

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ

Факультет дизайну
Кафедра мультимедійного дизайну

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка дизайну веб-сайту для музичної майстерні
Рівень вищої освіти перший (бакалаврський)
Спеціальність 022 Дизайн
Освітня програма Мультимедійний дизайн

Виконала: студентка групи БЗДм-21

Болгова Д.П.

Науковий керівник к.пед.н., доцент

Шаура А.Ю.

Рецензент к.т.н., доцент Хиневич Р.В.

Київ 2025

АНОТАЦІЯ

Болгова Д.П. Розробка дизайну веб-сайту для музичної майстерні. – Рукопис.

Кваліфікаційна робота за спеціальністю 022 Дизайн – Київський національний університет технологій та дизайну, Київ, 2025 рік.

У кваліфікаційній роботі надані результати розробки дизайну веб-сайту для музичної майстерні. В ході роботи було проведено аналіз предметної області, визначено основні вимоги до функціоналу та зовнішнього вигляду сайту. Розроблено макети інтерфейсу, підібрано відповідну кольорову гаму та шрифти, що відповідають тематиці та стилю музичної майстерні. На основі створеного дизайну було реалізовано повнофункціональний веб-сайт із сучасним адаптивним інтерфейсом, який забезпечує зручну навігацію для користувачів та ефективну презентацію послуг музичної майстерні.

Ключові слова: веб-дизайн, музична майстерня, музичні інструменти, адаптивний інтерфейс, розробка сайту, UX/UI.

SUMMARY

Bolghova D.P. Development of a website design for a music workshop. – The manuscript.

Qualification work in the specialty 022 Design – Kyiv National University of Technology and Design, Kyiv, 2025.

The qualification project presents the results of the development of a website design for a music workshop. During the project, an analysis of the subject area was conducted to define key functional and visual requirements for the website. Interface layouts were created, with careful selection of color schemes and fonts that align with the theme and style of the music workshop. Based on the developed design, a fully functional website with a modern, responsive interface was implemented, ensuring convenient navigation for users and effective presentation of the workshop's services.

Keywords: web design, music workshop, musical instruments, responsive interface, website development, UX/UI.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-САЙТУ ДЛЯ МУЗИЧНОЇ МАЙСТЕРНІ	9
1.1 Характеристика предметної області та її структурно-функціональних аспектів	9
1.2 Теоретичні основи розробки веб-сайту для музичних організацій	13
1.3 Проведення аналізу веб-сайтів конкурентів.....	24
Висновки до 1 розділу.....	28
РОЗДІЛ 2 АЛГОРИТМ СТВОРЕННЯ ВЕБ-САЙТУ ДЛЯ МУЗИЧНОЇ МАЙСТЕРНІ	30
2.1 Аналіз цільової аудиторії з метою розробки ефективного дизайну веб- сайту.....	30
2.2 Проектування об'єктів та зв'язків між ними.....	33
2.3 Вибір програмного забезпечення для розробки веб-сайту музичної майстерні.....	35
Висновки до 2 розділу.....	40
РОЗДІЛ 3 РОЗРОБКА ДИЗАЙНУ ВЕБ-САЙТУ ДЛЯ МУЗИЧНОЇ МАЙСТЕРНІ	42
3.1 Створення дизайну та прототипів сторінок веб-сайту музичної майстерні	42
3.2 Розробка та тестування веб-сайту музичної майстерні	46
3.3 Вимоги до технічного забезпечення та розгортання сайту музичної майстерні.....	52
Висновки до 3 розділу.....	56
ЗАГАЛЬНІ ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:	61
ДОДАТКИ.....	65

ВСТУП

В умовах динамічного розвитку інформаційних технологій та цифровізації бізнес-процесів наявність ефективного веб-представництва стає критичним фактором конкурентоспроможності для підприємств різних галузей. Особливо гостро ця проблема постає для спеціалізованих малих підприємств, зокрема музичних майстерень, які прагнуть розширити клієнтську базу та оптимізувати комунікацію з цільовою аудиторією. Створення функціонального веб-сайту для музичної майстерні дозволяє не лише презентувати послуги та продукцію, але й організувати ефективну взаємодію з клієнтами, автоматизувати бізнес-процеси та підвищити впізнаваність бренду.

Актуальність теми дослідження зумовлена зростаючою потребою музичних майстерень у цифровій присутності на ринку, що пов'язано з трансформацією споживчої поведінки та переходом значної частини комунікацій у онлайн-середовище. Згідно з дослідженням 1WorldSync, 87% споживачів використовують смартфони для пошуку інформації про продукти під час перебування в магазинах [1], що свідчить про важливість цифрових каналів у процесі прийняття рішень про покупку. Статистичні дані свідчать, що понад 80% споживачів регулярно шукають інформацію про місцеві підприємства в Інтернеті, а 61% використовують бізнес-інформаційні сайти для пошуку нових або незнайомих їм компаній [2]. Наявність функціонального веб-сайту підвищує довіру до підприємства на 62%, оскільки більшість споживачів ігнорують бізнеси без онлайн-присутності [3]. При цьому специфіка роботи музичних майстерень, що поєднує виробництво, ремонт, реставрацію інструментів та освітні послуги, вимагає створення спеціалізованих веб-рішень, які враховують унікальні бізнес-процеси та потреби таких підприємств.

Теоретичну основу дослідження складають праці вітчизняних науковців, зокрема Петренка В.М., Коваленко О.П., Дмитренко К.В., які досліджували особливості розробки веб-сайтів для малого бізнесу [4; 5; 6], а також зарубіжних дослідників – Нільсена Я., Круга С., Гарретта Дж., що розробили фундаментальні

принципи юзабіліті та користувацького досвіду [7; 8; 9]. Практичні аспекти розробки веб-рішень для творчих індустрій висвітлені у працях Шевченко Л.І., Морозової Т.О., а також Андерсона К. та Браун Д. [10; 11; 12; 13]

Вихідними даними для розробки теми є аналіз існуючих веб-рішень музичних майстерень, результати опитування потенційних користувачів, технічна документація сучасних веб-технологій та фреймворків, а також бізнес-вимоги конкретної музичної майстерні, для якої розробляється веб-сайт.

Метою дослідження є розробка функціонального та ергономічного веб-сайту для музичної майстерні, що забезпечить ефективну взаємодію з клієнтами, презентацію послуг та продукції, а також оптимізацію бізнес-процесів підприємства.

Завдання дослідження:

- Провести аналіз потреб та особливостей функціонування музичних майстерень в контексті їх представлення у веб-середовищі.
- Дослідити існуючі технології та підходи до розробки веб-сайтів для підприємств творчої індустрії.
- Розробити архітектуру та функціональну структуру веб-сайту для музичної майстерні.
- Реалізувати інтерфейсну частину веб-сайту з дотриманням принципів юзабіліті та доступності.
- Розробити серверну частину веб-сайту з організацією необхідних бізнес-процесів.
- Провести тестування та впровадження розробленого рішення.
- Оцінити ефективність розробленого веб-сайту за визначеними критеріями.

Об'єктом дослідження є процес цифровізації діяльності музичної майстерні та оптимізація її взаємодії з клієнтами.

Предметом дослідження є методи та технології розробки веб-сайту для музичної майстерні з урахуванням специфіки галузі та сучасних вимог до веб-рішень.

У процесі дослідження використані **методи** системного аналізу для визначення вимог до веб-сайту, методи об'єктно-орієнтованого проектування при розробці архітектури системи, методи юзабіліті-тестування для оцінки ергономічності інтерфейсу, а також методи статистичного аналізу при обробці результатів тестування системи.

Елементами наукової новизни дослідження є:

- розроблена модель структурування контенту для веб-сайтів музичних майстерень, що враховує специфіку представлення музичних інструментів та послуг;
- удосконалений метод інтеграції мультимедійних елементів для демонстрації звучання музичних інструментів;
- запропонований комплексний підхід до організації онлайн-комунікації між майстрами та клієнтами з урахуванням специфіки галузі.

Практична цінність отриманих результатів полягає у створенні функціонального веб-сайту, що дозволяє музичній майстерні ефективно представити свої послуги, налагодити комунікацію з клієнтами, автоматизувати процеси замовлення та консультацій, а також збільшити впізнаваність бренду. Розроблені в ході дослідження методи та підходи можуть бути використані при створенні веб-рішень для інших підприємств творчої індустрії.

Наукова новизна. На відміну від існуючих вітчизняних розробок, запропоноване рішення використовує сучасні технології та підходи до веб-розробки, поєднуючи унікальний дизайн та широку функціональність. У порівнянні з зарубіжними аналогами, розроблений веб-сайт адаптований до особливостей вітчизняного ринку музичних інструментів та послуг, включає локалізовані компоненти та оптимізований для роботи на різних пристроях з урахуванням особливостей місцевої інтернет-інфраструктури.

Апробація результатів дослідження. Участь та публікація в збірнику матеріалів Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем» (15 травня 2025 Житомир) (Додаток А, Б). Творчу роботу – розробку веб-сайту для

музичної майстерні було відправлено на конкурс Better Design Award (BDA) «Future Design» (м. Шеньчжень, Китай 2025) (Додаток В).

Структура роботи зумовлена логікою дослідження і складається із вступу, трьох розділів, висновків до кожного розділу, загальних висновків, списку використаних джерел (усього 40 джерел). Загальний обсяг роботи становить сторінок, з них основного тексту сторінок.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБ-САЙТУ ДЛЯ МУЗИЧНОЇ МАЙСТЕРНІ

1.1 Характеристика предметної області та її структурно-функціональних аспектів

Предметна область охоплює декілька взаємопов'язаних сфер та концепцій. Музичні майстерні — це спеціалізовані підприємства, які займаються виготовленням, ремонтом, реставрацією та налаштуванням музичних інструментів. В умовах цифрової трансформації бізнесу наявність власного веб-сайту стає необхідним елементом конкурентоспроможності таких підприємств [14].

Музичні майстерні поєднують у собі елементи виробництва та сфери послуг. Основними напрямками їх діяльності є виготовлення інструментів, що передбачає створення нових музичних інструментів за класичними або авторськими технологіями. Цей процес вимагає високої кваліфікації майстрів та використання якісних матеріалів. Крім того, майстерні займаються ремонтом та реставрацією інструментів, що включає відновлення їх функціональності, заміну деталей та усунення дефектів звучання.

Налаштування інструментів є ще однією важливою послугою, особливо для фортепіано, струнних та духових інструментів, адже правильне налаштування забезпечує їх бездоганне звучання. Консультаційні послуги допомагають клієнтам отримати професійні рекомендації щодо вибору, догляду та експлуатації інструментів. Окрім цього, майстерні часто займаються продажем супутніх товарів, таких як струни, смички, медіатори, чохли та інші аксесуари для музичних інструментів [15]. Специфіка цієї сфери полягає в поєднанні традиційного ремесла з сучасними технологіями, що вимагає від майстрів постійного вдосконалення своїх навичок та знань.

Клієнтами музичних майстерень є різні категорії споживачів. Це можуть бути професійні музиканти, які потребують регулярного обслуговування, налаштування та іноді ремонту інструментів високої якості. Музичні навчальні заклади — це школи, коледжі, консерваторії, що мають парк інструментів, які потребують систематичного догляду. Також початківці та аматори, що шукають доступні інструменти, часто потребують консультацій щодо вибору та догляду. Іноді колекціонери цікавляться рідкісними, антикварними або унікальними інструментами, часто потребують реставраційних послуг. Концертні зали та студії звукозапису потребують налаштування та обслуговування інструментів перед виступами або сесіями запису.

Сфера музичних інструментів та обслуговування зазнає суттєвих змін під впливом технологічного прогресу та зміни споживчих звичок. Через цифровізацію зростає попит на електронні та цифрові музичні інструменти, що поєднують традиційне звучання з сучасними технологіями. Персоналізація — збільшується кількість замовлень на виготовлення інструментів із індивідуальними характеристиками. Екологічність — підвищується увага до використання сталих матеріалів та екологічних технологій виробництва. Онлайн-присутність — клієнти все частіше шукають послуги музичних майстерень через інтернет, що робить веб-сайт необхідним інструментом комунікації та просування [16].

Веб-сайт стає ключовим елементом бізнес-стратегії музичної майстерні, виконуючи ряд важливих функцій:

- Інформаційна функція — надання повної інформації про послуги, ціни, графік роботи, контакти та місцезнаходження майстерні.
- Комунікаційна функція — забезпечення зручного каналу зв'язку з клієнтами через форми зворотного зв'язку, онлайн-чати та системи запису на обслуговування.
- Демонстраційна функція — представлення портфоліо виконаних робіт, фотографій майстерні та відгуків клієнтів для формування довіри.
- Освітня функція — розміщення корисної інформації про догляд за

інструментами, історію музичних інструментів, технології виготовлення тощо.

- Комерційна функція — можливість онлайн-замовлення послуг, бронювання часу для консультацій, продаж супутніх товарів.

Розробка веб-сайту для музичної майстерні має враховувати специфіку галузі та потреби цільової аудиторії. Архітектура сайту повинна забезпечити зручну навігацію та легкий доступ до основної інформації, що дозволить користувачам швидко знаходити потрібні відомості. Візуальний дизайн повинен відображати естетику музичного мистецтва, створюючи атмосферу творчості та майстерності, що підкреслює професіоналізм майстерні. Мультимедійні елементи, зокрема відео- та аудіоматеріали, можуть демонструвати процес роботи та результати реставрації інструментів, а також якість їх звучання.

Веб-сайт має бути адаптивним, щоб коректно відображатися на різних пристроях, що є особливо важливим для музикантів, які часто користуються мобільними телефонами та планшетами. Інтеграція з соціальними мережами дозволяє розширити канали комунікації з потенційними клієнтами та підвищити видимість сайту в Інтернеті. Важливою складовою є використання систем управління контентом (CMS), що дозволяє оперативно оновлювати інформацію без необхідності залучати програмістів. Також веб-сайт повинен мати можливості для масштабування, що дозволяє додавати нові функції в майбутньому, такі як онлайн-магазин, блог або форум.

У сучасних умовах високої конкуренції та зростаючих вимог користувачів, веб-сайт музичної майстерні повинен відповідати низці важливих критеріїв. Одним із ключових аспектів є швидкість завантаження, адже оптимізована продуктивність сайту дозволяє мінімізувати час очікування користувачів. Зручність використання є також критичним чинником; інтерфейс повинен бути інтуїтивно зрозумілим навіть для користувачів з різним рівнем технічної підготовки. Не менш важливою є безпека даних, зокрема захист особистої інформації клієнтів та забезпечення безпечної обробки онлайн-платежів. Для покращення видимості в Інтернеті веб-сайт повинен бути SEO-оптимізованим

[17], що включає правильну структуру та контент для високого ранжування в пошукових системах. Основні складові SEO-оптимізації відображено на рисунку 1.1.



Рис. 1.1 – Основні складові SEO-оптимізації

Інтеграція з CRM-системами дозволяє ефективно управляти взаємовідносинами з клієнтами, що значно підвищує рівень обслуговування. Крім того, використання аналітичних інструментів дає змогу відстежувати ефективність роботи сайту та аналізувати поведінку його відвідувачів, що дозволяє постійно покращувати користувацький досвід.

Таким чином, розробка веб-сайту для музичної майстерні вимагає комплексного підходу, що враховує специфіку галузі, потреби цільової аудиторії та сучасні технологічні можливості. Якісний веб-ресурс стає не лише інструментом інформування та залучення клієнтів, але й відображенням професіоналізму та унікальності послуг музичної майстерні в цифровому середовищі.

1.2 Теоретичні основи розробки веб-сайту для музичних організацій

Розробка веб-сайту для музичної майстерні передбачає створення зручного та функціонального інтерфейсу для користувачів, який забезпечить легкий доступ до послуг та інформації про майстерню. Алгоритм вирішення задачі включає кілька етапів, що охоплюють проектування, розробку, тестування та впровадження сайту.

На першому етапі важливо визначити основні вимоги до сайту, що включають кілька важливих аспектів, а саме: типи послуг, які надає музична майстерня (уроки музики, оренда інструментів, ремонт інструментів тощо). Важливо дослідити цільову аудиторію (учні, музиканти, любителі) для розуміння та розміщення структури інформації, яка буде представлена на сайті (наприклад, сторінки про послуги, контакти, відгуки, новини). Варто врахувати особливості інтерактивних функцій – це і реєстрація користувачів, і бронювання послуг, форма зворотного зв'язку тощо.

На основі зібраних вимог розробляється архітектура сайту. Це включає в себе створення карти сайту для визначення основних розділів та підрозділів, визначення структури бази даних для зберігання інформації про користувачів, послуги, відгуки, тощо. Проектування інтерфейсу користувача (UI), з урахуванням принципів зручності та доступності та планування системи управління контентом (CMS) для легкого оновлення інформації.

Фронтенд сайту розробляється з використанням сучасних веб-технологій, таких як HTML, CSS, JavaScript. Алгоритм розробки передбачає створення адаптивного дизайну для коректного відображення сайту на різних пристроях. Використання фреймворків, таких як Bootstrap, для пришвидшення процесу розробки та забезпечення єдиного стилю. Реалізація інтерактивних елементів таких, як форми зворотного зв'язку, онлайн-бронювання за допомогою JavaScript. Інтеграція з бібліотеками для візуалізації, наприклад, для відображення графіків або плану занять.

Серверна частина сайту розробляється на базі платформи Node.js з

використанням Express.js для обробки HTTP-запитів та взаємодії з базою даних MongoDB. Алгоритм включає реалізацію API для взаємодії з базою даних, забезпечення функціональності реєстрації та авторизації користувачів. Створення моделей для зберігання даних про послуги, користувачів, бронювання. Наступним кроком є реалізація механізмів захисту даних за допомогою JWT для авторизації та захисту сесій, а також організація зберігання та обробки зображень або відео, що будуть використовуватись на сайті.

Для зберігання інформації про користувачів, послуги, відгуки та інші дані використовуватиметься база даних MongoDB. Алгоритм інтеграції включає створення колекцій для кожного типу даних (наприклад, користувачі, послуги, бронювання); налаштування CRUD операцій для взаємодії з даними через сервер (створення, читання, оновлення, видалення); реалізація механізмів перевірки валідності введених даних перед їх збереженням.

Після розробки основних функцій проводиться тестування всіх компонентів системи. Тестування інтерфейсу користувача для виявлення помилок у відображенні на різних пристроях. Функціональне тестування серверної частини для перевірки коректності роботи API та взаємодії з базою даних. Навантажувальне тестування для оцінки стабільності сайту при великій кількості одночасних користувачів.

Після завершення тестування та усунення всіх виявлених помилок здійснюється деплоймент сайту на хостинг-платформу. Налаштування серверної частини для постійної роботи та забезпечення доступності сайту. Встановлення необхідних програмних засобів для забезпечення безпеки сайту (SSL-сертифікати, механізми резервного копіювання). Моніторинг працездатності сайту та підтримка його актуальності після запуску.

Сучасний стан розвитку веб-технологій характеризується динамічними змінами та стрімким удосконаленням підходів до створення веб-ресурсів. Аналізуючи наукову літературу з означеної проблематики, можна виділити кілька напрямів досліджень, що формують теоретичну базу для розробки веб-сайту музичної майстерні.

Фундаментальні принципи веб-дизайну та користувацького досвіду детально розглянуті у працях Нільсена Я. та Лоранжер Х. [7], які сформулювали базові принципи юзабіліті веб-сайтів. Автори акцентують увагу на необхідності створення інтуїтивно зрозумілих інтерфейсів, що не вимагають від користувача зайвих когнітивних зусиль. Цей підхід розвиває Круг С. [8] у своїй праці, де визначає принцип «Don't Make Me Think» як основоположний для успішного веб-дизайну. Дослідник підкреслює, що ефективний веб-сайт повинен дозволяти користувачам досягати своїх цілей з мінімальними зусиллями та без необхідності розуміти структуру або логіку системи.

Шість ключових принципів веб-дизайну та їхню суть коротко підсумовано на рисунку 1.2.

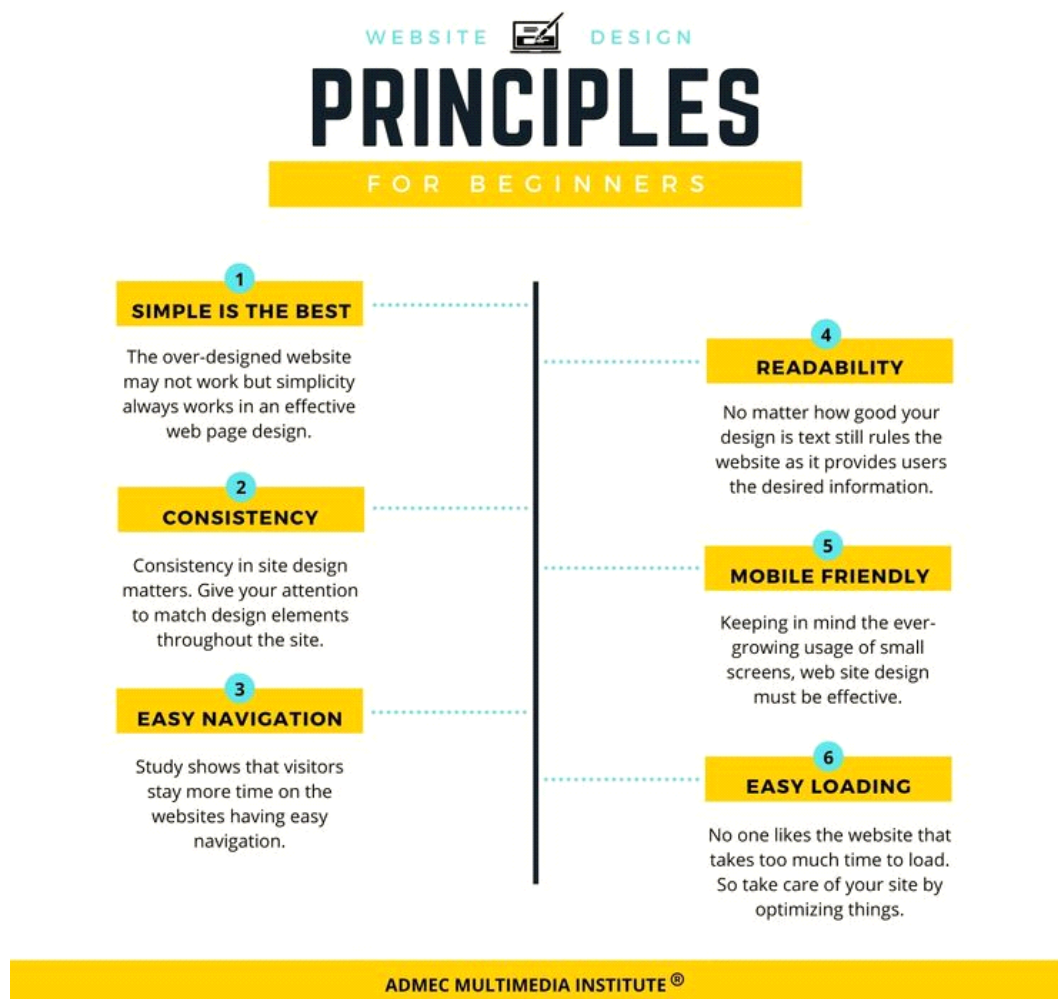


Рис. 1.2. Ключові принципи веб-дизайну

Системний підхід до проектування веб-інтерфейсів представлений у роботі Гарретта Дж., який запропонував п'ятирівневу модель користувацького досвіду, що включає стратегічний рівень, рівень можливостей, структурний рівень, рівень компонування та візуальний рівень [9]. Ця модель дозволяє комплексно підходити до проектування веб-ресурсів та враховувати всі аспекти взаємодії користувача з системою.

Серед вітчизняних науковців, що досліджують питання веб-розробки у контексті специфіки українського ринку, слід особливо відзначити роботи Петренка В.М., який здійснив ґрунтовну адаптацію провідних світових практик до умов українського бізнес-середовища [4]. У своїх працях дослідник висвітлює необхідність урахування соціально-економічних реалій України, зокрема нестабільності ринку, обмежених фінансових і кадрових ресурсів, що притаманні малому та середньому бізнесу. Автор пропонує цілісну методологію проектування веб-систем, яка базується на гнучких та економічно ефективних технологіях, з особливим акцентом на швидке розгортання MVP-рішень, подальше масштабування функціоналу за допомогою модульного підходу, а також автоматизацію бізнес-процесів. Такий підхід є надзвичайно актуальним для суб'єктів господарювання в умовах динамічного розвитку цифрової економіки.

Коваленко О.П. у своїх дослідженнях фокусується на розробці веб-рішень для підприємств культурної та креативної сфери [5]. Автор наголошує на важливості створення веб-продуктів, які одночасно відповідають високим вимогам естетичної привабливості, відображають мистецьку ідентичність організації та залишаються інтуїтивно зрозумілими для кінцевих користувачів. Коваленко підкреслює, що у цій сфері веб-дизайн не лише виконує інформативну функцію, а й є частиною іміджу організації, її візуальним брендом. У зв'язку з цим, особлива увага має приділятися глибокому аналізу цільової аудиторії, її культурного та соціального контексту, а також розробці дизайну, що резонує з очікуваннями користувача. Такий підхід дозволяє підвищити рівень взаємодії з аудиторією та сприяти популяризації культурних послуг у цифровому

середовищі.

Архітектурні аспекти розробки сучасних веб-додатків для спеціалізованих підприємств знайшли висвітлення у працях Дмитренка К.В., який обґрунтовує доцільність впровадження модульної архітектури в розробку адаптивних веб-систем [6]. Дослідник вказує на важливість використання компонентно-орієнтованого підходу, що забезпечує розширюваність, масштабованість та легкість у супроводі програмного забезпечення. У контексті обмежених ресурсів це дозволяє поступово вводити нові модулі, уникаючи необхідності переписування існуючої системи. Також Дмитренко підкреслює роль REST-архітектури, мікросервісів та хмарних технологій у створенні сучасних веб-продуктів, здатних адаптуватися до зростаючих потреб організацій.

Окрему нішу в наукових дослідженнях займають питання, пов'язані зі створенням веб-рішень для представників творчих індустрій. Зокрема, у роботах Шевченко Л.І. та Вороніної А.К. детально розглянуто роль цифрових технологій у розвитку креативного сектору економіки України [10]. Автори відзначають, що креативні підприємства все більше покладаються на онлайн-платформи як основний канал комунікації з аудиторією, що вимагає розробки оригінальних і водночас функціональних веб-рішень. Дослідження також підкреслюють значення інтеграції мультимедійних інструментів (відео, аудіо, анімацій) у веб-інтерфейс як засобу залучення користувачів та формування позитивного досвіду взаємодії.

Особливу увагу до специфіки веб-представництв музичних організацій приділяє Морозова Т.О., яка досліджує особливості структурування контенту, відображення творчої концепції на веб-ресурсі та шляхи підвищення залученості аудиторії за допомогою інтерактивного дизайну [11]. Автор відзначає, що для музичних майстерень важливо не лише інформувати, а й створити емоційний зв'язок з відвідувачем, використовуючи відповідну палітру кольорів, шрифти, динамічні елементи та аудіо-візуальні матеріали.

Серед зарубіжних джерел, які заслуговують на увагу, варто виділити дослідження Андерсона К., що зосереджуються на цифровому маркетингу для

підприємств креативного сектору [12]. Автор аналізує сучасні стратегії просування через соціальні мережі, впровадження SEO-практик та контент-маркетинг, які безпосередньо пов'язані з ефективністю функціонування веб-сайту. Водночас Браун Д. пропонує системний підхід до проектування веб-представництв для творчих бізнесів, з урахуванням фірмового стилю, інтеграції з інструментами онлайн-продажів, підключенням аналітичних сервісів та формуванням клієнтського досвіду на всіх етапах взаємодії з платформою [13].

Аналіз наукової літератури та технічної документації дозволяє виділити кілька ключових тенденцій у сучасній веб-розробці, які є актуальними для створення веб-сайту музичної майстерні.

Як відзначають Фаін М. та Мохер Д., прогресивні веб-додатки стали однією з найважливіших тенденцій останніх років, оскільки вони поєднують переваги веб-сайтів та нативних мобільних додатків [18]. PWA забезпечують швидке завантаження, можливість роботи офлайн та доступ до апаратних функцій пристрою, що робить їх ідеальним рішенням для бізнесів, які прагнуть забезпечити якісний користувацький досвід на різних пристроях.

Розвиток цього напрямку детально висвітлений у дослідженні Лалвані Т. та Сміта К., які аналізують вплив впровадження PWA на показники конверсії та залучення користувачів [19]. Автори наводять статистичні дані, що демонструють зростання часу перебування на сайті на 40% та збільшення конверсії на 25-30% після впровадження технології PWA.

Інноваційні підходи до організації архітектури веб-додатків представлені у роботах Геєра М., який досліджує концепцію мікрофронтендів як еволюцію мікросервісної архітектури на боці клієнта [20]. Цей підхід дозволяє розділити складний веб-додаток на незалежні модулі, що можуть розроблятися та розгортатися окремо, що особливо актуально для проектів з розподіленими командами.

Комінарець В. детально розглядає практичні аспекти впровадження модульної архітектури у веб-проектах різного масштабу, зокрема вказує на її високу ефективність як у малих, так і у великих системах [21]. Автор підкреслює,

що поділ функціональності на окремі незалежні модулі сприяє кращій організації коду, спрощує процес розробки, тестування та підтримки, а також забезпечує можливість масштабування системи без необхідності переробки її основних компонентів. Такий підхід дозволяє впроваджувати нові функції або оновлювати наявні з мінімальним ризиком порушити роботу інших частин веб-додатку.

Сучасні безсерверні технології, що дозволяють розробникам зосередитися на логіці додатку, абстрагуючись від управління серверною інфраструктурою, детально аналізуються у дослідженні Робертса М. [22]. У своїй роботі автор акцентує увагу на численних перевагах таких платформ, як AWS Lambda, Google Cloud Functions та Azure Functions. Робертс зазначає, що ці технології не лише скорочують витрати на інфраструктуру та обслуговування, але й забезпечують автоматичне масштабування, підвищену доступність і fault-tolerance, що особливо важливо в умовах високої динаміки цифрового середовища.

Практичні підходи до реалізації безсерверної архітектури у проєктах різної складності висвітлено у праці Збарські П. [23]. Автор докладно аналізує патерни проєктування безсерверних додатків, зокрема підхід event-driven architecture, який базується на взаємодії між мікросервісами через події. У роботі також розглянуто специфіку організації баз даних, моніторингу та безпеки у безсерверному середовищі, що є ключовими складовими успішного впровадження подібних рішень у виробниче середовище.

Тенденції розвитку веб-інтерфейсів, орієнтованих передусім на мобільних користувачів, розглядаються у дослідженні Вроблевського Л., який є одним з провідних теоретиків концепції «Mobile First» [24]. У своїй роботі автор обґрунтовує необхідність розробки інтерфейсів, у яких пріоритет надається мобільній версії, з подальшою адаптацією до ширших екранів. Такий підхід, за словами дослідника, забезпечує оптимальний досвід користування в умовах домінування мобільного трафіку, який, згідно з наведеними статистичними даними, у багатьох випадках перевищує 70% загального трафіку.

Розвиток адаптивного дизайну як необхідної складової сучасної веб-розробки висвітлюється у працях Марчотта Е. [25]. Автор пропонує методологію

побудови гнучких інтерфейсів, що ефективно адаптуються до характеристик пристрою користувача, незалежно від його розмірів та роздільної здатності екрана. Методологія базується на використанні гнучких сіток (flexbox, CSS grid), медіа-запитів, адаптивної типографіки та зображень з динамічним розміром. Особливу увагу приділено принципам user-centric design, що забезпечують комфорт і зручність у взаємодії з сайтом.

У дослідженні Бірнбаума Н. здійснено глибокий аналіз JavaScript-фреймворків, які сьогодні є основою сучасної фронтенд-розробки [26]. Автор порівнює найпопулярніші фреймворки — React, Angular та Vue — за критеріями продуктивності, гнучкості, простоти навчання, підтримки спільноти та екосистеми. На основі аналізу зроблено висновок, що вибір фреймворку повинен ґрунтуватися на специфіці поставленого завдання, обсязі проєкту та технічному рівні команди, що його реалізує.

Особливості односторінкових додатків (SPA), їх архітектурні характеристики та переваги у створенні інтерактивного користувацького досвіду описані у праці Холломана А. [27]. Автор наголошує на тому, що SPA забезпечують швидке реагування інтерфейсу, зменшення часу завантаження та покращення UX, оскільки сторінка не перезавантажується повністю при навігації. Також розглядаються виклики, пов'язані з SEO-оптимізацією та забезпеченням безпеки у SPA, що потребують застосування додаткових технологій, таких як SSR (server-side rendering) або ізоморфна розробка.

Сучасні підходи до побудови взаємодії між клієнтською та серверною частинами веб-додатків знайшли своє відображення у роботі Лайна М., де автор акцентує увагу на перевагах API-орієнтованої архітектури [28]. Такий підхід передбачає чіткий розподіл обов'язків між фронтендом і бекендом та використання RESTful або GraphQL API для взаємодії між ними. Це дозволяє досягти високої масштабованості, покращеної безпеки, а також гнучкості при інтеграції з іншими сервісами або мобільними додатками.

Особливості використання headless CMS (систем управління контентом без вбудованої фронтенд-частини) ґрунтовно розглядаються у праці Барклі Дж.

[29]. Автор висвітлює актуальну тенденцію до розділення логіки управління контентом та його відображення, що є важливим аспектом сучасної веб-архітектури. Такий підхід дозволяє централізовано зберігати та редагувати контент, забезпечуючи його повторне використання на різних платформах — від веб-сайтів і мобільних застосунків до пристроїв з підтримкою голосових інтерфейсів. Барклі зазначає, що headless CMS, зокрема такі системи, як Strapi, Contentful та Sanity, суттєво спрощують реалізацію омніканального досвіду для кінцевих користувачів, сприяють гнучкості у виборі технологій фронтенду та знижують залежність між командами контент-менеджменту і розробки.

Питання забезпечення доступності веб-ресурсів для всіх категорій користувачів, включаючи людей з порушеннями зору, слуху, моторики або когнітивних функцій, детально аналізуються у дослідженні Хейдона Х. [30]. Автор пропонує практичні рекомендації щодо створення інклюзивних інтерфейсів, зокрема через дотримання стандартів WCAG (Web Content Accessibility Guidelines), правильне використання семантичної розмітки, кольорових контрастів, альтернативного тексту до зображень та клавіатурної навігації. Особливу увагу приділено етичному виміру доступності — як прояву соціальної відповідальності веб-розробників та дизайнерів, що має на меті забезпечення рівних можливостей доступу до інформації для всіх.

Кларк А. розвиває концепцію інклюзивного дизайну, розглядаючи її як підхід, який виходить за межі суто технічної доступності [31]. Він пропонує методологію проектування, орієнтовану на потреби максимально широкого кола користувачів. У центрі його підходу — принцип «крайніх випадків», згідно з яким дизайн, розроблений для користувачів з особливими потребами, часто виявляється зручним і для основної аудиторії. Наприклад, субтитри, створені для людей з вадами слуху, корисні також у публічних місцях або в ситуаціях, коли користувач не може ввімкнути звук. Таким чином, інклюзивний дизайн стає важливим елементом підвищення загальної якості користувацького досвіду.

Інтеграція практик DevOps у процеси веб-розробки всебічно аналізується у роботі Кіма Дж. [32]. Автор підкреслює, що впровадження DevOps-культури

дозволяє досягти високої узгодженості між командами розробки та експлуатації, що, у свою чергу, позитивно впливає на швидкість розробки, стабільність продукту та його якість. Особливу увагу приділено автоматизації процесів безперервного тестування, розгортання (deployment) та моніторингу. Згідно з дослідженням, використання інструментів CI/CD, таких як Jenkins, GitLab CI, CircleCI, значно скорочує час виходу нових релізів на продакшн та мінімізує ризик помилок завдяки автоматизованим перевіркам і можливості швидкого відкоту до стабільної версії.

Практичні аспекти впровадження DevOps-підходів у веб-проекти різного масштабу конкретизуються у праці Хамбліна Р. [33], який пропонує набір перевірених інструментів і методологій для організації ефективного процесу розробки. Зокрема, автор розглядає застосування контейнеризації (Docker), оркестрації (Kubernetes), а також систем логування й моніторингу (ELK Stack, Prometheus, Grafana) як необхідні компоненти сучасного DevOps-стека. Завдяки цьому досягається не лише безперебійна робота веб-додатків, а й можливість оперативного реагування на інциденти, що забезпечує високий рівень надійності й довіри з боку кінцевих користувачів.

На основі аналізу літератури та сучасних тенденцій у веб-розробці можна визначити оптимальні технології та підходи для створення веб-сайту музичної майстерні.

Для розробки клієнтської частини веб-сайту музичної майстерні доцільно використовувати сучасні JavaScript-фреймворки, що забезпечують високу інтерактивність та швидкодію. Як відзначає Бірнбаум Н. [26], фреймворки React, Vue та Angular є найбільш популярними та добре підтримуваними інструментами для створення сучасних веб-інтерфейсів.

Для забезпечення адаптивності дизайну рекомендується застосовувати методологію «Mobile First», описану Вроблевським Л. [24], та використовувати фреймворки для створення адаптивних сіток, такі як Bootstrap або Tailwind CSS.

Особливу увагу слід приділити оптимізації завантаження мультимедійного контенту, що є критичним для веб-сайту музичної майстерні. Як зазначає Грігсбі

Дж. [34], використання сучасних форматів зображень (WebP, AVIF) та технологій адаптивного завантаження медіаконтенту дозволяє суттєво покращити швидкодію сайту.

Для розробки серверної частини веб-сайту доцільно використовувати сучасні фреймворки, що забезпечують швидку розробку та підтримку API. Згідно з дослідженням Лайна М. [28], найбільш ефективними інструментами для створення REST API є фреймворки Node.js (Express), Python (Django, Flask) та PHP (Laravel).

Для зберігання даних можуть використовуватися як реляційні (PostgreSQL, MySQL), так і нереляційні (MongoDB, Firebase) бази даних, вибір яких залежить від специфіки проекту та вимог до структури даних.

Для забезпечення зручного управління контентом веб-сайту музичної майстерні доцільно використовувати сучасні CMS. Як відзначає Барклі Дж., тенденція до використання headless CMS дозволяє розділити логіку управління контентом та його представлення, що забезпечує більшу гнучкість системи [29].

Серед популярних headless CMS можна виділити Strapi, Contentful та Sanity, які надають зручний інтерфейс для управління контентом та API для його отримання на клієнтській частині.

Специфіка веб-сайту музичної майстерні передбачає активне використання аудіо- та відеоконтенту. Як зазначає Морозова Т.О., ефективна інтеграція мультимедійних елементів є ключовим фактором успіху веб-представництва музичних підприємств [11].

Для роботи з аудіо на веб-сайті рекомендується використовувати сучасні API, такі як Web Audio API, що дозволяє реалізувати складну обробку аудіосигналу безпосередньо у браузері. Для роботи з відео можуть використовуватися HTML5 Video API та спеціалізовані бібліотеки, такі як Video.js.

Таким чином, розробка веб-сайту для музичної майстерні повинна базуватися на комплексному підході, що враховує як загальні принципи веб-розробки, так і специфічні вимоги, пов'язані з особливостями презентації

музичних інструментів та послуг онлайн.

1.3 Проведення аналізу веб-сайтів конкурентів

У процесі розробки веб-сайту для музичної майстерні важливо провести детальний аналіз існуючих рішень на ринку. Такий аналіз дозволяє виявити успішні підходи, які варто впровадити, та помилки, яких слід уникнути. Розглянемо найбільш примітні веб-сайти музичних майстерень та магазинів музичних інструментів, проаналізуємо їхні функціональні можливості, дизайн, зручність використання та інші важливі аспекти.

Для об'єктивного порівняння існуючих рішень було визначено наступні *критерії оцінювання*:

- Дизайн та візуальне представлення — естетика сайту, відповідність тематиці, якість візуальних матеріалів
- Юзабіліті та навігація — зручність користування, інтуїтивність інтерфейсу, доступність необхідної інформації
- Функціональність — наявність онлайн-каталогу, можливість замовлення послуг, інтеграція з системами оплати
- Інформативність — повнота представленої інформації про послуги, товари та майстерню
- Адаптивність — коректне відображення на різних пристроях (мобільних телефонах, планшетах, десктопах)
- Швидкодія — швидкість завантаження сторінок
- SEO-оптимізація — позиціонування в пошукових системах

Сайт «*Crimson Guitars*» [35] має наступні переваги та недоліки:

Переваги:

- Преміальний дизайн з відмінними фотографіями інструментів ручної роботи
- Детальна презентація процесу створення гітар з високоякісними зображеннями

- Інтегрований освітній контент (курси по створенню гітар, відеоматеріали)

- Зручний онлайн-магазин з продуманою системою фільтрації інструментів та аксесуарів

- Розділ портфоліо з унікальними проєктами та кастомними моделями

Недоліки:

- Складна структура сайту для нових користувачів
- Відносно довгий час завантаження сторінок з великою кількістю зображень

- Відсутність онлайн-конструктора для попередньої конфігурації інструменту

- Обмежений функціонал взаємодії для мобільних пристроїв

Сайт «StewMac» [36] має наступні переваги та недоліки:

Переваги:

- Надзвичайно повний каталог інструментів, комплектуючих та матеріалів для лютьєрів

- Докладні інструкції та посібники з ремонту та виготовлення інструментів

- Структурована база знань з відео-тutorіалами та статтями
- Зручний інтерфейс покупки з розширеними фільтрами пошуку
- Професійні інструменти для підбору комплектуючих

Недоліки:

- Переважання комерційної складової над інформаційною
- Перевантаженість головної сторінки, що може відволікати користувача
- Недостатньо персоналізований підхід до взаємодії з клієнтами
- Іноді надмірна кількість товарів ускладнює вибір для початківців

Вигляд головної сторінки сайту StewMac зображено в додатку Г.

Сайт «*Vienna Music & Sound Arts*» [37] має наступні переваги та недоліки:

Переваги:

- Вишуканий, класичний дизайн, що відображає високу майстерність виробництва

- Акцент на історії майстерні та традиціях виготовлення інструментів
- Детальні портфоліо майстрів з їхніми біографіями та спеціалізацією
- Якісні фотографії процесу реставрації та виготовлення
- Розділ з експертними статтями про догляд за інструментами

Недоліки:

- Обмежена функціональність онлайн-замовлення
- Недостатньо розвинені соціальні взаємодії (відгуки, коментарі)
- Слабка адаптація для мобільних пристроїв
- Відсутність інтерактивних елементів для залучення аудиторії

Сайт «Thomann» [38] має наступні переваги та недоліки:

Переваги:

- Масштабний онлайн-магазин з величезним асортиментом музичних інструментів та обладнання

- Потужна система фільтрації та порівняння товарів
- Детальні технічні характеристики кожного продукту
- Багатомовний інтерфейс з локалізацією для різних країн
- Інтегрована система рейтингів та відгуків клієнтів

Недоліки:

- Недостатній фокус на послугах з ремонту та обслуговування інструментів

- Комерційна спрямованість переважає над інформаційною цінністю
- Складність навігації через великий обсяг товарів
- Швидкість завантаження сторінок може знижуватись у годину пік

Вигляд головної сторінки сайту Thomann зображено в додатку Д.

Для наочного порівняння проаналізованих веб-сайтів було проведено оцінювання за вищезазначеними критеріями за 10-бальною шкалою, результати аналізу наведено в таблиці 1.1.

Таблиця 1.1.

Порівняння веб-сайтів музичних майстерень

Критерій	Crimson Guitars	StewMac	VMSA	Thomann
Дизайн	9	7	8	7
Юзабіліті	7	8	6	8
Функціональність	8	9	5	9
Інформативність	8	9	7	8
Адаптивність	7	8	5	8
Швидкодія	6	7	7	6
SEO-оптимізація	8	9	6	9
Середній бал	7.6	8.1	6.3	7.9

Проведений аналіз існуючих рішень дозволяє виділити ключові аспекти, які варто врахувати при розробці веб-сайту для музичної майстерні. Важливим є баланс між естетикою та функціональністю. Наприклад, Crimson Guitars та VMSA демонструють, як важливо створити візуально привабливий сайт, який підкреслює майстерність та традиції, при цьому StewMac та Thomann показують перевагу потужної функціональності. Освітній компонент є ще однією важливою складовою успіху, оскільки сайти на кшталт Crimson Guitars та StewMac пропонують цікаві відеоуроки, статті та майстер-класи, що підвищує залученість їхньої аудиторії та сприяє активності користувачів на сайті.

Детальний візуальний контент, зокрема якісні фотографії інструментів і процесів виробництва та реставрації, допомагає створювати довіру клієнтів, як це видно на всіх проаналізованих сайтах. Спеціалізований функціонал, такий як фільтри, конфігуратори та інструменти підбору комплектуючих, що присутні на StewMac та Thomann, покращує користувацький досвід. Адаптивність для мобільних пристроїв також є важливою, оскільки сучасні користувачі очікують повноцінного функціоналу на всіх типах пристроїв, що враховують Thomann та StewMac, але, на жаль, не завжди є на сайті VMSA.

Не менш важливою є оптимізація швидкодії, адже сайти з великою кількістю візуального контенту повинні приділяти особливу увагу оптимізації швидкості завантаження. Загалом, рекомендується щоб час завантаження

сторінки не перевищував 2-3 секунди [39]. На основі цього аналізу можна визначити оптимальний набір функцій та підходів для розробки веб-сайту музичної майстерні: поєднання естетичного дизайну Crimson Guitars, інформаційної насиченості StewMac, культурної цінності VMSA та функціональності Thomann дозволить створити сучасний, зручний і інформативний ресурс, який відповідатиме потребам цільової аудиторії та вирішуватиме бізнес-задачі майстерні музичних інструментів.

Висновки до 1 розділу

У підсумку проведеного аналізу теоретичних засад створення веб-сайтів, а також огляду особливостей сучасної веб-розробки, можна стверджувати, що розвиток інформаційних технологій, зокрема веб-технологій, відіграє вирішальну роль у цифровізації бізнесу, культури та повсякденного життя. Сьогодні веб-сайти перестали бути просто інформативними візитівками — вони перетворилися на повноцінні платформи взаємодії між користувачами та сервісами. Відтак, створення веб-сайту для музичної майстерні не є лише технічним завданням, а комплексною проблемою, яка охоплює аспекти дизайну, зручності користування, безпеки, оптимізації продуктивності та відповідності потребам цільової аудиторії.

У першому розділі було розглянуто основні підходи до організації клієнт-серверної архітектури, що є фундаментом сучасного веб-програмування. Було встановлено, що найефективнішим для реалізації динамічного та функціонального веб-сайту є поєднання технологій HTML, CSS і JavaScript на клієнтській стороні із серверною логікою, реалізованою за допомогою Node.js, Express та бази даних MongoDB. Важливо також зазначити, що використання шаблонізаторів, таких як EJS, а також застосування RESTful API та JSON Web Token (JWT) забезпечують не лише гнучкість розробки, але й високий рівень безпеки доступу до персоналізованих функцій.

Аналіз сучасних веб-фреймворків та підходів до побудови інтерфейсів

користувача підтвердив доцільність впровадження адаптивного дизайну, що дозволяє веб-додатку коректно відображатися на пристроях із різними розмірами екранів, що особливо важливо для залучення ширшої аудиторії. Також було наголошено на актуальності дотримання принципів UI/UX-дизайну, що сприяє зручності взаємодії користувача з веб-ресурсом та формує позитивне враження від сервісу загалом.

У процесі дослідження було визначено, що застосування бази даних MongoDB є доцільним для зберігання гнучко структурованої інформації про користувачів, розташування, пости, що публікуються, тощо. Її документно-орієнтована структура забезпечує простоту масштабування та ефективну обробку запитів, що відповідає потребам веб-сайту для музичної майстерні. Крім того, особливу увагу було приділено аспектам реєстрації, аутентифікації та авторизації користувачів, що є важливою умовою безпечного функціонування системи, особливо якщо вона передбачає індивідуальний доступ до певних функцій чи контенту.

Отримані результати теоретичного аналізу стали основою для формування вимог до розробки веб-сайту музичної майстерні, що будуть реалізовані у наступних розділах роботи.

РОЗДІЛ 2

АЛГОРИТМ СТВОРЕННЯ ВЕБ-САЙТУ ДЛЯ МУЗИЧНОЇ МАЙСТЕРНІ

2.1 Аналіз цільової аудиторії з метою розробки ефективного дизайну веб-сайту

У процесі розробки веб-сайту для музичної майстерні критично важливим етапом є визначення та детальний аналіз цільової аудиторії. Чітке розуміння потреб, очікувань та поведінкових патернів потенційних користувачів дозволяє створити інтуїтивно зрозумілий інтерфейс та функціональність, що відповідає запитам аудиторії [40].

На основі проведеного дослідження ринку та потенційних користувачів, цільову аудиторію музичної майстерні можна розділити на декілька основних сегментів.

Професійні музиканти характеризуються високими потребами в плані обслуговування, як-от купівля якісних інструментів, ремонт та налаштування інструментів, пошук індивідуальних рішень. Вони вирізняються високою технічною обізнаністю у сфері музичних інструментів, специфічними вимогами, потребою в фаховій консультації. Аматори та хобісти шукають придбання інструментів середнього цінового сегменту, навчальні матеріали, регулярне обслуговування інструментів. Їх характеризує середній рівень обізнаності, потреба в рекомендаціях, цінова чутливість.

Батьки дітей-музикантів найчастіше зацікавлені у придбанні початкових/навчальних інструментів, періодичному обслуговуванні, консультаціях щодо вибору. Для них характерний низький рівень обізнаності в музичних інструментах, акцент на співвідношення ціна-якість, потреба в детальних консультаціях. Навчальні заклади, такі як музичні школи, консерваторії, гуртки, мають потреби у гуртових закупівлях, спеціалізованому обслуговуванні, довгостроковій співпраці. Їх характеризують формалізовані

процеси закупівель, потреба в документації, особливі умови оплати. Музичні колективи та студії, включаючи оркестри, гурти, звукозаписуючі студії, шукають специфічне обладнання, комплексне обслуговування, швидкий ремонт. Для них важлива технічна обізнаність, встановлені вимоги, важливість оперативності.

На основі проведених опитувань та аналізу поведінки відвідувачів схожих веб-ресурсів, визначено ключові потреби користувачів веб-сайту музичної майстерні (див. таб. 2.1).

Таблиця 2.1.

Аналіз потреб цільової аудиторії

Сегмент аудиторії	Головні потреби	Очікування від веб-сайту
Професійні музиканти	Якісне, спеціалізоване обслуговування	Повнота інформації, висока функціональність, бажана можливість здійснення онлайн-замовлень
Аматори	Зрозумілість навчальних матеріалів, доступні ціни	Наявність навчальних матеріалів, доступність закладів
Батьки дітей-музикантів	Інформація для музикантів-початківців, надійність, гарантії	Зрозумілість написаного, відсутність складної термінології, лаконічне подання інформації
Навчальні заклади	Спеціальні пропозиції для навчальних закладів, зворотній зв'язок	Можливість зворотнього зв'язку, окремий розділ B2B
Музичні колективи	Спеціалізоване обладнання, оперативне обслуговування	Функціонал для оперативної комунікації з клієнтом

Дослідження особливостей онлайн-поведінки цільової аудиторії дозволило визначити ключові патерни. За даними, 58% користувачів заходять з мобільних пристроїв, 37% використовують настільні комп'ютери, а 5% - планшети та інші пристрої. Шлях користувача на сайті відрізняється залежно від сегменту: професійні музиканти шукають інструменти серед популярних брендів і готові багато платити за високу якість; аматори читають відгуки та порівнюють моделі інструментів; батьки в першу чергу оглядають бюджетні варіанти і часто зв'язуються з консультантом.

Середній час сесії становить 5-7 хвилин, при цьому професіонали проводять на сайті до 15 хвилин, аматори та батьки – 3-5 хвилин, а корпоративні клієнти – 10-12 хвилин. Щодо поведінки при здійсненні замовлення, 45% користувачів віддають перевагу онлайн-замовленню з наступним самовивозом, 30% замовляють з доставкою, а 25% використовують сайт для ознайомлення, але купують офлайн.

Основні фактори, що впливають на прийняття рішень різними сегментами аудиторії були оцінені за шкалою від 1 до 5 і структуровані в таблиці 2.2.

Таблиця 2.2.

Аналіз факторів прийняття рішень цільовою аудиторією

Фактор	Професіонали	Аматори	Батьки	Навчальні заклади
Ціна	3	4	5	4
Якість	5	4	3	5
Бренд	5	3	2	3
Сервіс	4	3	4	5
Швидкість доставки	4	2	2	3
Гарантія	3	4	5	5
Відгуки	3	5	4	3

На основі проведеного аналізу цільової аудиторії можна сформулювати рекомендації для розробки веб-сайту музичної майстерні. Архітектура сайту має включати окремі розділи для різних сегментів аудиторії, забезпечувати інтуїтивну навігацію з урахуванням різного рівня технічної обізнаності користувачів, а також реалізувати адаптивний дизайн з пріоритетом на мобільні пристрої. У сфері контенту важливо розробити детальні картки товарів з технічними специфікаціями для професіоналів, створити базу знань та глосарій музичних термінів для початківців, підготувати серію навчальних матеріалів з вибору та догляду за інструментами.

Функціональність сайту має включати систему онлайн-запису на обслуговування та ремонт, функцію порівняння характеристик інструментів, розширені фільтри пошуку (за ціною, типом, брендом, призначенням) та форму

запиту на індивідуальну консультацію. Маркетингові інструменти повинні інтегрувати систему відгуків та рейтингів товарів, функціонал персоналізованих рекомендацій, програму лояльності для постійних клієнтів та можливість підписки на новини та спеціальні пропозиції. Технічна оптимізація має забезпечити швидке завантаження сторінок для зменшення показника відмов, оптимізувати пошук на сайті для обробки специфічних музичних термінів та реалізувати можливість перегляду сайту в офлайн-режимі для подальшого відвідування магазину.

Врахування особливостей цільової аудиторії дозволить створити веб-сайт, що максимально відповідає потребам користувачів, забезпечить високу конверсію та сприятиме формуванню лояльності клієнтів до музичної майстерні [41].

2.2 Проектування об'єктів та зв'язків між ними

На етапі проектування веб-сайту для музичної майстерні особливу увагу приділено формуванню об'єктної моделі системи та взаємозв'язку між її основними компонентами. Використання бази даних MongoDB зумовлює документно-орієнтований підхід до зберігання інформації, що забезпечує гнучкість у структурі даних і високу масштабованість.

У системі передбачено три основні колекції, кожна з яких відповідає окремому логічному блоку функціоналу. Колекція – це аналог реляційної таблиці в середовищі MongoDB.

Колекція Users зберігає облікові записи адміністраторів, які мають право додавати інформацію на сайт. Структура документа має такі поля:

- username — унікальне ім'я користувача (тип: рядок);
- password — хешований пароль (тип: рядок);
- profilePicture — посилання на зображення профілю (тип: рядок).

Колекція Locations відображає інформацію про розташування музичних майстерень чи пов'язаних об'єктів:

- name — назва закладу;
- address — поштова адреса;
- phone — контактний номер телефону;
- email — електронна пошта;
- createdBy — ідентифікатор користувача, який створив запис (зв'язок з Users).

Колекція Posts зберігає записи блогу, новини або інформаційні статті:

- title — заголовок публікації;
- desc — короткий опис;
- content — основний текст статті;
- image — зображення або ілюстрація до публікації;
- createdAt — дата та час створення;
- createdBy — ідентифікатор автора (зв'язок з Users).

У системі реалізовано логічні зв'язки між сутностями за допомогою посилань на ідентифікатори документів. У MongoDB такі зв'язки зазвичай реалізуються через збереження `_id` документа однієї колекції як поля іншої. Зокрема, `createdBy` у колекціях `Locations` і `Posts` є зовнішнім ключем до `_id` з `Users`.

Це дозволяє:

- встановлювати авторство публікацій та записів;
- реалізувати контроль доступу до редагування;
- формувати особисті кабінети користувачів.

Ці зв'язки логічно відображають модель "один до багатьох", де один користувач може створити багато записів у колекціях `Locations` і `Posts`.

На рисунку 2.1 представлено діаграму взаємозв'язку між сутностями.

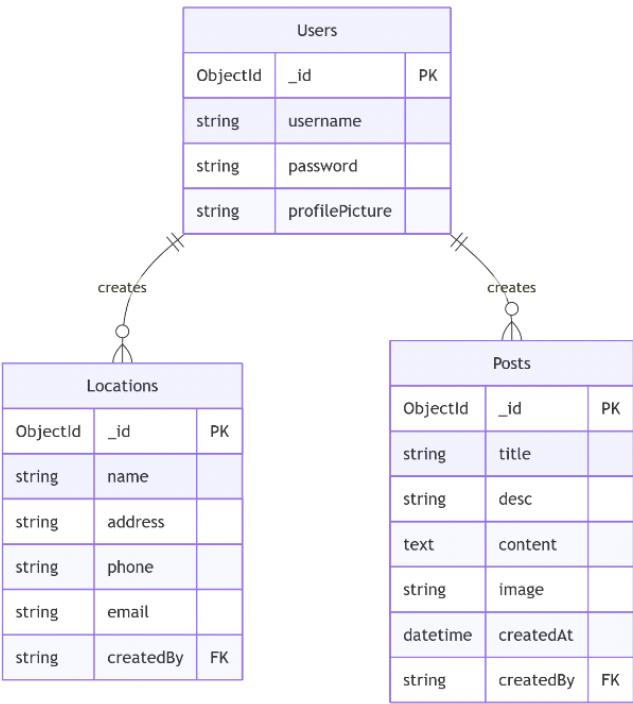


Рис. 2.1. ER-діаграма бази даних сайту

Дана структура сприяє модульності та гнучкому масштабуванню додатку, що особливо важливо для динамічних проєктів з перспективою розширення.

2.3 Вибір програмного забезпечення для розробки веб-сайту музичної майстерні

У процесі розробки веб-сайту для музичної майстерні було прийнято рішення використовувати сучасний технологічний стек MERN, що поєднує MongoDB, Express.js, React та Node.js [42]. Цей стек обрано через його повну орієнтованість на JavaScript як на стороні клієнта, так і на стороні сервера, що значно спрощує обмін даними, уніфікує підхід до розробки та скорочує кількість необхідних технологій.

На рисунку 2.2 зображено схему роботи додатків, що працюють на стеку MERN.

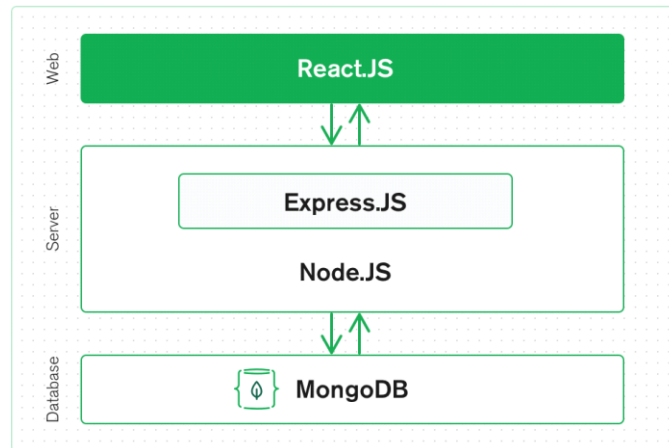


Рис. 2.2. Схема роботи стеку MERN

Для зберігання даних використано базу даних MongoDB, яка відома своєю масштабованістю, гнучкістю в обробці неструктурованих даних та підтримкою документно-орієнтованої моделі. На відміну від реляційних СКБД, таких як MySQL або PostgreSQL, MongoDB не вимагає суворого дотримання схем, що дозволяє швидше розробляти прототипи та адаптувати модель даних до змін [43]. Завдяки бібліотеці Mongoose було реалізовано обробку моделей з валідацією та зручним API для взаємодії з колекціями.

Для серверної частини застосовано Node.js у поєднанні з фреймворком Express.js. Node.js забезпечує неблокуючу обробку запитів, що особливо важливо для систем із великою кількістю одночасних з'єднань. Express.js, у свою чергу, спрощує створення REST API, дозволяє легко налаштовувати маршрутизацію та інтегрувати middleware. У порівнянні з альтернативами на зразок Django (Python) чи Laravel (PHP), рішення на основі Node.js має перевагу в продуктивності для задач, орієнтованих на події.

В таблиці 2.3 представлена порівняльна таблиця рішень на основі Node.js та Express з Django та Laravel.

Порівняльна таблиця засобів розробки

Критерій	Node.js	Django	Laravel
Мова програмування	JavaScript	Python	PHP
Парадигма	Мінімалістичний фреймворк	Зручність розробки	Висока функціональність
Крива навчання	Помірна	Відносно полого	Помірна
Швидкість розробки	Висока при невеликих проектах, гнучка	Дуже висока завдяки вбудованим компонентам	Висока, багато готових рішень
Продуктивність	Висока, з асинхронністю	Висока, але синхронна	Нормальна
Система шаблонів	Зовнішні рішення	Django Template Language	Blade Template Engine
ORM	Sequelize, Mongoose (зовнішні бібліотеки)	Вбудований Django ORM	Eloquent ORM
Автентифікація та авторизація	Реалізовується вручну або через пакети (Passport)	Вбудовано	Вбудовано
Маршрутизація	Дуже гнучка, вручну налаштовується	Вбудована, декларативна	Вбудована, декларативна
Вбудовані засоби адміністрування	Немає, потрібно створювати вручну	Є (admin panel з коробки)	Є пакети, але не з коробки
Підтримка REST API	Відмінна, легко створюється вручну	Вбудована через Django REST Framework	Є вбудована підтримка (Laravel API Resources)
Ком'юніті та документація	Дуже велике	Велике	Велике
Хостинг / Розгортання	Гнучкий (VPS, Vercel, Render, Heroku)	Потребує налаштування, добре працює на VPS	Добре працює на традиційних хостингах і VPS
Найкраще підходить для	SPA, реального часу (chat, IoT, API)	CMS, аналітика, API, наукові додатки	Блогів, eCommerce, CRM, API

Для забезпечення взаємодії між клієнтом і сервером, які можуть працювати на різних доменах або портах під час розробки, застосовано middleware cors. Цей модуль дозволяє явно вказувати, які джерела мають право надсилати запити до серверу, що запобігає блокуванню запитів через політику однакового

походження (Same-Origin Policy). У порівнянні з ручним налаштуванням заголовків, використання `cors` є швидшим і безпечнішим способом конфігурування доступу.

З метою оптимізації процесу розробки на серверній частині використано `nodemon` — інструмент, який автоматично перезапускає сервер при зміні коду. Це значно пришвидшує цикл розробки, зменшуючи потребу в ручному перезапуску після кожної зміни. На відміну від базового запуску через `Node.js`, `nodemon` забезпечує більш зручне та ефективне тестування серверної логіки в реальному часі.

Для безпечної авторизації використано бібліотеку `jsonwebtoken`, яка дозволяє реалізувати безстейтову автентифікацію за допомогою JWT [44]. Обробка та зберігання паролів здійснюється через `bcryptjs`, що забезпечує достатній рівень криптографічного захисту. Модуль `multer` використано для завантаження зображень, що було необхідним для публікацій і профілів користувачів. Усі ці інструменти інтегруються з `Express` і дозволяють швидко впроваджувати функціональність із дотриманням безпекових стандартів.

Клієнтська частина веб-застосунку створена з використанням `React` — однієї з найпопулярніших бібліотек для розробки інтерфейсів. `React` надає компонентну структуру, що дозволяє ефективно повторно використовувати елементи та організувати код. Замість `Create React App` обрано `Vite`, який демонструє значно вищу швидкодію при збірці та запуску проекту, а також забезпечує кращу оптимізацію у `production`-режимі [45]. У порівнянні з `Angular` чи `Vue`, `React` має ширшу підтримку серед розробників та більшу кількість додаткових бібліотек (рис. 2.3).

Для стилізації застосовано `Tailwind CSS`, який забезпечує утилітарний підхід до оформлення, дозволяє швидко прототипувати компоненти без написання окремих CSS-файлів і зменшує дублювання стилів. Альтернативи на кшталт `Bootstrap` або `Material UI` мають більшу кількість вбудованих компонентів, однак `Tailwind` забезпечує більшу гнучкість і компактність інтерфейсного коду.

Factor	Vue	React	Angular
 Community and Development	Community-based development	Backed by Facebook	Backed by Google
 How it all Started	Developed by ex-Google employee and maintained by community	Developed by Facebook for their products	Developed by Google
 License	Open-source MIT License	Open-source MIT License	Open-source MIT License
 Popularity	18.82% developers use it	42.64% developers use it	20.39% developers use it
 Job Market	It has relatively low market	Biggest market for jobs	Niche and advanced market
 Migrations	Easier with the migration helper tool	Very easy	Slightly complex
 Size/Load times	Lightweight and swift	Medium sized and nominal load time	Slightly heavy
 Components	Customisable components	Components dictate UI and interactions (behavior)	Components dictate the behaviour of the interface
 Learning Curve	Low	Very low	Steep
 Ready to Use Libraries	Very few libraries available	Extensive libraries available	Very few libraries are available

Рис. 2.3. Порівняльна таблиця популярних фреймворків

Для підтримання єдиного стилю коду та виявлення потенційних помилок у JavaScript-кодi впроваджено використання ESLint. Цей інструмент статичного аналізу дозволяє виявляти синтаксичні та логічні помилки ще до виконання програми, що підвищує стабільність системи. Порівняно з ручною перевіркою, ESLint забезпечує автоматизацію контролю якості та легко інтегрується з більшістю сучасних редакторів коду.

Для взаємодії з бекендом обрано Axios — легку та надійну бібліотеку для HTTP-запитів, яка в багатьох випадках зручніша за Fetch API завдяки простішій обробці помилок і конфігурації. Форматування дат реалізовано за допомогою date-fns, що дозволяє легко працювати з датами в сучасному JavaScript без надлишкової ваги, як у випадку з Moment.js. Для реалізації слайдерів і каруселей використано Swiper — одну з найпотужніших бібліотек з адаптивною підтримкою та широкими можливостями кастомізації [46].

Таким чином, вибір кожного компонента програмного забезпечення був обґрунтований критеріями продуктивності, гнучкості, сумісності між технологіями та відповідності сучасним тенденціям у веб-розробці. Такий підхід дозволив створити ефективну, адаптивну та розширювану архітектуру веб-сайту, що повністю відповідає завданням дипломного проєкту.

Висновки до 2 розділу

У другому розділі кваліфікаційної роботи було проведено всебічне технічне проєктування веб-сайту для музичної майстерні, що стало основою для його подальшої реалізації. Розробка функціональних вимог до системи дала змогу чітко визначити її ключові можливості: реєстрацію та авторизацію користувачів, створення й перегляд публікацій, управління інформацією про розташування майстерні, взаємодію з клієнтами тощо. Ці вимоги відображають реальні потреби користувачів і забезпечують зручність у користуванні веб-сайтом.

У межах архітектурного проєктування було обґрунтовано вибір сучасного та продуктивного стеку технологій MERN (MongoDB, Express.js, React, Node.js). Кожен із компонентів цієї архітектури відповідає за певну частину функціональності: React — за динамічний та адаптивний інтерфейс користувача, Express.js і Node.js — за серверну логіку, MongoDB — за ефективне зберігання та обробку даних у форматі JSON. Такий вибір технологій забезпечує високу масштабованість, легкість у підтримці та розширенні веб-додатку.

Особливу увагу приділено розробці структури бази даних. Було створено моделі MongoDB для трьох основних сутностей — Users (Користувачі), Locations (Розташування) та Posts (Публікації). Кожна модель чітко визначає атрибути, які зберігаються в системі, а також забезпечує логічні зв'язки між сутностями, необхідні для коректного функціонування системи в цілому. Для взаємодії клієнтської та серверної частин створено REST API, що забезпечує обмін даними через HTTP-запити та дозволяє реалізувати гнучку структуру маршрутизації.

На рівні клієнтської частини веб-сайту спроектовано логічну структуру інтерфейсу, яка забезпечує інтуїтивно зрозумілу навігацію, привабливий дизайн і відповідність принципам UX/UI. Було також враховано вимоги до безпеки: реалізовано механізми автентифікації та авторизації з використанням JWT (JSON Web Token), що дозволяє надійно обмежити доступ до персоналізованих розділів сайту та захистити дані користувачів.

Таким чином, у результаті технічного проектування було створено чітку й послідовну модель веб-додатку, яка відповідає як технічним вимогам, так і очікуванням цільової аудиторії. Розроблене проєктне рішення є надійною основою для практичної реалізації сайту музичної майстерні, спрямованого на підвищення ефективності комунікації з клієнтами, популяризацію послуг майстерні та покращення організації її діяльності.

РОЗДІЛ 3

РОЗРОБКА ДИЗАЙНУ ВЕБ-САЙТУ ДЛЯ МУЗИЧНОЇ МАЙСТЕРНІ

3.1 Створення дизайну та прототипів сторінок веб-сайту музичної майстерні

У процесі розробки веб-сайту музичної майстерні особлива увага була приділена створенню унікального та функціонального дизайну, який гармонійно поєднує естетичні аспекти з практичною зручністю використання. Дизайн сайту відображає сутність музичної майстерні як місця, де традиції зустрічаються з інноваціями, а творчість втілюється у довершених музичних інструментах.

При розробці дизайну веб-сайту музичної майстерні було дотримано кількох ключових принципів. Насамперед, це принцип функціональності, який передбачає, що кожен елемент дизайну має чітке призначення та сприяє навігації користувача по сайту. Другим важливим принципом став принцип гармонії, адже музика та музичні інструменти самі по собі є втіленням гармонії, тому веб-сайт повинен був передати це відчуття через збалансоване розташування елементів, узгодження кольорів та шрифтів.

Кольорова палітра сайту була ретельно підібрана з урахуванням психології сприйняття кольорів та їх асоціативного зв'язку з музикою. Основним акцентним кольором було обрано глибокий червоний (AD0000), який використовувався для кнопок та варіацій логотипу. Цей колір символізує пристрасть, енергію та творчий запал, що є невід'ємними складовими процесу створення музики та музичних інструментів. Червоний колір також привертає увагу користувачів до найважливіших елементів інтерфейсу, таких як кнопки дії, заклики до дії та логотип.

В якості фонових кольорів були використані класичні чорний (000) та білий (fff). Ця комбінація забезпечує оптимальний контраст для читабельності тексту та чіткого сприйняття візуальних елементів. Білий фон створює враження

простору та чистоти, що асоціюється з акустичним звучанням інструментів, а чорний текст на білому фоні забезпечує максимальну читабельність контенту. У деяких розділах, зокрема у блозі та галереї, використовувалося інверсне поєднання — білий текст на чорному фоні, що додає варіативності дизайну та підкреслює різні настрої, пов'язані з музикою — від світлої радості до глибокої меланхолії.

Типографіка відіграє особливу роль у дизайні сайту музичної майстерні. В якості основного шрифту був обраний Roboto Serif, який поєднує класичну елегантність засічок із сучасною читабельністю. Цей шрифт має достатню кількість варіацій товщини, що дозволяє створювати ієрархію контенту без необхідності використання додаткових шрифтів. Засічки Roboto Serif створюють відчуття традиційності та майстерності, що відповідає характеру діяльності музичної майстерні, яка поєднує вікові традиції із сучасними технологіями.

Для логотипу був обраний шрифт Kurale, який відрізняється витонченістю та має каліграфічні елементи. Цей шрифт має виразний характер і надає логотипу майстерні унікальності та запам'ятовуваності. Плавні лінії Kurale асоціюються з музичними нотами та хвилями звуку, що підкреслює спеціалізацію майстерні. Контраст між структурованим Roboto Serif для основного тексту та більш творчим Kurale для логотипу створює цікаву типографічну напругу, яка відображає поєднання технічної майстерності та творчого натхнення у процесі створення музичних інструментів.

Для музичної майстерні було обрано витончену назву «Ліра», із svg-зображенням відповідного інструменту у якості логотипу. Загалом, логотип має мінімалістичний, зрозумілий, але водночас доволі витончений дизайн, що є естетично привабливим і сприяє зануренню клієнтів в атмосферу музичної майстерні. Такий підхід цілком відповідає сучасним стандартам [47]. Логотип було розроблено у двох варіаціях: чорно-білій та бордово-білій. На рисунку 3.1 зображено логотип у кольоровій варіації.



Рис. 3.1. Логотип сайту

Іконографіка сайту була реалізована за допомогою плагіну Iconify, який надає доступ до широкого спектру іконок різних стилів. Для сайту музичної майстерні були обрані лінійні іконки з мінімалістичним дизайном, які гармонійно вписуються у загальну естетику сайту. Особлива увага приділялася іконкам, пов'язаним з музикою та музичними інструментами, які використовувалися для навігації по розділам сайту та візуального підкріплення текстової інформації. Іконки також були стилізовані у червоному кольорі (AD0000) для акцентування уваги користувачів на важливих елементах інтерфейсу.

Головна сторінка сайту музичної майстерні включає наступні розділи:

- Вступ
- Про нас
- Галерея
- Послуги
- Пошук магазинів
- Блог

Розділ «Вступ» призначений для того, щоб зацікавити користувача та занурити його в атмосферу музичної майстерні. Розділ «Про нас» розповідає про історію та цінності компанії. Розділ «Послуги» описує ті послуги, які надає музична майстерня. Розділ «Пошук магазинів» переводить користувача на

сторінку, де він має можливість шукати магазини за їхньою адресою.

Блог музичної майстерні був розроблений як простір для обміну знаннями та новинами у світі музичних інструментів. Дизайн блогу поєднує функціональність з естетикою через чітку структуру статей та виразні заголовки. Кожна стаття представлена у вигляді картки з зображенням, заголовком напівжирного накреслення та короткою анотацією. При розробці дизайну блогу особлива увага приділялася типографіці для забезпечення комфортного читання довгих текстів. Для цього був встановлений оптимальний розмір шрифту, міжрядковий інтервал та довжина рядка.

У шапці сторінки розташовані навігаційні елементи, які дозволяють легко переключатись між розділами. У підвалі сторінки знаходяться посилання на соц мережі компанії.

На рисунку 3.2 зображено головну сторінку сайту в графічному редакторі Figma. Як видно на рисунку, сторінку розбито на окремі елементи, що дозволяє легко адаптувати дизайн до нових вимог, якщо це необхідно. Повну версію головної сторінки наведено в додатку Е.

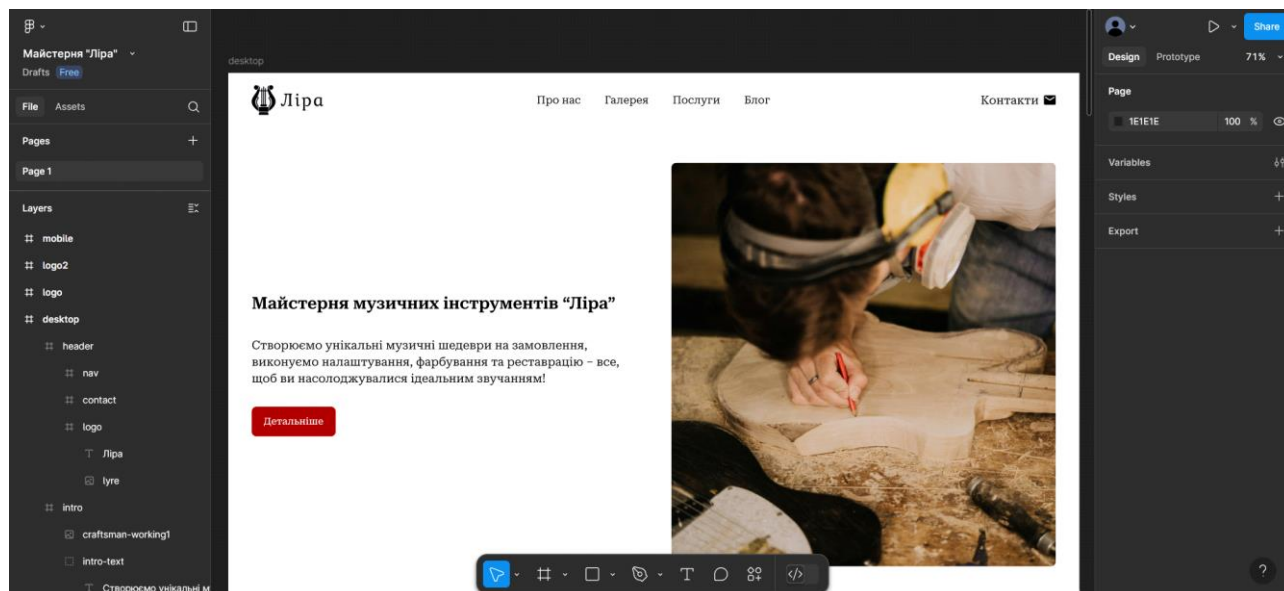


Рис. 3.2. Головна сторінка сайту в графічному редакторі

Важливим аспектом дизайну сайту музичної майстерні стала адаптивність під різні пристрої. Кожен розділ сайту був спроектований з урахуванням

можливості перегляду як на великих екранах, так і на мобільних пристроях. Адаптивний дизайн реалізований через систему гнучких сіток та медіа-запитів, які забезпечують оптимальне відображення контенту на екранах різних розмірів.

Дизайн сайту музичної майстерні є результатом ретельного аналізу потреб цільової аудиторії, трендів веб-дизайну та специфіки галузі. Він поєднує класичну елегантність із сучасними технологіями, створюючи онлайн-представництво, яке гідно представляє майстерність та традиції музичної майстерні у цифровому просторі. Використання стриманої кольорової палітри, елегантною типографіки з поєднанням шрифтів Roboto Serif та Kurale, а також мінімалістичної іконографіки через плагін Iconify створює цілісний візуальний образ, який запам'ятовується та викликає довіру у відвідувачів сайту.

Кожне дизайнерське рішення на сайті музичної майстерні було прийняте з урахуванням не лише естетичних міркувань, але й практичних аспектів юзабіліті та доступності. Це дозволило створити інтерфейс, який є не лише візуально привабливим, але й інтуїтивно зрозумілим для користувачів різного рівня технічної підготовки. Дизайн сайту успішно комунікує цінності та унікальні особливості музичної майстерні, створюючи міст між традиційним ремеслом та сучасними технологіями.

3.2 Розробка та тестування веб-сайту музичної майстерні

Процес розробки веб-сайту для музичної майстерні складався з кількох ключових етапів, що охоплювали планування, вибір технологій, реалізацію функціоналу, налаштування серверної частини, створення клієнтського інтерфейсу, забезпечення зручного адміністрування вмісту, а також налаштування середовища розробки й запуску. Основна мета полягала у створенні сучасного, адаптивного, функціонального та безпечного веб-ресурсу, що дозволяє користувачам ознайомлюватися з послугами майстерні, читати тематичні дописи, переглядати записи та публікації, а адміністраторам – керувати контентом через захищену панель управління.

Проект було реалізовано на базі стеку MERN (MongoDB, Express, React, Node.js), що забезпечив ефективне розділення логіки між фронтендом та бекендом. Проект структуровано у дві частини — client (фронтенд) і server (бекенд). Для забезпечення одночасного запуску обох частин у режимі розробки використано інструмент concurrently, який за допомогою команди `npm run dev` запускає фронтенд через Vite, а сервер через nodemon.

На стороні серверу реалізовано REST API на основі фреймворку Express. Для обробки HTTP-запитів та побудови маршрутизації створено окремі файли у папці routes, зокрема: `auth.js`, `blog.js`, `places.js` та головний маршрутизатор `routes.js`. Робота з базою даних MongoDB здійснюється за допомогою бібліотеки `mongoose`, що дозволяє створювати моделі сутностей — User, Post, Place. Кожна з них відповідає окремому типу інформації: користувачі, публікації блогу, заклади або сервіси, пов'язані з музичною майстернею. Для завантаження медіа-файлів у систему інтегровано middleware `m multer`, що відповідає за обробку та збереження файлів у директорії `uploads`.

Аутентифікація користувачів реалізована через `jsonwebtoken` у поєднанні з `bcryptjs` для хешування паролів. Система підтримує реєстрацію та вхід користувачів, захищаючи окремі маршрути за допомогою middleware `authMiddleware.js`, яке перевіряє валідність токенів. Конфігурація сервера та середовища зберігається у файлі `.env`, що містить змінні середовища, такі як URI підключення до бази даних або секретний ключ для JWT.

На клієнтській стороні застосовано бібліотеку React (версія 19), що забезпечує побудову компонентного інтерфейсу та керування станом додатка. Основні сторінки зберігаються у папці `pages`, зокрема: `Home.jsx`, `PostPage.jsx`, `SearchPage.jsx`. Окрім цього, реалізовано набір багаторазових компонентів у папці `components`, які відповідають за візуальні блоки, як-от Header, Footer, BlogSlider, BlogVideo тощо. Для організації маршрутизації використано `react-router-dom`, що дозволяє користувачеві переміщатися між сторінками без перезавантаження сайту.

Особливу увагу приділено створенню адміністративної панелі (admin), яка

дозволяє авторизованим користувачам додавати нові пости (AddPost.jsx), редагувати наявні (EditPost.jsx), шукати записи (SearchManager.jsx), а також керувати загальним вмістом сайту через компонент AdminPanel.jsx. Сторінки входу та реєстрації (Login.jsx, Register.jsx) надають базову функціональність для керування доступом до адмін-панелі.

Для HTTP-запитів між клієнтом і сервером використано бібліотеку axios, а для зручного форматування дат — date-fns. Візуальне оформлення реалізовано за допомогою CSS-фреймворку Tailwind CSS, який дає змогу швидко створювати адаптивний дизайн із чистим кодом. Під час розробки використовувалась flexbox-верстка, яка допомогла створити резинову верстку без необхідності використання зайвих медіа-запитів у CSS. Також інтегровано слайдер-каруселі через бібліотеку swiper.

Для керування якістю коду на фронтенді застосовано eslint разом із відповідними плагінами, що забезпечує відповідність коду сучасним стандартам. Конфігурації для ESLint зберігаються у файлі eslint.config.js.

Збірка та підготовка до продакшну здійснюється за допомогою команди npm run prod, що включає в себе побудову фронтенду (vite build) та запуск продакшн-сервера (cross-env NODE_ENV=production node server/index.js).

Таким чином, сайт музичної майстерні реалізовано як повнофункціональний веб-додаток з розділеною архітектурою, що забезпечує ефективну взаємодію користувача з ресурсом, просту адмін-роботу та гнучкість у подальшому розширенні функціоналу. Враховано сучасні вимоги до дизайну, безпеки, масштабованості та продуктивності.

На рисунку 3.3 зображено файлову структуру розробленого веб-сайту, що демонструє грамотну архітектуру та гнучкий підхід до організації додатку.

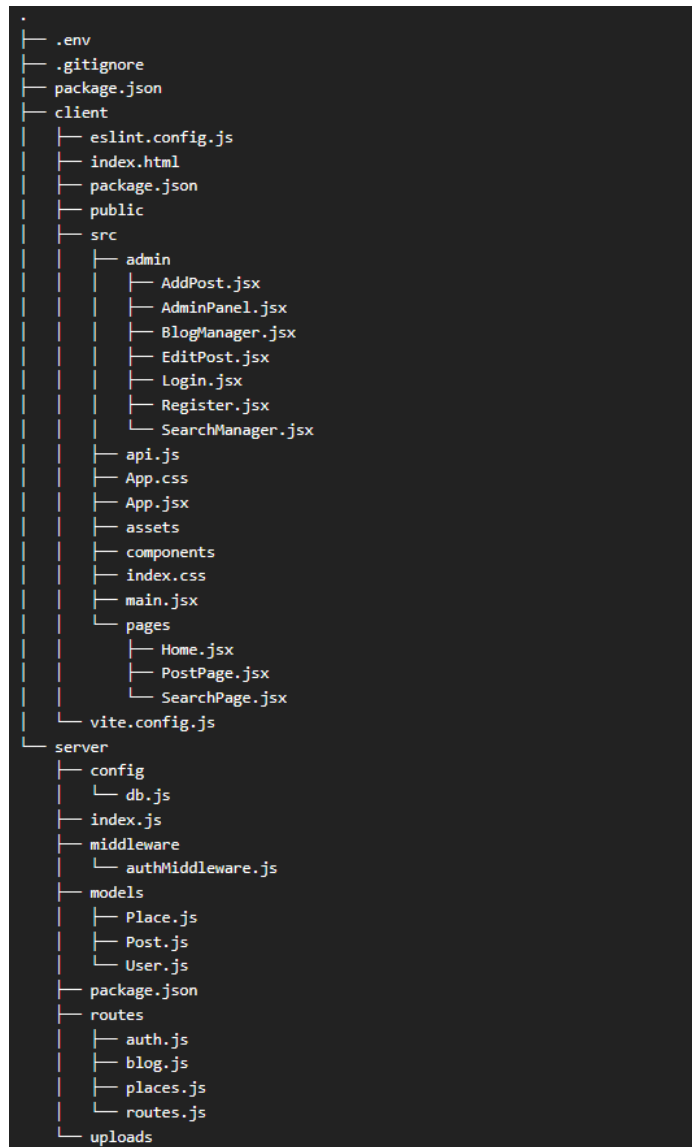


Рис. 3.3. Файлова структура сайту

Після завершення основного етапу розробки веб-сайту музичної майстерні важливим етапом стала процедура тестування. Мета тестування — виявити можливі помилки у роботі як серверної, так і клієнтської частини проєкту, перевірити коректність реалізації функціоналу, забезпечити безперебійну взаємодію між компонентами системи, а також оцінити загальну зручність та інтуїтивність інтерфейсу для кінцевого користувача. Тестування здійснювалося вручну та частково автоматизовано, з урахуванням як функціональних, так і нефункціональних вимог.

Першим етапом було локальне тестування серверної частини, яка відповідала за обробку запитів, зберігання та пошук даних у базі MongoDB, а

також авторизацію користувачів. За допомогою утиліт Postman та Insomnia були протестовані всі основні API-ендпоінти, що включають запити GET, POST, PUT та DELETE.

Під час тестування виявлено декілька критичних і незначних помилок. Зокрема, на початковому етапі API для реєстрації не перевіряв повторне використання одного і того ж імені користувача або номера телефону, що дозволяло створювати дублікати. Було додано перевірку у відповідний маршрут `auth.js`, яка попереджає користувача про те, що акаунт з такими даними вже існує.

Також виявлено помилку в `middleware authMiddleware.js`, яка некоректно обробляла відсутність токена — замість повернення статусу 401 поверталось внутрішнє повідомлення про помилку. Це було виправлено шляхом додавання додаткової перевірки на наявність заголовка `Authorization`.

На етапі тестування роботи з файлами через `multer` виявлено проблему з обробкою одночасних запитів на завантаження: при відсутності правильного MIME-типу зображення система зберігала порожні файли. Було впроваджено перевірку типу файлу перед збереженням, а також додано логування у консоль для відслідковування таких запитів.

Далі проводилося фронтенд-тестування, яке включало перевірку роботи інтерфейсу в реальних умовах. Було протестовано основні сценарії використання, включно з:

- Навігацією між сторінками (Home, PostPage, SearchPage);
- Роботою слайдерів та інтерактивних компонентів (побудованих на бібліотеці `swiper`);
- Відображенням медіаконтенту;
- Заповненням та обробкою форм реєстрації/входу;
- Додаванням нового поста через панель адміністратора;
- Пошуком та редагуванням наявних записів.

Однією з основних проблем була некоректна робота React Router з динамічними маршрутами: при оновленні сторінки напряму (наприклад, `site.com/post/123`) сайт не завантажувався. Це було вирішено налаштуванням

проксі та редиректів у конфігурації `vite.config.js`, щоб сервер віддавав `index.html` для всіх маршрутів, що не є API-запитами.

Ще однією помилкою була відсутність повідомлень про помилки у формах (наприклад, при введенні неправильного пароля). Було додано систему сповіщень, яка виводить повідомлення користувачеві про успішність або невдачу авторизації, помилки валідації тощо. Це значно покращило досвід взаємодії з сайтом.

Також було проведено перевірку адаптивності інтерфейсу за допомогою інструментів DevTools у браузерях Google Chrome та Firefox. Було виявлено, що деякі елементи (зокрема, блоки зображень та футер) не масштабувалися відповідно до ширини екрану на мобільних пристроях. Було відкориговано класи TailwindCSS, додано правила `flex-wrap` і адаптивні брейки (`sm`, `md`, `lg`), що забезпечило повну підтримку мобільних та планшетних пристроїв.

Особлива увага приділялася перевірці безпеки: переконанося, що токени не зберігаються у відкритому вигляді, доступ до адмін-панелі мають лише авторизовані користувачі, запити не можуть бути підроблені. Було реалізовано обробку помилок на рівні серверу, що запобігає виведенню `stack trace` у відповідь користувачеві.

Також здійснювалося тестування продуктивності. Було перевірено час завантаження головної сторінки, реакцію на запити, рендеринг компонентів. За допомогою інструменту Lighthouse [48] проведено аудит, який показав високу швидкість завантаження (індекс продуктивності перевищував 90 балів), однак були рекомендації щодо оптимізації розміру зображень, що в подальшому також було враховано шляхом компресії графічних файлів.

На завершальному етапі виконувалося тестування інтеграції — перевірялася злагоджена робота всіх частин системи: від заповнення форм і авторизації, до додавання контенту, його збереження в базу даних, відображення на головній сторінці, і зворотної взаємодії з користувачем. Усі виявлені неузгодженості були усунуті.

Таким чином, тестування веб-сайту музичної майстерні дозволило не лише

виявити та усунути критичні недоліки у функціональності та інтерфейсі, а й забезпечити стабільну, захищену та зручну роботу системи як для адміністратора, так і для звичайного користувача. Після успішного завершення всіх тестових сценаріїв сайт було підготовлено до розгортання у продакшн-середовищі.

3.3 Вимоги до технічного забезпечення та розгортання сайту музичної майстерні

Для ефективного розгортання та стабільної роботи веб-сайту музичної майстерні необхідно забезпечити відповідне технічне середовище на сервері, яке відповідатиме характеристикам і потребам проєкту. Веб-сайт створено на основі стеку MERN (MongoDB, Express.js, React, Node.js), що передбачає наявність серверного середовища з підтримкою JavaScript/Node.js, а також бази даних MongoDB. Оскільки очікуване навантаження є помірним, а обсяг даних — обмеженим, вимоги до обладнання є відносно невисокими, однак повинні враховувати базові принципи продуктивності, стабільності та безпеки.

На момент завершення розробки повний обсяг сайту (включаючи зображення, JavaScript-компоненти, стилі та ресурси бази даних) становить приблизно 280 МБ. Цей обсяг включає клієнтську збірку React-додатку, завантажені мультимедійні файли (зображення до блогів і сторінок закладів), а також дані користувачів, публікацій і локацій, які зберігаються в MongoDB.

Беручи до уваги потребу в одночасному обслуговуванні декількох користувачів, а також необхідність підтримки постійного з'єднання для авторизації, перегляду публікацій, пошуку та взаємодії з адміністративною панеллю, можна обчислити мінімальні технічні вимоги до сервера.

Процесор (CPU) повинен мати 2 ядра з тактовою частотою не менше 2.0 ГГц (наприклад, Intel Xeon, AMD Ryzen або еквівалент ARM-процесор для VPS/VDS). Це забезпечить достатню обчислювальну потужність для одночасного оброблення запитів і обслуговування Node.js-програми.

Оперативна пам'ять (RAM) – не менше 2 ГБ. У випадку одночасної роботи Node.js-сервера, MongoDB та клієнтського сервера (наприклад, при попередньому рендерингу або використанні SSR), рекомендовано 4 ГБ для уникнення перевантаження пам'яті.

Дисковий простір – мінімум 1 ГБ вільного простору з урахуванням резерву для бази даних та завантаження нових медіафайлів. Формально достатньо 500 МБ поверх уже зайнятих 280 МБ, однак для стабільної роботи рекомендовано SSD-диск обсягом 10 ГБ або більше, що забезпечує швидке читання/запис.

Операційна система – не має великого значення, втім бажано, щоб це був Linux-дистрибутив (наприклад, Ubuntu Server 20.04 або вище). Linux забезпечує надійність, безпеку та широку підтримку для роботи з Node.js та MongoDB.

Підключення до Інтернету – стабільний канал з пропускнуою здатністю не менше 10 Мбіт/с. Враховуючи наявність мультимедійного контенту, бажано мати безлімітний трафік і не менше 100 Мбіт/с у пік навантаження.

Програмне забезпечення:

- Node.js версії 18 або новішої;
- MongoDB 6.0 або новіша (локально або як віддалений хостинг);
- PM2 або інший процес-менеджер для запуску та моніторингу Node.js-програми;
- Сертифікати SSL (Let's Encrypt або аналогічні) для захисту трафіку;
- Система керування файлів (наприклад, scp, rsync, FileZilla або CI/CD-інтеграція).

За наявності додаткових вимог до масштабування або надання доступу кільком адміністраторам (наприклад, для додавання контенту), може розглядатися варіант із контейнеризацією (Docker) або розгортанням на хмарних платформах (Heroku, Vercel, DigitalOcean, AWS EC2 тощо).

З урахуванням помірного потоку користувачів і невеликого обсягу даних, достатньо базового віртуального сервера (VPS) із наведеними параметрами. Такий сервер забезпечить швидкий час відгуку, доступність сервісу цілодобово, а також можливість розширення в разі зростання потреб проєкту.

Таким чином, обрана конфігурація дозволяє зберігати стабільну роботу веб-сайту, забезпечити безпечний доступ до контенту, підтримку авторизації та обробку запитів без затримок, навіть у разі одночасного підключення кількох користувачів [49].

Для комфортної роботи з веб-сайтом музичної майстерні з боку користувача-клієнта не потрібно спеціалізованого або високопродуктивного обладнання. Оскільки сайт реалізовано як SPA-додаток (single-page application) з використанням бібліотеки React, усі основні обчислення виконуються на боці клієнта в браузері. Тому основні вимоги стосуються сумісності з сучасними веб-технологіями та стабільного інтернет-з'єднання.

Мінімальні вимоги до клієнтського пристрою:

- Операційна система: Windows 10, macOS 10.13, Linux (будь-який сучасний дистрибутив), Android 9.0+, iOS 12+.
- Оперативна пам'ять (RAM): від 2 ГБ.
- Процесор: будь-який двоядерний процесор з тактовою частотою від 1.5 ГГц.
- Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari — останніх двох-трьох версій.
- Інтернет-з'єднання: стабільне підключення зі швидкістю не менше 5 Мбіт/с для швидкого завантаження зображень, відео та сторінок.

За таких умов користувачі зможуть безперешкодно переглядати сторінки, шукати заклади, читати публікації та взаємодіяти з інтерфейсом веб-сайту на всіх типах пристроїв — як настільних, так і мобільних.

Після завершення етапів розробки, тестування та підготовки клієнтської і серверної частин веб-сайту музичної майстерні постало завдання впровадження проєкту в реальне середовище для забезпечення його доступності користувачам. З метою розгортання сайту було обрано хостинг-платформу Render [50], яка пропонує зручний інтерфейс, автоматичне масштабування, підтримку Node.js, MongoDB та можливість безперебійного розміщення як статичних сайтів, так і повноцінних веб-сервісів. У рамках цього проєкту сайт було розгорнуто як Web

Service — це дозволило окремо керувати серверною частиною (Node.js + Express) та базою даних, а також налаштувати середовище виконання, змінні середовища, автоматичне розгортання з Git-репозиторію та моніторинг ресурсу.

Render було обрано з кількох причин. По-перше, платформа забезпечує безкоштовний стартовий план, що є оптимальним для проєктів із помірним навантаженням, таких як цей сайт. По-друге, Render підтримує автоматизацію процесу CI/CD: кожне оновлення в репозиторії GitHub або GitLab може автоматично ініціювати нове розгортання, що значно спрощує внесення змін і їх тестування на реальному сервері. Крім того, Render надає інтегровані механізми захисту, такі як автоматичне SSL-шифрування та ізоляція середовищ, що забезпечує базову безпеку даних користувачів і стабільність сервісу. Окремою перевагою є просте налаштування змінних середовища, необхідних для роботи з JWT, MongoDB, CORS і іншими параметрами, що впливають на роботу API.

Щоб підтримувати стабільну роботу сайту після його впровадження, рекомендується періодично перевіряти логи сервера, які Render зберігає в реальному часі. Це дозволяє оперативно виявляти помилки, пов'язані з мережею, базою даних або некоректними запитами. Також доцільно контролювати використання ресурсів — оперативної пам'яті, процесора та трафіку, особливо у випадку зростання кількості користувачів. Регулярне резервне копіювання бази даних MongoDB забезпечить збереження важливої інформації навіть у разі непередбачених збоїв. Для забезпечення безперервної підтримки сайту бажано оновлювати залежності проєкту відповідно до нових версій бібліотек, особливо тих, що відповідають за безпеку та маршрутизацію запитів, таких як jsonwebtoken, mongoose або express.

Таким чином, впровадження веб-сайту на платформі Render забезпечило ефективний і безпечний запуск проєкту в мережі Інтернет, відкривши доступ до ресурсу для цільової аудиторії з будь-яких пристроїв. Розгортання як Web Service стало технічно доцільним рішенням для MERN-додатку, дозволяючи легко масштабувати його функціональність у майбутньому.

Висновки до 3 розділу

У цьому розділі кваліфікаційної роботи було реалізовано основний функціонал веб-сайту для музичної майстерні, що передбачає створення повноцінного онлайн-ресурсу з підтримкою авторизації, керування контентом та збереженням даних у базі даних. Основну увагу було зосереджено на побудові клієнтської та серверної частин застосунку з урахуванням сучасних вимог до безпеки, ефективності та зручності використання.

На етапі розробки функціоналу було впроваджено механізм реєстрації та входу користувачів із використанням технології JSON Web Token (JWT). Такий підхід дозволяє надійно ідентифікувати користувача в системі, зберігаючи при цьому безпеку персональних даних. Важливою перевагою використання JWT є можливість розмежування доступу до ресурсів системи відповідно до рівня авторизації, що відкриває перспективи для подальшого розвитку сайту, наприклад, упровадження ролей (адміністратор, редактор, гість тощо).

Для збереження інформації про користувачів, публікації та заклади було використано нереляційну базу даних MongoDB, яка добре підходить для веб-проектів з динамічною структурою даних. Створено відповідні моделі — User, Location та Post, які забезпечують ефективне структурування інформації та зручний доступ до неї з боку сервера. Всі операції з даними — створення, редагування, видалення — реалізовані з урахуванням REST-архітектури, що підвищує гнучкість системи та спрощує її підтримку й масштабування.

Крім серверної логіки, розроблено і реалізовано клієнтську частину веб-додатку з використанням бібліотеки React. Інтерфейс розроблений таким чином, щоб забезпечити інтуїтивно зрозуміле користування для відвідувачів сайту, з мінімальною кількістю кроків для досягнення цільових дій (перегляду закладів, створення публікацій тощо). Було реалізовано функціонал додавання нових постів і закладів із зазначенням автора, який автоматично підставляється на основі токена, що зберігається після входу в систему. Це забезпечує персоналізовану взаємодію з платформою.

Загалом, розроблений функціонал становить міцну основу для подальшого розвитку веб-сайту музичної майстерні. Система є масштабованою, модульною і такою, що відповідає сучасним вимогам до розробки веб-застосунків. У наступних етапах реалізації можна передбачити додаткові можливості, такі як рейтингування закладів, коментування постів, інтеграцію з картами для геолокації майстерень, розширення профілів користувачів тощо. Це дозволить перетворити сайт не лише на інформаційний ресурс, а й на повноцінну соціальну платформу для музикантів, викладачів і клієнтів.

ЗАГАЛЬНІ ВИСНОВКИ

У процесі виконання кваліфікаційної роботи на тему «Розробка дизайну веб-сайту для музичної майстерні» було комплексно розглянуто, проаналізовано та реалізовано всі етапи створення сучасного інформаційного ресурсу, орієнтованого на потреби музичної спільноти. Робота охоплює як теоретичні аспекти розробки веб-додатків, так і практичну реалізацію повноцінного веб-сайту з урахуванням особливостей предметної області, вимог цільової аудиторії та актуальних технологічних рішень.

У першому розділі було визначено специфіку предметної області — діяльності музичних майстерень. Проведено аналіз функціональних і структурних особливостей організацій, що надають послуги у сфері музики, зокрема індивідуального навчання, ремонту музичних інструментів, організації подій тощо. Було з'ясовано, що сучасні музичні майстерні потребують ефективного онлайн-представництва, яке дозволяло б не лише поширювати інформацію про послуги, а й забезпечувати зручний канал зв'язку з потенційними клієнтами.

Розглянуто теоретичні засади створення дизайну для веб-сайтів подібного типу. Проаналізовано принципи UX/UI-дизайну, адаптивності інтерфейсу, візуальної привабливості, а також вимоги до структури сторінок, що сприяють зручності користування ресурсом. Було проведено аналіз сайтів-конкурентів із метою виявлення сильних і слабких сторін уже існуючих рішень, що дозволило сформулювати рекомендації для розробки власного унікального продукту, конкурентоспроможного в умовах сучасного цифрового простору.

У другому розділі акцент було зроблено на побудові логіки та структури майбутнього веб-сайту. Детально проаналізовано цільову аудиторію — музикантів, викладачів, учнів музичних шкіл та їхніх батьків, потенційних клієнтів майстерні. Це дало змогу визначити ключові функції, які мають бути реалізовані на сайті та ухвалити рішення щодо концепції майбутнього програмного продукту. Було спроектовано об'єкти системи (користувачі, пости,

заклади) та зв'язки між ними, що лягло в основу створення структури бази даних.

Окрему увагу приділено вибору інструментів і технологій для реалізації веб-додатку. Зокрема, для клієнтської частини обрано бібліотеку React, яка забезпечує динамічність і гнучкість інтерфейсу, а для серверної частини — середовище Node.js з Express. У якості бази даних використано MongoDB, що дозволило реалізувати гнучку схему зберігання даних. Вибір цих технологій обґрунтовано їхньою популярністю, ефективністю, активною підтримкою спільноти розробників та відповідністю вимогам проєкту.

Третій розділ присвячено безпосередній розробці веб-сайту. Було розроблено концепцію дизайну, який було реалізовано у вигляді макетів сторінок через Figma з урахуванням попередніх теоретичних напрацювань. Інтерфейс орієнтовано на зручність використання, інтуїтивну навігацію та привабливий зовнішній вигляд. Реалізовано адміністративну панель з функціоналом для авторизації з JWT-токенами, можливість створення та перегляду публікацій і закладів, що забезпечує базові можливості взаємодії користувача з системою. Усі операції з даними реалізовано за REST-принципами, що забезпечує модульність, розширюваність та зручність у тестуванні.

Під час розробки значна увага приділялася питанням безпеки (захист даних користувачів, перевірка прав доступу), зручності підтримки коду (розділення на модулі та компоненти, дотримання принципів чистої архітектури) та подальшого масштабування проєкту. Проведено тестування окремих компонентів, оцінено відповідність інтерфейсу сучасним вимогам доступності.

У результаті виконання кваліфікаційної роботи було:

- здійснено всебічний аналіз предметної області та конкурентного середовища;
- визначено цільову аудиторію та сформульовано вимоги до майбутнього продукту;
- обґрунтовано вибір технологій і засобів реалізації;
- спроектовано логіку взаємодії об'єктів системи та структуру бази

даних;

- розроблено повноцінний веб-сайт для музичної майстерні з функцією авторизації, публікації інформації та зручним інтерфейсом;
- забезпечено технічні та візуальні умови для подальшого розширення функціоналу платформи.

Таким чином, поставлену мету кваліфікаційної роботи — створення сучасного веб-сайту для музичної майстерні — досягнуто повною мірою. Результати виконаної роботи можуть бути використані як основа для впровадження реального веб-продукту в комерційну або просвітницьку діяльність у сфері музики. Крім того, запропоноване рішення може бути масштабованим на інші типи творчих майстерень або освітніх установ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. 1WorldSync. Shopping Trends 2023: 66% of Consumers Migrate from In-Store to Online Purchases. URL: <https://retailsee.com/shopping-trends-2023> (дата звернення: 29.04.2025).
2. BrightLocal. 31 Local SEO Statistics You Need for 2025. URL: <https://www.brightlocal.com/resources/local-seo-statistics/> (дата звернення: 29.04.2025).
3. Dastweb. Statistics of small businesses on the Internet from 2023 (From the point of consumers and service providers). URL: <https://www.dastweb.me/blog/statistics-of-small-businesses-on-the-internet-from-2023-from-the-point-of-consumers-and-service-providers-1> (дата звернення: 29.04.2025).
4. Петренко, В. М. Методологія проектування веб-систем для малого та середнього бізнесу. Київ: Видавництво "ІТ-Освіта", 2023. 312 с.
5. Коваленко, О. П. Специфіка розробки веб-рішень для підприємств культурної сфери. Вісник Національного університету "Львівська політехніка". Серія: Інформаційні системи та мережі. 2023. № 4: С. 117–128.
6. Дмитренко, К. В. Архітектурні підходи до розробки адаптивних веб-систем для спеціалізованих підприємств. Збірник наукових праць Інституту проблем моделювання в енергетиці ім. Г. Є. Пухова. 2022. № 12: С. 89–97.
7. Nielsen, J., Loranger, H. Prioritizing Web Usability. New Riders Press, 2021. 432 p.
8. Krug, S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. Pearson Education, 2023. 216 p.
9. Garrett, J. J. The Elements of User Experience: User-Centered Design for the Web and Beyond. New Riders, 2022. 240 p.
10. Шевченко, Л. І., Вороніна, А. К. Інформаційні технології в креативних індустріях України. Наукові записки НаУКМА: Економічні науки. 2023. № 8: С. 76–83.

11. Морозова, Т. О. Особливості веб-представництва підприємств музичної індустрії. Інформаційні технології і засоби навчання. 2022. № 5(91): С. 156–168.
12. Anderson, K. Digital Marketing for Creative Industries. Routledge, 2023. 384 p.
13. Brown, D. Designing for the Arts: Web Solutions for Creative Businesses. A Book Apart, 2022. 176 p.
14. Ніколаєва, Л. В. Цифровізація малого та середнього бізнесу в умовах глобалізації. Економіка та суспільство. 2023. № 2(47): С. 32–40.
15. Дорошенко, І. С. Музичні інструменти: технології виробництва та методики реставрації. Київ: Музична Україна, 2022. 248 с.
16. Кострицький, В. В. Цифрова трансформація бізнес-моделей у креативних індустріях. Економіка та держава. 2021. № 5: С. 82–88.
17. Бойчук, І. В., Музика, О. М. Інтернет в маркетингу: підручник. Київ: Центр учбової літератури, 2020. 512 с.
18. Fain, M., Moher, D. Progressive Web Applications: Architecture and Development. O'Reilly Media, 2023. 320 p.
19. Lalwani, T., Smith, K. PWA Conversion Rates: A Statistical Analysis. Journal of Web Engineering. 2023. Vol. 18, № 3: P. 247–262.
20. Geyer, M. Micro Frontends in Action. Manning Publications, 2022. 328 p.
21. Комінарець, В. Модульна архітектура у веб-проектах: практичні аспекти впровадження. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2023. № 6: С. 112–121.
22. Roberts, M. Serverless Architectures on AWS. Manning Publications, 2023. 400 p.
23. Sbarski, P. Serverless Architectures on AWS: With examples using AWS Lambda. Manning Publications, 2022. 376 p.
24. Wroblewski, L. Mobile First. A Book Apart, 2021. 144 p.
25. Marcotte, E. Responsive Web Design: Patterns and Principles. A Book

Apart, 2022. 160 p.

26. Birnbaum, N. Modern JavaScript Frameworks: A Comparative Analysis. International Journal of Web Development. 2023. Vol. 7, № 2: P. 78–95.

27. Holloman, A. Single Page Applications: Benefits and Implementation Strategies. Web Technologies Today. 2022. Vol. 14, № 4: P. 218–234.

28. Line, M. API-First Development: Principles and Practices. O'Reilly Media, 2023. 256 p.

29. Barclay, J. Headless CMS: The Future of Content Management. Journal of Content Strategy. 2022. Vol. 8, № 1: P. 34–48.

30. Heydon, H. Inclusive Design Patterns: Coding Accessibility Into Web Design. Smashing Magazine, 2023. 312 p.

31. Clark, A. Designing for All: Principles of Inclusive Web Interfaces. International Journal of Human-Computer Interaction. 2022. Vol. 12, № 4: P. 312–328.

32. Kim, G. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press, 2023. 480 p.

33. Hamblin, R. DevOps for Web Development: Practical Approaches. O'Reilly Media, 2022. 328 p.

34. Grigsby, J. Optimizing Web Performance: A Practical Guide. O'Reilly Media, 2023. 296 p.

35. Crimson Guitars. URL: <https://www.crimsonguitars.com/> (дата звернення: 29.04.2025).

36. Stewart-MacDonald. Guitar tools, parts and supplies. URL: <https://www.stewmac.com/> (дата звернення: 29.04.2025).

37. Violin Makers School of America. URL: <https://vmsa.net/> (дата звернення: 29.04.2025).

38. Thomann. Music Shop – the Online Music Store. URL: <https://www.thomann.de/intl/index.html> (дата звернення: 29.04.2025).

39. BrowserStack. How fast should a website load? URL: <https://www.browserstack.com/guide/how-fast-should-a-website-load> (дата звернення: 03.05.2025).
40. Цільова аудиторія – що це: приклади, види, аналіз ЦА / Ukrainian Digital. URL: <https://ukrainiandigital.com/tsilova-audytoriia/> (дата звернення: 02.05.2025).
41. Цільова аудиторія: що це, види та як визначити / WebTune. URL: <https://webtune.com.ua/statti/internet-marketing/czilova-audytoriia-v-internet-marketyngu/> (дата звернення: 02.05.2025).
42. Herron A. Full-Stack React Projects: Modern Web Development Using React 16, Node, Express, and MongoDB. Packt Publishing, 2018. 354 p.
43. Murray S. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017. 616 p.
44. Omri Luzon. Understanding JSON Web Tokens (JWT) in JavaScript. Dev.to. URL: <https://dev.to/omriluz1/understanding-json-web-tokens-jwt-in-javascript-12je> (дата звернення: 03.05.2025).
45. NamasteDev. Webpack vs Vite: Which is better?. Namaste Dev. URL: <https://namastudev.com/blog/webpack-vs-vite-which-is-better/> (дата звернення: 03.05.2025).
46. NPM Compare. @glidejs/glide, slick-carousel, swiper. URL: <https://npm-compare.com/@glidejs/glide,slick-carousel,swiper> (дата звернення: 03.05.2025).
47. Романюк О.Н., Катільніков Д.І., Косовець О. П. Веб-дизайн і комп'ютерна графіка. Навчальний посібник. Вінниця: ВНТУ, 2007. 142 с.
48. Google Developers. Lighthouse: інструмент для аудиту веб-додатків. URL: <https://developer.chrome.com/docs/lighthouse/> (дата звернення: 06.05.2025).
49. DreamHost. Хостинг додатка: що потрібно знати для першого запуску. 2024. URL: <https://www.dreamhost.com/blog/uk/hosting-dodatka/> (дата звернення: 06.05.2025).
50. Render. URL: <https://render.com/> (дата звернення: 06.05.2025).

ДОДАТКИ

Додаток А

Всеукраїнська науково-практична конференція здобувачів вищої освіти і молодих вчених
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА МОДЕЛЮВАННЯ СИСТЕМ»

ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ

Болгова Д. П., здобувач¹
першого (бакалаврського) рівня
ОП «Мультимедійний дизайн»

Київський національний університет технологій та дизайну

ОСОБЛИВОСТІ UX/UI ДИЗАЙНУ ВЕБ-САЙТУ МУЗИЧНОЇ МАЙСТЕРНІ ДЛЯ
СТВОРЕННЯ ЕМОЦІЙНОГО ЗВ'ЯЗКУ З КОРИСТУВАЧАМИ

Вступ. У сучасному цифровому середовищі користувачі очікують не лише ефективного функціонування веб-сайту, а й емоційного залучення, що забезпечує глибший рівень взаємодії з брендом. Особливо це актуально для сайтів музичних майстерень, які представляють не просто продукт, а творчість, філософію та особисту майстерність. Емоційний UX/UI-дизайн стає ключовим інструментом у створенні тривалого враження та формуванні емоційного зв'язку з користувачем.

Особливо важливим аспектом є дизайн користувацького досвіду (UX) та користувацького інтерфейсу (UI), які визначають спосіб взаємодії користувачів із ресурсом та їхню емоційну залученість. Враховуючи емоційний і творчий характер музичної діяльності, особливості UX/UI дизайну мають не лише сприяти зручності навігації, а й викликати позитивні емоції, створювати атмосферу натхнення та довіри.

Мета даної роботи полягає у виявленні ключових особливостей UX/UI дизайну веб-сайту музичної майстерні, які сприяють формуванню глибокого емоційного зв'язку між користувачами та ресурсом. Важливим є аналіз сучасних трендів та практик дизайну, що враховують психологічні особливості цільової аудиторії та створюють неповторну атмосферу, котра підсилює інтерес та залучення. Саме через гармонійне поєднання функціональності, естетики та емоційної складової можливо досягти високої ефективності комунікації та активації позитивних асоціацій з брендом.

Виклад основного матеріалу. Однією з ключових особливостей UX/UI дизайну для веб-сайту музичної майстерні є створення емоційно привабливої та інтуїтивно зрозумілої навігації, що дозволяє користувачам легко орієнтуватися у просторі ресурсу та швидко знаходити необхідну інформацію. Візуальні рішення, такі як кольорова палітра, типографіка та ілюстрації, відіграють важливу роль у формуванні настроїв та настрою, що відповідає характеру музичної діяльності. Наприклад, теплі та насичені кольори сприяють створенню відчуття затишку і натхнення, тоді як мінімалістичний дизайн підкреслює професійність і сучасність.

Ще одним важливим аспектом є використання елементів руху та анімацій, що підсилюють емоційний вплив сайту. Наприклад, плавні переходи, динамічні інтерфейсні ефекти або інтерактивні візуальні підказки сприяють залученню користувача та створюють відчуття живої та енергійної музичної атмосфери. Крім того, важливим є оптимізація мобільної версії — сучасний користувач часто взаємодіє з сайтом через смартфон або планшет, тому адаптивний дизайн має забезпечувати комфорт та емоційний комфорт на будь-якому пристрої.

Особливу увагу слід приділити елементам взаємодії, що викликають позитивні емоції: інтерактивним формам, можливості залишати коментарі чи відгуки, а також персоналізації контенту (наприклад, рекомендації на основі попередніх дій користувача). Всі ці підходи сприяють формуванню довіри, залученості та прихильності до бренду музичної майстерні.

¹ Науковий керівник – доцент кафедри мультимедійного дизайну Київського національного університету технологій та дизайну, кандидат педагогічних наук, доцент Шаура А. Ю.

Рис. А.1. Збірник наукових тез



Рис. Б.1. Сертифікат учасника конференції

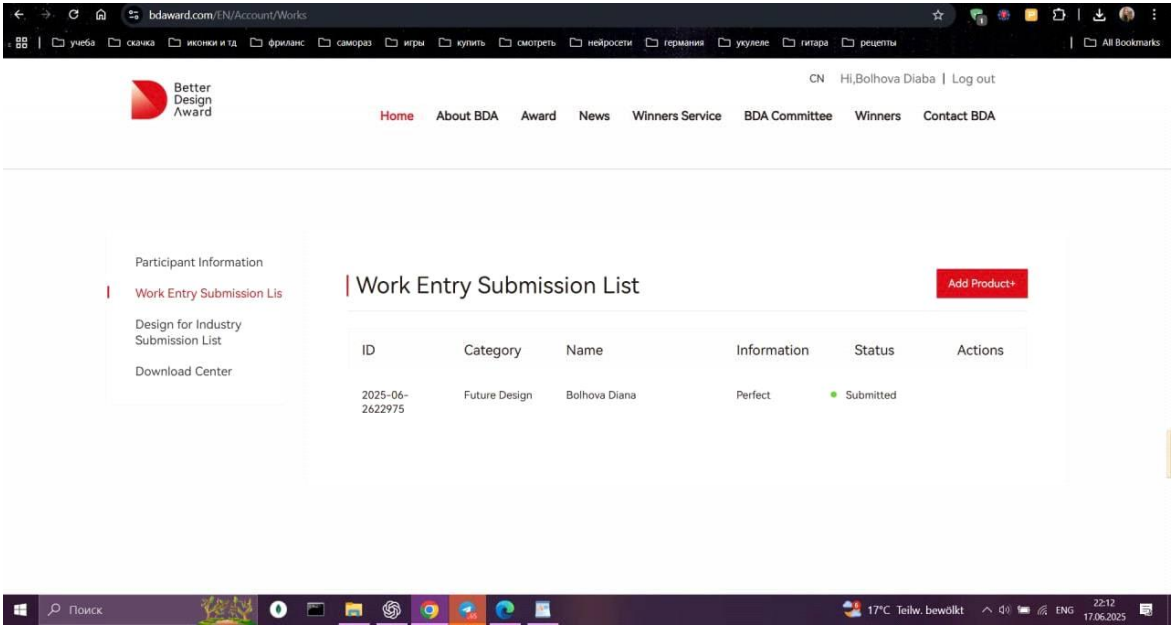


Рис. В.1. Лист-підтвердження учасника конкурсу

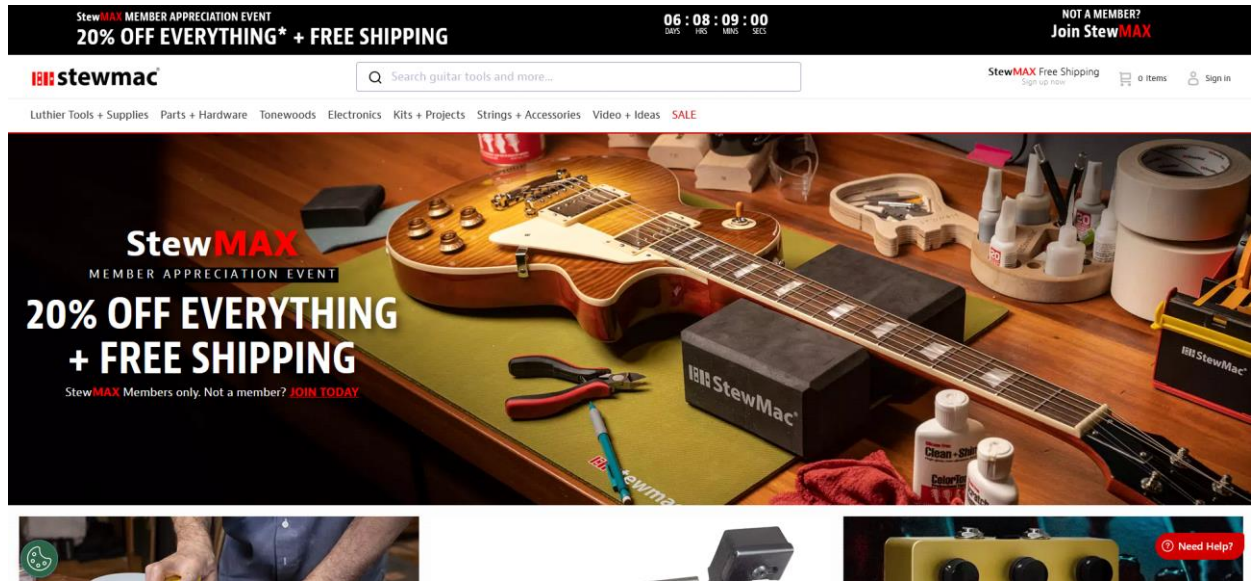
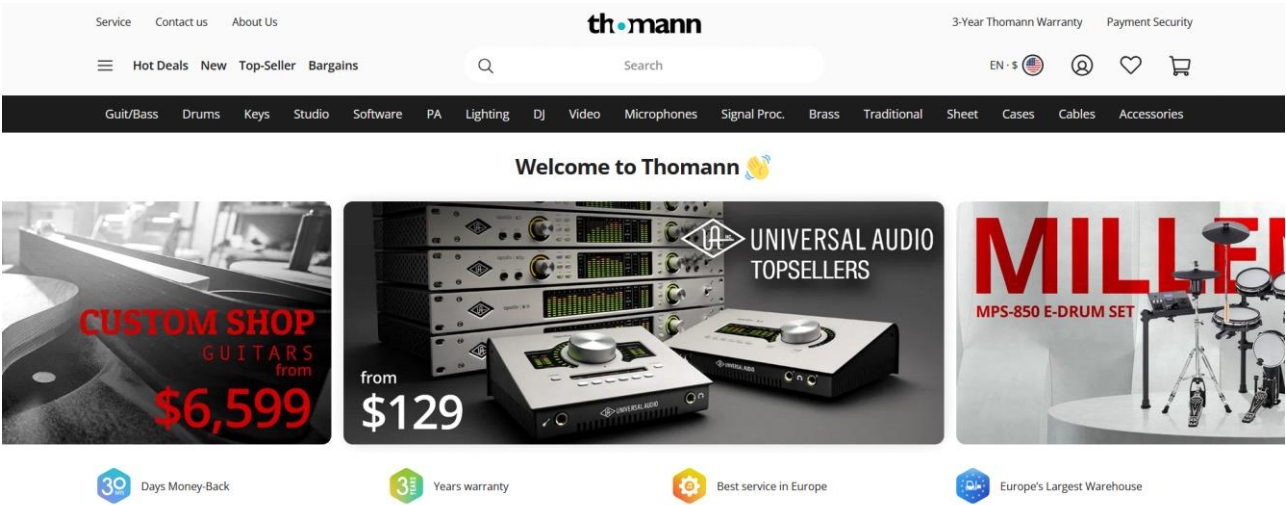


Рис. Г.1. Головна сторінка сайту StewMac



Our Categories

Рис. Д.1. Головна сторінка сайту Thomann

