квитків.

ABSTRACT

Thesis: 59 pages, 25 figures, 7 tables, 6 appendixs, 28 literary sources.

This thesis is devoted to the development of a web service for viewing cultural events.

The purpose of the thesis is building a web-service for viewing cultural events, for this you need to solve the following tasks: To achieve this goal you need to solve the following tasks:

- Explore the theoretical foundations of building web services online
- Analyze software and hardware solutions, different types of web service derivation of shortcomings in these systems
- Design, implement and implement a web service based on engineering requirements

Object of research.

Software and technical, organizational principles, conceptual approaches to building a web service with a certain functionality.

Subject of research.

Software and technical, organizational principles, principles, approaches to building a web service for viewing cultural events.

Research methods.

System analysis, theory of interface design, methods of implementation of interfaces, analysis of basic functions, creation of software application of web service.

Keywords: web service, viewing of cultural events, CMS-system, WordPress, Figma, ticket booking.

ЗМІСТ

Вступ	8
Розділ 1. СУЧАСНІ ПІДХОДИ ДО РОЗРОБЛЕННЯ ТА ВПРОВАДЖЕННЯ ВЕБ-СЕРВІСІВ ДЛЯ ПЕРЕГЛЯДУ КУЛЬТУРНИХ ЗАХОДІВ	9
1.1 Актуальність веб-сервісів і технологій	9
1.2 Технології для розробки веб-сервісів	13
1.3 Функціональні рішення, принципи та підходи для організації веб- сервісу для перегляду культурних заходів	20
Розділ 2. АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-СЕРВІСІВ	24
2.1 Збір інформації та формування вимог до веб-сервісу	24
2.2 Створення дизайн-концепції	27
2.3 Проектування та дизайн інтерфейсу	32
Розділ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ ДЛЯ ПЕРЕГЛЯДУ КУЛЬТУРНИХ ЗАХОДІВ	39
3.1 Програмна реалізація веб-сервісу	39
3.2 Розробка адміністративної частини веб-сервісу	42
3.3 Огляд фіналізованого функціоналу веб-сервісу	47
ВИСНОВОК	53
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	57

ВСТУП

Веб-сервіси допомагають людям розвивати бізнес, доносити інформацію, займатись мистецтвом, тощо. Тому є актуальною задача створити веб-сервіс для перегляду культурних заходів.

Актуальність дослідження цієї дипломної роботи зумовлено потребою бізнесу існувати в мережі internet, веб-сервіс орієнтований на всі категорії користувачів та на вирішення задач для бізнесу.

Метою дипломної роботи є побудова веб-сервісу для перегляду культурних заходів, для цього необхідно вирішити такі завдання:

- Дослідити теоретичні основи побудови веб-сервісів у режимі онлайн;
- Проаналізувати програмо-технічні рішення, різні види забезпечення веб-сервісу
- Спроекувати і впровадити веб-сервіс.

Об'єктом дослідження дипломної роботи є веб-сервіс для перегляду культурних заходів. Його функціональні рішення, принципи та підходи для організації основного веб-сервісу.

Предметом дослідження дипломної роботи є програмно-технічні, організаційні принципи, концептуальні підходи до побудови веб-сервісу з визначеним функціоналом.

Методи дослідження – системний аналіз, теорія проектування інтерфейсів, методи реалізації інтерфейсів.

Практичне значення одержаних результатів – виконана робота надасть можливість користувачами залишати заявки для зворотнього зв'язку та користуватись навігацією веб-сайту.

Структура роботи: дипломна робота складається зі вступу, трьох розділів, розподілених на підрозділи, висновку, додатків та списку використаних джерел.

Розділ 1. СУЧАСНІ ПІДХОДИ ДО РОЗРОБЛЕННЯ ТА ВПРОВАДЖЕННЯ ВЕБ-СЕРВІСІВ ДЛЯ ПЕРЕГЛЯДУ КУЛЬТУРНИХ ЗАХОДІВ

1.1 Актуальність веб-сервісів і технологій

Перш за все, веб-сервіси - це технологія. І як будь-яка інша технологія, вони мають досить чітко окреслену середу застосування.

Якщо подивитись на веб-сервіси в розрізі стека мережевих протоколів, ми побачимо, що це, в класичному випадку, ще одна надбудова над протоколом HTTP.

З іншого боку, якщо гіпотетично розділити інтернет на кілька шарів, ми зможемо виділити, як мінімум, два концептуальних типу додатків - обчислювальні вузли, які реалізують нетривіальні функції та прикладні веб-ресурси. При цьому другі, часто зацікавлені в послугах перших.

Але і сам інтернет - різномірний, тобто різні додатки на різних вузлах мережі функціонують на різних апаратно-програмних платформах і використовують різні технології та мови [1].

Серверна частина у більшості випадків реалізована за допомогою мови програмування PHP.

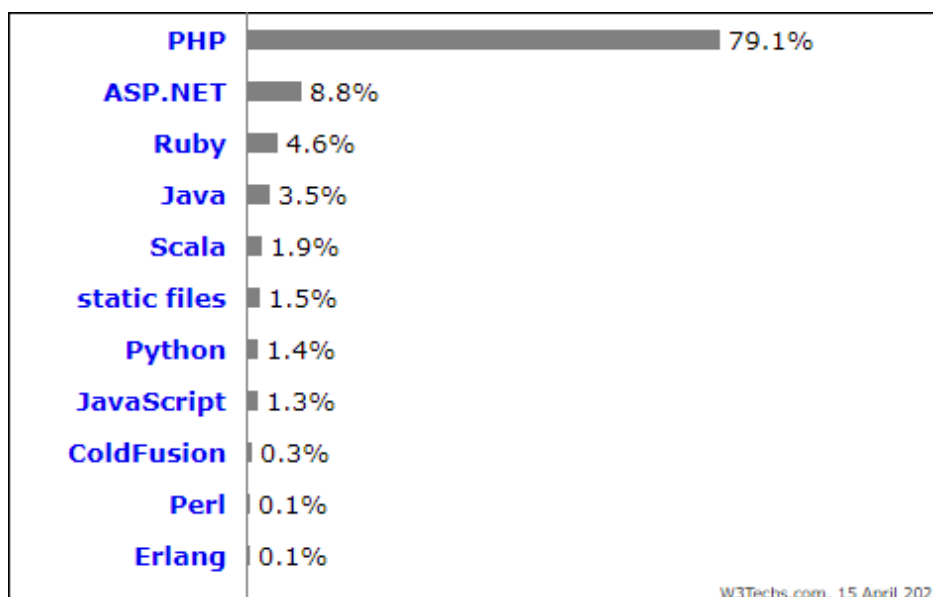


Рисунок 1.1 — Статистика мов програмування для серверної частини

Мова програмування звісно залежить ще від задач для розробника. В свою чергу back-end може реалізуватись двома різними підходами:

1. ООП;
2. Процедурний стиль

Який з цих варіантів використовувати залежить від складності проекту. Найчастіше, якщо проект складний та буде масштабуватися в майбутньому використовується підхід ООП.

Якщо проект простий та не вимагає майбутнього розвитку, або весь back-end розробляє один програміст використовується процедурний стиль. Це не являє собою аксіому, всі ці моменти можуть обговорюватись до початку реалізації [2].

Клієнтська частина як правило використовує JavaScript.

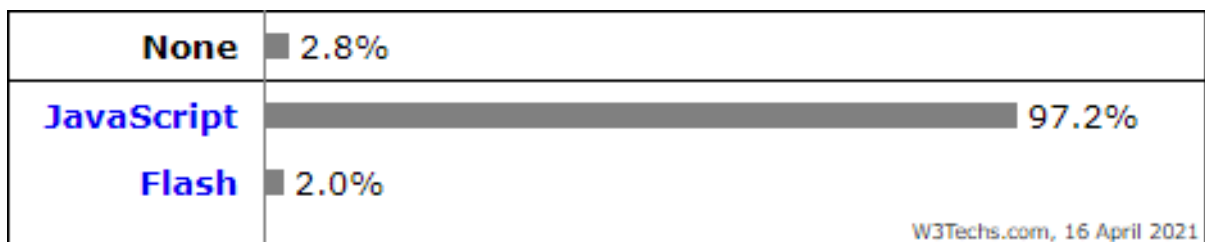


Рисунок 1.2 — Статистика мов програмування для клієнтської частини

2,8% веб-сайтів не використовують жоден з вищепоказаних мов програмування на стороні клієнта. JavaScript використовується на 97,2% всіх веб-сайтів.

Окрім класичного підходу до реалізації проекту для прискорення та здешевлення розробки та взаєморозуміння розробників, працюючих в одній команді використовуються різні бібліотеки як для back-end, так і для front-end.

Найпоширеніші фреймворки для розробки back-end:

1. Laravel
2. CodeIgniter
3. Symfony
4. Yii

5. CakePHP

Найпоширеніші фреймворки для розробки front-end:

1. Bootstrap
2. Semantic UI
3. Foundation
4. Materialize

Веб-сервіси - це реалізація абсолютно чітких інтерфейсів обміну даними між бізнесом та клієнтами, які написані не тільки на різних мовах, але й розподілені на різних вузлах мережі.

До веб-сервісів відносяться:

- Дошки оголошень,
- Каталоги,
- Онлайн-консультації,
- Пошук товарів та порівняння цін,
- Рішення для бізнесу,
- Торгові маркетплейси
- Системи бронювання

Для реалізації типових задач в веб-розробці ми можемо використовувати вже готові рішення (CMS – системи управління контентом). Які в більшості є безкоштовними та мають відкритий вихідний код.

За допомогою веб-сервісу користувач в режимі реального часу отримує послугу або здійснює: купує квиток, обробляє фотографію, збирає презентацію або перераховує одну валюту в іншу за поточним курсом, чи записується до лікаря. З цієї точки зору веб-сервіси нагадують програми або мобільні додатки. Поки візитки або лендінги просто продають послуги і розповідають про компанії, веб-сервіси несуть практичну користь.

Сервіси мають вирішувати проблему користувача за короткий проміжок часу та за допомогою нескладних маніпуляцій. Якщо вам потрібно замовити банківську карту по ідеї ви би мали піти в банк, простояти в черзі, пояснити консультанту які умови обслуговування вам потрібні, підписати необхідні

документи, та лише після цих дій, не завжди одночасно з їх завершенням ви зможете отримати завітну кредитну карту. Але в сучасному світі завжди присутній альтернативний варіант, який дозволяє вам скористатися сервісами передових банків, ознайомитись зі всією інформацією в одному місці та замовити потрібний вам продукт онлайн [3].

Якісний сервіс як правило має мінімалістичний зовнішній вигляд. Інтерфейс має бути не нагромадженим та не відволікати користувача від основних функцій, чим менше вибору, тим краще. Гарний приклад це відомі пошукові системи та карти, на сторінках з картами ви знайдете тільки карті, а на сайті в з пошуком інформації тільки пошук, без надмірних рекламних матеріалів та сторонніх ресурсів.

Для зручності користувача, ми маємо провести багато досліджену в області UX-аналітики, а витрати на це можуть бути досить великими, більше ніж в невеликих сайтах, наприклад односторінкових [4].

Правильний набір технологій - перший крок до створення будь-якого продукту, в тому числі і веб-сервісу. Одні з найпоширеніших технологій для створення веб-сервісів:

- Мови програмування PHP, Python, Ruby.
- Фреймворки Laravel, Yii2, Symfony, Django, Ruby On Rails.
- Бази даних MySQL, PostgreSQL, Redis, MongoDB.
- Операційні системи Ubuntu, CentOS.

На сьогоднішній день найбільшого поширення набули такі протоколи реалізації веб-сервісів:

- SOAP (Simple Object Access Protocol) — по суті це трійка стандартів SOAP / WSDL / UDDI
- REST (Representational State Transfer)
- XML-RPC (XML Remote Procedure Call)

SOAP - це аббревіатура від Simple Object Access Protocol. Це протокол обміну повідомленнями на основі XML для обміну інформацією між комп'ютерами. SOAP - це програма специфікації XML.

Хоча SOAP може використовуватися в різних системах обміну повідомленнями і може бути доставлений за допомогою різноманітних транспортних протоколів, початковий фокус SOAP - це виклики віддалених процедур, що транспортуються через HTTP.

Інші фреймворки, включаючи CORBA, DCOM та Java RMI, надають аналогічну функціональність SOAP, але повідомлення SOAP повністю написані у форматі XML і, отже, унікально незалежні від платформи та мови.

1.2. Технології для розробки веб-сервісів

Хмарні послуги - це інфраструктура, платформи або програмне забезпечення, яке розміщується сторонніми провайдерами та стає доступним для користувачів через Інтернет.

Хмарні сервіси полегшують потік даних користувачів від інтерфейсних клієнтів (наприклад, серверів користувачів, планшетів, настільних комп'ютерів, ноутбуків - будь-чого, що є на кінцях користувачів), через Інтернет, до систем провайдера та назад. Користувачі можуть отримати доступ до хмарних служб лише за допомогою комп'ютера, операційної системи та підключення до Інтернету або віртуальної приватної мережі (VPN).

Для розвитку бізнесу в інтернеті найкращими будуть:

1. Wix
2. Prom.ua
3. Facebook
4. Instagram
5. Tilda

Завдяки тому, що від користувача прихована абсолютно вся інфраструктура хмари йому не потрібно володіти спеціальними навичками або знаннями для використання і управління даною технологією.

Такі технології передбачають швидкий старт, або надання послуг за мінімальний бюджет.

З основних недоліків можна виділити те, що ві не є власником програмного коду та функціонал обмежений самим хмарним сервісом, який не завжди повністю безкоштовний.

Крім того, хмарні сервіси популярні серед невеликих і середніх компаній тому, як вони "демократизують" CRM-системи [6].

Плюси хмарних CRM-систем:

1. У більшості організацій глибоко вкорінений страх, що встановлення CRM - це тривалий і складний процес, але це вже не відповідає дійсності, коли йдеться про хмарний CRM. Сучасний хмарний CRM може бути запущений і запущений в найкоротші терміни. Все, що вам насправді потрібно - це підключення до Інтернету. Як тільки ви ввійдете в Інтернет, ви зможете негайно розпочати використання програмного забезпечення та максимально використати його можливості. Вам не потрібно налаштовувати апаратне забезпечення, платити за обслуговування сервера та програмного забезпечення, мати ІТ-спеціаліста на місці цілодобово або турбуватися про складні установки, міграцію даних або навіть оновлення - про все це піклується ваша хмара Постачальник послуг CRM;
2. Найбільш суттєвими інвестиціями в традиційні CRM-системи були попередні капітальні видатки, але вже ні. Подібно до інших хмарних рішень, хмарні CRM працюють за моделлю передплати під час оплати. За допомогою мінімальних попередніх інвестицій хмарні CRM зменшують шанси ризику і дозволяють меншим організаціям швидше досягти економії від масштабу. Оскільки малий бізнес, швидше за все, не виділяє великі суми грошей на систему підприємств, ці менші регулярні внески допомагають їм використовувати багато можливостей, які є, одночасно покращуючи свої результати;
3. На відміну від думки, що хмарні рішення не дуже безпечні, компанії, які пропонують хмарний CRM, дуже серйозно ставляться до безпеки.

Завдяки вдосконаленим політикам автоматизованого резервного копіювання та надійним планам відновлення даних, хмарні CRM виводять кібербезпеку на абсолютно новий рівень. Більше того, ви можете додатково покращити безпеку та захист даних, використовуючи двоетапну автентифікацію та надійні паролі;

4. Традиційні, самостійні продукти часто не можуть забезпечити гнучкості та інтеграції, яких вимагає більшість команд. Особливо це стосується середовища малого бізнесу, де обмін інформацією між наявними технологіями є першорядним. Хмарний CRM може бути легко інтегрований з іншими програмами та програмним забезпеченням, наприклад, електронними поштовими клієнтами та інструментами продуктивності, що дозволяє безперешкодно виконувати завдання та забезпечувати наскрізну видимість протягом усього процесу продажів.;

Мінуси хмарних CRM-систем:

1. Одним з найбільших недоліків Cloud CRM є те, що доступність, як би це не було доступно, безпосередньо залежить від кількості користувачів у віддалених середовищах. Це віддалене середовище може бути просто в іншому розділі закладу, але воно повинно включати мобільних користувачів, що мають доступ через Інтернет. Це реальне використання, яке надає перевага Cloud CRM. Це може суттєво підвищити рівень відповідності та продуктивності роботи мобільних користувачів.;
2. Хоча це програмне рішення є безпечним, той факт, що дані зберігаються в централізованій точці, створює загрозу для будь-якої компанії, яка впроваджує рішення CRM. Що станеться, якщо живлення вийде з ладу і ваше програмне забезпечення не збереже дані в хмарі? Або що трапляється, якщо незадоволений працівник вводить дані, які не є точними, як спосіб помсти? Повинні бути

системи резервного копіювання та моніторингу, щоб можна було підтримувати цілісність даних;

3. Встановлення нового рішення CRM може бути дуже захоплюючим. Ідея автоматизації багатьох процесів та впорядкування даних може бути дуже привабливою. Але потім приходить процес впровадження, і все змінюється. Хоча для цих систем не потрібен фахівець, персонал може витратити час, щоб звикнути до нього та навчитися користуватися новим програмним забезпеченням;
4. Процеси автоматизуються, коли встановлюється програмне забезпечення CRM, а це також означає, що потенційні клієнти втрачають частину людського елемента, котрий іноді вимагає комерційне завдання. Існує різниця між тим, що при здійсненні телефонного дзвінка допомагає реальна людина, ніж автоматизована система обслуговування клієнтів. Іноді добре зробити завдання більш автоматизованими, але також добре мати невелику персоналізовану допомогу, щоб уникнути втрати цінності людського елемента;

Коробочні продукти, як правило розроблені для неточної кількості користувачів з класичним функціями під них. В цьому випадку стандартний набір рішень має ряд стандартизація та обмежень, посамому функціоналу, по спробам масштабування та залежить від оновлень розробником. Такі програми продаються як вже фіналізовані системи, тому вони важко адаптуються під вимоги та завдання компаній, які їх впроваджують на своїх підприємствах [7].

Переваги коробочних CRM-систем:

1. Зручність та швидкість, можна користуватись одразу після покупки та базових налаштувань;
2. Ліцензовані та продаються у великій кількості магазинів;

3. Зручно користуватись, тому що ліцензовані продукти завжди випускаються з інструкцією, що створить низький поріг входу для початку користування продуктом;
4. Працюють як з доступом до інтернету, так і без нього, що робить їх досить гнучкими у використанні.

Мінуси коробочних CRM-систем:

1. Треба мати високий рівень технічних знань, для того щоб самостійно установити та налаштувати систему;
2. Іноді виникають проблема при роботі паралельно з іншими програмами;
3. Іноді в коробочних системах за замовчуванням знаходиться дуже велика кількість функціоналу, що впливає на їх ціну, як правило клієнти не можуть скоротити не потрібний їм функціонал, відповідно і ціна для них виходить завищеною;
4. Як правило ліцензії видаються на один пристрій та не дозволяють докупити їх на більшу кількість.

Система управління контентом, яку часто скорочують як CMS, - це програмне забезпечення, яке допомагає користувачам створювати, керувати та модифікувати вміст на веб-сайті без необхідності спеціальних технічних знань.

Простішою мовою, система управління контентом - це інструмент, який допомагає вам створити веб-сайт без необхідності писати весь код з нуля (або навіть взагалі знати, як кодувати).

Замість того, щоб створювати власну систему для створення веб-сторінок, зберігання зображень та інших функцій, система управління контентом обробляє всі основні інфраструктурні матеріали для вас, щоб ви могли зосередитись на більш зовнішніх частинах вашого веб-сайту.

Крім веб-сайтів, ви також можете знайти системи управління вмістом для інших функцій, таких як управління документами [8].

Основні плюси CMS:

1. Великий вибір безкоштовних CMS під різні задачі;
2. Гнучкість в доопрацюванні;
3. Відкритий вихідний код;
4. Документація;
5. Велика кількість написаних доповнень (плагінів) для розширення функціоналу;
6. Готовий front-end.

Основні мінуси:

1. Відкритий вихідний код – це проблеми з безпекою. Будь-який зловмисник може з'ясувати яка CMS використовується для вашого сайту, будучи з нею ознайомленим він може спробувати отримати доступ до вашого сайту;
2. Масштабованість в межах можливостей back-end'у самої CMS.

Перед тим як починати роботу над створенням веб-сервісу для перегляду культурних заходів необхідно спочатку вивчити пропозиції на ринку: існує дуже багато варіантів під різні потреби, умови використання та нюанси. Умовно усі CMS можна поділити на відкриті, які працюють безкоштовно та дозволяють користувачи вносити зміни до вихідного коду та закриті, як правило платні системи. Найпопулярніші системи зараз:

- Joomla
- Drupal
- Magento (for eCommerce stores)
- Squarespace
- Wix
- TYPO3
- WordPress

Підбираючи для себе систему необхідно обирати з власних уподобань та потреб, хтось надає перевагу одному рішення, комусь набагато зручніше

ббуде користуватись іншим. Все залежить від задач та потреб компанії і підходить для вирішення її задач.

Слад пам'ятати також і про раціональність: продукт з великим нагромадження функціоналу, більшість з якого ви можливо не будете ніколи використовувати немає сенсу купувати. Але й навпаки, інколи прості CMS можуть просто не впоратись з велики сайтами, які складаються зі складного функціоналу.

Таблиця 1.1 — Класифікаційна таблиця технологій

Назва	Чат-підтримка	Ціна	Кастомізація	Швидкість реалізації	Шаблонний дизайн
WordPress	Ні	Безкоштовно	Так	Швидко	Так
Wix	Так	Платно	Ні	Швидко	Так
Prom.ua	Так	Платно	Ні	Швидко	Так
Tilda	Так	Платно	Ні	Швидко	Так
Власний back-end (PHP)	Ні	Безкоштовно	Так	Довго	Ні

В цій таблиці розглядаються функціональні можливості різних технологій, після ознайомлення з ними, можна зробити висновок, що для веб-сервісу по перегляду культурних заходів найкраще підходить WordPress, тому що на ньому зручно відтворити постійно оновлюваний блог з можливістю масштабування та досить швидка реалізація будь-якого функціоналу, також він дає змогу реалізувати бронювання квитків користувачами та отримати зворотній зв'язок.

1.3. Функціональні рішення, принципи та підходи для організації веб-сервісу для перегляду культурних заходів.

Перегляд культурних заходів являє собою сайт, присвячений культурним заходам. Такі сайти як правило створюють у форматі новинних сайтів, або блогів. Для обох цих категорій є схожі патерни проектування.

Зазвичай сайт складається з трьох ключових сторінок:

1. Головна сторінка;
2. Сторінка сортування статей;
3. Сторінка статті.

Виходячи з цього можна зробити висновок, що сайт має бути адаптивним під всі сучасні розміри екранів (1200px, 960px, 640px, 320px). Також мають використовуватись сучасні технології верстки згідно до вимог SEO-просування та пошукових систем. Враховуючи тематику сайту користувач повинен отримати якомога швидший доступ до інформації, тому необхідно звести до мінімуму використання різних скриптів та бібліотек, які можуть вплинути на швидкість завантаження сторінок, це також стосується й дизайну, він повинен бути мінімалістичним та не перешкоджати користувачу отримувати доступ до інформації. Повинні бути сверстані всі сторінки, які відрізняються між собою.

Вимоги щодо back-end'у:

1. Зручна адміністративна панель;
2. Підтримка декількох адміністраторів, з різними рівнями доступу;
3. Реалізація коментарів
4. SEO-оптимізація;
5. Гнучкість. Можливість створювати необмежену кількість сторінок, розділів, новин і т.д;
6. Простота в обслуговуванні.

В результаті аналізу можна стверджувати, що веб-сервіс являє собою типову задачу в веб-розробці, тому ми можемо зупинити свій вибір на готових

системах управління контентом, що дозволить нам прискорити реалізацію проекту та спростити подальше обслуговування.

Щоб дати вам уявлення про те, як працює система управління контентом, ми здійснимо поверхневий огляд по інтерфейсу WordPress (WordPress є гарним прикладом системи управління вмістом).

Почнемо зі створення частини вмісту. Без системи управління вмістом вам потрібно буде написати статичний HTML-файл і завантажити його на свій сервер (звучить складно, правда?).

За допомогою системи управління вмістом, такої як WordPress, ви можете просто писати свій вміст в інтерфейсі, схожому на Microsoft Word:

Так само, щоб завантажувати та керувати мультимедіа, як зображення, ви можете просто переглядати медіатеку, замість того, щоб насправді взаємодіяти безпосередньо з вашим веб-сервером.

Однак система управління вмістом - це не просто інтерфейс внутрішнього управління. Крім того, весь створений вами вміст відображається для відвідувачів саме таким, яким ви цього хочете.

Інформація завжди зберігається в базі даних (стандартна база даних в WordPress – це MySQL), або в файлах. Для коректної роботи багатьох CMS потрібні певні хостинги. Якщо back-end написаний на мові PHP і йому потрібна база даних MySQL для належного функціонування, то вам потрібен хостинг, який задовольняє такі потреби [10].

В даному випадку із всіх CMS найбільше за своїм функціоналом нам підходить WordPress, тому що:

1. Це одна з найпопулярніших систем управління контентом, а отже є велика кількість кадрів на ринку, яких можна залучити для обслуговування, або доробки проекту;
2. Великий вибір готових розширень, за допомогою яких в майбутньому можна розширювати функціонал;
3. Проста та зрозуміла адмін-панель;

4. Підтримка декількох адміністраторів, з різними рівнями доступу до проекту;
5. Можливість створювати необмежену кількість сторінок, розділів, новин і т.д.
6. Докладна документація в вільному доступі для інтеграції front-end'у на дану CMS.

WordPress - це найпростіший, найпопулярніший спосіб створити власний веб-сайт або блог. Насправді WordPress забезпечує понад 40,0% всіх веб-сайтів в Інтернеті. Так - більше, ніж кожен четвертий веб-сайт, який ви відвідуєте, працює на основі WordPress.

На дещо більш технічному рівні WordPress - це система управління контентом з відкритим кодом, ліцензована, що означає, що кожен може безкоштовно використовувати або модифікувати програмне забезпечення WordPress. Система управління контентом - це в основному інструмент, який дозволяє легко керувати важливими аспектами вашого веб-сайту, без необхідності мати поглиблені навички в програмуванні.

Кінцевим результатом є те, що WordPress робить створення веб-сайту доступним для всіх - навіть людей, які не є розробниками.

Багато років тому WordPress був насамперед інструментом для створення блогу, а не більш традиційних веб-сайтів. Однак це давно не відповідає дійсності. На сьогоднішній день, завдяки змінам основного коду, а також масивній екосистемі плагінів та тем WordPress, ви можете створити веб-сайт будь-якого типу за допомогою WordPress.

Наприклад, WordPress не тільки працює з величезною кількістю бізнес-сайтів та блогів, це ще й найпопулярніший спосіб створити магазин електронної комерції. За допомогою WordPress ви можете створювати:

- Бізнес веб-сайти
- Інтернет-магазини
- Блоги
- Сайти-портфоліо

- Резюме
- Форуми
- Соціальні мережі

WordPress був створений як самостійний проект ще в 2003 році, виникши як відгалуження попереднього проекту, який називався b2 / safelog.

WordPress - це програмне забезпечення з відкритим кодом, тож сьогодні його створює величезна спільнота авторів. Але якщо ми мали простежити походження WordPress назад до його коренів, його оригінальним творінням стала співпраця між Меттом Малленвегом та Майком Літглом.

З тих пір Метт Малленвег здебільшого став обличчям WordPress. А ще він є засновником Automattic, що є компанією, що стоїть за комерційною службою WordPress.com.

Історія WordPress між його заснуванням як платформи для блогів у 2003 році і сьогодні є довгою ...

Досить сказати, WordPress просунувся вперед і завдяки своїм авторам та величезному співтовариству перетворився на найпопулярніше рішення для створення веб-сайтів будь-якого типу.

Висновок до розділу 1

У цьому розділі було досліджено актуальність веб-сервісів, розглянуто програмні засоби для створення веб-сервісу для перегляду культурних заходів. Проведено порівняльний аналіз актуальних технологій, з усіх варіантів реалізації моєї дипломної роботи найбільш оптимальним буде WordPress, тому що з усіх перелічених технологій він має таку саму гнучкість, як розробка власного backend'у, при цьому він є повністю відкритою та безкоштовною платформою, на якій можна швидко створити та запустити MVP-версію веб-сервісу.

WordPress дає змогу створити повноцінний портал для перегляду культурних заходів, реалізувати функцію бронювання квитків та зручні налаштування для подальшого масштабування веб-сервісу.

Розділ 2. АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ І ВІБИР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-СЕРВІСІВ

2.1 Збір інформації та формування вимог до веб-сервісу

Перший етап розробки веб-сервісу – це брифінг. Бриф – це спеціальний документ для опитування клієнта, багато відповідних питань щодо проекту, відповівши на які клієнт дає картину продукту, який він хоче отримати. Бриф заповнюється клієнтом, але в нашому випадку ми зможемо попрацювати за клієнта та самостійно його заповнити. У брифі зберігаються основні пункти, які допомагають отримати максимальну картину бачення продукту. Це один з найважливіших документів перед початком розробки будь-якого продукту.

Всі наступні етапи виконуються на основі того, що написав клієнт. Бриф слід створювати таким чином, щоб до виконавця була мінімальна кількість запитань з уточненнями [13].

Таблиця 2.1 Бриф

Контактна інформація	
Компанія	Cultural events
Контактна особа	Вкажіть ПІБ контактної особи та її посаду. Ми будемо зв'язуватися з цією людиною для уточнення інформації та затвердження етапів
Телефон	Телефон контактної особи
E-mail	mail@example.ru
Інші контакти	Skype, WhatsApp
Про компанію	
Сфера діяльності	Афіша культурних заходів
Продукт/послуга	Новинний сайт для перегляду культурних заходів

Продовження таблиці 2.1

Опис послуг	Веб-сервіс, який дозволяє переглядати та бронювати квитки на майбутні культурні заходи.
Унікальна торгова пропозиція	У нас гарне співвідношення ціни та якості, унікальні пропозиції на ринку
Географія	Україна, Київ
Кратка інформація про компанію	5 років на ринку, 3 партнери, 7 співробітників, 10 сертифікатів
Сайт компанії	Example.ru
Посилання на соціальні мережі	ВКонтакте, Facebook, Instagram, LinkedIn
Запроси в пошуковиках, за якими індексуються ваші послуги	Культурні заходи, афіша.
Користувач	
Яку проблему вирішує ваш продукт для користувача	Продукт – допомагає користувачами знаходити актуальні культурні події у їхньому місті, ознайомлюватись з описом подій та бронювати квитки.
Стать	Чоловіки – 50%, жінки – 50%
Вік	Від 25 років
Фінансове положення	Середній рівень заробітної платні
Інтереси	Мистецтво, гумор, концерти, музика, кіно, телебачення, політика.
Сайт	

Продовження таблиці 2.1

Цілі, які повинен вирішити сайт	- Залучення нових клієнтів - Підвищення рівня конверсії - Збільшення поточних обсягів продажів - Популяризація позитивного іміджу компанії
Конверсійна дія, очікувана від користувача	Забронювати квиток
Бажаний стиль майбутнього веб-сайту	Стильний, легкий для сприйняття
Особливі переваги щодо колірної гами	Білий кольор
Які розділи хотіли б бачити?	1. Хедер 2. Категорії заходів 3. Слайдери з актуальними заходами 4. Сторінка заходу 5. Сторінка бронювання квитків 6. Медіаблок 7. Контакти 8. Сертифікати
Додаткові побажання по сайту	Зробити йогу легким та зручним для використання
Технічні аспекти	Спроекувати унікальний дизайн та розробити веб-сервіс за допомогою CMS WordPress

Окрім цих розділів брифу необхідно також розписати функціонал веб-сайту, такі параметри як структура, навігація, тип оплати, реєстрацію та наявність особистого кабінету.

Також рекомендується знайти 3-10 конкурентів та розписати, що в їх сервісах вам подобається, а що – ні. Після цього потрібно обов'язково запросити матеріали для сервісу. До матеріалів належать:

- Логотип;
- фірмовий стиль;
- фото продукту;
- відеоролики;
- сертифікати, дипломи;
- акції, знижки;
- новини, статті, події;
- відгуки;
- тексти/презентації;

Наприкінці файлу брифу як правило пишуть невеликий дескриптор, де дякують клієнту за те, що він знайшов час його заповнити. Бриф являє собою першу версію технічного завдання та потім еволюціонує в нього.

2.2 Створення дизайн-концепції

Першим етапом для створення дизайн-концепції є функціональний прототип.

Функціональний прототип – це прототип, у якому створюються блоки сайту та їх складові частини, без застосування графічного методу відображення прототипів, що дає змогу зосередити увагу на контенті, а не на його візуалізації.

Як правило такий прототип складається після конкурентного аналізу, що дає змогу побачити які блоки використовують конкуренти та на основі їх сайтів запропонувати клієнту свою структуру та блоки.

Після створення функціонального прототипу необхідно його презентувати клієнту, у форматі workshop, тому що спочатку можна було не врахувати деякі нюанси та були можливість їх виправити разом з клієнтом.

Під час прототипування фактично визначається кількість блоків на сайті та їх складові частини [14].

Таблиця 2.2 — Функціональний прототип головної сторінки

Обкладинка	<p>Опис</p> <ul style="list-style-type: none"> ● Промо-подія для залучення уваги <p>Склад</p> <ul style="list-style-type: none"> ● Велике фото ● Заголовок події ● Короткий опис ● Дата ● Категорія
Категорія мистецтво	<p>Опис</p> <ul style="list-style-type: none"> ● Набір карточок статей <p>Склад</p> <ul style="list-style-type: none"> ● Заголовок статті ● Категорія ● Дата ● Фото
Категорія концерти	<p>Опис</p> <ul style="list-style-type: none"> ● Набір карточок статей <p>Склад</p> <ul style="list-style-type: none"> ● Заголовок статті ● Категорія ● Дата ● Фото
Категорія гумор	<p>Опис</p> <ul style="list-style-type: none"> ● Набір карточок статей <p>Склад</p> <ul style="list-style-type: none"> ● Заголовок статті ● Категорія ● Дата ● Фото
Footer	<p>Опис</p> <ul style="list-style-type: none"> ● Підвал сайту <p>Склад</p> <ul style="list-style-type: none"> ● Логотип ● Навігація ● Контакти

Після завершення етапу створення функціонального прототипу необхідно розробити технічне завдання.

Технічне завдання - це суміш брифу та функціонального прототипу. Як правило це постійно оновлюваний документ, який являє собою фіксування вимог та умов проекту.

До складу технічного завдання входять:

1. Інформація про контактну особу клієнта (Контактна особа, посада, телефон, e-mail);
2. Інформацію про компанію (сфера діяльності, послуга, опис послуги, унікальна торгова пропозиція, географія, кратка інформація про компанію, сайт компанії, посилання на соціальні мережі);
3. Дані про користувача (яку проблеми вирішує продукт для користувача, стать, вік, фінансове становище, інтереси);
4. Відомості стосовно бажаного сайту (цілі, які повинен вирішувати сайт, конверсійна дія, очікувана від користувача, бажаний стиль майбутнього сайту, особові вподобання щодо кольорової гама сайту);
5. Матеріали, які клієнт надає розробнику. (Логотип, фірмовий стиль, фотографії, відео, сертифікати, акції, новини, відгуки, тощо.);
6. Функціональний прототип сайту.

Загальне визначення візуальної концепції – це графічна ідея майбутнього проекту, яка виражена за допомогою ескізів ключових розділів.

Формальне визначення – дизайн-макет, який дає достатнє уявлення про фінальний результат, щоб після його затвердження можна було почати фіналізувати роботу по дизайну [15].

Процес створення візуального дизайну:

1. Створення концепту;
2. Затвердження концепту;
3. Створення фінального дизайн-макету;
4. Затвердження фінального дизайн-макету.

Концепція – це гіпотеза, ніхто не може стверджувати, чи буде вона успішно працювати (виконувати завдання, які були прописані в технічному завданні), доки її не запустять. Можна лише відштовхуватись від свого особистого досвіду, або чужого (конкурентний аналіз).

Концепція відображає дві суті: логіка та емоція.

Логіка:

- Цілі, які вирішуються для клієнта;
- Потреби, які закриваються для користувача;
- Порядок оповідання (фіксується на етапі прототипу);

Вирішивши ці задачі, користувач буде лояльним та вдячним.

Емоція:

- Створити асоціацію емоції з продуктом;
- Виділити продукт на тлі конкурентів;
- Дати приємні емоції при взаємодії;
- Розважити, дати контент в приємному форматі (не переборщити);

Вирішивши ці задачі, користувач запам'ятовує сайт та асоціює з чимось приємним (буде приємно взаємодіяти повторно).

Послідовність кроків:

1. Оцінити, що вже є:
 - a. Фірмовий стиль;
 - b. Фото/відео;
 - c. Текстові матеріали;
 - d. Інформація про компанію в інтернеті;
2. Проаналізувати ринок зовні:
 - a. Логічні рішення (конкурентний аналіз);
 - b. Дизайн рішення (кольори, форми);
 - c. Контент (фото, відео, тексти);
 - d. Аудиторія;

Раніше було популярно складати портрет користувача, за наступним принципом:

Бізнесмен Андрій, 30 років, живе в Києві, заробляє вище середнього рівня заробітної плати, двічі на рік їздить у відпустку, тощо. Але вже настав час про це забути, зараз актуальними є тільки 3 параметри:

1. Інтернет-грамотність (наскільки користувач володіє UI елементами);
2. Рівень креативності (наскільки користувач зацікавлений у тому, щоб отримати естетичну насолоду від користування сайтом)
3. Зацікавленість у продукті;

Кожен з цих параметрів потрібно оцінити по шкалі від 0 до 10. Це робиться для того щоб максимально адаптувати дизайн під цільову аудиторію (наприклад для людей літнього віку).

Приклад адаптації інтерфейсу під інтернет грамотність аудиторії:

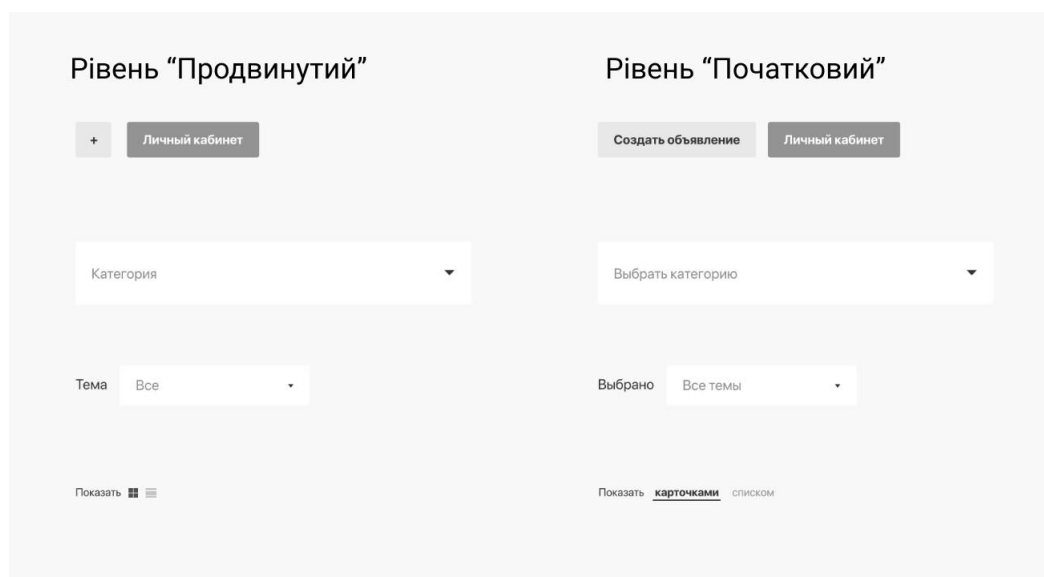


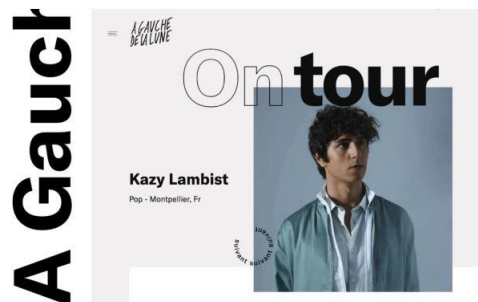
Рисунок 2.5 — Адаптація інтерфейсу під рівень інтернет-грамотності аудиторії

Якщо є наприклад п'ять груп користувачів і чотири з них сучасного рівня, а одна початкового, то потрібно робити дизайн зрозумілим для аудиторії початкового рівня, тому що завжди треба відштовхуватись від найслабшої ланки.

Подібним чином можна визначити рівень креативності аудиторії. Якщо розробляється наприклад інтернет-магазин з побутовою технікою, то деяких покупців краса може відштовхнути, тому що вони подумують, що це для них дорого, тощо.

Так само й якщо ви сайт-галерея, орієнтований на креативних людей – ви просто не можете собі дозволити зробити візуально не естетичний, шаблонний сайт [16].

Рівень “Дизайнер”



Рівень “Банкір”

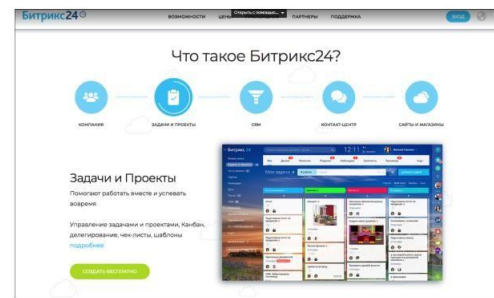


Рисунок 2.6 — Адаптація інтерфейсу під рівень креативності аудиторії

Для зацікавленої аудиторії потрібно представити переваги над конкурентами та інформацію про проект. Аудиторія, яка не зацікавлена у продукті проведе на сайті до 20 секунд, зацікавлена 5-10 хвилин. На основі цієї інформації можна коректувати структуру сайту: послідовність блоків та їх складові, для того щоб зачепити всі групи [17].

2.3 Проектування та дизайн інтерфейсу

Для того щоб визначити стилістику бажано завжди дивитись на конкурентів та адаптувати їх рішення під себе (не копіювати, а саме адаптувати).

Об’єкти, які визначають стиль:

- Форми (лінії, шейпи);
- Сітка;
- Шрифти;

- Кольори;
- Медіа-контент (фото, відео, ілюстрації);
- Текст (стиль тексту).

Індустрії, які визначають стиль:

- Мода;
- Мистецтво;
- Юриспруденція;
- Фінанси;
- Нерухомість;
- Технології;
- Їжа;
- Реклама/дизайн;

Контент – це основа концепції. Якщо на сайті неякісні фото, або погані тексти, то в першу чергу необхідно працювати з ними, а не з дизайном.

Дизайн – це скелет для контенту. Завжди перед тим, як почати роботу над інтерфейсом необхідно врахувати складові частини вже існуючої айдентики сайту, такі як:

- Логотип;
- Кольорова гама;
- Шрифт;
- Сітка;
- Медіа-контент;
- Елементи;

Після аналізу існуючих компонентів можна приступати до створення нового дизайну, інтегруючи його у вже існуючу середу сайту, інакше існує ризик створити несумісні за стилістикою блоки сайту, які будуть викликати стрес у користувача [18].

Проектування інтерфейсу складається з наступних етапів:

1. Постановка задачі;
2. Прототипування;

3. Пошук референсів;
4. Дизайн блоків;
5. Пошук фото;
6. Фіналізація;

На етапі постановки задачі в першу чергу потрібно ознайомитись з контентом для сайту (логотип, тексти, фото, тощо). Далі рекомендується скористатись конкурентним аналізом, який допоможе створити більш яскраві та цікаві рішення, адже без аналізу контенту дизайнер не зможе навіть обґрунтувати свої рішення. Тому що лише проаналізувавши конкурентів можна заявити про свою усвідомленість на ринку. Конкурентів бажано брати не менше п'яти (бажано 5-10), від лідерів ринку до аутсайдерів [19].



Рисунок 2.7 — Логотип

Наступним пунктом йде прототипування, насправді цей крок вже був частково виконаний, коли створювався функціональний прототип, який являється основою для основного прототипу.

Використовуючи функціональний прототип можна перейти до наступного кроку у прототипуванні – першого наближення макету. Починаємо малювати блоки нашого сайту вже у редакторі Figma. При першому наближенні не обов'язково все робити одразу ідеально та застосовувати потім на сайті, достатньо буде зробити лише начерк майбутнього сайту. Цей етап потрібен лише для того щоб візуалізувати особисте сприйняття сайту, його не потрібно нікому показувати.



Рисунок 2.8 — Перше наближення сайту

Наступним пунктом йде інтеграція текст у макет. При необхідності допускається рихтування текстів, наприклад “наші послуги” можна замінити на “послуги”, якщо це покращить візуальне сприйняття сайту [20].

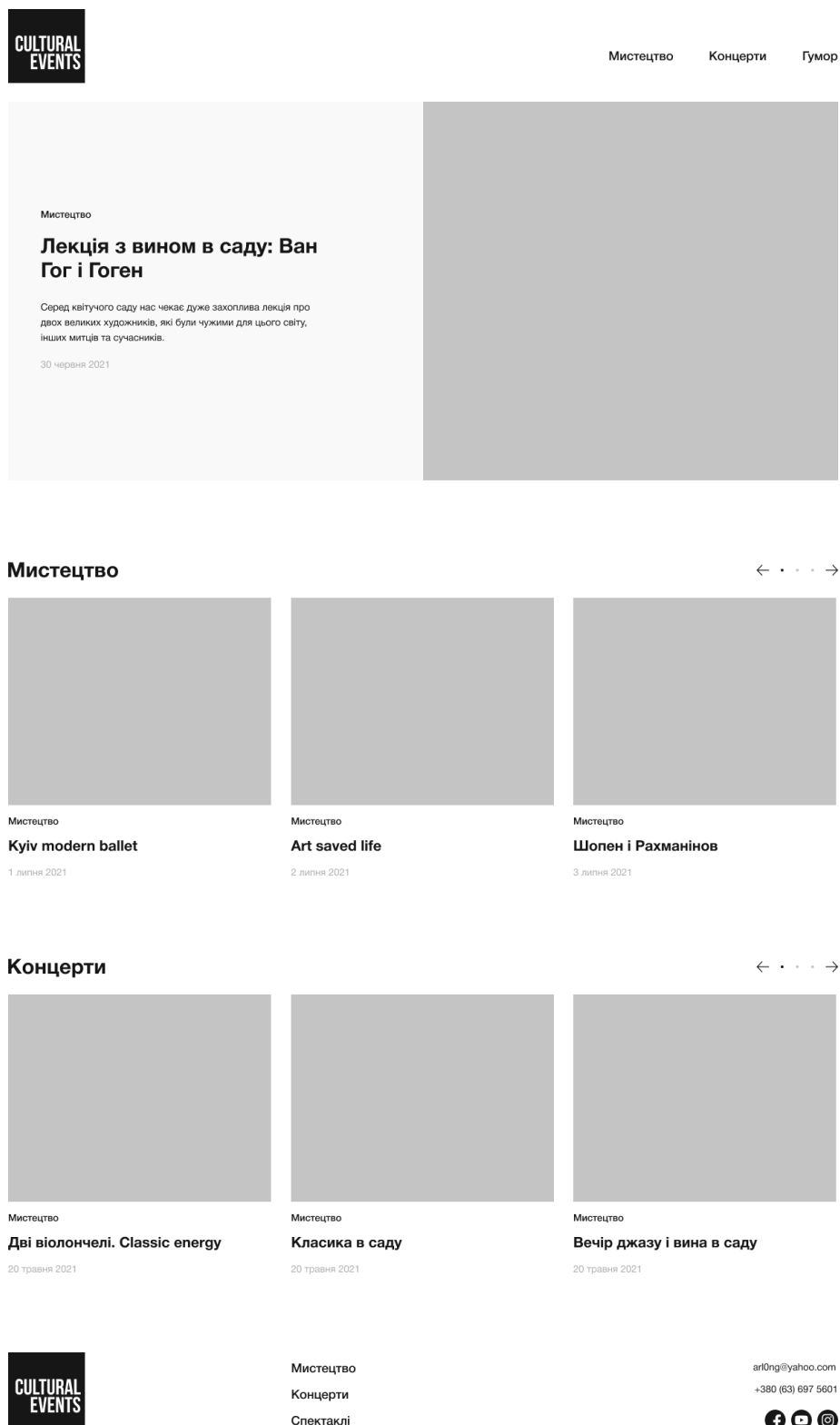


Рисунок 2.9 — Інтеграція тексту

Після вдалої інтеграції тексту нам потрібно знайти референси для нашого веб-сайту. Пошук референсів – це підбір вже існуючих рішень інших

дизайнерів інтерфейсу під наш проект, з майбутньою їх адаптацією. Наприклад, на обкладинці сайту є один великий заголовок та кнопка, тому дивитися рішення інших бажано з такими ж умовами.

Цей етап дозволяє методом перебору знайти найкращі візуальні рішення для нашого інтерфейсу. Після підбору 4-5 сайтів для прикладу можна переходити до дизайну блоків [21].

На мою думку дизайн блоків - це найцікавіший етап створення інтерфейсу, пошук візуальних зв'язків.

Блоки сайту можна розділити на 2 категорії:

- Ключові блоки;
- Шаблонні блоки;

До шаблонних блоків відносяться такі блоки, як: обкладинка, підвал, слайдер, хедер. Їх відносять до зрозумілих, тому що їх форма вже багато разів намальована великою кількістю людей, залишається знайти лише варіанти, які максимально підходять до концепції.

Ключові блоки – це блоки, в яких знаходиться основний контент сайту. Дизайн блоків рекомендується починати в першу з ключових блоків та шукати візуальні зв'язки між ними.

Під час процесу дизайну блоків зовнішній вигляд не повинен вас повністю задовільняти, тому що пізніше їх ще потрібно буде шліфувати. Також потрібно враховувати те, що через деякий час клієнт може замінити контент (текст, фото), тому треба створювати дизайн не зважаючи на контент, а концентруючись лише на загальній структурі блоків[22].

Фото як правило використовують самого продукту, емоцію людей від користування продуктом, або абстракцію на цей продукт.

Бажано, щоб фото підходили одне до одного по кольоровій гамі. Гарні фото можна знайти на безкоштовних ресурсах, наприклад unsplash.com.

Першим кроком фіналізації є додавання кольорів, вирівнювання відступів, покращення макету.

Другий крок – це вставка фото, з фотографіями макет починає набувати закінченого для користувача зовнішнього виду.

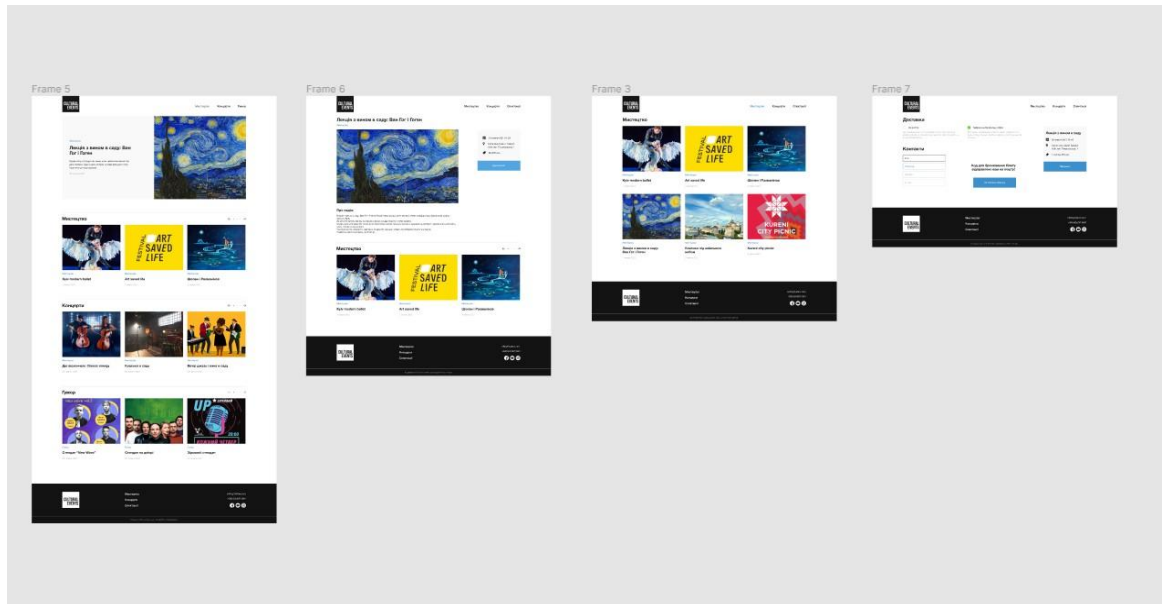


Рисунок 2.10 — Макети після фіналізації

Після того, як закінчиться фіналізація макету рекомендується взяти паузу на 1-2 дні, для того щоб свіжим поглядом ще раз подивитись на нього та можливо внести корективи [23].

Коли макети вже фіналізовано макетом необхідно зробити їх версії для всіх розмірів екрану. Взагалі прийнято малювати 4 розміри по ширині: 1600 px, 1000 px, 640 px та 320 px.

В першу чергу після десктопної версії рекомендується починати мобільну (320 px), тому що планшетна версія - це мобільна, тільки вдвічі ширше [24].

Висновок до розділу 2

У цьому розділі було створено бриф та функціональний прототип веб-сервісу для перегляду культурних заходів, проведено дослідження рівню інтернет-грамотності та потреб користувачів, на основі цих даних було створено дизайн-концепцію, після цього в графічному редакторі figma були спроектовані прототипи, інтегрований текст та розроблений фінальний дизайн для веб-сервісу. Веб сервіс був адаптований під всі розміри екранів.

Розділ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ ДЛЯ ПЕРЕГЛЯДУ КУЛЬТУРНИХ ЗАХОДІВ

3.1 Програмна реалізація веб-сервісу

Для верстки веб-сервісу використовувались HTML5/CSS3 та бібліотека jQuery. Використовувався сучасний стандарт в верстці з використанням логічних тегів розмітки що позитивно вплине на подальше просування даного сайту.

Для реалізації адаптивного меню, і прокрутки заходів на сторінках була обрана бібліотека jQuery - з огляду на її популярність таким чином ми забезпечили можливість доопрацювання сайту іншими розробниками [25].

Зміст файлів CSS-папки:

1. Деактивація стилів браузера за замовчуванням (для більш точного управління версткою);
2. Графічний шрифт (гамбургер для адаптивного меню, кнопка “Закрити меню”, соціальні мережі.). Цей файл зменшив вагу веб-сервісу, що має позитивний вплив на швидкість завантаження сайту;
3. Винесені в окремі файли стилі слайдерів для більш зручної структури верстки;

Зміст JS-папки:

1. Скрипт для слайдеру;
2. Script.js – файл спільних налаштувань для скриптів, кастомізації, реалізації адаптивного меню;
3. Webfonts – зберігає файли графічних та текстових шрифтів.

При верстці не використовувались front-end фреймворки, тому що сам проект не являється складним в реалізації з боку дизайну, була можливість використовувати мінімальну кількість стилів, класів та тегів.

Під час реалізації back-end частини була взята одна з найпопулярніших CMS - WordPress.

З огляду на що дана система управління написана на серверній мові програмування PHP з використанням мови запитів до БД SQL був встановлений локальний сервер який інтерпретує серверну мову і віддає в браузер контент сторінок в тегах.

Згідно з документацією локального сервера в папці domains був створений віртуальних домен куди були вивантажені файли wordpress, була створена база даних для зберігання всієї необхідної інформації по сайту і майбутніх заходів.

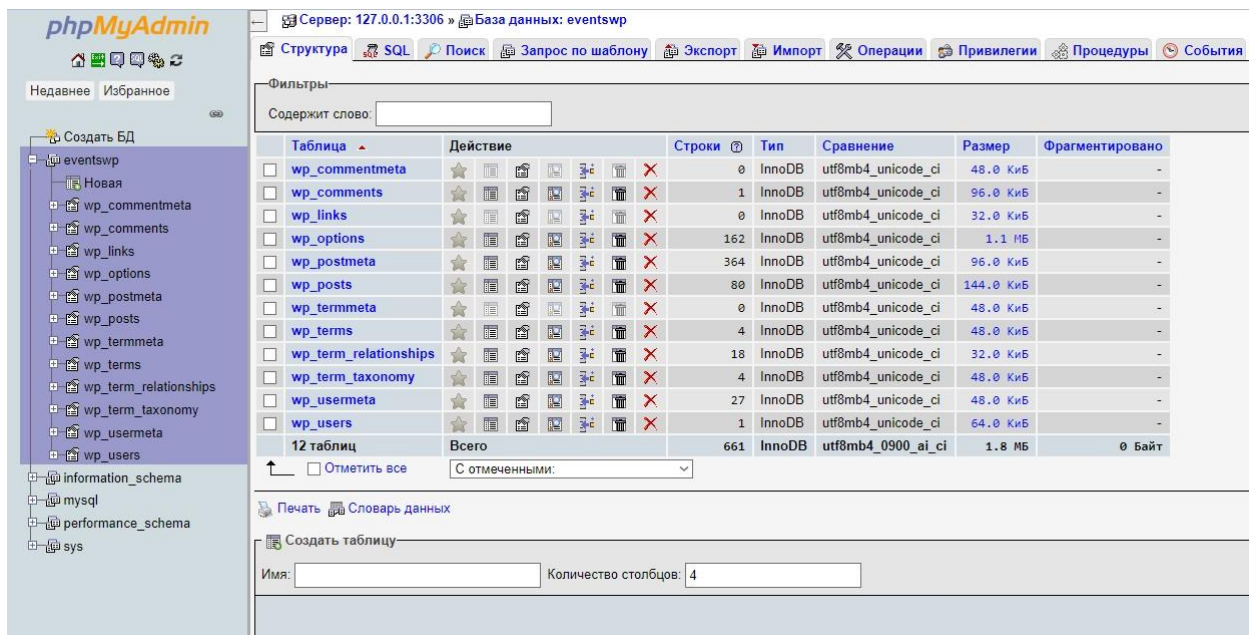


Рисунок 3.7 — База даних

WordPress дозволяє працювати з базою даних власними засобами, без необхідності створювати окреме з'єднання з базою даних. Ваша база даних WordPress розміщується на хості бази даних, який потім поміщає її на сервер MySQL. Думайте про хості бази даних як про реальний будинок, який ви можете знайти знаючи певний адреса (ім'я хоста). Ви можете отримати щось з розділу «кімнати» (ім'я бази даних). Звичайно, вам знадобиться дозвіл на це, тому вам потрібно ввести ім'я користувача і пароль перед входом в будинок.

Сервер: 127.0.0.1:3306 » База данных: eventswp » Таблица: wp_options

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отображение строк 0 - 49 (544 всего, Запрос занял 0,0003 сек.)

```
SELECT * FROM `wp_options`
```

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодного

+ Параметры

	option_id	option_name	option_value	autoload
<input type="checkbox"/>	1	siteurl	http://eventswp.loc	yes
<input type="checkbox"/>	2	home	http://eventswp.loc	yes
<input type="checkbox"/>	3	blogname	Портал событий	yes
<input type="checkbox"/>	4	blogdescription	Ещё один сайт на WordPress	yes
<input type="checkbox"/>	5	users_can_register	0	yes
<input type="checkbox"/>	6	admin_email	admin@admin.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
<input type="checkbox"/>	19	default_comment_status	open	yes
<input type="checkbox"/>	20	default_ping_status	open	yes
<input type="checkbox"/>	21	default_pingback_flag	1	yes
<input type="checkbox"/>	22	posts_per_page	10	yes
<input type="checkbox"/>	23	date_format	d.m.Y	yes
<input type="checkbox"/>	24	time_format	H:i	yes
<input type="checkbox"/>	25	links_updated_date_format	d.m.Y H:i	yes
<input type="checkbox"/>	26	comment_moderation	0	yes
<input type="checkbox"/>	27	moderation_notify	1	yes
<input type="checkbox"/>	28	permalink_structure	/%postname%/	yes
<input type="checkbox"/>	29	rewrite_rules	a:182:{s:27:"events/{d(4)-{d(2)-{d(2)}} <td>yes</td>	yes

Рисунок 3.8 — Приклад бази даних для задання параметрів wordpress

В кореневій папці наступні файли відповідають за:

- Index.html - файл головної сторінки сайту;
- Catalog.html - файл зовнішнього вигляду сторінки категорії заходів;
- Page.html - файл зовнішнього вигляду кожного заходу ;
- Delivery.html - файл зовнішнього вигляду для сторінки бронювання білету.

Тема WordPress складається з трьох основних типів файлів, на додаток до зображень. Перший - це таблиця стилів під ім'ям style.css, яка контролює зовнішній вигляд сторінок сайту. Другий надає собою файл додаткового

функціоналу (`functions.php`). Інші файли - це файли шаблонів, які визначають, яким чином виводиться інформація з бази даних на веб-сторінку. Давайте розглянемо кожен тип окремо.

Крім власне таблиці стилів вашої теми, файл `style.css` обов'язково повинен містити інформацію про вашу тему у вигляді коментарів. Кожна тема повинна мати свою власну унікальну інформацію в коментарях заголовка, в зворотному випадку виникнуть проблеми в діалозі вибору теми. Створюючи нову тему на основі вже існуючої, в першу чергу змініть коментарі заголовка [26].

При написанні коду теми WordPress, я дотримувався наступних стандартів:

1. Використав коректний структурований, який не містить помилок PHP-код і сучасний HTML-код.
2. Використав правильно складений, пройдений стандартизацію CSS-код.
3. Дотримувався рекомендацій з основ дизайну сайту.

В папці `wp-content-> themes` створив назву нашої теми `events`. У цю папку переніс папки з нашої верстки `css`, `images`, `js`, `webfonts` і два файли `index.html`, `style.css`.

Тема може додатково використовувати файл з необхідними функціями для роботи теми, він повинен розташовуватися в каталозі теми і називатися `functions.php`. Цей файл в основному працює подібно плагіну, і якщо він присутній в каталозі теми, яку ви використовуєте, то він автоматично завантажується під час ініціалізації WordPress. Це підходить як для сторінок адміністратора, так і для інших сторінок [27].

3.2 Розробка адміністративної частини веб-сервісу

Веб-сервіс виконує функцію ознайомлювального характеру з культурними заходами та бронюванням квитків.

Було впроваджено просто та інтуїтивно зрозуміле наповнення сторінки події адміністратором, яке не потребує глибоких знань у використанні ПК. На панелі записів є можливість додати новий запис, сортувати записи за категоріями, датами та рубриками. Є можливість кастомізувати сторінку, видаляти, редагувати шукати потрібні події. Є можливість додавати адміністраторів з різними рівнями доступу, для правильного розподілу функціональних можливостей між ними.

Заголовок	Автор	Рубрики	Метки	Дата
Комедійне шоу Improv Show	admin	Гумор	—	Опубліковано 21.05.2021 в 01:07
Зірковий стендап	admin	Гумор	—	Опубліковано 21.05.2021 в 01:04
Стендап на днірі	admin	Гумор	—	Опубліковано 21.05.2021 в 01:02
Стендап "New Wave"	admin	Гумор	—	Опубліковано 21.05.2021 в 01:01
Джаз на даху	admin	Концерти	—	Опубліковано 21.05.2021 в 00:58
Класика в саду	admin	Концерти	—	Опубліковано 21.05.2021 в 00:53
Вечір джазу і вина в саду	admin	Концерти	—	Опубліковано 21.05.2021 в 00:50
Шопен і Рахманінов	admin	Мистецтво	—	Опубліковано 21.05.2021 в 00:44
Fluid art	admin	Мистецтво	—	Опубліковано 21.05.2021 в 00:39
Дві віолончелі. Classic energy	admin	Концерти	—	Опубліковано 20.05.2021 в 18:48
Лекція з вином в саду: Ван Гог і Гоген	admin	Мистецтво	—	Опубліковано 20.05.2021 в 17:28
Kureni city рісніс	admin	Мистецтво	—	Опубліковано 20.05.2021 в 17:26

Рисунок 3.1 — Сторінка записів

На панелі наповнення події було впроваджено можливість змінювати назву та опис, за допомогою плагіну “Advanced custom fields” задавати дату, місце проведення та ціну квитка. Також є можливість змінювати посилання та категорію заходу.

Тобто, крім стандартних полів Тема, Редактор, Мініатюра і так далі, в редакторі Постів, Сторінок і інших типів записів, ви можете додати свої власні кастомні поля. Якщо у вас на сайті десятки або сотні сторінок з однаковим висновком інформації, краще використовувати кастомними поля. Якщо ви хочете опублікувати на сайті, наприклад, якийсь захід, ви можете це зробити це у вигляді простого тексту в статті або на сторінці.

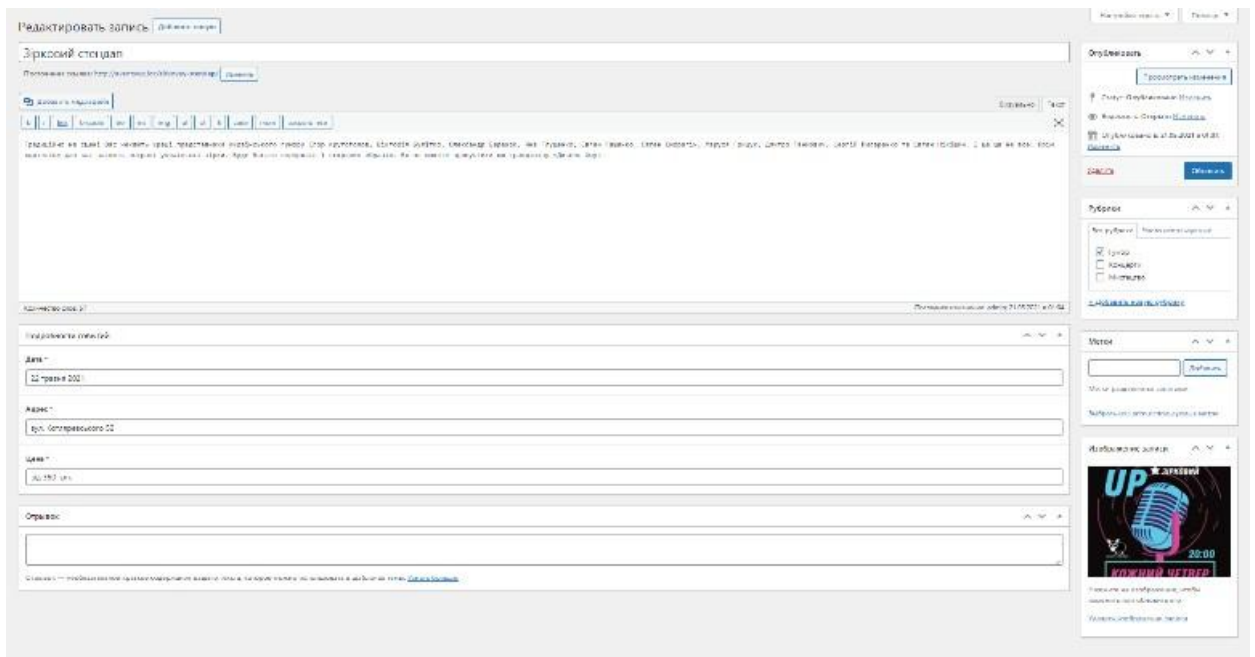


Рисунок 3.2 — Наповнення сторінки заходу

Адміністрування сайту можна виконувати з будь-якого пристрою, з підтримкою всіх розмірів екранів.

За допомогою плагіну “Contact form 7” було впроваджено форму, яка дозволяє збирати дані (ім’я, прізвище, номер телефону, e-mail) про особу, яка здійснює бронювання квитків. Також плагін відправляє на e-mail замовнику підтвердження про бронювання квитку та інструкції, як квиток отримати.

Contact Form 7 може управляти численними контактними формами, де ви можете гнучко налаштовувати вміст форм і пошти з досить простою розміткою. Форми мають вбудовану підтримку Ajax відправки, CAPTCHA, спам фільтра Akismet і не тільки. Один з найпопулярніших плагінів для створення контактних форм на сайті. Плагін дозволяє створювати форми будь-яких видів; дуже гнучкий і зручний в налаштуванні; розвивається вже багато років і містить багато напрацювань. Для верстки можна використовувати html-теги і шорткоди плагіна. Шорткоди додають поля форми, а html теги дозволяють створити довільну HTML структуру.

Шаблон форми Письмо Уведомления при отправке формы Дополнительные настройки

Письмо
 Вы можете редактировать шаблон письма здесь. Детально смотрите [Setting up mail](#).
 В полях ниже вы можете использовать эти почтовые теги:
 [radio-269] [text-361] [text-362] [tel-146] [email-83]

Кому:

От кого:

Тема:

Additional headers:

Тело письма:

Исключить вывод строк с пустыми тегами сообщения
 Использовать HTML-формат письма

Рисунок 3.3 — Налаштування розсилки

Отримання даних користувачів також корисне тим, що їх можна використовувати для подальшого маркетингу.

На основі HTML5/CSS3 та бібліотеки jQuery була створена та активована унікальна тема, під наш веб-сервіс.

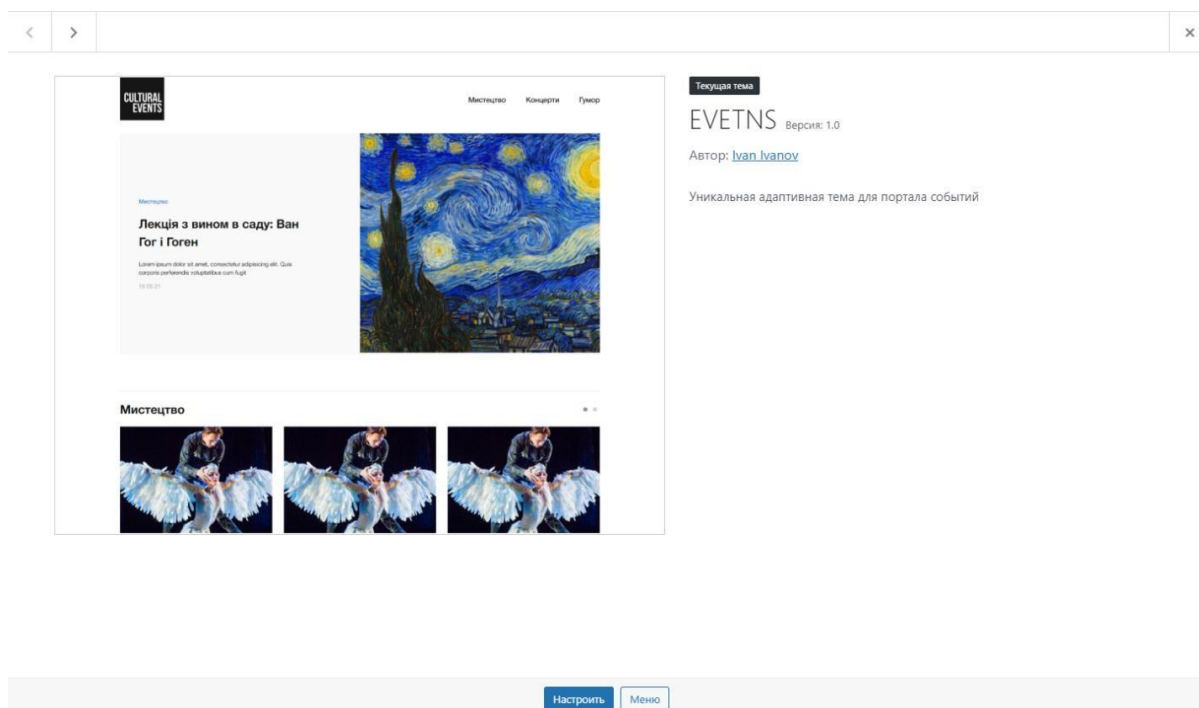


Рисунок 3.4 — Сторінка теми

Створені 3 різні рубрики, за допомогою яких можна поділити заходи на категорії, це дозволяє проводити сортування заходів та отримувати швидкий доступ та інформацію до сторінок рубрик. В залежності від налаштувань можна встановлювати обмеження, або унікальні параметри для заходів, які відносяться до конкретних категорій.

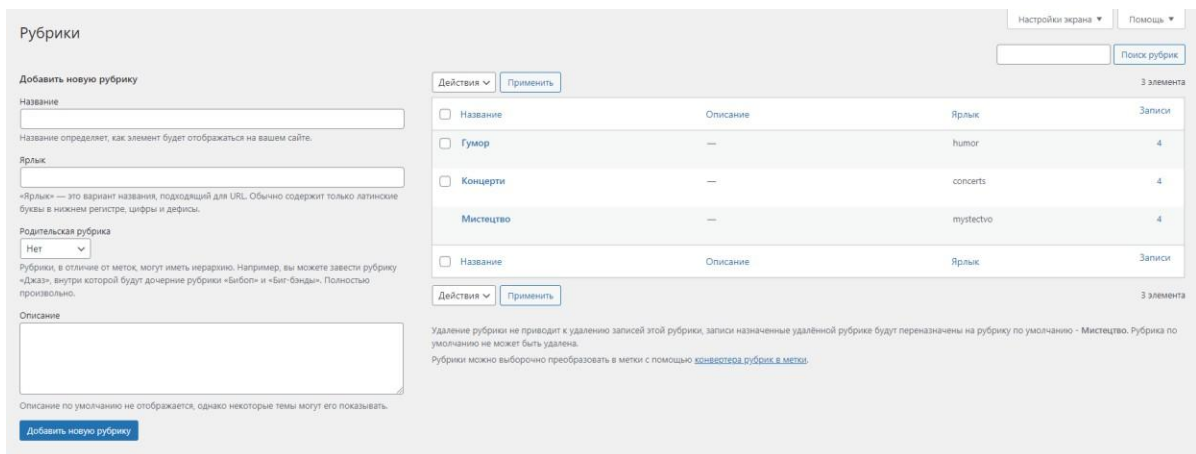


Рисунок 3.5 — Сторінка налаштування категорій

Для роботи з файлами (зображеннями, документами, аудіо- та відеозаписами) використовується меню Медіафайли. Завантажити новий файл можна за допомогою пункту меню Медіафайли> Додати новий.

Як і на інших сторінках консолі, тут в правому верхньому куті є кнопка «Налаштування екрану», що дозволяє вибрати стовпці, які відображаються в таблиці.

Під заголовком Бібліотека файлів відображаються відомості про зайняте і вільне місце. При необхідності можна оплатити додаткове місце для файлів.

Медіафайли wordpress - це певний програмний комплекс і щоб використовувати завантажене зображення, його потрібно зареєструвати в бібліотеці медіафайлів. А щоб зареєструвати, необхідно картинку завантажити через інтерфейс медіафайлів.

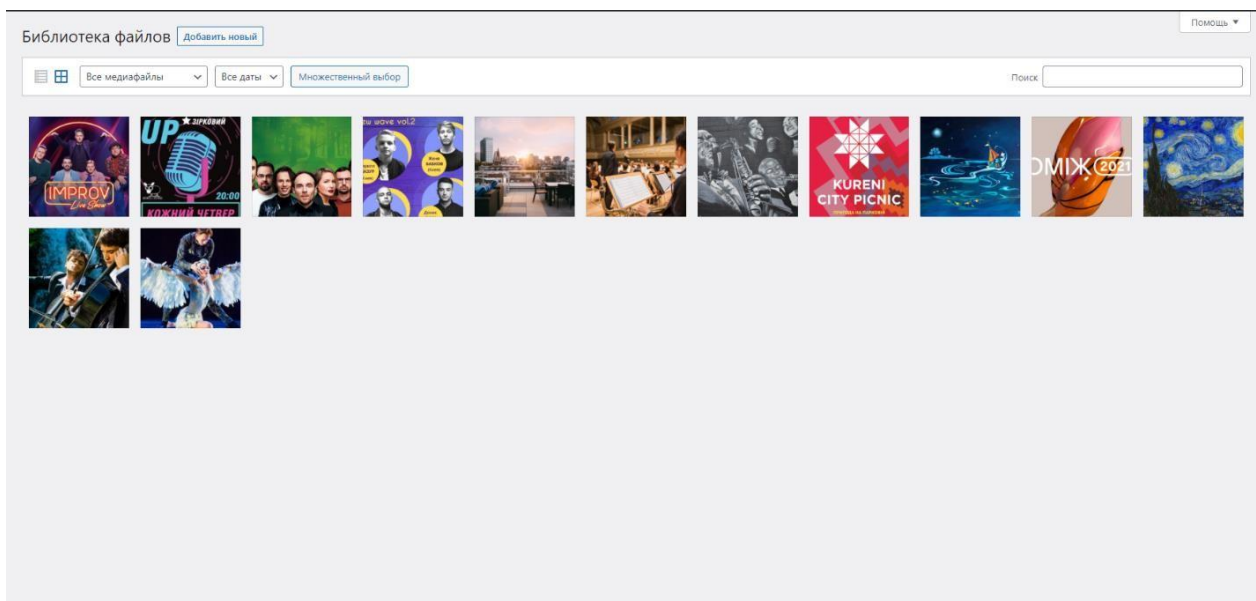


Рисунок 3.5 — Бібліотека медіа-файлів.

Власна медіатека допомагає вам керувати медіа. Організуйте тисячі завантажених файлів у папки, колекції та галереї. Справжній файловий менеджер, який дозволяє управляти великою кількістю файлів, таких як зображення, відео чи документи, у WordPress. Папки медіатеки для всіх [28].

3.3 Огляд фіналізованого функціоналу веб-сервісу

Враховуючи інформацію з технічного завдання було складено mindmap веб-сервісу, з урахуванням зручності та швидкості користування інтерфейсом. На рисунку відображено шлях користувача для здійснення кінцевої дії на сайті.

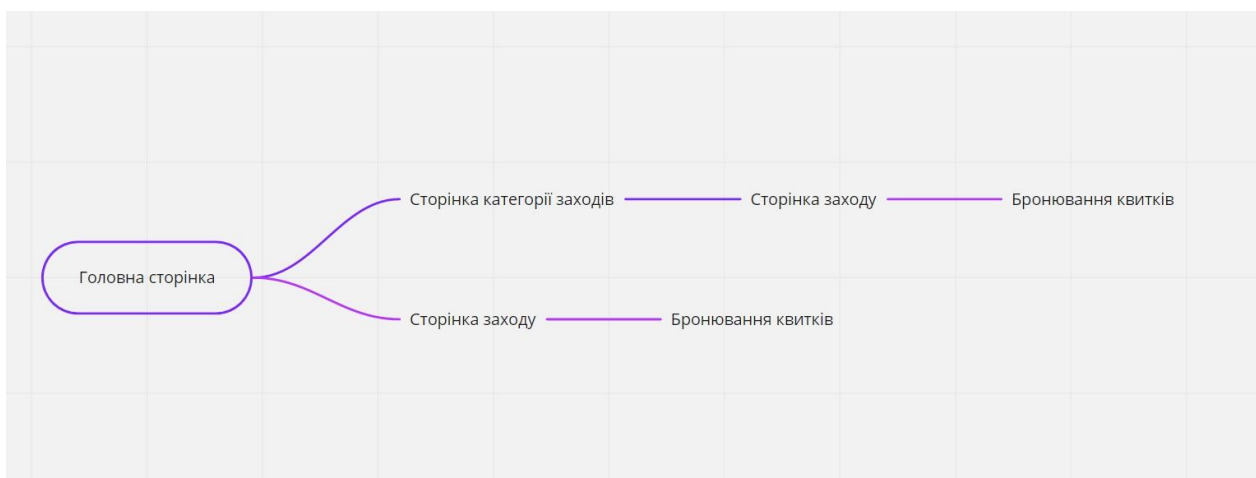


Рисунок 3.9 — Mindmap веб-сервісу.

Для обкладинки головної сторінки я вирішив обрати промо-блок, який включає в себе картку заходу, цей блок може використовуватись у комерційних цілях, для виводу на нього промо-події, або як в нашому випадку подію з найсвіжішою датою додавання на сайт.

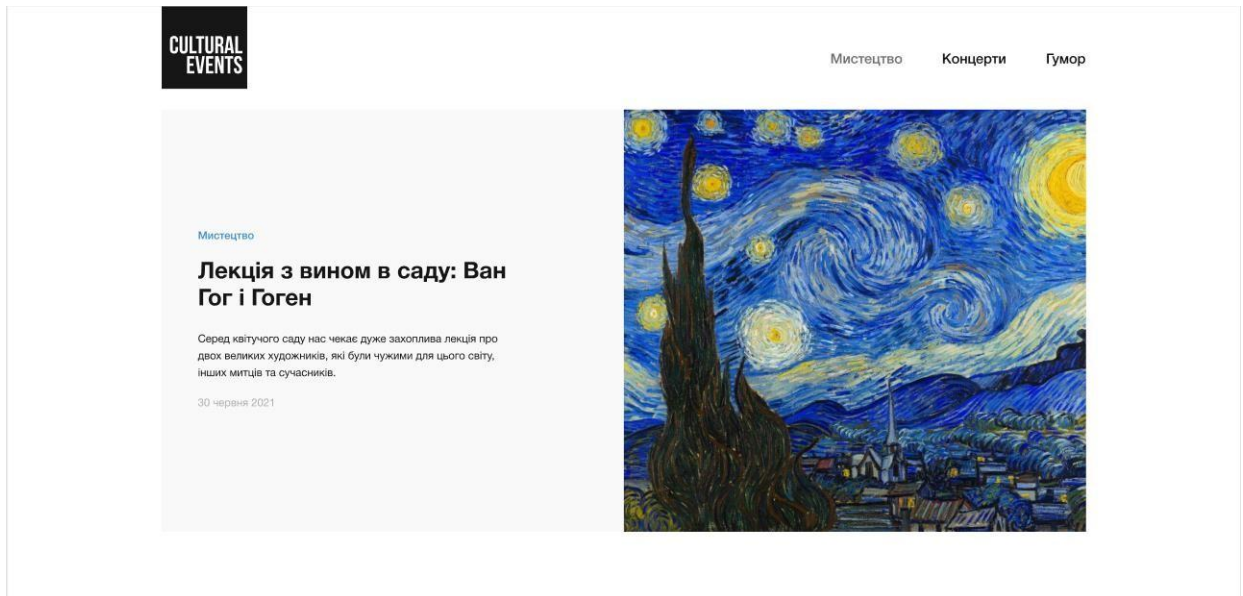


Рисунок 3.10 — Обкладинка сайту на головній сторінці.

Наступним блоком на головній сторінці було розміщено три слайдери з заходами, по різних категоріям. Таке рішення було прийнято на основі аналізу поведінки користувачів, які заходять на сайт культурних заходів. В більшості випадків для користувачів важливий саме контент веб-сайту та можливість швидкого доступу до нього, тому задля того щоб не відволікати користувача від цільової дії було вирішено одразу надати можливість обрати подію з якоїсь з категорії.

Також планується зробити сортування за різними параметрами, наприклад: дата події, місто проведення, ціна. Це спростить життя користувачам веб-сервісу та зробить його набагато більш зручним.

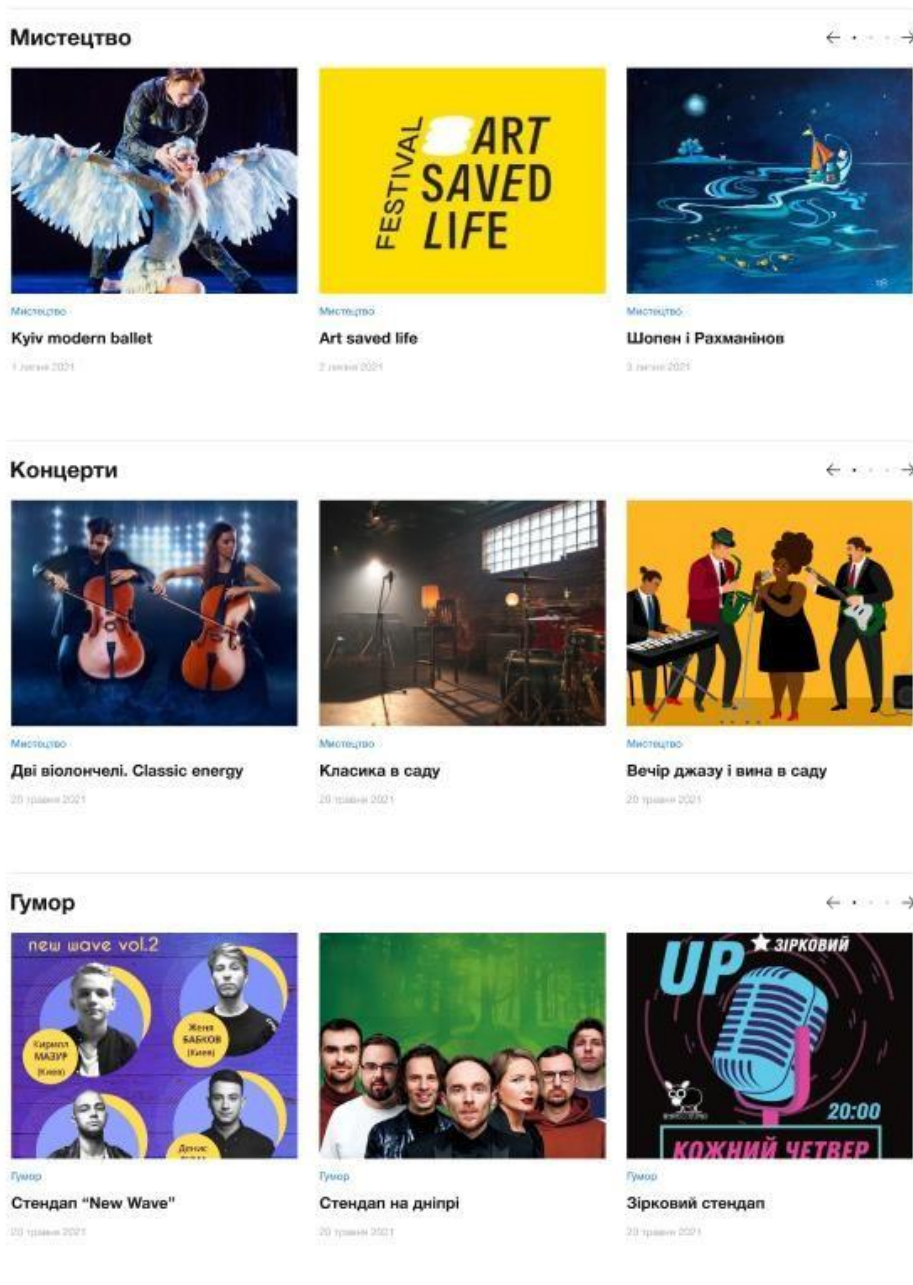


Рисунок 3.11 — Слайдери

В підвалі (footer) веб-сайту було вирішено продублювати зміст шапки (логотип, навігація по категоріям) та додати актуальні e-mail, номер телефону та посилання на соціальні мережі.

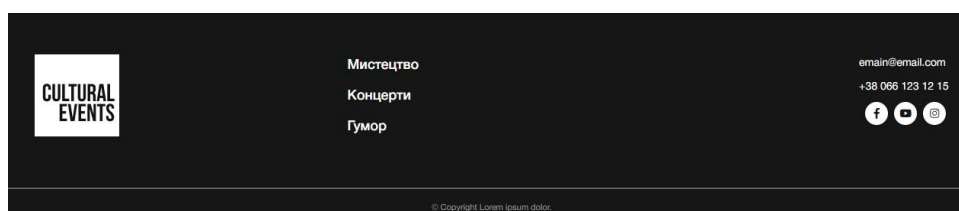


Рисунок 3.11 — Footer.

Сторінка категорії включає в себе всі заходи з відповідної категорії, також структурою сторінки передбачене подальше інтегрування рекламних матеріалів (банери, тощо). По аналогії з головною сторінкою ця сторінка також спроектована, орієнтуючись на потреби користувача отримати якомога швидший доступ до контенту сайту.

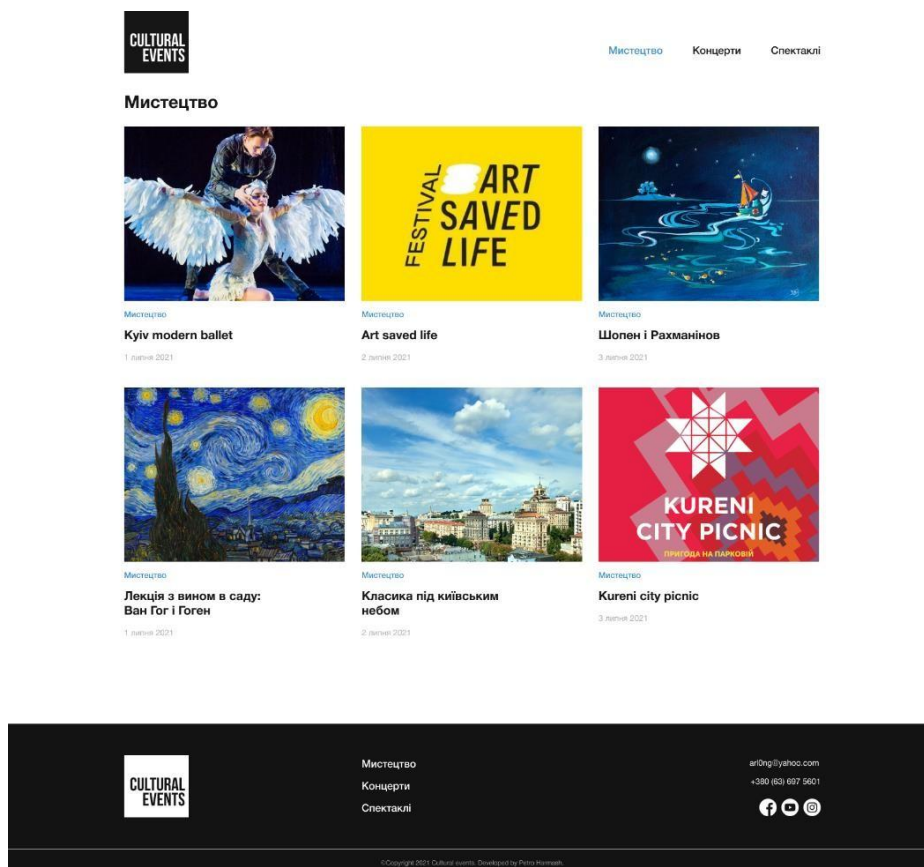


Рисунок 3.12 — Сторінка категорії

Сторінка події складається з фото самої події, детальної інформації про подію (дата, місце проведення, ціна), детальним описом події, слайдеру з подіями тієї ж категорії та кнопкою “Забронювати квиток”, однією з конверсійних дій на сайті.

Вона включає в себе всі необхідні функції для перегляду культурних заходів.



Мистецтво Концерти Спектаклі

Лекція з вином в саду: Ван Гог і Гоген

Мистецтво



📅 20 травня 2021, 19:00
📍 Ботанічний сад ім. Гришко
Київ, вул. Тимірязєвська, 1
💰 Від 375 грн.

Купити білет

Про подію

Лекція з вином в саду: Ван Гог і Гоген в Києві! Чекаємо вас 2021-05-20 о 19:00 на майданчику Ботанічний сад ім. Гришко, Київ.

Ви можете купити квитки на Лекція з вином в саду: Ван Гог і Гоген онлайн.

Серед квітучого саду нас чекає дуже захоплива лекція про двох великих художників, які були чужими для цього світу, інших митців та сучасників.

Про їхнє життя, творчості, картинах, їх дружбу, фактах і міфах, постімпресіонізму і мистецтві.

У вартість квитка входить келих вина.

Мистецтво



Мистецтво

Київ modern ballet

1 липня 2021



Мистецтво

Art saved life

2 липня 2021



Мистецтво

Шопен і Рахманінов

3 липня 2021



Мистецтво

Концерти

Спектаклі

art0ng@yahoo.com

+380 (63) 697 5601



©Copyright 2021 Cultural events. Developed by Petro Harmash.

Рисунок 3.13 — Сторінка події

Після натискання кнопки “Забронювати білет” користувач потрапляє на сторінку з бронюванням білету, де йому необхідно ознайомитись зі способами доставки та заповнити форму (Ім’я, Прізвище, e-mail, номер телефону) для бронювання білету.

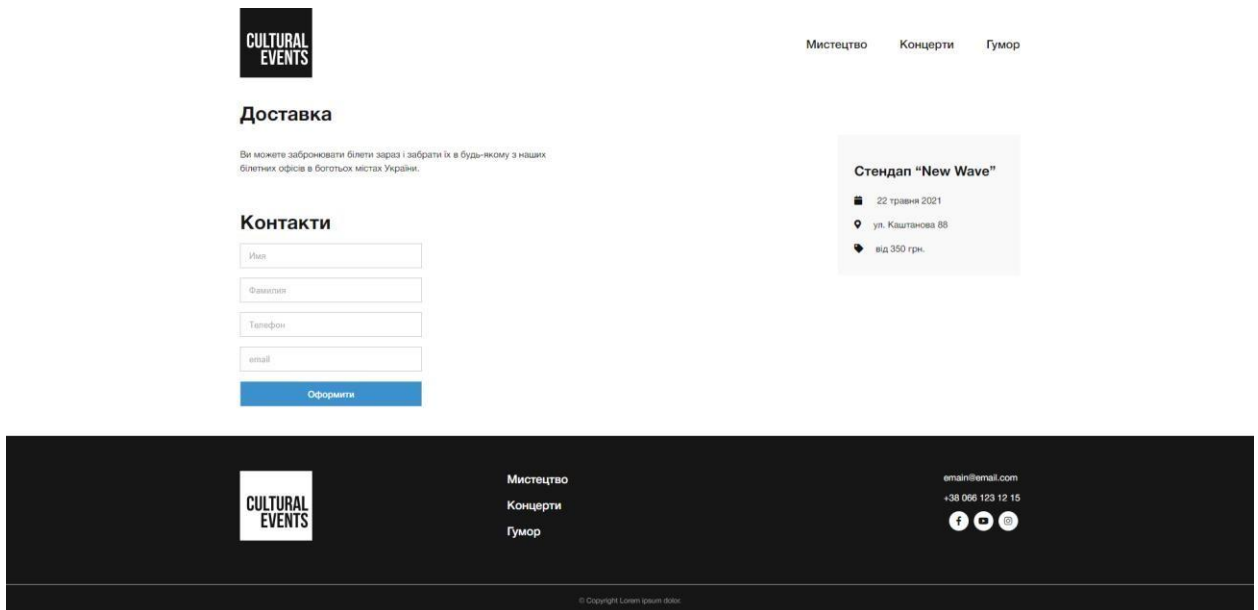


Рисунок 3.14 — Сторінка бронювання білету

Після успішної відправки форми на e-mail адміністратора будуть відправлені дані з полів, заповнюваних користувачем. Після чого на e-mail особи, яка забронювала білет буде відправлено посилання з підтвердженням бронювання та подальшими інструкціями.

Висновки до розділу 3

У цьому розділі було проведено аналіз безпосередньої розробки продукту, розглянуто панель управління адміністратора, яка дозволяє виконувати пошук по сайту, створювати та редагувати контент в існуючих рубриках, можливості та застосування БД, надано опис компонентів веб-сервісу та продемонстровано роботу програмного продукту. Була здійснена практична реалізація веб-сервісу для перегляду культурних заходів. Робота була виконана за допомогою Wordpress та суміжних технологій HTML, CSS та JavaScript.

Сервіс отримав функціонал у вигляді перегляду культурних заходів, бронювання квитків, створені можливості для подальшого масштабування кількості статей.

ВИСНОВКИ

У результаті виконаної дипломної роботи було досліджено актуальність веб-сервісів, розглянуто програмні засоби для створення веб-сервісу для перегляду культурних заходів. Проведено порівняльний аналіз актуальних технологій.

Сервіс для перегляду культурних заходів був створений за допомогою брифу та функціонального прототипу, проведено дослідження рівню інтернет-грамотності та потреб користувачів, на основі цих даних було створено дизайн-концепцію, після цього в графічному редакторі figma були спроектовані прототипи, інтегрований текст та розроблений фінальний дизайн для веб-сервісу. Веб сервіс був адаптований під всі розміри екранів.

Були спроектовані та адаптовані наступні сторінки:

- Головна сторінка;
- Сторінка заходу;
- Сторінка категорії;
- Сторінка з формою для бронювання квитків;

Було проведено аналіз безпосередньої розробки продукту, розглянуто панель управління адміністратора, яка дозволяє виконувати пошук по сайту, створювати та редагувати контент в існуючих рубриках, можливості та застосування БД, надано опис компонентів веб-сервісу та продемонстровано роботу програмного продукту. Була здійснена практична реалізація веб-сервісу для перегляду культурних заходів. Робота була виконана за допомогою Wordpress та суміжних технологій HTML, CSS та JavaScript.

Сервіс отримав функціонал у вигляді перегляду культурних заходів, бронювання квитків, створені можливості для подальшого масштабування кількості статей.

Розроблений проект працює на будь-яких пристроях та операційних системах, має можливість масштабованості, що передбачає можливість впровадження нових функцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-служба [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0> (дата звернення: 08.03.2021);
2. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language [Електронний ресурс]. – Режим доступу : <https://www.w3.org/TR/wsd120/>;
3. Веб-сервисы – Краткое руководство [Електронний ресурс]. – Режим доступу : <https://coderlessons.com/tutorials/veb-razrabotka/izuchite-veb-servisy/veb-servisy-kratkoe-rukovodstvo>;
4. Моррісон М. – Вивчаємо JavaScript / Моррісон М// Вивчаємо JavaScript 2013. – С. 1 –55.
5. Чарльза Уайк–Сміт – Стильний сайт за допомогою CSS / Чарльза Уайк–Сміт // Стильний сайт за допомогою CSS 2007. – С. 12 – 51.
6. Специфікація HTML та CSS [Електронний ресурс] / HTMLBOOK 2019 – Режим доступу : <http://htmlbook.ru/css>.
7. HTML5 Developer guides. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>.
8. jQuery. URL: <https://jquery.com/>.
9. WordPress создание тем https://codex.wordpress.org/Theme_Development
10. WordPress Editor <https://wordpress.org/support/article/wordpress-editor/>
11. Tilda Education. Figma <https://tilda.education/articles-figma>
12. Uprock. Приоритизация функций <https://www.uprock.ru/articles/prioritizaciya-funkciy>
13. Uprock. Читают ли пользователи весь текст на странице? <https://www.uprock.ru/articles/chitayut-li-polzovateli-ves-tekst-na-stranice>

- 14.Uprock. Коммуникация в дизайне как важнейший навык
<https://www.uprock.ru/articles/kommunikaciya-v-dizayne-kak-vazhneyshiy-navyk>
- 15.Uprock. 3 стратегии для улучшения взаимодействия с клиентом.
<https://www.uprock.ru/articles/uluchshit-rekonstruirovat-perestroit-zanovo-3-strategii-dlya-uluchsheniya-vzaimodeystviya-s-klientom>
- 16.Miro documentation. <https://help.miro.com/hc/en-us/articles/360017730533-What-Is-Miro->
- 17.HTML Academy. Когда и зачем использовать jQuery
https://htmlacademy.ru/blog/boost/frontend/why-jquery?utm_campaign=fb_blog1003_03092020&utm_medium=social&utm_source=fb
- 18.Специфікація HTML та CSS [Електронний ресурс] // HTMLBOOK 2020 – Режим доступу: <http://htmlbook.ru/css>.
- 19.Грабер М., Введение в SQL: книга / пер. з англ. Ястребов В.Ф. Москва, 2016, 396 с.
- 20.Роббінс Д. HTML5, CSS3 і Java. Script. Вичерпне керівництво / Роббінс Дженіфер.; пер. з англ. Н.А. Райтмана. - М.: Ескімо, 2016 - 323 - 391с.;
- 21.Ізварін І.В., Гайдаржи В.С. Базы даних в інформаційних системах: книга. Університет "Україна", 2018, 37-38с.
- 22.Меліник Р.В. Програмування веб-застосунків (фронт-енд та бек-енд): книга. Львівська політехніка, 2018, 53-57.
- 23.Томсон Л., Веллинг Л. Разработка web-приложений на PHP и MySQL. Москва, 2019, 215-217 с.
- 24.Девід Фленаган. JavaScript. Карманный справочник, 3-е издание: книга. Диалектика, 254-257 с.
- 25.Ferraiolo DF, Cugini JA, Kuhn DR Рольовий контроль доступу: особливості та мотивація. 11-а щорічна конференція програм комп'ютерної безпеки; 2015; Новий Орлеан, Лос-Анджелес, США. С. 241–248 с.

- 26.HTML5 Developer guides. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>.
- 27.CSS: Cascading Style Sheets Developer guides. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
- 28.Choose Between Traditional Web Apps and Single Page Apps (SPAs). Microsoft. URL: <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps>.

ДОДАТКИ

Додаток А — Приклад виведення довільних рядків

```

</div><!--wrap-single-text-->
<div class="wrap-single-order">
  <p><i class="fas fa-calendar"></i><?php the_field('дата'); ?></p>
  <p><i class="fas fa-map-marker-alt"></i><?php the_field('адрес'); ?></p>
  <p><i class="fas fa-tag"></i><?php the_field('цена'); ?></p>
  <a class="order" id="clickorder" href="#">Забронювати</a>
</div><!--wrap-single-order-->
</div><!--wrap-content-single-article-->
</article><!--single-article-->
<section class="art-main-page hidden-order">

```

Додаток Б — Динамічне виведення меню із WordPress

```

<?php wp_nav_menu([
  'theme_location' => 'all_menu',
  'menu_class'     => 'menu-header'
]); ?>

</nav>
</header>

```

Додаток В — Приклад виведення записів з категорії

```

1  <?php get_header();?>
2  <div class="wrap-category">
3      <h1><?php single_cat_title();?></h1>
4  <div class="wrap-catalog">
5
6  <?php if ( have_posts() ) : ?>
7  <?php while ( have_posts() ) : the_post(); ?>
8  <article>
9  <a href="<?php the_permalink();?>"
10     <?php the_post_thumbnail('full');?>
11     </a>
12     <?php
13     $category = get_the_category();
14     $category_id = $category->cat_ID;
15     ?>
16
17     <a href="<?php echo get_category_link( $category_id ); ?>"><?php
18     single_cat_title();?></a>
19     <a href="<?php the_permalink();?>"><h3><?php the_title();?></h3></a>
20     <span class="datearticle"><?php the_field('дата'); ?></span>
21     </article>
22 <?php endwhile; ?>
23 <?php else: ?>
24 <!-- no posts found -->
25 <?php endif; ?>
26
27     </div><!--wrap-catalog-->
28 </div><!--wrap-category-->
29

```

Додаток Г — Приклад коду для підключення стилів та скриптів

```

3
4 ▽ function load_style_script(){
5
6     wp_enqueue_style('reset_css', get_template_directory_uri() .'/css/reset.css');
7     wp_enqueue_style('font_css', get_template_directory_uri().'/css/all.min.css');
8     wp_enqueue_style('carousel_css', get_template_directory_uri().'/css/owl.carousel.min.css');
9     wp_enqueue_style('carousel_def_css', get_template_directory_uri(). '/css/owl.theme.default.css');
10    wp_enqueue_style('style_css', get_template_directory_uri().'/style.css');
11    wp_enqueue_style('media_css', get_template_directory_uri() .'/css/media.css');
12
13    wp_enqueue_script('lib_jquery', 'https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js',
14    array(), false, true);
15    wp_enqueue_script('carousel_js', get_template_directory_uri().'/js/owl.carousel.min.js', array(),
16    false, true);
17    wp_enqueue_script('script_js', get_template_directory_uri().'/js/script.js', array(), false, true);
18
19 }
20 add_action('wp_enqueue_scripts', 'load_style_script');
21
22 add_theme_support('post-thumbnails');
23
24 ▽ add_action( 'after_setup_theme', function(){
25 ▽     register_nav_menus( [
26         'all_menu' => 'Меню в шапке и в подвале'
27     ] );
28 } );
29
30
31

```

Додаток Г — Стандартний цикл WordPress для виведення контенту

```

<article class="single-article show-order" >
    <h1>Доставка</h1>
    <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>

        <?php the_content();?>

    <?php endwhile; else : ?>

    <?php endif; ?>

```

Додаток Д — Структура теми

