

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ
Веб-сервіс культурних заходів міста**

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Прикладне програмування»**

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи ПП-42

Бабійчук С. І.

(прізвище та ініціали)

Керівник Білий Р. О.

(прізвище та ініціали)

(науковий ступінь, звання)

Унікальність тексту – 87.3%

Випускна кваліфікаційна робота бакалавра допущена до захисту

Рішенням кафедри *прикладних інформаційних систем*

Протокол № 14 від 23.05.2023 р.

зав. кафедри  Плескач В. Л.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
 Факультет інформаційних технологій
 Кафедра прикладних інформаційних систем

НАЗВА ТЕМИ: «Веб-сервіс культурних заходів міста»

Освітня програма: Прикладне програмування
 Спеціальність: Комп'ютерні науки

ПІБ	Підпис
Бабійчук Сергій Ігорович	

ТЕМА РОБОТИ

Веб-сервіс культурних заходів міста

Web service of cultural events of the city

МЕТА БАКАЛАВРСЬКОЇ РОБОТИ, ЗАВДАННЯ

Мета бакалаврської роботи: створення ефектного та зручного сайту для перегляду культурних заходів міста, що надає корисну інформацію про культурні події, такі як концерти, вистави, виставки, фестивалі та інші заходи

План роботи:

1. Сучасні технології розробки веб-сайтів
2. Аналіз та вибір технологій розроблення застосунку
3. Проектування та програмна реалізація веб-сервісу культурних заходів міста

ПІБ, ступінь, звання наукового керівника роботи: Білий Роман Олександрович, асистент

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Номер	Назва етапів кваліфікаційної роботи бакалавра	Термін виконання етапів кваліфікаційної роботи бакалавра	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	14.10.2022	виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	24.10.2022	заява
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	31.10.2022	виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	01.11.2022	виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	08.11.2022	виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	21.12.2022	виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	31.01.2023	виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	30.03.2023	виконано
9.	Подання роботи у першому варіанті	28.04.2023	виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	03.05.2023	виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	22.05.2023	виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедрі	26.05.2023	виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботи)	12.06.2023	виконано
14.	Захист кваліфікаційної роботи бакалавра	27.06.2023	виконано

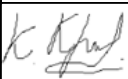
Здобувач вищої освіти _____

(підпис)

Керівник _____ (підпис)

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1
Календарний план дипломної роботи	1
Відомість дипломної роботи	1
Анотація	1
Анотація (іноземною мовою-англійською)	1
Зміст	1
Перелік скорочень, умовних позначень, термінів	1
Вступ	2
1	18
2	14
3	12
Висновки	1
Додатки	10
Перелік використаних джерел	4

				ДП ХХХХ 00.000.00		
	ПБ	Підп.	Дата			
Розробн.	Бабійчук С. І.			Відомість дипломної роботи	Лист	Листів
Керівн.	Білий Р. О.					
Н/контр.	Кравченко К.В.					
Зав.каф.	Плескач В. Л.					

АНОТАЦІЯ(РЕФЕРАТ)

Дипломна робота: 66 с., 15 рис., 3 табл., 47 джерел, 16 дод.

Ця дипломна робота присвячена проектуванню та реалізації веб-застосунку культурних заходів міста.

Метою дипломної роботи є створення ефективного та зручного сайту для перегляду культурних заходів міста, що надає корисну інформацію про культурні події, такі як концерти, вистави, виставки, фестивалі та інші заходи. Люди можуть знайти всю необхідну інформацію в одному місці та вибрати події, які їх цікавлять.

Для досягнення поставленої мети потрібно вирішити такі **завдання**:

- забезпечити відвідувачів актуальною інформацією про культурні заходи та місця відпочинку. Сайт повинен містити описи, дати проведення, адреси та іншу необхідну інформацію, щоб відвідувачі могли зручно спланувати свій відпочинок;
- створити привабливий та естетичний дизайн, який привертає увагу та стимулює інтерес до культурних заходів та місць відпочинку;
- забезпечити простоту навігації та користувацької зручності на сайті. Відвідувачі повинні легко знайти необхідну інформацію та взаємодіяти з функціями сайту.

Об'єкт дослідження Процеси поширення інформації щодо культурних заходів.

Предмет дослідження Програмно-технічні принципи та підходи щодо побудови сайту культурних заходів міста.

Методи дослідження:

Теорія розробки веб-застосунків, дослідження конкурентів, їх подання інформації та функціонал. Порівняння з сайтами зі схожою тематикою, тестування веб-застосунку, вдосконалення функціоналу та користувацького інтерфейсу.

Ключові слова: сайт, фронт-розробка, культурні заходи, місця для

ВІДПОЧИНКУ.

ABSTRACT

Thesis: 66 pp., 15 figures, 3 tables, 47 sources, 16 appendices

This thesis is devoted to the design and implementation of the city's cultural events web application.

The **aim** of the thesis is to create an effective and convenient site for viewing the cultural events of the city, which provides useful information about cultural events, such as concerts, plays, exhibitions, festivals and other events. People can find all the information they need in one place and select events that interest them.

To achieve this aim, I have the following **tasks**:

- To provide visitors with up-to-date information about cultural events and places of rest. The site should contain descriptions, dates, addresses and other necessary information so that visitors can conveniently plan their vacation;
- To create an attractive and aesthetic design that attracts attention and stimulates interest in cultural events and places of rest;
- To ensure ease of navigation and user-friendliness on the site. Visitors should easily find the information they need and interact with site features.

Object of study Processes of dissemination of information on cultural events

Subject of study Software and technical principles and approaches to building a site for cultural events of the city.

Research methods:

The theory of developing web applications, already existing software solutions for the set goals for the site, competitor research, their presentation of information and functionality. Comparison with sites with a similar theme, testing the web application, improving the functionality and user interface.

Keywords: site, front-end development, cultural events, places for recreation.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GIMP – GNU Image Manipulation Program, редактор зображень.

SEO – Search Engine Optimization, оптимізація пошукових систем.

HTML – HyperText Markup Language, мова розмітки гіпертексту.

CSS – Cascading Style Sheets, кускові таблиці стилів.

WCAG Web Content Accessibility Guidelines, Рекомендації щодо доступності веб-контенту.

URL – Uniform Resource Locator, узагальнений локатор ресурсу.

MVC – Model-View-Controller, Модель-Вид-Контролер.

ORM – Object-Relational Mapping, технологія, яка дозволяє взаємодіяти з базою даних, використовуючи об'єктно-орієнтований підхід.

API – Application Programming Interface, набір визначень і протоколів, що визначає, як різні програми можуть взаємодіяти одна з одною.

БД – база даних.

DOM – Document Object Model, структура документа веб-сторінки, яка представляється у вигляді дерева об'єктів.

ЗМІСТ

ВСТУП

РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ

- 1.1 Веб-дизайн як інструмент в реалізації сайтів.**
- 1.2 Принципи веб-дизайну та розробки веб-сайтів**
- 1.3 Роль технологій та веб-сервісів у розвитку культурної сфери**
- 1.4 Цілі та вимоги до функціональності веб-сайту**
- 1.5 Огляд можливих програмних рішень**

РОЗДІЛ 2. АНАЛІЗ РІШЕНЬ І СТЕК ТЕХНОЛОГІЙ ДЛЯ РОЗРОБЛЕННЯ ВЕБ-САЙТУ.

- 2.1 Основні програмні рішення розробки сайту**
- 2.2 Вибір технологій для розробки веб-сайту**

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ КУЛЬТУРНИХ ЗАХОДІВ МІСТА

- 3.1 Архітектура веб-сайту та його компоненти**
- 3.2 Веб-дизайн та використання користувацького інтерфейсу**
- 3.3 Розробка базового функціоналу веб-сайту**
- 3.4 Відображення детальної інформації**
- 3.5 Компонент «інтерактивна галерея»**
- 3.6 Пошук та сортування**
- 3.7 Дизайн та користувацький інтерфейс**

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

ВСТУП

Сьогодні стає відправною точкою для розквіту культури та туризму Прикарпаття. Івано-Франківськ, малий Львів, багатий на культурні події, виставки, концерти, фестивалі та інші заходи, що пропонують розваги та розвиток для мешканців і гостей міста. На вирішення цієї проблеми приходять веб-сайти. Вони – невід'ємна частина нашого життя. У сучасному світі, де доступ до інформації швидко зростає, веб-сайти стають невід'ємною частиною нашого повсякденного життя.

Цей дипломний проект присвячений розробці та оптимізації веб-сайту культурних заходів міста. Я надаю перевагу реалізації сучасного, стильного та візуально привабливого дизайну сайту. Я хочу, щоб користувач проводив якомога більше часу на сайті, захоплений його дизайном та інтерактивними елементами.

З іншого боку, я хочу надати користувачеві зручний інтерфейс – легкий у використанні та швидкий у роботі. Буду розробляти такий інтерфейс, який буде водночас інтуїтивним та мінімалістичним з фокусом на основну тему веб-сайту.

У цій дипломній роботі будуть розглянуті основні аспекти розробки веб-сайту культурних заходів міста, включаючи аналіз вимог, проектування користувацького інтерфейсу, розробку функціональності та інші важливі аспекти.

АКТУАЛЬНІСТЬ ТЕМИ

Наразі Івано-Франківськ є одним з найбільш привабливих туристичних напрямків на заході України. Місто має багатий культурний спадок, численні пам'ятки архітектури, музеї, парки та інші місця відпочинку. Зростання туристичного потоку створює потребу в зручному та інформативному веб-сайті.

Але часто так трапляється, що інформація про культурні заходи та місця відпочинку Івано-Франківську розсіяна по різних джерелах. Створення централізованого веб-сайту, що об'єднує всю актуальну інформацію, спростить пошук та зробить його більш зручним для користувачів.

Розвиток культурних заходів та місць відпочинку сприяє підтримці культурного життя міста. Веб-сайт, що присвячений цим аспектам, допоможе привернути увагу громадськості, популяризувати культурні події та сприятиме розвитку місцевих культурних ініціатив.

Отже, створення веб-сайту культурних заходів Івано-Франківську та місць відпочинку є актуальним завданням, що відповідає потребам користувачів, сприяє розвитку туризму та популяризації культурного життя міста.

МЕТА

Метою дипломної роботи є створення ефектного та зручного сайту для перегляду культурних заходів міста, що надає корисну інформацію про культурні події, такі як концерти, вистави, виставки, фестивалі та інші заходи. Люди можуть знайти всю необхідну інформацію в одному місці та вибрати події, які їх цікавлять.

ЗАВДАННЯ

Для досягнення поставленої мети потрібно вирішити такі **завдання**:

- забезпечити відвідувачів актуальною інформацією про культурні заходи та місця відпочинку. Сайт повинен містити описи, дати проведення, адреси та іншу необхідну інформацію, щоб відвідувачі могли зручно спланувати свій відпочинок;
- створити привабливий та естетичний дизайн, який привертає увагу та стимулює інтерес до культурних заходів та місць відпочинку;
- забезпечити простоту навігації та користувацької зручності на сайті. Відвідувачі повинні легко знайти необхідну інформацію та взаємодіяти з функціями сайту.

РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1 Веб-дизайн як інструмент в реалізації сайтів.

Веб-дизайн – галузь веб-розробки, що охоплює цілий ряд напрямів і дисциплін із створення та супроводу сайтів або веб-застосунків, таких як графічний веб-дизайн, проектування інтерфейсів, авторинг, використовність та оптимізація для пошукових систем. Термін веб-дизайн зазвичай використовують для опису проектування й реалізації клієнтської частини веб-сайту, включаючи верстку.

У розрізі веб-дизайну, термін «інструмент» використовується для позначення програмних засобів та ресурсів, які веб-дизайнери використовують для створення та редагування веб-сайтів. Ці інструменти допомагають веб-дизайнерам перетворити свої творчі ідеї у візуально привабливий та функціональний веб-інтерфейс.

У процесі розробки веб-сайту, веб-дизайнери використовують різноманітні інструменти, які постійно оновлюються відповідно до нових стандартів і програмного забезпечення. Незважаючи на це, основні принципи їх застосування залишаються незмінними.

У графічному веб-дизайні, для створення прототипів зображень або дизайну у форматі для вебу, використовують пакети векторної та растрової графіки.

Пакети графіки у веб-дизайні використовуються для створення графічних елементів, зображень та дизайну веб-сайту. Їх можна поділити на дві основні категорії: пакети векторної графіки та пакети растрової графіки.

Інструменти векторної графіки дозволяють створювати векторні зображення, що складаються з математичних форм та кривих, замість пікселів. Вони зберігають інформацію про форму, розмір, колір та інші атрибути графічних об'єктів.

Найпоширенішими пакетами векторної графіки є Adobe Illustrator, CorelDRAW і Inkscape. Вони надають широкі можливості для малювання, маніпуляції формами, редагування кольорів та створення прототипів дизайну.

Інструменти растрової графіки призначені для роботи з растровими зображеннями, які складаються з пікселів і мають визначену розмірність. Вони часто використовуються для обробки фотографій, створення комплексних графічних ефектів та ретуші зображень. Найвідомішими пакетами растрової графіки є Adobe Photoshop, GIMP і Corel PaintShop Pro. Вони надають різноманітні інструменти для редагування, обрізання, коригування кольору та додавання спеціальних ефектів до растрових зображень.

Для перевірки позиціонування сайту у пошукових системах та його оптимізації використовуються інструменти оптимізації для пошукових систем.

Інструменти оптимізації для пошукових систем (SEO-інструменти) є набором програмних засобів, які допомагають веб-розробникам та маркетологам аналізувати, вдосконалювати та оптимізувати веб-сайт для покращення його видимості в пошукових системах, таких як Google, Bing або Yahoo. Ці інструменти дозволяють проводити дослідження ключових слів, аналізувати структуру сайту, відстежувати позиції у пошукових результатах, перевіряти технічні аспекти оптимізації та багато іншого.

Основні функції та можливості інструментів оптимізації:

1. аналіз ключових слів;
2. аналіз конкурентів;
3. аналіз технічних аспектів;
4. відстеження позицій;
5. аналіз зв'язків та зворотного посилання;
6. звітність та моніторинг.

До інструментів веб-дизайну також належать валідатори розмітки та інші засоби для тестування використовності й доступності, які допомагають переконатися, що сайт відповідає веб-стандартам.

Валідатори розмітки та інші засоби для тестування використовності й доступності є важливими інструментами веб-дизайну, які допомагають перевірити та забезпечити відповідність сайту веб-стандартам і забезпечити оптимальну взаємодію з користувачами.

Ці інструменти перевіряють валідність коду HTML, CSS і інших мов розмітки. Вони перевіряють, чи дотримується код встановлених веб-стандартів, чи відповідає правилам синтаксису і чи використовуються допустимі елементи та атрибути. Використання валідаторів допомагає уникнути помилок і забезпечити коректну роботу сайту на різних браузерах та платформах.

Серед популярних онлайн редакторів лідером є онлайн редактор FIGMA. Він має багато переваг, включаючи крос-платформеність, безкоштовний тарифний план для одного користувача, а також можливість спільної роботи в режимі реального часу.

Оскільки дизайн містить елементи мистецтва та інженерії, на нього розповсюджуються їхні правила, що можуть забезпечити створення якісних макетів для Інтернет-видань. Одне з основних правил вказує, що дизайн має ціль і залежно від задач, обираються засоби. Таким чином дизайном є не оформлення веб-сторінок, а конструкторська діяльність, що покликана реалізувати заздалегідь визначені цілі. Існують правила, щодо необхідної кількості кольорів та шрифтів у макеті, симетрії, відступів, порядку подачі матеріалів тощо. В Україні правилами веб-дизайну займався дизайнер та викладач кафедри Мережєвих видань і медіа-дизайну Інституту журналістики Київського Національного Університету ім. Тараса Шевченка Салига Павло.

1.2 Принципи веб-дизайну та розробки веб-сайтів

Принципи веб-дизайну та розробки веб-сайтів визначають набір основних принципів та рекомендацій, які використовуються для створення ефективних, зручних у використанні та привабливих веб-сайтів. Вони допомагають забезпечити коректне функціонування сайту, його зручність для користувачів і відповідність сучасним веб-стандартам.

Зручність використання. Веб-сайт повинен бути легким у навігації та зрозумілим для користувачів. Його структура, розміщення елементів та інтерфейс повинні бути логічними та інтуїтивно зрозумілими. Важливо враховувати потреби цільової аудиторії та забезпечити простоту та зручність

взаємодії з сайтом.

Візуальна привабливість. Естетика веб-сайту має значення. Гармонійне поєднання кольорів, використання візуальних ефектів та відповідний дизайн створюють привабливий вигляд сайту. Важливо дотримуватися сучасних тенденцій веб-дизайну, але також враховувати брендові особливості та цінності компанії.

Адаптивність та відзивчивість. Веб-сайт повинен бути адаптивним та добре працювати на різних пристроях та роздільних здатностях екранів. Важливо використовувати респонсивний дизайн, який забезпечує автоматичну адаптацію контенту та макету під різні пристрої.

Швидкість завантаження. Веб-сайт повинен завантажуватись швидко, оскільки користувачі мають низьку терпимість до довгих часів очікування. Важливо оптимізувати розмір зображень, використовувати кешування, компресію файлів та інші методи для покращення швидкості завантаження.

Доступність. Веб-сайт повинен бути доступним для користувачів з різними фізичними та когнітивними обмеженнями. Дотримання стандартів доступності, таких як WCAG, дозволяє забезпечити, що веб-сайт може бути використаний широкою аудиторією, включаючи людей з вадами зору, слуху, руховими обмеженнями та іншими особливостями.

SEO-оптимізації. Важливо розробляти веб-сайт з урахуванням SEO-принципів, щоб забезпечити високу видимість сайту в пошукових системах. Це включає оптимізацію метатегів, використання ключових слів, структурування контенту та інші техніки для покращення ранжування в пошукових системах.

Нижче наведена схема, що включає основні фактори формування влучних веб-дизайну:

Схема 1. Фактори формування сайтів

Веб-дизайн – це процес створення та організації веб-сайтів, включаючи графічний дизайн, розмітку сторінок, взаємодію з користувачем та інші аспекти, щоб забезпечити естетичний, функціональний та зручний веб-сайт.

Дизайн інтерфейсу – це процес проектування і створення елементів інтерфейсу, таких як кнопки, меню, форми, значки тощо, з метою забезпечення зручної та зрозумілої взаємодії користувача з програмою чи веб-сайтом.

Інтерфейс – це точка контакту між користувачем і системою, де відбувається обмін інформацією та взаємодія. Він може включати елементи дизайну, функціональність та способи навігації.

Візуальний інтерфейс – це аспект інтерфейсу, який включає в себе графічні елементи, кольори, типографіку, ілюстрації та інші візуальні компоненти, що впливають на сприйняття та зовнішній вигляд інтерфейсу.

HTML реалізація – це процес перетворення дизайну та інтерфейсу веб-сайту в кодову реалізацію за допомогою мови розмітки HTML. Вона використовується для створення структури, розміщення елементів та форматування веб-сторінок.

Інформаційна архітектура – це процес організації та структурування інформації на веб-сайті, щоб забезпечити логічну навігацію, легкість пошуку та

зрозумілість для користувачів.

Користувацький досвід – це спосіб, яким користувачі сприймають та взаємодіють з веб-сайтом чи програмою. Він охоплює емоційні, практичні та ергономічні аспекти, які впливають на задоволення та ефективність користування.

Карта сайту – це структуроване візуальне представлення ієрархії сторінок веб-сайту. Вона допомагає користувачам легко орієнтуватися, знаходити потрібну інформацію та навігувати по сайту.

Контент – це інформація, яка міститься на веб-сайті, така як тексти, зображення, відео, аудіо та інші медіа-елементи. Контент є важливим елементом, який привертає увагу користувачів та передає їм необхідну інформацію.

Створення контенту – це процес створення, редагування та оформлення контенту для веб-сайту. Це включає написання текстів, створення графіки, редагування відео та інші дії, щоб забезпечити якісний та привабливий контент.

Стратегія контенту – це планування та управління контентом на веб-сайті з метою досягнення певних цілей. Вона включає визначення цільової аудиторії, вибір форматів контенту, розклад публікацій та інші аспекти, які допомагають ефективно просувати веб-сайт та залучати користувачів.

1.3 Роль технологій та веб-сервісів у розвитку культурної сфери

У сучасному цифровому світі технології та веб-сервіси займають все більш важливе місце у розвитку культурної сфери. Вони впливають на спосіб, яким ми сприймаємо, досліджуємо та спілкуємося з культурою. Роль технологій у цій сфері є незаперечною, прискорюючи доступ до культурних змістів, розширюючи можливості їхнього зберігання, передачі та інтерпретації.

Одним із головних аспектів, на який впливають технології та веб-сервіси, є доступ до культурних ресурсів. Завдяки Інтернету та веб-платформам, ми маємо можливість швидкого та зручного отримання інформації про культурні події, мистецтво, літературу, музику та інші аспекти культури. Онлайн-ресурси,

такі як сайти, застосунки та соціальні мережі, створюють віртуальні простори, де люди можуть ділитися своїми творчими досягненнями, взаємодіяти з іншими людьми та знаходити нові джерела натхнення.

Технології також сприяють збереженню та відтворенню культурної спадщини. Цифрові архіви, віртуальні музеї та онлайн-колекції стають місцями, де можна досліджувати та вивчати культурні артефакти, які можуть бути розташовані у віддалених регіонах світу. Це відкриває нові можливості для освіти, досліджень та збереження культурної спадщини, а також дозволяє зберегти цінність цих матеріалів для майбутніх поколінь.

Крім того, веб-сервіси сприяють популяризації культурних заходів та місць відпочинку. Організатори культурних подій можуть використовувати веб-платформи для промоції своїх заходів, продажу квитків та спілкування з аудиторією. Сервіси, що пропонують інформацію про місця відпочинку, туристичні об'єкти та культурні маршрути, допомагають людям знаходити нові місця для відпочинку та пізнавальних подорожей.

Необхідно відзначити, що технології та веб-сервіси мають свої виклики і ризики. Наприклад, питання щодо цифрової приватності, авторських прав та безпеки інформації стають актуальними у контексті використання цих технологій у культурній сфері. Важливо забезпечити збалансоване використання технологій, щоб вони сприяли розвитку культури, захисту прав та інтересів учасників цієї сфери.

Провівши дослідження, можна сказати, що технології та веб-сервіси відіграють значну роль у розвитку культурної сфери. Вони розширюють доступ до культурних ресурсів, сприяють збереженню культурної спадщини, популяризують культурні заходи та місця відпочинку. Проте, їх використання потребує уважного підходу, щоб забезпечити етичне, безпечне та стійке використання технологій у культурній сфері.

Основні функції сайтів у розвитку культурної сфери можуть включати:

Представлення культурних подій. Сайти можуть використовуватись для розміщення інформації про культурні події, такі як концерти, виставки,

фестивалі, театральні вистави тощо. Вони надають можливість ознайомитися з датами, часами, локаціями та деталями подій.

Продаж квитків. Багато сайтів у культурній сфері надають можливість придбати квитки на культурні події онлайн. Це спрощує процес придбання квитків і дозволяє людям зручно забронювати місця на заходах.

Поділ інформації про культурні об'єкти. Сайти можуть містити інформацію про музеї, галереї, пам'ятки архітектури та інші культурні об'єкти. Вони дозволяють користувачам дізнатися про відвідування, режим роботи, виставки та експозиції.

Віртуальні тури. Деякі сайти у культурній сфері надають можливість віртуальної екскурсії. Це дозволяє відвідувачам досліджувати культурні об'єкти, навіть якщо вони знаходяться в інших місцях або країнах.

Веб-платформи для творчих проектів. Деякі сайти створені для того, щоб збирати та показувати творчі проекти, такі як література, живопис, фотографія, музика тощо. Вони допомагають артистам, художникам і творцям знайти аудиторію та отримати відгуки на свої роботи.

Спілкування та обмін думками. Сайти можуть створювати можливість для спілкування і обміну думками між користувачами. Це може включати форуми, коментарі, соціальні мережі або спеціалізовані платформи для обговорення культурних тем і подій.

Онлайн-ресурси та навчальні матеріали. Багато сайтів у культурній сфері надають освітні матеріали, відео-уроки, вебінари та ресурси для самоосвіти у галузі мистецтва, культури, історії та інших суміжних областей.

Популяризація культурних традицій. Сайти можуть використовуватися для популяризації культурних традицій, народних звичаїв та інших аспектів культурного спадщини. Вони можуть розповідати про різноманітні свята, фольклор, ремесла та інші аспекти культури.

Інтерактивність та захоплюючий контент. Сайти можуть використовувати інтерактивні елементи, відео, аудіо, фотографії та інші формати контенту для залучення аудиторії. Це дозволяє створити захоплюючі

експерименти, які привертають увагу і зацікавлення користувачів.

Зручність і доступність. Однією з важливих функцій сайтів у культурній сфері є забезпечення зручного та доступного досвіду для користувачів. Це включає простоту навігації, зрозумілу структуру, розміщення актуальної інформації та адаптивний дизайн для різних пристроїв.

Такі функції веб-сайтів в культурній сфері допомагають розширити доступ до культурних ресурсів, залучити аудиторію та сприяють розвитку культури в сучасному цифровому середовищі. Вони створюють можливості для обміну ідеями, навчання, спілкування та популяризації культурних цінностей.

1.4 Цілі та вимоги до функціональності веб-сайту

Мій майбутній веб-сайт матиме важливу роль у сприянні розвитку та просуванні культурної сфери. Для досягнення моїх бажань щодо сайту важливо визначити конкретні цілі та вимоги до функціональності веб-сайту.

Веб-сайт має надавати користувачам доступ до актуальної та повної інформації про культурні заходи, події та місця відпочинку в місті. Інформація повинна бути легко знаходжуваною та зрозумілою для різних категорій користувачів.

Веб-сайт повинен містити інформацію про розклад роботи організацій, відповідальних за культурні заходи, а також контактні дані, щоб користувачі мали можливість зв'язатися з організаторами та отримати додаткову інформацію.

Веб-сайт повинен мати зручну функцію пошуку, яка дозволяє користувачам швидко знаходити потрібні події чи місця відпочинку за різними критеріями, такими як дата, місце, тематика тощо.

Важливим аспектом культурних заходів є візуальна привабливість. Веб-сайт повинен містити фотографії, які демонструють атмосферу подій та місць відпочинку.

Веб-сайт може мати функції поширення контенту через соціальні медіа, що допомагає просувати культурні заходи та залучати більше аудиторії.

1.5 Огляд можливих програмних рішень

Однією з важливих тем, на які варто звернути увагу є те, що існує багато програм та інструментів для створення веб-сайтів, які допомагають простим Інтернет-користувачам, без глибоких знань у галузі комп'ютерних технологій, та непрофесійним новачкам зробити цей процес більш доступним і зручним. Ось кілька популярних програм для створення веб-сайтів: WordPress, Wix, Shopify, Adobe Dreamweaver, Joomla, Squarespace, Google Sites, Weebly, Webflow, Adobe Muse.

Ці програми та інструменти надають широкі можливості для створення веб-сайтів без необхідності в глибоких знаннях програмування. Вони пропонують різні функції, готові шаблони, інтуїтивний інтерфейс та інтеграцію з іншими сервісами, дозволяючи вам створити професійний веб-сайт за короткий час.

При виборі програми-конструктора для веб-сайту важливо враховувати наші потреби, рівень гнучкості, налаштування та можливості для подальшого розвитку, адже такі програми мають значні обмеження в порівнянні зі стандартним підходом.

Ось такими основними перевагами, на мій погляд, володіють програми-конструктори веб-застосунків:

- простота використання;
- готові шаблони та компоненти;
- швидкість розробки.

Сервіси-конструктори зазвичай мають інтуїтивний і легкий у використанні інтерфейс, що дозволяє навіть новачкам створювати веб-сайти без необхідності знання програмування чи дизайну. Багато програм-конструкторів пропонують велику кількість готових шаблонів та компонентів, які можна використовувати для створення веб-сайту. Це полегшує процес розробки та дозволяє швидше створити функціональний сайт. Завдяки готовим шаблонам та компонентам, програми-конструктори дозволяють прискорити процес розробки

застосунку. Вам не потрібно створювати все з нуля, а просто вибрати потрібні елементи та налаштувати їх за потребами.

Але такі програми мають свої недоліки:

- обмежена гнучкість;
- залежність від платформи;
- недостатня унікальність кінцевого продукту.

Оскільки програми-конструктори призначені для широкого кола користувачів, вони можуть мати обмежені можливості налаштування. Ви не завжди зможете реалізувати складні функціональність чи унікальний дизайн, які можуть бути нестандартними для конкретного конструктора. Багато програм-конструкторів працюють на певних платформах чи хостингах, що може обмежити вашу свободу вибору. Вам потрібно розглянути питання масштабованості та переносу сайту, якщо ви плануєте змінити платформу у майбутньому. Завдяки використанню готових шаблонів та компонентів, може виникнути проблема нестачі унікальності сервісу. Ви можете зіткнутися з ситуацією, коли інші веб-сайти, що використовують ту ж саму програму-конструктор, мають схожий вигляд та функціонал.

Я ознайомився з функціоналом найбільш популярних допоміжних застосунків для створення сайтів застосунків та можу підбити результати.

WordPress є однією з найпопулярніших платформ для створення веб-сайтів. Він використовується мільйонами людей по всьому світу, від особистих блогерів до великих корпорацій. Однією з найбільших переваг *WordPress* є його простота в використанні. Він має інтуїтивний і зрозумілий інтерфейс, що дозволяє навіть новачкам швидко освоїти його функціонал.

WordPress має широкий вибір тем і плагінів, які дозволяють налаштувати зовнішній вигляд і функціональність вашого веб-сайту. Ви можете вибрати тему, яка відповідає вашому стилю та вимогам, і додати плагіни, що розширяють можливості вашого сайту. *WordPress* також підтримує різноманітні типи контенту, такі як статті, фотогалереї, відео, аудіо тощо.

Однією з сильних сторін WordPress є його велика спільнота користувачів. Ви можете знайти безліч ресурсів, форумів, підручників і підтримки від інших користувачів, що допоможуть вам вирішити будь-які проблеми або запитання, що виникають під час роботи з WordPress.

WordPress також пропонує зручні інструменти для управління контентом, такі як редактор блоків, який дозволяє вам легко створювати та редагувати вміст на своєму сайті. Ви можете легко додавати текст, зображення, відео, таблиці та інші елементи до своїх сторінок без необхідності знань кодування.

Як висновок, WordPress є потужною та гнучкою платформою для створення веб-сайтів, яка надає вам велику свободу та контроль над вашим проектом. Незалежно від того, чи ви новачок у веб-розробці, чи досвідчений користувач, WordPress дозволяє вам швидко створити професійний веб-сайт з мінімальними зусиллями.

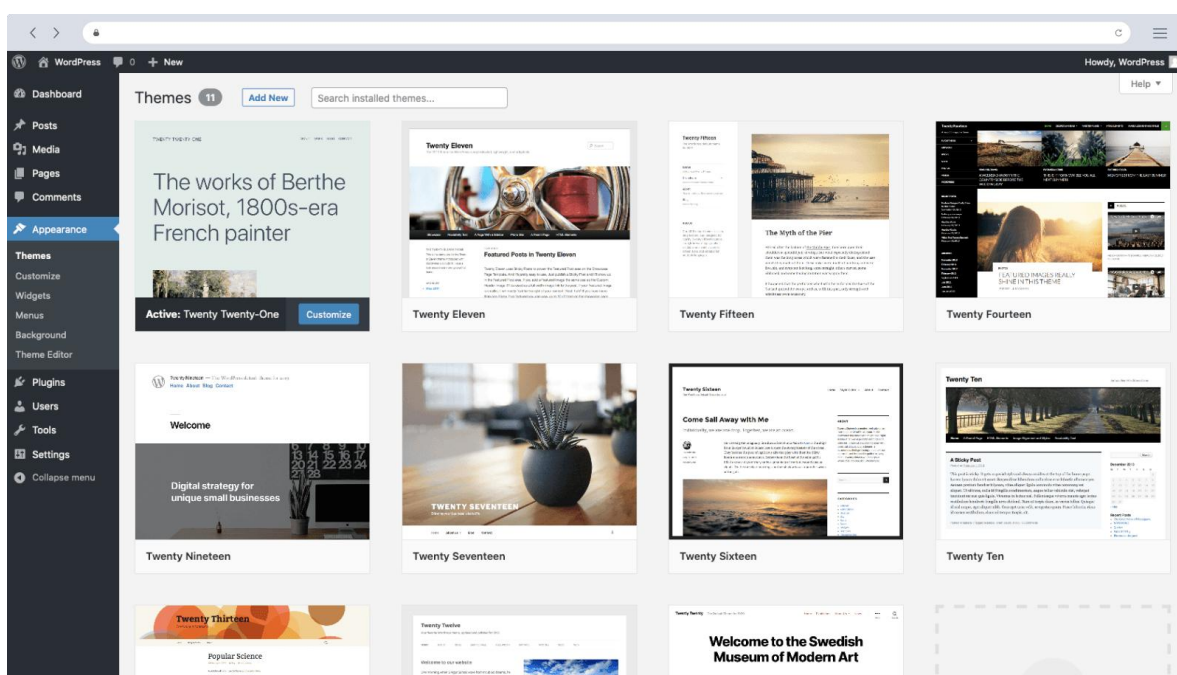


Рисунок 1.1 – Головне вікно WordPress

Wix – це популярний веб-конструктор, який дозволяє створювати професійні веб-сайти без необхідності знань програмування або дизайну. Однією з головних переваг Wix є його легкість в використанні. Інтерфейс Wix інтуїтивно зрозумілий і простий, що дозволяє навіть початківцям швидко розібратися з його функціоналом.

У Wix великий вибір шаблонів, з якими ви можете почати свій проект. Шаблони розділені на різні категорії, що дозволяє вибрати той, що найкраще відповідає вашим потребам і стилю. Ви також можете використовувати редактор перетягування та опускання, щоб змінювати розміщення елементів на своєму сайті та налаштовувати його вигляд.

Wix пропонує багато вбудованих функцій і можливостей, які дозволяють розширити можливості вашого веб-сайту. Ви можете додати фотогалереї, блоги, форми зворотного зв'язку, соціальні медіа-кнопки та багато іншого. Також є можливість інтегрувати сторонні сервіси, такі як платіжні системи або інструменти аналітики.

Wix також має мобільний редактор, що дозволяє вам налаштовувати вигляд вашого сайту на мобільних пристроях. Ви можете змінювати макети, додавати анімацію, оптимізувати вміст для мобільних пристроїв, щоб ваш сайт виглядав і працював на різних пристроях належним чином.

Однак, важливо враховувати, що Wix має свої обмеження. Хоча ви можете налаштувати вигляд і функціонал свого сайту, деякі більш складні зміни можуть бути обмежені. Також, використання деяких розширень і функцій може вимагати платного плану.

Отже, Wix є потужним і простим у використанні інструментом для створення веб-сайтів, який підходить для різних типів проектів. Його широкий вибір шаблонів, легкість використання та вбудовані функції роблять його популярним вибором для тих, хто шукає швидкий спосіб створити професійний веб-сайт без труднощів програмування.

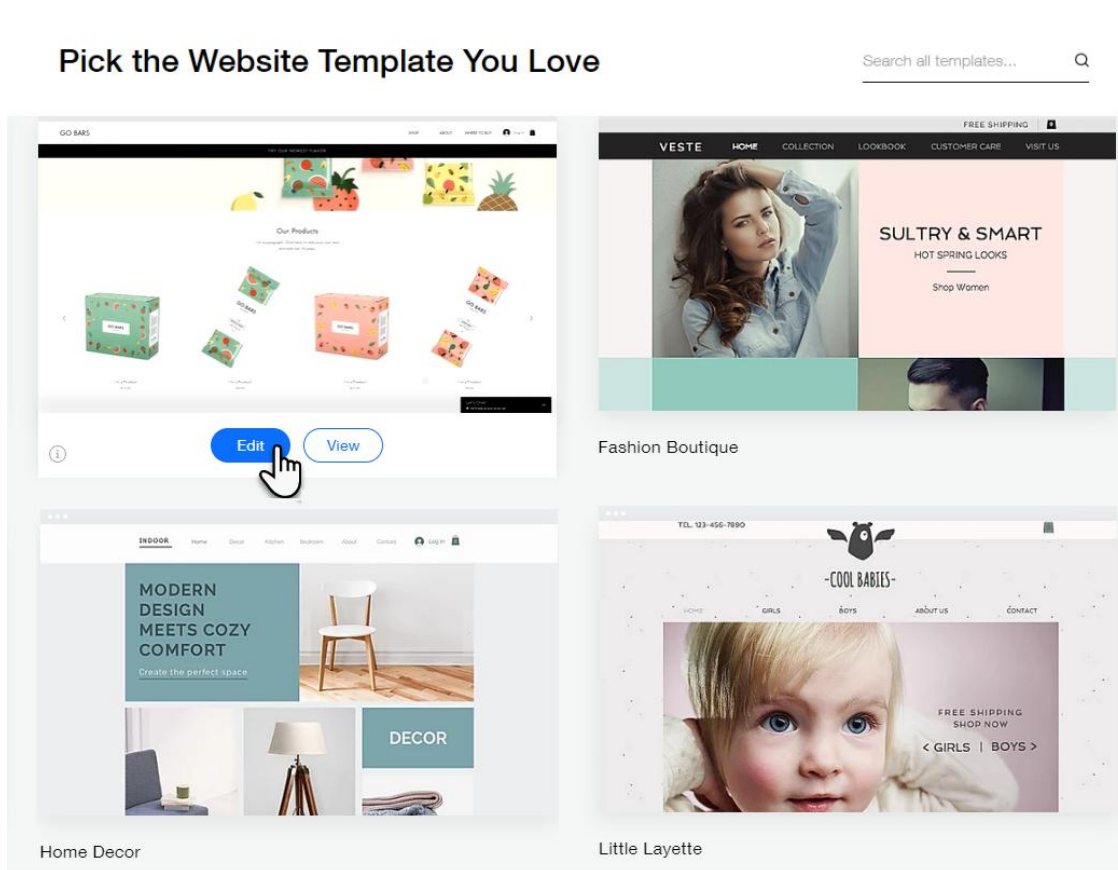


Рисунок 1.2 – Головне вікно Wix

Adobe Dreamweaver – це професійна програма для розробки веб-сайтів, що надає розширені можливості для дизайну, розмітки та програмування. Вона відома своєю візуальною редакцією коду та вбудованими інструментами для підтримки різних технологій веб-розробки.

Однією з основних переваг *Adobe Dreamweaver* є його можливість працювати як візуальний редактор, де ви можете перетягувати та розташовувати елементи на сторінці, так і як кодовий редактор, що дозволяє вам працювати з HTML, CSS, JavaScript та іншими мовами програмування. Це робить його універсальним інструментом для веб-дизайнерів та програмістів, які шукають зручний спосіб розробки веб-сайтів.

Dreamweaver надає широкі можливості налаштування та розширення, що дозволяє вам адаптувати його до ваших потреб. Ви можете додавати власні плагіни, створювати власні шаблони та розширювати функціональність програми. Крім того, *Dreamweaver* інтегрується з іншими продуктами *Adobe*,

такими як Photoshop чи Illustrator, що дозволяє зручно працювати з графічним вмістом та експортувати його безпосередньо у веб-сайти.

Однак, важливо відзначити, що Adobe Dreamweaver є потужним інструментом, який вимагає певного рівня знань та досвіду у веб-розробці. Його широкий функціонал може бути перевагою для професіоналів, але для початківців він може виявитися складним і перевантаженим. Також, програма є комерційним продуктом і вимагає придбання ліцензії.

Загалом, Adobe Dreamweaver є потужним інструментом для розробки веб-сайтів з великим набором функцій та можливостей. Він підходить для професіоналів, які шукають інструмент для розробки складних веб-проектів і мають відповідні навички у веб-розробці.

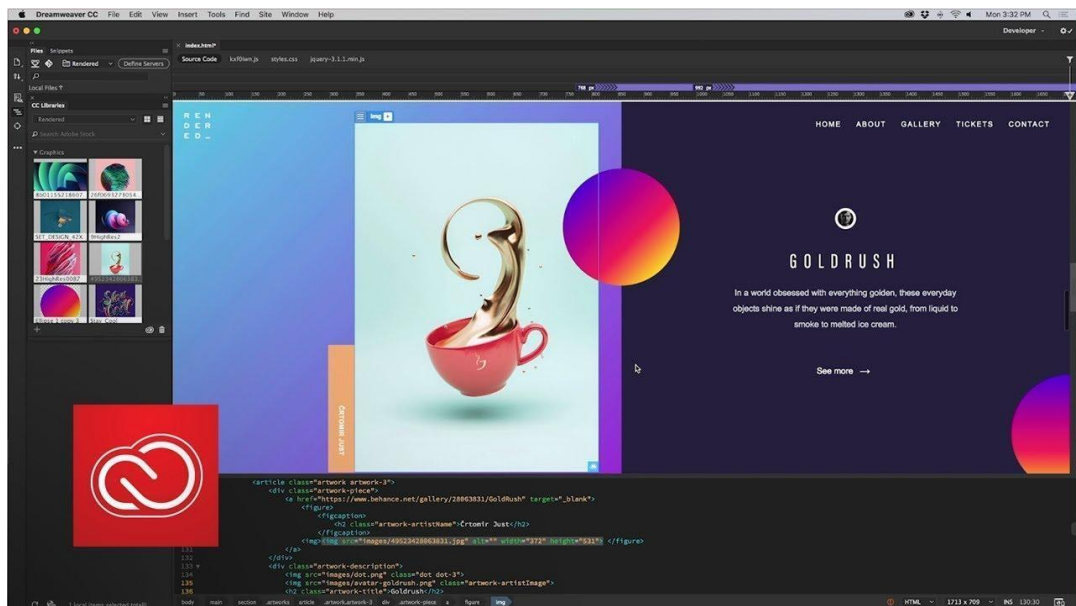


Рисунок 1.3 – Процес створення сайту за допомогою Adobe Dreamweaver

Таблиця 1.1. – порівняльна таблиця застосунків WordPress, Wix та Adobe Dreamweaver.

Платформа	Гнучкість	Унікальність	Шаблони	Простота в розробці	Швидкість розробки
WordPress	Середня	Висока	Висока	Середня	Середня
Wix	Низька	Середня	Висока	Висока	Висока
Adobe Dreamweaver	Висока	Висока	Середня	Низька	Середня

Пояснення до таблиці:

Wix є досить обмеженим у відношенні налаштування та редагування веб-сайту. Ви маєте обмежені можливості налаштування дизайну та функціональності.

WordPress та Adobe Dreamweaver надають більше гнучкості щодо налаштування. Ви можете впливати на вигляд та функціональність веб-сайту шляхом налаштування тем та коду.

Wix надає велику кількість шаблонів та блоків, що можуть призвести до меншої унікальності результату. У випадку використання популярних шаблонів, ваш сайт може виглядати схожим на інші сайти, створені на Wix.

WordPress має велику кількість тем і розширень, що дозволяє створювати більш унікальні веб-сайти. Однак, використання популярних тем може призвести до деякої схожості з іншими сайтами.

Adobe Dreamweaver дозволяє вам створювати повністю унікальний дизайн та функціональність вашого веб-сайту, оскільки ви можете редагувати HTML, CSS та інші файли без обмежень.

Adobe Dreamweaver не надає велику кількість готових шаблонів. Ви повинні самостійно створювати дизайн вашого веб-сайту.

WordPress має велику кількість безкоштовних і платних тем, які можна використовувати для створення веб-сайту.

Wix надає найбільшу кількість шаблонів і готових блоків зі всіх платформ, які розглядаються. Шаблони можна використовувати для швидкого створення веб-сайту.

Wix має простий редактор перетягування та розміщення, що робить процес розробки досить простим і швидким.

WordPress має дещо вищий рівень складності у порівнянні з Wix. Він вимагає певних знань управління темами, плагінами та налаштуванням.

Adobe Dreamweaver вимагає глибоких знань у веб-розробці, оскільки ви повинні вручну створювати код HTML, CSS та код інших мов програмування за потребою.

Важливо вказати, що при виборі програми-конструктора для веб-сайту важливо враховувати наші потреби щодо сайту, рівень гнучкості, налаштування та можливості для подальшого розвитку.

РОЗДІЛ 2. АНАЛІЗ РІШЕНЬ І СТЕК ТЕХНОЛОГІЙ ДЛЯ РОЗРОБЛЕННЯ ВЕБ-САЙТУ.

2.1 Основні програмні рішення розробки сайту

Вибір правильних застосунків є важливою складовою розробки технологічних проєктів. У світі, де технології швидко розвиваються, надання переваги відповідним та сучасним інструментам є ключовим фактором успіху. Від вибору правильних застосунків залежить продуктивність, ефективність та якість роботи застосунку.

Вибір застосунків має вплив на багато аспектів, включаючи функціональність, масштабованість, продуктивність, безпеку та зручність в розробці. Необхідно враховувати особливості нашого проєкту, його мету та вимоги, щоб знайти технології, які найкраще відповідають потребам.

Вибір правильних застосунків дозволяє забезпечити ефективну розробку, легке масштабування та підтримку вашого проєкту у майбутньому. Також, він впливає на користувацький досвід, дозволяючи створити інтуїтивний та зручний інтерфейс.

У цьому швидкоплинному світі технологій, вибір правильних застосунків є необхідним для підтримки конкурентоспроможності та успіху. Незалежно від його розміру та складності, вибір правильних застосунків важливий крок у досягненні ваших цілей та наданні якісних продуктів та сервісів.

При виборі технологій для розробки веб-застосунку було приділено увагу таким трьом основним аспектам:

1. Веб-фреймворк
2. Бази даних
3. Front-end технології

Веб-фреймворк – це набір інструментів, бібліотек та стандартів, які спрощують розробку веб-застосунків. Він надає розробникам зручний набір функцій та шаблонів для побудови веб-сайтів або веб-застосунків швидше та ефективніше. Веб-фреймворки допомагають забезпечити структуру, організацію та повторне використання коду, що дозволяє прискорити розробку

та зменшити кількість написаного коду.

Основна ідея веб-фреймворків полягає в тому, щоб відокремити логіку бізнес-логіки веб-застосунку від деталей реалізації, таких як маршрутизація, обробка запитів та відповідей, робота з базами даних тощо. Веб-фреймворки часто мають свою специфічну архітектуру та правила розробки, що дозволяють розробникам більш організовано та структуровано писати код.

Django – це веб-фреймворк, розроблений на мові програмування Python. Він надає розробникам потужні інструменти для швидкого та ефективного створення веб-застосунків. Django пропонує комплексний підхід до веб-розробки, забезпечуючи все необхідне для створення функціональних та масштабованих веб-застосунків.

Однією з основних переваг Django є його простота в використанні та швидкість розробки. Він має чистий і зрозумілий синтаксис, що дозволяє розробникам швидко впроваджувати нові функціональні можливості. Django також надає широкий спектр готових компонентів і бібліотек, які допомагають розробникам ефективно працювати над своїми проектами.

Одним з головних принципів Django є «кожен застосунок повинен мати свою відповідність». Це означає, що веб-застосунок розбивається на невеликі компоненти, які можуть бути повторно використані та розширені. Ця модульна структура дозволяє розробникам організувати свій код більш структуровано і зручно управляти функціональністю застосунку.

Django забезпечує вбудовану підтримку аутентифікації, управління користувачами, адміністративний інтерфейс та безпеку. Він також пропонує велику кількість інструментів для роботи з базами даних, URL-маршрутизацію, шаблони та інші компоненти, що допомагають розробникам створювати потужні веб-застосунки зі складною функціональністю.

Загалом, Django є потужним і гнучким веб-фреймворком, який спрощує розробку веб-застосунків на Python. Він надає розробникам всі необхідні інструменти для швидкого розгортання проектів і забезпечує ефективну роботу з різними компонентами веб-застосунку. Django є популярним вибором для

розробників, які цінують продуктивність, зручність використання та масштабованість своїх веб-застосунків.

Ruby on Rails – це веб-фреймворк, розроблений на мові програмування Ruby. Він використовує концепцію Model-View-Controller для побудови веб-застосунків. Rails пропонує конвенції над конфігурацією, що дозволяє розробникам швидше розгортати та підтримувати свої проекти.

Однією з ключових особливостей Ruby on Rails є його фокус на зручності розробки та простоті. Фреймворк ставить на перше місце конвенції, що дозволяють розробникам швидше створювати функціональні веб-застосунки. Rails забезпечує стандартні шаблони та практики, що роблять розробку більш передбачуваною і менш складною для команди розробників.

Ще одна суттєва перевага Ruby on Rails – це його активна спільнота розробників і велика кількість готових модулів та плагінів. Це дозволяє розробникам використовувати готові рішення для швидкого впровадження певної функціональності в їх веб-застосунки. Rails також надає зручний інструментарій для роботи з базами даних, маршрутизації URL, управління сесіями та багатьма іншими аспектами розробки.

Завдяки своїй гнучкості та здатності до швидкого розгортання проектів, Ruby on Rails є популярним вибором для розробки веб-застосунків різного масштабу. Він надає розробникам зручні інструменти для швидкої розробки прототипів та ефективного впровадження функціональності. Багато успішних веб-проектів, таких як GitHub, Airbnb та Shopify, були побудовані на Ruby on Rails.

Laravel - це веб-фреймворк, розроблений на мові програмування PHP, який пропонує елегантну та зручну платформу для розробки веб-застосунків. Він заснований на концепції Model-View-Controller, що дозволяє розробникам легко організувати код, розділяти логіку застосунку та відображення даних.

Однією з головних переваг Laravel є його простота в використанні та чистота коду. Фреймворк надає інтуїтивний синтаксис, чітку структуру та багато готових компонентів, що допомагають розробникам швидко створювати

функціональні веб-застосунки. Laravel також пропонує розширені можливості для маршрутизації, управління базами даних, роботи з сесіями та багатьма іншими аспектами розробки.

Іншою значною перевагою Laravel є його розширюваність та розширений набір інструментів. Він має вбудовану систему пакетного менеджера Composer, яка дозволяє легко встановлювати та керувати залежностями веб-застосунків. Laravel також має широкий спектр готових розширень та пакетів, які розробники можуть використовувати для додавання специфічної функціональності до своїх проєктів.

Завдяки своїй гнучкості, Laravel є популярним вибором для розробки веб-застосунків різного рівня складності. Він пропонує широкі можливості для розробників, починаючи від невеликих веб-сайтів до великих корпоративних проєктів. Laravel також має активну спільноту розробників, що підтримує його розвиток та надає велику кількість документації, пакетів та ресурсів для розробників.

Express.js - це легкий і гнучкий веб-фреймворк для розробки веб-застосунків на мові програмування JavaScript. Він побудований на базі Node.js і надає простий спосіб створити серверну частину застосунку, використовуючи потужні можливості JavaScript.

Однією з основних переваг Express.js є його простота в використанні. Він має мінімальний набір функцій, який дозволяє розробникам швидко створювати веб-сервери та обробляти запити. Express.js надає просту й інтуїтивну структуру застосунку, яка дозволяє легко маршрутизувати запити, встановлювати обробники маршрутів та виконувати різноманітні дії.

Друга перевага Express.js – це також його гнучкість. Він не накладає жорсткі правила та обмеження на розробку застосунків і дозволяє розробникам вибрати ті інструменти та бібліотеки, які вони вважають найкращими для своїх потреб. Express.js також підтримує використання шаблонів для відображення динамічного контенту та може бути поєднаний з різними базами даних та іншими інструментами.

Express.js має активну спільноту розробників, яка постійно вносить коректив його розвитку та надає різноманітні пакети та модулі, які полегшують розробку. Це дозволяє розробникам швидко знайти рішення для своїх завдань та отримати підтримку та поради від інших членів спільноти.

Отже, Express.js є потужним і простим у використанні веб-фреймворком, який дозволяє розробникам швидко створювати веб-застосунки на основі JavaScript. Його гнучкість та підтримка широкого спектру рішень роблять його популярним вибором для розробки сучасних веб-застосунків.

Таблиця 2.1. Порівняльна таблиця між Django, Ruby on Rails, Laravel і Express.js.

	Django	Ruby on Rails	Laravel	Express.js
Мова програмування	Python	Ruby	PHP	JavaScript
Архітектура	MVC	MVC	MVC	Мінімальний фреймворк
Документація	Добра	Добра	Добра	Прийнятна
Швидкість	Швидка	Швидка	Швидка	Швидка
Розширюваність	Легко розширюється	Легко розширюється	Легко розширюється	Легко розширюється
Комплектність	Великий набір функціональності	Великий набір функціональності	Великий набір функціональності	Мінімальний набір функціональності
Безпека	Міцна безпека	Міцна безпека	Міцна безпека	Потребує додаткових заходів безпеки
ORM	Наявний	Наявний	Наявний	Відсутній, але може бути інтегрований
Тестування	Вбудована	Вбудована	Вбудована	Відсутня

	підтримку тестів	підтримку тестів	підтримку тестів	підтримка
--	------------------	------------------	------------------	-----------

Пояснення до таблиці:

MVC є скороченням від Model-View-Controller. Це архітектурний шаблон, що використовується у багатьох веб-фреймворках для побудови програмного забезпечення.

ORM є скороченням від Object-Relational Mapping. Це технологія, яка дозволяє взаємодіяти з базою даних з використанням об'єктно-орієнтованого підходу.

Окремо хочу наголосити на тому, що Django застосовується в різних проектах, від невеликих до великих, включаючи сайти, веб-застосунки, електронну комерцію та інше; Ruby on Rails широко використовується веб-розробкою, особливо для створення соціальних мереж, комерційних платформ та стартапів; Laravel використовується в різних проектах, зокрема управління контентом, електронна комерція, CRM-системи та інші; Express.js використовується для розробки швидких та масштабованих веб-застосунків, в тому числі API, односторінкових застосунків та реального часу.

Бази даних є невід'ємною частиною багатьох веб-сервісів і відіграють важливу роль у збереженні та управлінні даними. Вони дозволяють нам зберігати структуровану інформацію, отримувати до неї доступ, виконувати пошук та аналіз даних, здійснювати зв'язки між різними сутностями та багато іншого.

При розробці веб-сервісу, ми можемо використовувати різні типи баз даних, такі як реляційні бази даних (наприклад, MySQL, PostgreSQL), нереляційні бази даних (наприклад, MongoDB, CouchDB), а також гібридні рішення, які поєднують реляційні та нереляційні підходи (наприклад, PostgreSQL з підтримкою JSONB).

Вибір бази даних залежить від різних факторів, таких як потреби проекту, тип даних, вимоги до масштабованості та продуктивності, наші власні знання та досвід. Реляційні бази даних зазвичай використовуються для даних з чіткою структурою та зв'язками між сутностями, в той час як нереляційні бази даних

зручні для зберігання неструктурованих або змінюваних даних.

Розробка та управління базою даних є критичним елементом веб-сервісу, і правильний вибір технологій та налагодження оптимальної архітектури допоможуть нам створити потужний та ефективний веб-сервіс з можливістю масштабування.

Реляційна база даних є типом бази даних, який використовує модель даних, що ґрунтується на реляційній теорії. У реляційній базі даних дані організовані у вигляді таблиць, які складаються з рядків і стовпців.

Реляційна модель даних базується на концепції взаємозв'язків між таблицями. У реляційній базі даних дані утримуються у взаємозв'язаних таблицях з допомогою ключів. Кожна таблиця має унікальний ідентифікатор, відомий як первинний ключ, який ідентифікує кожен рядок у таблиці. Ключі також використовуються для встановлення зв'язків між таблицями.

Реляційні бази даних використовують мову структурованих запитів SQL для створення, модифікації та записування даних. SQL надає потужні засоби для роботи з реляційними базами даних, включаючи можливість створення таблиць, вставки, оновлення та видалення даних, а також складні запити для отримання потрібної інформації.

Реляційні бази даних є дуже поширеними і використовуються у багатьох сферах, включаючи бізнес, науку, освіту та веб-розробку. Вони забезпечують надійне зберігання та організацію даних, а також можливість ефективної обробки та запитування інформації.

Нереляційна база даних – це тип бази даних, який відрізняється від реляційних баз даних за своєю структурою та підходом до зберігання та організації даних. У нереляційних базах даних дані можуть бути збережені у різних форматах, таких як ключ-значення, документи, стовпчики або графи.

Нереляційні бази даних створені для розв'язання специфічних вимог і завдань, які необхідно вирішити в деяких сценаріях. Вони зазвичай забезпечують гнучкість, швидкість та масштабованість, що робить їх популярними в проектах, де потрібно зберігати та обробляти великі обсяги

даних або коли структура даних може змінюватися з часом.

Основні види нереляційних баз даних включають ключ-значення, документні, стовпчикові та графові бази даних. Кожен з цих типів нереляційних баз даних має свої особливості та використовується для різних типів застосунків.

Нереляційні бази даних можуть бути корисними в ситуаціях, де потрібна висока швидкість доступу до даних, гнучкість у зберіганні та організації даних, а також при необхідності масштабування системи. Вони дозволяють розробникам ефективно працювати з невстановленою або змінною структурою даних, а також зберігати та обробляти великі обсяги даних у розподілених системах.

Порівняльна таблиця 2.2. Реляційна та Нереляційна бази даних

Критерій	Реляційні бази даних	Нереляційні бази даних
Структура даних	Таблична структура зі зв'язками між таблицями	Гнучка структура, включаючи ключ-значення, документи, стовпці, графи
Схема	Статична	Динамічна, може бути мінімально визначеною або відсутньою
Масштабованість	Обмежена вертикальним та горизонтальним масштабуванням	Висока масштабованість, горизонтальне масштабування
Запити	SQL-запити	Зазвичай використовуються спеціальні мови запитів або API
Сховище даних	Використовуються таблиці у вигляді файлів	Використовуються різні формати та сховища даних
Залежність від структури	Надмірно залежить від структури даних та схеми	Гнучкість у роботі зі змінною структурою даних

Продовження таблиці 2.2

Підтримка транзакцій	Повна підтримка транзакцій	Обмежена підтримка транзакцій або відсутність такої підтримки
Складність налаштування	Потребує попереднього опису схеми та визначення відношень	Менша складність налаштування, не вимагає жорсткого визначення структури
Використання	Широко використовується у традиційних системах та бізнес-застосунках	Використовується у застосунках, де потрібна висока швидкодія та гнучкість в роботі з даними
Приклади	MySQL, PostgreSQL, Oracle, SQL Server	MongoDB, Cassandra, Redis, Elasticsearch

Важливо зазначити, що кожен вид баз даних має свої переваги та використовується в залежності від конкретних вимог та характеристик проекту. Реляційні бази даних підходять для ситуацій, де важлива структурованість даних та потреба в повноцінних транзакціях, в той час як нереляційні бази даних надають гнучкість, швидкодію та масштабованість для роботи зі змінною структурою даних та великими обсягами інформації.

2.2 Вибір технологій для розробки веб-сайту

Я обрав *Express* для розробки веб-сервісу з кількох важливих причин. Перш за все, було оцінено його популярність та широке використання в розробці веб-застосунків на платформі Node.js. Express має велику спільноту розробників, що означає, що є багато ресурсів, документації та підтримки доступних для мене.

Express пропонує мінімалістичний підхід до розробки, що дуже підходить мені. Він надає основні функції та структуру, але залишає волю розробнику для вибору технологій та підходів. Це означає, що є можливість побудувати веб-

сервіс згідно зі своїми унікальними потребами та вимогами.

Також, важливою перевагою Express є його простота використання та швидкість розробки. Він надає зручні функції для створення маршрутів, обробки запитів та відповідей, що дозволяє мені швидко реалізовувати функціональність мого веб-сервісу.

Нарешті, розширюваність Express. Завдяки великій кількості плагінів та модулів, можна легко додавати нові функції до веб-сервісу та розширювати його можливості в майбутньому.

З усіх цих причин було впевнено обрано Express для розробки веб-сервісу. Він надає мені потужні інструменти, гнучкість та швидкість, що необхідні для створення високоякісного та ефективного веб-сервісу.

Я обрав *MySQL* для розробки веб-сервісу з кількох причин. Перш за все, *MySQL* є однією з найпопулярніших та надійних реляційних баз даних, що використовується в широкому спектрі проектів.

Однією з переваг *MySQL* є його швидкодія та висока продуктивність. Він може обробляти великі обсяги даних та виконувати складні запити швидко і ефективно. Це особливо важливо для мого веб-сервісу, де швидкість взаємодії з базою даних є ключовим фактором.

Крім того, *MySQL* має широкий набір функцій і можливостей, що дозволяють мені ефективно управляти даними. З її допомогою можна створювати складні структури бази даних, встановлювати зв'язки між таблицями та виконувати різноманітні операції з даними, такі як сортування, фільтрація та групування.

MySQL також має велику спільноту розробників, що забезпечує підтримку, оновлення та безпеку бази даних. За потреби можна легко знайти рішення на форумах та отримати допомогу в разі виникнення проблем або питань.

Крім того, *MySQL* є безкоштовним та відкритим за ліцензією, що дозволяє мені використовувати його без обмежень та зменшує витрати на розробку.

Загалом, обравши MySQL, я отримав потужний та надійний інструмент для зберігання та управління даними мого веб-сервісу. Він відповідає моїм потребам щодо продуктивності, функціональності та підтримки, що робить його ідеальним вибором для мого проекту.

HTML (HyperText Markup Language) – це мова розмітки, яка використовується для створення структури і візуального вигляду веб-сторінок. HTML використовує теги для визначення елементів сторінки, таких як заголовки, абзаци, зображення, посилання та багато інших. Він визначає структуру документа, а також містить відомості для веб-браузерів про те, як правильно відображати вміст сторінки.

CSS (Cascading Style Sheets) – це мова стилів, яка використовується для визначення зовнішнього вигляду веб-сторінок. CSS визначає кольори, шрифти, розміри, відступи, межі та інші властивості елементів сторінки. Він дозволяє розміщувати елементи на сторінці, створювати анімацію та респонсивний дизайн, що забезпечує зручне відображення на різних пристроях і розмірах екранів.

JavaScript (JS) – це мова програмування, яка використовується для реалізації інтерактивності на веб-сторінках. Він дозволяє додавати функціональність до сторінок, взаємодіяти з користувачем, обробляти події, виконувати асинхронні запити до сервера та багато іншого. JavaScript може працювати в браузері користувача, виконувати скрипти на стороні клієнта, а також використовуватися на сервері (за допомогою платформи Node.js).

Ці три технології (HTML, CSS, JS) часто використовуються разом для розробки фронтенду веб-застосунків. HTML відповідає за структуру сторінки, CSS – за вигляд і стилізацію, а JavaScript – за інтерактивність і поведінку сторінки. Вони взаємодіють між собою, дозволяючи створювати багатофункціональні та зручні для користувача веб-застосунки.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ КУЛЬТУРНИХ ЗАХОДІВ МІСТА

3.1 Архітектура веб-сайту та його компоненти

Архітектура веб-сайтів – це планування та проектування технічних, функціональних та візуальних компонентів веб-сайту – до його розробки, розробки та розгортання.

Я використав архітектуру сайту – клієнт-сервер, оскільки вона дозволяє ефективно розподілити завдання та забезпечити гнучкість, розширюваність та надійність системи.

У цій моделі, клієнт є інтерфейсом користувача або програмою, яка взаємодіє безпосередньо з користувачем. Він відповідає за збір та обробку вхідних даних, відображення інформації користувачу та виконання локальних операцій. Клієнт може бути веб-браузером, мобільним застосунком, настільною програмою тощо.

Сервер, з свого боку, забезпечує надання ресурсів та послуг клієнту. Він обробляє запити від клієнтів, виконує необхідні операції та повертає результати.

Клієнтський інтерфейс – це фронтенд, з яким буде взаємодіяти користувач. Реалізований за допомогою HTML, CSS та JS.

Веб-сервер – це серверна частина сайту, що обробляє запити від клієнтського інтерфейсу. Веб-сервер отримує запити від користувача, обробляє їх і виконує необхідні функції. Він повинен взаємодіяти з базою даних для отримання і збереження інформації.

База даних – це місце, де зберігаються дані про культурні заходи, місця відпочинку, відгуки користувачів сайту тощо. База даних забезпечує доступ до даних і дозволяє зберігати, оновлювати та видаляти інформацію.

Навбар (або навігаційна панель) та футбар (або підвал) – це два важливих компонента веб-сайту, які забезпечують навігацію та інформацію про сайт.

Навбар розташовується зазвичай у верхній частині сторінки і містить навігаційні елементи, які дозволяють користувачам легко переміщатися по

різних розділах та сторінках сайту. Навбар також містить логотип сайту, що переміщує користувача на головну сторінку.

Футбар розташовується у нижній частині сторінки і містить додаткову інформацію про контактні дані у вигляді посилань на додаткові соціальні медіа міста Івано-Франківськ.

Головна сторінка сайту – це основна сторінка, яка вітає користувачів і надає загальну інформацію про місто Івано-Франківськ, у моєму випадку. Тут представлений список оглядів всіх культурних заходів, їх місце та час проведення. Клацнувши на місце проведення, користувач переходить на сторінку з гугл картами, де вже знайдено місце проведення того чи іншого заходу. На цій же сторінці є попап з додатковою інформацією, який надає користувачам можливість отримати додаткову інформацію про певну подію, її детальний опис. Це популярний інтерактивний елемент, який додає зручність і ефективність взаємодії з користувачем.

Друга сторінка з історією створення міста. Вона складається з заголовка, блоку з історією міста та інтерактивною галереєю.

Третя сторінка містить компонент пошуку та фільтрації визначних місць Івано-Франківську. Компонент пошуку та фільтрації є важливою частиною багатьох сайтів. Він дозволяє користувачам знайти потрібні елементи на сайті шляхом введення ключових слів, використання фільтрів або комбінації обох методів.

Остання сторінка містить форму відгуків та пропозицій. Це важливий аспект сайту, який дозволяє відвідувачам сайту надсилати свої коментарі, питання, пропозиції або відгуки щодо культурних подій та заходів. Отримані відгуки та пропозиції зберігаються в базі даних для подальшої обробки адміністраторами. Застосування форми для відгуків та пропозицій на сайті культурних заходів міста сприяє активній взаємодії з користувачами, забезпечує можливість збору цінних відгуків та пропозицій, а також покращує комунікацію між аудиторією та організаторами культурних заходів.

3.2 Веб-дизайн та використання користувацького інтерфейсу

Веб-дизайн сайту культурних заходів повинен привертати увагу своєю естетикою, яскравими фотографіями, графічними ефектами та загальною візуальною привабливістю, що спонукає користувачів до подальшого дослідження сайту та участі в культурних подіях. Тому я вирішив використати яскраві фотографії та графічні елементи, щоб відтворити енергію та атмосферу культурних подій, що відбуваються в нашому місті. Навігація на сайті повинна бути логічно організована з легкодоступним меню та пошуковою панеллю, що допомагає користувачам швидко знайти потрібну інформацію. Розділи відповідають різним категоріям культурних заходів, що дозволяє легко орієнтуватися на сайті.

Інтерактивні елементи, такі як кнопки «Детальніше», слайдери з фотографіями та фільтри визначних місць, створюють динамічну взаємодію з користувачами.

Я хочу забезпечити незабутній користувацький досвід на сайті культурних заходів, де відвідувачі можуть знайти, досліджувати та взаємодіяти з різноманітними культурними подіями, що відбуваються в Івано-Франківську.

3.3 Розробка базового функціоналу веб-сайту

До базового функціоналу можна віднести навбар та футбар, адже вони реалізують навігацію та перехід до соціальних мереж і присутні на кожній сторінці.

Зовнішній вигляд навбару(рис. 3.1). Його програмна реалізація знаходиться в Додатку А.

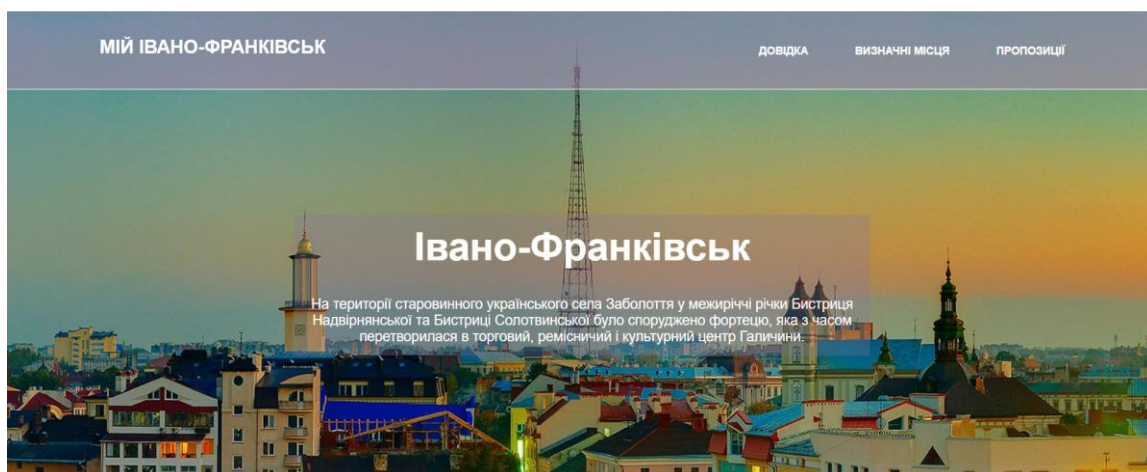


Рисунок 3.1 – Навбар головної сторінки

Верхня частина навбару містить логотип, який є візуальним ідентифікатором сайту. Логотип містить посилання на головну сторінку.

У навбарі є три кнопки меню, розташовані зверху. Кожна кнопка представляє окрему сторінку сайту. При наведенні курсору на кнопку, вона змінює колір, щоб показати активний стан.

Навбар має задній фон, на якому відображається фото Івано-Франківська. Це додає візуальний ефект і допомагає створити унікальний стиль для навбару.

Другий варіант навбару(рис. 3.2). Його програмна реалізація знаходиться в Додатку Б.

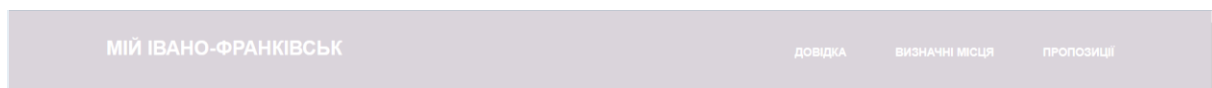


Рисунок 3.2 – Другий варіант навбару

Футбар є важливою частиною веб-сайту, яка знаходиться в нижній частині сторінки. Його зовнішній вигляд(рис. 3.3.). Його програмна реалізація знаходиться в Додатку В.



Рисунок 3.3 – Футбар

Футбар включає п'ять значків-посилань на соціальні мережі міста, що дозволяє відвідувачам легко знайти і зв'язатися з містом через різні платформи. Кожен значок представлена відповідною соціальною мережею і має візуальне відображення, що розпізнається користувачами.

Для однакового стилю значків я використовував сайт, що безкоштовно надає доступ до великої кількості зображень на будь-який смак для веб-розробників. Я підключив його за допомогою функції, застосування якої знаходиться в Додатку Г.

3.4 Відображення детальної інформації

На головній сторінці реалізована невелика функція, що викликає попап з детальною інформацією при натисканні на кнопку «Дізнатися більше». Зовнішній вигляд такого попапу(рис. 3.4.). Реалізація такого попапу в HTML описана в додатку Г.

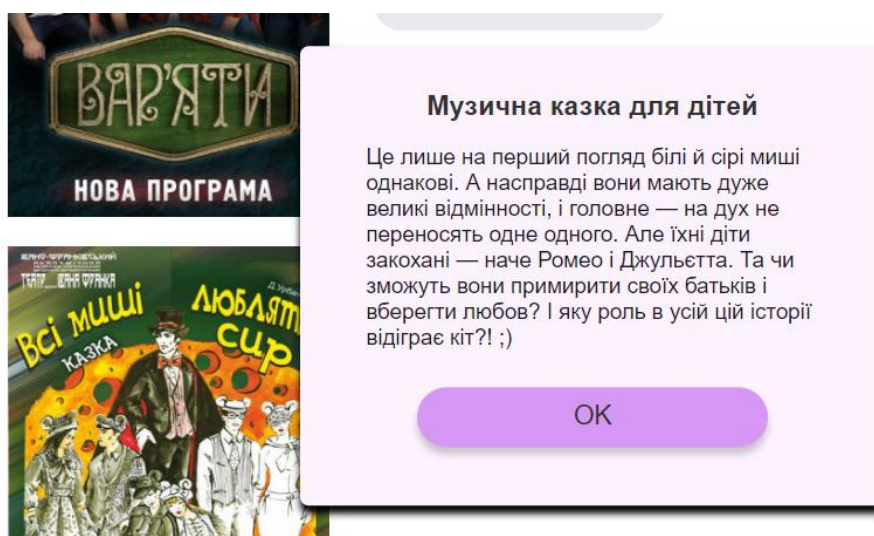


Рисунок 3.4 – Попап на головному вікні

А реалізація функції `openPopup()` та `closePopup()` в JS в додатку Д.

3.5 Компонент «інтерактивна галерея»

Компонент інтерактивної галереї (`Draggable Image Slider`) є цікавим

елементом на веб-сайті, який дозволяє відвідувачам переглядати зображення з використанням функції перетягування (drag) і листання (swipe). Зовнішній вигляд галереї на моєму сайті:



Рисунок 3.5 – Інтерактивна галерея без змін

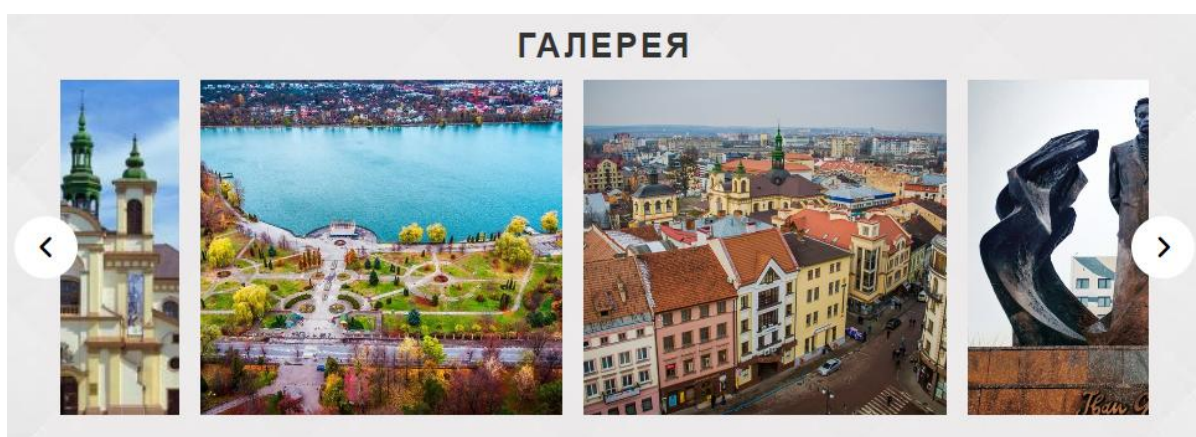


Рисунок 3.6 – «Перетягнуті» слайди

Програмна реалізація в HTML знаходиться в додатку Е.

Для відтворення функції перетягування мишею та переходу до наступної картинки потрібно реалізовувати функції за допомогою скриптів JavaScript, що показані в додатку Є:

З функції, що викликаються обробкою подій миші, а саме

- dragStart викликається при натисканні кнопки миші на компоненті "carousel". Вона ініціалізує початок перетягування, фіксує початкові координати миші та може виконувати додаткові дії, пов'язані з початком перетягування.

- `dragging` викликається під час руху миші по компоненту `"carousel"`. Вона слідкує за зміною координат миші та виконує відповідні дії, наприклад, зміщує елементи компоненту відповідно до руху миші.
- `dragStop` викликається при відпусканні кнопки миші на компоненті `"carousel"`. Вона завершує перетягування, фіксує кінцеві координати миші та може виконувати додаткові дії, пов'язані з завершенням перетягування.

В цьому скрипті функції `dragStart`, `dragging` і `dragStop` використовуються для реалізації механізму перетягування на компоненті `"carousel"`. Вони дозволяють відстежувати рух миші та зміщувати елементи компоненту відповідно до цього руху, створюючи ефект перетягування слайдів у галереї.

Далі скрипт, що додає обробку подій кліку на стрілочках у компоненті `"carousel"`.

`()=>{ }` – це стрілкова функція, яка викликається при кліку на стрілку.

`? -firstImgWidth: firstImgWidth` – це тернарний оператор, який вираховує зміщення галереї вліво або вправо. Якщо ідентифікатор значка дорівнює `"left"`, то відбувається зміщення вліво, в іншому випадку відбувається зміщення вправо.

Отже, цей скрипт реалізує функціонал перемикання слайдів у `"carousel"` за допомогою стрілок. Таким чином, виконується перехід між картинками в галереї.

3.6 Пошук та сортування

Реалізація пошуку та сортування на сайті має на меті полегшити користувачам знаходження необхідної інформації та покращити їх взаємодію з сайтом. Для реалізації пошуку визначних місць я підключив базу даних, та написав запит, що показано в додатку Ж:

Для реалізації функціоналу виводу блоків інформації я записав такий скрипт, який показано в додатку З.

Цей код проходить по всім рядкам з БД та на основі інформації формує `div`'и, надаючи їм необхідні класи для додавання редагування в CSS.

HTML файл з елементами фільтру та пошуку виглядає таким чином, як вказано в додатку І.

Далі розглянемо функцію фільтрації у додатку І.

Основна ідея полягає в тому, що при виклику функції з аргументом `value`, вона перебирає елементи на сторінці і застосовує фільтр до них.

Функція спочатку вибирає всі кнопки з класом `.button-value` і перевіряє, які з них мають значення, ідентичне переданому `value`. Якщо значення співпадає, то кнопці додається клас `active`, в іншому випадку він видаляється.

Потім функція вибирає всі елементи з класом `.atraction` і перевіряє, чи вони відповідають заданому значенню `value`. Якщо `value` дорівнює «all», то всі елементи з класом `.atraction` видаляють клас `hide`, що показує їх на сторінці. У протилежному випадку, якщо елемент має клас, що співпадає з `value`, то клас `hide` видаляється, і елемент відображається на сторінці. У іншому випадку, якщо елемент не має класу `value`, то йому додається клас `hide`, і він приховується.

Загалом, ця функція виконує фільтрацію елементів на сторінці засновану на заданому значенні `value` і змінює їх видимість відповідно до встановлених умов.

Наступною буде реалізація функції пошуку в додатку І.

Цей код встановлює лістелер на кнопку з `id` «search». При кліку на цю кнопку виконується функція-обробник.

У функції спочатку отримується значення введеного тексту з поля вводу з `id` «search-input». Значення зберігається у змінній `searchInput`.

Далі вибираються всі елементи з класом «product-name» і зберігаються в змінній `elements`. Також вибираються всі елементи з класом "atraction" і зберігаються в змінній `at`.

Потім за допомогою циклу `forEach` перебираються всі елементи `elements` і перевіряється, чи містить текст елемента підрядок, який збігається з `searchInput`.

Якщо такий підрядок знайдено, то відповідний елемент `at[index]` видаляє клас "hide", що робить його видимим на сторінці. У протилежному випадку, якщо підрядок не знайдено, елементу `at[index]` додається клас «hide», і він приховується.

Цей код реалізує пошук елементів на сторінці за введеним користувачем текстом та змінює їх видимість відповідно до результатів пошуку.

Тестування функцій наведених вище:

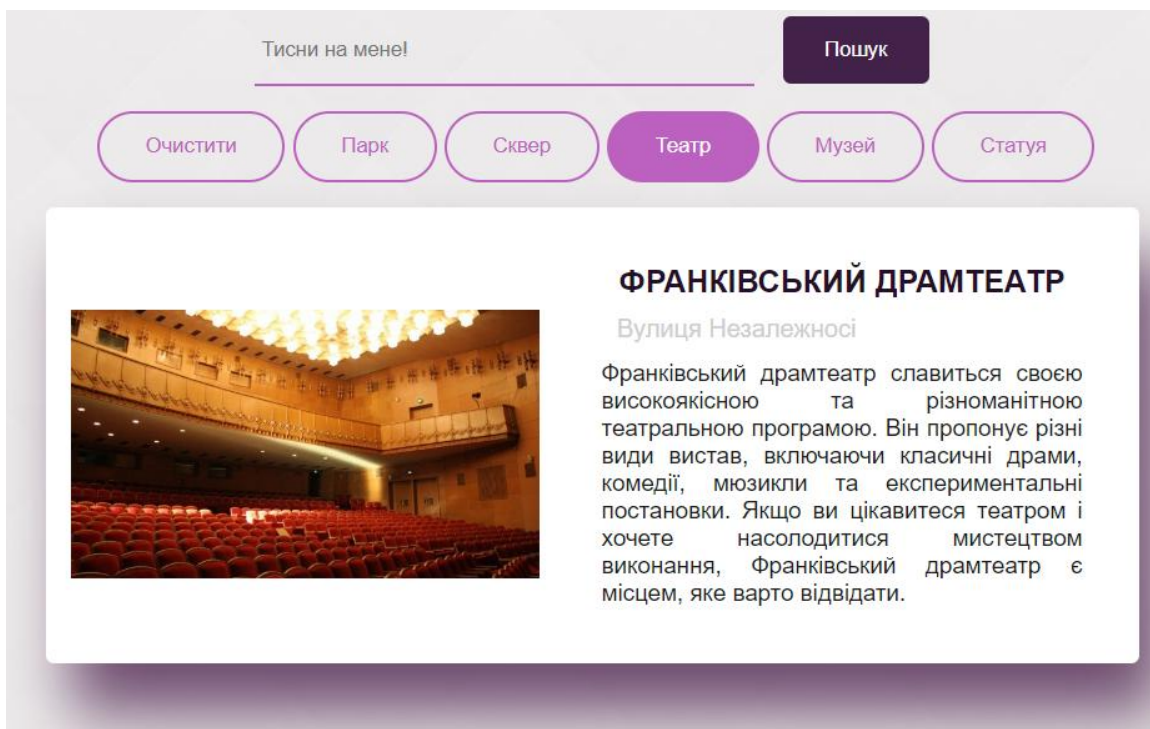


Рисунок 3.7 – Перевірка фільтру

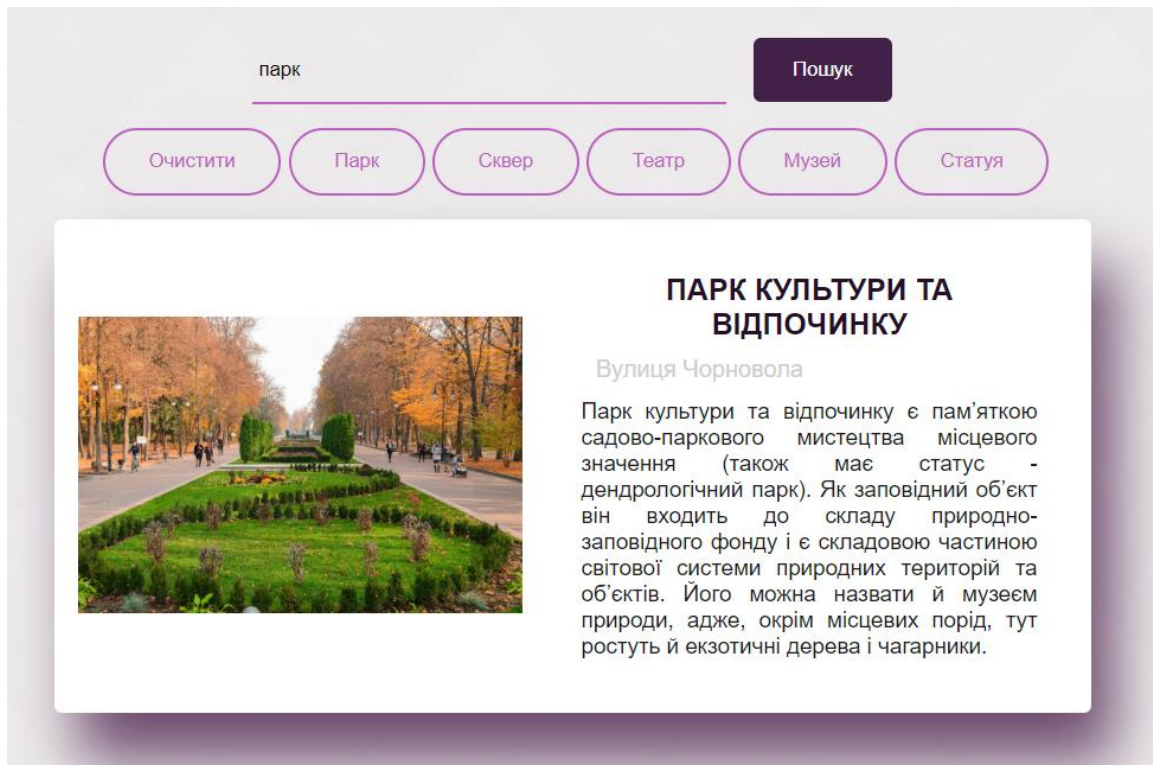


Рисунок 3.8 – Перевірка пошуку

Я реалізував зворотну форму шляхом відправки введених користувачем даних мені на пошту. Для цього я використав серверну частину на сайті, що відповідає за розсилку електронних листів SmtпJS.com. HTML код виглядає, як звичайна форма з полями та кнопкою, а ось на JS скрипт я б хотів звернути більше уваги знаходиться в додатку Й.

`const form = document.querySelector('.contact_form');` Знаходить елемент форми за допомогою селектора класу `.contact_form` і зберігає його в змінну `form`.

`function sendMsg(e) { ... }` Це функція, яка обробляє подію надсилання форми.

`e.preventDefault();` Виклик `preventDefault()` зупиняє стандартну відправку форми при натисканні на кнопку або натисканні клавіші `Enter`.

`const name = document.querySelector('.name');` Знаходить елемент із класом `.name` і зберігає його в змінну `name`. Аналогічно, знаходяться елементи з класами `.email`, `.t` та `.msg`.

`msgV = "Ім'я: " + name.value + " .Текст повідомлення: " + msg.value + " .`

Email: " + email.value; Формується текстове повідомлення msgV, яке включає значення полів.

Email.send({ ... }) - Виклик функції send з об'єктом параметрів, який містить дані для надсилання електронної пошти. Вказується відправник, отримувач, тема та текст повідомлення.

then(message => alert(message)); Після успішної відправки, виводиться повідомлення з результатом.

name.value = ""; email.value = ""; t.value = ""; msg.value = ""; Очищаються поля форми шляхом присвоєння пустих значень.

form.addEventListener('submit', sendMsg); Додається прослуховувач події submit до форми, який викликає функцію sendMsg при натисканні кнопки "Відправити" або натисканні клавіші Enter.

Пропозиція від користувача надійшла мені на пошту:

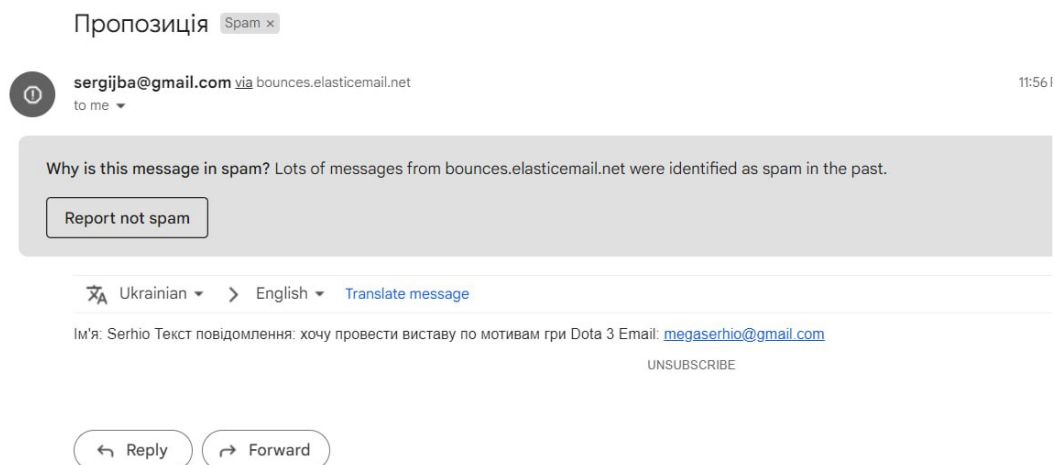


Рисунок 3.9 – Результат користування формою

3.7 Дизайн та користувацький інтерфейс

CSS є надзвичайно важливим інструментом для створення привабливого та функціонального веб-дизайну. Він визначає зовнішній вигляд та стиль веб-сторінок, включаючи кольори, шрифти, розміщення елементів, анімацію та багато іншого.

Одним з головних аспектів CSS є його роль у забезпеченні способу

форматування та розташування елементів на сторінці. Використання CSS дозволяє розташувати елементи точно на сторінці, задати їх розміри, відступи, межі та інші властивості, що забезпечує структурованість та зручність використання веб-сайту для користувачів.

CSS дозволяє створювати привабливий дизайн і візуальну привабливість веб-сторінок. За допомогою CSS можна задавати кольори, фонові зображення, тіні, анімацію, переходи та багато інших ефектів, що надають веб-сайту унікальність та привертають увагу користувачів.

Далі наведено деякі важливі приклади реалізації декількох компонентів дизайну користувацького інтерфейсу.

Поведінка та стиль іконок з посиланнями на соціальні мережі Івано-Франківську. Його реалізація знаходиться в додатку К.

Зовнішній вигляд іконок:



Рисунок 3.10 – У стані спокою

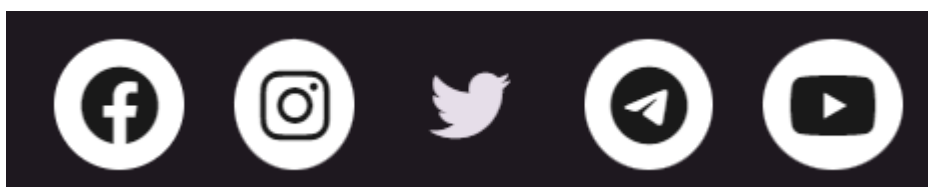


Рисунок 3.11 – При наведенні мишкою.

Меню в навігаційному полі запрограмоване схожим чином.

Отже, давайте розглянемо кожну з властивостей і їх призначення.

`.social_icons` – це клас, який застосовується до контейнера іконок соціальних мереж.

- `display: flex`; встановлює контейнеру властивість "flex", що дозволяє розташовувати елементи всередині нього в одному рядку.
- `justify-content: center`; вирівнює елементи по горизонталі по центру контейнера.

`.social_icons a` – це стиль для посилань, які містять значки соціальних мереж.

- `text-decoration: none`; вимикає підкреслення тексту посилання.
- `padding: 10px`; встановлює внутрішні відступи для посилань.
- `background-color: #fff`; встановлює білий колір фону для посилань
- `margin: 10px`; встановлює зовнішні відступи для посилань.
- `border-radius: 50%`; заокруглює кути посилань, щоб створити круглу форму.

`.social_icons a i` – це стиль для іконок в межах посилань.

- `font-size: 2em`; встановлює розмір шрифту для іконок.
- `color: #000000`; встановлює чорний колір тексту.
- `opacity: 0.9`; встановлює прозорість для іконок (0.9 означає 90% прозорості).

`.social_icons a:hover` – це стиль, який застосовується до посилань при наведенні курсору.

- `background-color: #1e181f`; змінює колір фону посилань на темно фіолетовий при наведенні.
- `transition: 0.5s`; встановлює плавний перехід для зміни стилів з анімацією тривалістю 0.5 секунди.

`.social_icons a:hover i` – це стиль, який застосовується до іконок при наведенні курсору.

- `color: #fcf3ff`; змінює колір тексту іконок на світло-фіолетовий при наведенні.
- `transition: 0.5s`; встановлює плавний перехід для зміни стилів з анімацією тривалістю 0.5 секунди.

Отже, цей CSS-код встановлює стилі для соціальних іконок, змінюючи їх

фон, кольори тексту та ефекти при наведенні курсору, щоб створити зручний та привабливий вигляд для відображення соціальних мереж на моєму веб-сайті.

Варто також звернути увагу на стилізацію галереї – слайдера фотографій Івано-Франківська. Реалізація знаходиться в додатку Л.

`.carousel_wrapper` – це клас, який застосовується до обгортки каруселі.

- `padding: 10px 10px`; встановлює внутрішні відступи для обгортки.
- `align-items: center`; вирівнює елементи по вертикалі по центру.
- `display: flex`; встановлює контейнеру властивість "flex", що дозволяє розташовувати елементи всередині нього в одному рядку.
- `max-width: 1200px`; встановлює максимальну ширину для обгортки каруселі.

`.carousel_wrapper i` – це стиль для стрілок (іконок) каруселі.

- `position: absolute`; встановлює позицію стрілок абсолютно відносно обгортки.
- `height: 45px`; встановлює висоту стрілок.
- `width: 45px`; встановлює ширину стрілок.
- `background: #fff`; встановлює білий колір фону для стрілок.
- `text-align: center`; вирівнює текст в центрі стрілок.
- `line-height: 45px`; встановлює висоту рядка для центрування тексту в стрілках.
- `border-radius: 50%`; заокруглює кути стрілок, щоб створити круглу форму.
- `cursor: pointer`; встановлює курсор вигляду "вказівник" для стрілок.
- `font-size: 1.2rem`; встановлює розмір шрифту для стрілок.

`.carousel_wrapper i:first-child` та `.carousel_wrapper i:last-child` – ці стилі встановлюють позицію лівої та правої стрілок відповідно.

`.carousel_wrapper .carousel` – це стиль для самої каруселі.

- `overflow: hidden`; обрізає зайві частини слайдів, щоб забезпечити відображення лише одного слайда одночасно.
- `font-size: 0px`; встановлює розмір шрифту 0, щоб усунути проміжки між слайдами.
- `cursor: pointer`; встановлює курсор вигляду "вказівник" для каруселі.
- `white-space: nowrap`; забороняє перенесення слайдів на новий рядок.

`.carousel img` – це стиль для зображень слайдів каруселі.

- `margin-left: 15px`; встановлює відступ зліва для зображень слайдів.
- `height: 240px`; встановлює висоту зображень слайдів.
- `object-fit: cover`; змінює розмір зображень таким чином, щоб вони повністю заповнили свої контейнери, зберігаючи при цьому свої пропорції.
- `width: calc(100%/3)`; встановлює ширину зображень, яка займає третину ширини контейнера.

`.carousel img:first-child` – це стиль, що встановлює нульовий відступ зліва для першого зображення слайда, щоб уникнути зайвих відступів.

CSS-код використовується для стилізації галереї на веб-сайті. Він встановлює властивості для обгортки каруселі, стрілок, зображень слайдів та самої каруселі, щоб створити бажаний вигляд та поведінку компонента.

Структура проекту складається з таких файлів: `index.html`, `about.html`, `attractions.html`, `contact.html`, `index.css`, `about.css`, `attractions.css`, `contact.css`, `script_about.js`, `script_attractions.js`.

Файли з розширенням HTML відповідають за структуру та вміст веб-сторінки. Вони використовуються для описування розмітки елементів на сторінці, таких як заголовки, параграфи, таблиці, форми та інші.

Файли з розширенням CSS відповідають за стилізацію веб-сторінки. Вони використовуються для задання зовнішнього вигляду елементів, таких як кольори, шрифти, розташування, розміри, анімації та інші візуальні ефекти.

Файли з розширенням JS (JavaScript) відповідають за програмування веб-сторінки. Вони використовуються для додавання інтерактивності, обробки подій, взаємодії з користувачем та здійснення різних операцій на веб-сторінці, таких як валідація форм, маніпуляції з DOM-елементами, виконання запитів на сервер та багато іншого.

ВИСНОВКИ

В дипломній роботі з веб-застосунку культурних заходів міста Івано-Франківськ була проведена важлива робота з розробки цифрової платформи, яка сприяє просуванню культурного туризму та покращенню інформаційного середовища міста.

Я зосередив свою увагу на зовнішньому вигляді сайту. Моєю ціллю було розробити таку інформаційну платформу, котра буде привертати увагу користувачів простотою і в той же час гармонічністю інтерфейсу.

На розробку та вдосконалення користувацького інтерфейсу була витрачена найбільша кількість роботи над дипломним проектом, тому наразі перед нами предстає зручний інтерфейс – легкий у використанні та швидкий у роботі, що є водночас інтуїтивним та мінімалістичним.

В цілому застосунок задовольняє потреби користувачів, забезпечуючи інформацію про події, їх розклад, розташування та опис. Користувачі мають змогу знайти місця для відпочинку, що відповідають їхнім інтересам, скористувавшись пошуком або фільтрацією. Також на сайті користувачі можуть насолодитися реалізованою галереєю захоплюючих краєвидів та визначних місць Івано-Франківська.

На сайті реалізована форма зворотнього зв'язку, що створює можливість для комунікації та взаємодії між адміністраторами сайту та організаторами заходів та простими користувачами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гунченко О. Веб-дизайн: засоби та методи: навчальний посібник, 2013, 368 стор.(date of access: 15.11.2022)
2. Мінін В. Веб-дизайн з нуля до професіонала: навчальний посібник, 2020, 512 стор.(date of access: 29.11.2022)
3. Хошелтон Р. Веб-дизайн: вступ до красивого та функціонального інтерфейсу: навчальний посібник, 2016, 432 стор.(date of access: 12.12.2022)
4. Драбик А. Веб-дизайн з елементами UX/UI: навчальний посібник, 2019, 320 стор.(date of access: 31.12.2022)
5. Постолов В. Дизайн інтерфейсів для веб-застосунків: навчальний посібник, 2018, 264 стор.(date of access: 24.01.2023)
6. Демчук М. Веб-дизайн для початківців: навчальний посібник, 2015, 304 стор.(date of access: 10.01.2023)
7. Нуñez В. та Болком Дж. Ілюстрований путівник з веб-дизайну: практичний посібник, 2019, 352 стор.(date of access: 24.01.2023)
8. Чумаров В. Веб-дизайн та веб-програмування: навчальний посібник, 2017, 304 стор.(date of access: 01.02.2023)
9. Овчаров О. Проектування користувальницького інтерфейсу: науково-практична книга, 2012, 380 стор.(date of access: 04.02.2023)
10. Мандрик Є. Графічний дизайн веб-сайтів: навчальний посібник, 2014, 328 стор.(date of access: 03.02.2023)
11. Офіційна документація на JavaScript <https://developer.mozilla.org/uk/docs/Web/JavaScript>(date of access: 26.02.2023)
12. Морган Н. JavaScript для дітей: навчальний посібник, 2018, 336 стор.(date of access: 25.02.2023)
13. Девід Ф. JavaScript. Підручник з програмування: практичний посібник, 2013, 1088 стор.(date of access: 09.03.2023)
14. Мінковський С. JavaScript: Повний курс: навчальний посібник, 2021, 580 стор.(date of access: 13.03.2023)
15. Дакетт Дж. JavaScript та jQuery: навчальний посібник, 2017, 464

стор.(date of access: 26.03.2023)

16. Флегг Д. JavaScript. Навчальний посібник: навчальний посібник, 2019, 672 стор.(date of access: 25.11.2022)

17. Чунг Л. JavaScript і HTML5: Пошукова оптимізація для веб-розробників: практичний посібник, 2013, 456 стор.(date of access: 02.01.2023)

18. Атанасіо Л. JavaScript: Завдання для розробників: практичний посібник, 2020, 354 стор.(date of access: 22.01.2023)

19. Івченко Є. JavaScript. Завдання та вправи: практичний посібник, 2017, 192 стор.(date of access: 22.01.2023)

20. Фріман Е. та Фріман Е. JavaScript: Пошагове навчання: навчальний посібник, 2012, 700 стор.(date of access: 24.01.2023)

21. Ульман Л. та Моррісон М. JavaScript і AJAX: Розробка веб-застосунків: практичний посібник, 2008, 840 стор.(date of access: 03.02.2023)

22. Офіційна документація на Node.js <https://nodejs.org/uk/docs/>(date of access: 04.02.2023)

23. Офіційна документація на Express <https://expressjs.com/uk/>(date of access: 11.12.2022)

24. Швартц М. Node.js та Express: Практичний вступ: практичний посібник, 2019, 492 стор.(date of access: 13.12.2022)

25. Шуфф Б. Node.js, Express і MongoDB. Розробка веб-застосунків: практичний посібник, 2019, 418 стор.(date of access: 13.12.2022)

26. Ульман Д. Node.js: Розробка веб-застосунків з Express та MongoDB: практичний посібник, 2018, 284 стор.(date of access: 23.12.2022)

27. Міллер Дж. Node.js та Express. Повний курс: навчальний посібник, 2020, 368 стор.(date of access: 15.02.2023)

28. Шарун В. Node.js: Вступ до розробки веб-застосунків: навчальний посібник, 2017, 330 стор.(date of access: 16.02.2023)

29. Офіційна документація на HTML <https://developer.mozilla.org/uk/docs/Web/HTML>(date of access: 16.02.2023)

30. Офіційна документація на CSS

<https://developer.mozilla.org/uk/docs/Web/CSS>(date of access: 19.02.2023)

31. Дакетт Д. HTML та CSS. Конструювання веб-сайтів: практичний посібник, 2018, 432 стор.(date of access: 26.02.2023)
32. Мензак І. Реляційні бази даних. Комп'ютерні системи: навчальний посібник, 2014, 270 стор.(date of access: 02.03.2023)
33. Кривко Д., Ковальова О., Кучеренко В. SQL: Навчальний посібник: навчальний посібник, 2019, 352 стор.(date of access: 09.03.2023)
34. Мейерс Д. та Ліберті Дж. HTML5 і CSS3. Веб-розробка з нуля: навчальний посібник, 2017, 576 стор.(date of access: 27.03.2023)
35. Хайнріх Н. та Кардінейл Р. HTML та CSS. Професійний підхід: практичний посібник, 2016, 528 стор.(date of access: 09.03.2023)
36. Дакетт Д. HTML, CSS та JavaScript. Веб-розробка для початківців: навчальний посібник, 2021, 816 стор.(date of access: 03.03.2023)
37. Хоган Б. П. HTML5 та CSS3. Пошукова оптимізація для веб-розробників: практичний посібник, 2016, 432 стор.(date of access: 04.03.2023)
38. Нідам-Вайнер Дж. HTML та CSS. Комп'ютерна графіка: навчальний посібник, 2013, 400 стор.(date of access: 14.03.2023)
39. Шмедтманн Й. HTML та CSS. Розробка і дизайн веб-сайтів: практичний посібник, 2015, 482 стор.(date of access: 06.02.2023)
40. Хоган П. Б. HTML5 та CSS3. Сучасна веб-розробка: практичний посібник, 2014, 500 стор.(date of access: 23.02.2023)
41. Джаяретна Й. К. Бази даних: Повний курс: навчальний посібник, 2019, 428 стор.(date of access: 24.02.2023)
42. Колодій А. та Бондар В. Бази даних: Практикум: навчальний посібник, 2016, 336 стор.(date of access: 24.02.2023)
43. Бурунов С. Реляційні бази даних: Теорія та практика: навчальний посібник, 2021, 288 стор.(date of access: 23.02.2023)
44. Скрипник В. та Діденко І. Реляційні бази даних: Навчальний курс: навчальний посібник, 2017, 352 стор.(date of access: 26.02.2023)
45. Шевченко О. Бази даних: Практичний підхід: практичний посібник,

2018, 192 стор.(date of access: 01.03.2023)

46. Кобзар А. Бази даних: Навчальний посібник: навчальний посібник,
2018, 376 стор.(date of access: 05.03.2023)

ДОДАТКИ

Додаток А

```

14     <section>
15         <nav>
16             <div class="logo"><a href="/">Мій Івано-Франківськ</a></div>
17             <ul>
18                 <li><a href="about.html">Довідка</a></li>
19                 <li><a href="atraction.ejs">Визначні місця</a></li>
20                 <li><a href="contact.html">Пропозиції</a></li>
21             </ul>
22         </nav>
23         <div class="info_if">
24             <h1>Івано-Франківськ </h1>
25             <br>
26             <p>На території старовинного українського села Заболоття у
27             межиріччі річки Бистриця Надвірнянської та Бистриці Солотвинської
28             було споруджено фортецю, яка з часом перетворилася в торговий, ремісничий
29             і культурний центр Галичини.</p>
30         </div>
31     </section>
32

```

Додаток Б

```

16     <section>
17         <nav>
18             <div class="logo"><a href="/">Мій Івано-Франківськ</a></div>
19             <ul>
20                 <li><a href="about.html">Довідка</a></li>
21                 <li><a href="atraction.html">Визначні місця</a></li>
22                 <li><a href="contact.html">Пропозиції</a></li>
23             </ul>
24         </nav>
25         <div class="info_if">
26             <h1>Івано-Франківськ </h1>
27             <br>
28             <p>Молоде європейське місто, одне з відомих економічних,
29             культурних та архітектурних центрів Західної України.</p>
30         </div>
31     </section>
32

```

Додаток В

```

71     <footer>
72         <div class="footer_container">
73             <div class="social_icons">
74                 <a href="https://www.facebook.com/news.ivanofrankivsk/?locale=ru_RU"><i class="fa-brands fa-facebook"></i></a>
75                 <a href="https://www.instagram.com/franik_ua/?hl=ru"><i class="fa-brands fa-instagram"></i></a>
76                 <a href="https://twitter.com/ifcityua"><i class="fa-brands fa-twitter"></i></a>
77                 <a href="https://t.me/s/IF112"><i class="fa-brands fa-telegram"></i></a>
78                 <a href="https://www.youtube.com/channel/UCZBA00e1ECmizjмNh4Xhwg/videos"><i class="fa-brands fa-youtube"></i></a>
79             </div>
80         </div>
81     </footer>

```

Додаток Г

```

9 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
10 integrity="sha512-iecdLmaskl7CVkqkXNO/ZH/XLlvwZOJyj7Yy7tcenmpD1ypASozpmT/E01PtmFIB46ZmdtAc9eNBVH0H/Zpibw==
11 crossorigin="anonymous" referrerpolicy="no-referrer" />

```

Додаток Г

```

<button onclick="openPopup(popup2)">Дізнатися більше</button>
<div class="popup" id="popup2">
  <h3>Один день у гуцулів</h3>
  <p><b>Косів</b> здавна називають столицею народного мистецтва</p>
  <p>У <b>садибі Святого Миколая</b> можна відвідати кімнату-му</p>
  <p>Річка Пістинька перетинає потужну скельну товщу, утворюючи</p>
  <p>По дорозі зупинка в с. Город, підйом на <b>Гору</b> Остра, </p>
  <p><b>Яворів</b>- унікальне гуцульське село з давніми славни</p>
  <button onclick="closePopup(popup2)">ОК</button>
</div>

```

Додаток Д

```

function openPopup(p){
  p.classList.add("open-popup");
}
function closePopup(p){
  p.classList.remove("open-popup");
}

```

Додаток Е

```
<h2>Галерея</h2>

<div class="carousel_wrapper">
  <i class="fa-solid fa-angle-left" id="left"></i>
  <div class="carousel">
    
    
    
    
    
    
    
    
    
    
  </div>
  <i class="fa-solid fa-angle-right" id="right"></i>
</div>
```

Додаток Є

```

const carousel = document.querySelector(".carousel");
firstImg = carousel.querySelector("img")[0];
arrowIcons = document.querySelectorAll(".carousel_wrapper i")

let isDragStart = false, prevPageX, prevScrollLeft;
let firstImgWidth = firstImg.clientWidth + 15;

arrowIcons.forEach(icon =>{
  icon.addEventListener("click", ()=>{
    carousel.scrollLeft += icon.id == "left" ? -firstImgWidth: firstImgWidth;
  });
});

const dragStart = (e) => {
  isDragStart = true;
  prevPageX = e.pageX;
  prevScrollLeft = carousel.scrollLeft;
}

const dragging = (e) => {
  if (!isDragStart) return;
  e.preventDefault();
  let positionDiff = e.pageX - prevPageX;
  carousel.scrollLeft = prevScrollLeft - positionDiff;
}

const dragStop = () =>{
  isDragStart = false;
}

carousel.addEventListener("mousedown", dragStart);
carousel.addEventListener("mousemove", dragging);
carousel.addEventListener("mouseup", dragStop);

```

Додаток Ж

```

12  const conn = mysql.createConnection({
13    host: "localhost",
14    user: "root",
15    database: "if",
16    password: ""
17  });
18  let query = "SELECT * FROM `attractions`";
19
20  app.get('/attraction.ejs', (req, res) => {
21    conn.query(query, (error, results) => {
22      if (error) throw error;
23      console.log
24      res.render('attraction.ejs', { data: results });
25    });
26  });

```

Додаток 3

```
184     for (let i of attractions.data) {
185
186         let attraction = document.createElement("div");
187         attraction.classList.add("attraction", i.type, "hide");
188
189         let imgContainer = document.createElement("div");
190         imgContainer.classList.add("img_container");
191
192         let img = document.createElement("img");
193         let src_img = "images/db/" + i.img;
194         img.setAttribute("src", src_img);
195         imgContainer.appendChild(img);
196         attraction.appendChild(imgContainer);
197
198         let infoContainer = document.createElement("div");
199         infoContainer.classList.add("info_container");
200
201         let name = document.createElement("h5");
202         name.classList.add("product-name");
203         name.innerText = i.name.toUpperCase();
204         infoContainer.appendChild(name);
205
206         let location = document.createElement("p");
207         location.innerText = i.location;
208         infoContainer.appendChild(location);
209
210
211         let discription = document.createElement("h6");
212         discription.innerText = i.discription;
213         infoContainer.appendChild(discription);
214         attraction.appendChild(infoContainer);
215     }
```

Додаток И

```

<div class="wrapper">
  <div id="search-container">
    <input type="search" id="search-input" placeholder="Тисни на мене!"/>
    <button id="search">Пошук</button>
  </div>
  <div id="buttons">
    <button class="button-value" onclick="filterProduct('all')">ОЧИСТИТИ</button>
    <button class="button-value" onclick="filterProduct('Парк')">
      Парк
    </button>
    <button class="button-value" onclick="filterProduct('Сквер')">
      Сквер
    </button>
    <button class="button-value" onclick="filterProduct('Театр')">
      Театр
    </button>
    <button class="button-value" onclick="filterProduct('Музей')">
      Музей
    </button>
    <button class="button-value" onclick="filterProduct('Статуя')">
      Статуя
    </button>
  </div>
</div id="atractions"></div>

```

Додаток І.

```

function filterProduct(value) {
  let buttons = document.querySelectorAll(".button-value");
  buttons.forEach((button) => {
    if (value.toUpperCase() == button.innerText.toUpperCase()) {
      button.classList.add("active");
    } else {
      button.classList.remove("active");
    }
  });
  let elements = document.querySelectorAll(".attraction");
  elements.forEach((element) => {
    if (value == "all") {
      element.classList.remove("hide");
    } else {
      if (element.classList.contains(value)) {
        element.classList.remove("hide");
      } else {
        element.classList.add("hide");
      }
    }
  });
}

```

Додаток І

```

document.getElementById("search").addEventListener("click", () => {
  let searchInput = document.getElementById("search-input").value;
  let elements = document.querySelectorAll(".product-name");
  let at = document.querySelectorAll(".atraction");
  elements.forEach((element, index) => {
    if (element.innerText.includes(searchInput.toUpperCase())) {
      at[index].classList.remove("hide");
    } else {
      at[index].classList.add("hide");
    }
  });
});

```

Додаток Й

```

const form = document.querySelector('.contact_form');

function sendMsg(e){
  e.preventDefault();

  const name = document.querySelector('.name');
  const email = document.querySelector('.email');
  console.log(email.value)
  const t = document.querySelector('.t');
  const msg = document.querySelector('.msg');
  msgV = "Ім'я: " + name.value + " .Текст повідомлення: "+ msg.value + " . Email: " + email.value;
  console.log(msgV)
  Email.send({
    Host : "smtp.elasticemail.com",
    Username : "sergijba@gmail.com",
    Password : "884BD0215545699A65EA06C9124CAF47687A",
    To : 'sergijba@gmail.com',
    From : 'sergijba@gmail.com',
    Subject : t.value,
    Body : msgV
  }).then(
    message => alert(message)
  );
  name.value = '';
  email.value = '';
  t.value = '';
  msg.value = '';
}

form.addEventListener('submit', sendMsg);

```

Додаток К

```
189 footer{
190     background-color: #1e181f;
191 }
192 .footer_container{
193     width: 100%;
194     padding: 40px 30px 20px;
195 }
196 .social_icons{
197     display: flex;
198     justify-content: center;
199 }
200 .social_icons a{
201     text-decoration: none;
202     padding: 10px;
203     background-color: #fff;
204     margin: 10px;
205     border-radius: 50%;
206 }
207
208 .social_icons a i{
209     font-size: 2em;
210     color: #000000;
211     opacity: 0.9;
212 }
213
214 .social_icons a:hover{
215
216     background-color: #1e181f;
217     transition: 0.5s;
218 }
219 .social_icons a:hover i{
220     color: #fcf3ff;
221     transition: 0.5s;
222 }
223
```

Додаток Л

```
117 .carousel_wrapper{
118     padding: 10px 10px;
119     align-items: center;
120     display: flex;
121     max-width: 1200px;
122 }
123 .carousel_wrapper i{
124     position: absolute;
125     height: 45px;
126     width: 45px;
127     background: #fff;
128     text-align: center;
129     line-height: 45px;
130     border-radius: 50%;
131     cursor: pointer;
132     font-size: 1.2rem;
133 }
134 .carousel_wrapper i:first-child{
135     left: -23px;
136 }
137 .carousel_wrapper i:last-child{
138     right: -23px;
139 }
140 .carousel_wrapper .carousel{
141     overflow: hidden;
142     font-size: 0px;
143     cursor: pointer;
144     white-space: nowrap;
145 }
146 .carousel img{
147     margin-left: 15px;
148     height: 240px;
149     object-fit: cover;
150     width: calc(100%/3);
151 }
152 .carousel img:first-child{
153     margin-left: 0px;
154 }
```