МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РОЗРОБКА ВЕБСАЙТУ ПОСВЯЧЕНОМУ УКРАЇНСЬКІЙ МУЗИЦІ»

Виконав: здобувач освіти 4 курсу, групи К121-21
Спеціальність 121 Інженерія
програмного забезпечення
(шифр і спеціальність)
Олександр ГУСЄВ
(ім'я та ПРІЗВИЩЕ)
Керівник Альона ХОДАКОВСЬКА
(ім'я та ПРІЗВИЩЕ)
доцент кафедри програмної інженерії ЗНУ, к.ф.-м.н., доцент
Рецензент Олексій КУДІН
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2025

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо-професійний ступінь фаховий молодший бакалавр

Спеціальність 121 Інженерія програмного забезпечення

(шифр і назва)

ЗАТВЕРДЖУЮ
Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Тетяна СМОЛЯНКОВА
(підпис)

“ 14 ” червня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ГУСЄВУ Олександру Сергійовичу
(ПРІЗВИЩЕ ім’я та по–батькові)

1. Тема роботи «Розробка вебсайту посвяченому українській музиці»

Керівник роботи викладач вищої категорії, викладач-методист, PhD ХОДАКОВСЬКА Альона Валеріївна
(посада, науковий ступінь, вчене звання, ПРІЗВИЩЕ ім’я по-батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від «26» листопада 2024 р. № 1852-с

2. Строк подання студентом роботи 13.06.2025

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Огляд сучасних інформаційних систем
2. Розробка проєкту вебсайту
3. Програмна реалізація

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.12.2024

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	Постановка задачі	Грудень 2024	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2025	виконано
3.	Розробка першого розділу:		
	Огляд інформаційних систем та технологій для їхнього створення	Лютий 2025	виконано
4.	Розробка другого розділу:		
	Розробка проєкту інформаційної системи та вибір технологій	Квітень 2025	виконано
5.	Розробка третього розділу:		
	Розробка вебсайту	Травень 2025	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2025	виконано
7.	Захист кваліфікаційної роботи	20.06.2025	виконано

Здобувач освіти _____ Олександр ГУСЄВ
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Альона ХОДАКОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Кваліфікаційна робота: 46 сторінок, 20 ілюстрацій, 11 позицій у переліку посилань.

Метою кваліфікаційної роботи є розробка інформаційного ресурсу посвяченому українській музиці, що популяризує національну культурну спадщину та забезпечує зручний доступ до структурованої інформації про українських виконавців.

Об’єктом дослідження є процес розробки інформаційного ресурсу в системі управління контентом Atom.

Предметом дослідження є розробка інформаційного ресурсу української музики за допомогою системи управління контентом Atom.

Середовище розробки – Atom.

Метод дослідження – аналітичний, із залученням порівняльного аналізу інструментів веброзробки та систем управління контентом.

У кваліфікаційній роботі виконано огляд сучасних систем управління контентом для розробки інформаційного ресурсу, розглянуто переваги та недоліки системи управління контентом Atom, наведено етапи створення інформаційного ресурсу з використанням системи управління контентом Atom. Спроєктовано та розроблено сайт інформаційного ресурсу посвяченому українській музиці в системі управління контентом Atom.

ІНТЕРНЕТ-РЕСУРС, HTML, CSS, JAVASCRIPT, BOOTSTRAP, PHP7, ATOM.

SUMMARY

Qualification work: 46 pages, 20 illustrations, 11 items in the list of references.

The aim of the qualification work is to develop an informational web resource dedicated to Ukrainian music, which promotes national cultural heritage and provides convenient access to structured information about Ukrainian performers.

Object of the research: the process of developing an informational resource using the Atom content management system.

Subject of the research: the development of an informational resource about Ukrainian music using the Atom content management system.

Development environment: Atom.

Research method: analytical, involving comparative analysis of web development tools and content management systems.

The qualification work includes a review of modern content management systems for developing informational resources, examines the advantages and disadvantages of the Atom content management system, and outlines the stages of creating a web resource using Atom. The website dedicated to Ukrainian music was designed and developed using this system..

INTERNET RESOURCE, HTML, CSS, JAVASCRIPT, BOOTSTRAP, PHP7, ATOM.

ЗМІСТ

Завдання на кваліфікаційну роботу студенту 2
Реферат 4
Summary 5
Вступ 7
1 Аналіз предметної області 9
1.1 Поняття вебдизайну 9
1.2 Типи вебсайтів 10
1.3 Структура і ергономіка вебсайтів 12
1.4 Основні принципи розробки макету сайта (wireframe) 18
1.5 Засоби розробки інформаційного ресурсу 19
1.6 Аналіз предметної області 25
2 Проєктування вебсайту 28
2.1 Технічне завдання на розробку вебсайту 28
2.2 Вимоги до вебсайту 30
3 Реалізація музичного сайту посв'яченому українській музиці 35
3.1 Створення головної сторінки та вторинних 35
3.2 Створення адаптивності сайту та згортаючого меню 38
3.3 Викладення сайту на хостинг 40
3.4 Тестування музичного сайту на персональному комп'ютері 42
Висновки 43
Додаток А 45

ВСТУП

У сучасному цифровому суспільстві інформаційні ресурси в Інтернеті стали одним із ключових інструментів для отримання знань, комунікації та культурного самовираження. Швидкий доступ до даних, зручність використання, інтерактивність і мінімальні технічні вимоги зробили вебресурси надзвичайно популярними серед користувачів у всьому світі. Це зумовлює постійну потребу у створенні нових якісних сайтів, які відповідають запитам різних аудиторій.

Особливої актуальності набувають вебсайти, присвячені українській культурі, зокрема музиці. В умовах повномасштабної війни, питання збереження національної ідентичності, культурної спадщини та інформаційного суверенітету стали як ніколи важливими. Українська музика сьогодні виконує не лише естетичну чи розважальну функцію – вона стала формою спротиву, способом консолідації суспільства та духовної підтримки. Актуальність теми кваліфікаційної роботи полягає також у потребі створення якісного україномовного контенту в інтернет-просторі, який відповідає сучасним вимогам дизайну, зручності користування та функціональності. Враховуючи інтерес молоді до цифрових медіа, вебсайт, присвячений українській музиці, сприятиме підвищенню обізнаності про національну культуру, підтримці сучасних виконавців та розвитку цифрової грамотності.

Мета дослідження полягає у створенні сучасного, інформативного та зручного у використанні вебсайту, присвяченого українській музиці, який забезпечуватиме доступ до актуального контенту, популяризуватиме українську музичну культуру та відповідатиме потребам користувачів у швидкому та якісному отриманні інформації. Для реалізації поставленої мети було сформульовано наступні завдання:

− проаналізувати наявні вебресурси, присвячені українській музиці, з метою виявлення їх сильних та слабких сторін;

− сформулювати вимоги до структури, функціональності та дизайну майбутнього вебсайту;

− розробити інформаційну архітектуру ресурсу та структуру бази даних для зберігання музичного контенту;

− реалізувати вебсайт із використанням сучасних вебтехнологій (HTML, CSS, JavaScript, PHP7, Bootstrap), забезпечивши адаптивність та зручність навігації;

− провести тестування сайту на різних пристроях та оцінити його ефективність.

Об’єкт дослідження – процес розробки інформаційного ресурсу.

Предмет дослідження – методи та засоби розробки вебсайту, присвяченого українській музиці, з акцентом на структуру, функціональність, дизайн та інтеграцію інформаційного контенту.

Для виконання поставлених задач використовувалися як електронні так і друковані ресурси, розробка інформаційного ресурсу відбувалась у безкоштовному середовищі розробки Atom.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та переліку посилань (11 найменувань).

У першому розділі розглянуто поняття інформаційного ресурсу, проведено класифікацію типів вебресурсів, виконано огляд сучасних CMS (систем управління контентом) та інструментів розробки. Другий розділ присвячено опису обраної технології створення сайту, технічному завданню, проєктуванню діаграм варіантів використання та розгортання системи. У третьому розділі детально описано процес налаштування сайту, його запуск, тестування функціоналу на персональному комп’ютері та мобільному пристрої.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття вебдизайну

Вебдизайн – це спеціалізована галузь цифрового дизайну, що охоплює процеси створення, оформлення та структурного упорядкування вебресурсів у середовищі Інтернету. Основною метою вебдизайну є формування зручного, естетично привабливого та функціонально ефективного інтерфейсу для користувача [5].

На відміну від вебпрограмування, що відповідає за технічну реалізацію функціоналу, вебдизайн зосереджується на візуальній складовій сайту, побудові логічної структури сторінок, зручній навігації, а також адаптивності ресурсу до різних типів пристроїв – комп'ютерів, планшетів, смартфонів [9].

До ключових складників сучасного вебдизайну належать:

- інформаційна архітектура – логічна організація контенту та структура сторінок;
- UX/UI-дизайн – досвід взаємодії користувача з інтерфейсом та його візуальне оформлення;
- типографіка й колористика – добір шрифтів, кольорів і графічних елементів відповідно до тематики ресурсу;
- дотримання стандартів W3C – для забезпечення доступності сайту для осіб із порушеннями зору або слуху, а також сумісності з різними браузерами;
- інтеграція з SEO та цифровим брендингом – для підвищення впізнаваності ресурсу в мережі [3].

У контексті сучасних викликів, вебдизайн набуває ще більшого значення як засіб цифрової комунікації, культурного спротиву та популяризації національної ідентичности. Інтернет-ресурси, присвячені українській музиці, відіграють важливу роль у збереженні мистецької

спадщини, підтримці митців, а також формуванні патріотичного світогляду в молодого покоління.

Отже, якісний вебдизайн є не лише інструментом ефективної взаємодії користувача з сайтом, а й потужним каналом трансляції культурних смислів і національних цінностей. У процесі створення вебсайту, присвяченого українській музиці, його роль є важливою для досягнення як технічної, так і суспільно-культурної мети.

1.2 Типи вебсайтів

Сучасне інформаційне середовище в Інтернеті включає велику кількість вебресурсів, які поділяються на різні типи залежно від їхньої мети, функціоналу, цільової аудиторії та способу подачі інформації. Класифікація вебсайтів дозволяє краще орієнтуватися в різноманітті онлайн-ресурсів та обирати відповідні рішення під час їх розробки [8].

1. Інформаційні сайти. Основна мета таких ресурсів – надання користувачам актуальної та достовірної інформації з певної тематики. Це можуть бути новинні портали, енциклопедії, освітні платформи, сайти організацій або особисті блоги. Зазвичай вони мають статичну або динамічну структуру, просту навігацію та текстово-графічний контент.

2. Комерційні сайти (корпоративні) призначені для представлення компаній, брендів чи послуг в Інтернеті. Такі ресурси містять інформацію про діяльність підприємства, каталог продукції, контактні дані та інші комунікаційні інструменти. Їхньою метою є встановлення довіри до компанії та залучення клієнтів [5].

3. Інтернет-магазини розраховані на продаж товарів і послуг через Інтернет. Вони мають функціонал для перегляду товарів, оформлення замовлення, онлайн-оплати та відстеження доставки. Важливою складовою

таких сайтів є зручність користування, безпека платежів та система фільтрації товарів.

4. Соціальні мережі та спільноти орієнтовані на створення та підтримку взаємодії між користувачами. Вони надають можливість обміну контентом, коментування, участі в обговореннях та формування особистих профілів.

5. Розважальні сайти призначені для надання мультимедійного контенту – відео, музики, ігор, мемів, стрімів тощо. Часто такі ресурси мають велику кількість інтерактивних елементів та візуальних ефектів.

6. Особисті або персональні сайти представлені у вигляді онлайн-щоденника, портфоліо, сайта-резюме тощо. Вони використовуються для самопрезентації, творчої реалізації або ведення блогу.

7. Промо-сайт є прекрасним рекламним інструментом. Особливо, якщо ви плануєте провести рекламну компанію окремого товару або послуги. Структура промо-сайту сама по собі виконує функцію рекламного проспекту або буклету, та привертає увагу потенційних клієнтів. Також на такому сайті можна розмістити ту інформацію, яка не вміщається на сторінці журналу, в радіоповідомленні, телевізійному ролику. Обсяг промо-сайту складає до 10 сторінок, на яких можна розмістити дані про компанію, напрямок діяльності, просування послуги, контактні дані. Окремо повідомляється інформація про акцію, із зазначеним термінів й місця проведення.

Промо-сайт - це:

− використання можливого інтернету для впливу на цільову аудиторію;

− донесення до потенційних клієнтів максимум інформації про товар та послугу;

− потужній й ефективний інструмент реклами компанії та її продуктів.

8. Тематичні вебсайти – це спеціалізовані ресурси, присвячені певній тематиці, наприклад, культурі, мистецтву, науці чи музиці. Саме до цього типу належить сайт, розроблений у межах даної кваліфікаційної роботи – вебсайт, присвячений українській музиці, що поєднує інформаційний, культурний та мультимедійний аспекти.

Таким чином, розуміння типів вебсайтів дозволяє ефективно обрати засоби реалізації, враховуючи потреби майбутніх користувачів.

1.3 Структура і ергономіка вебсайтів

1.3.1 Структура вебсайтів

Зовнішній вигляд кожного сайту є унікальним, проте в усіх сайтів можна знайти спільні за функціональністю частини. На будь-якому сайті першою відкривається головна сторінка. Її розробці приділяють особливу увагу, оскільки дослідження показали, що люди не здатні читати інформацію, що відображається на моніторі, не так уважно, як книжки або журнали. Вони зазвичай лише поверхнево переглядають її, наприклад, як рекламу. Якщо головна сторінка містить те, що шукає відвідувач, він читає її далі, а якщо ні – переходить до інших сайтів, яких в Інтернеті дуже багато [7].

У верхній частині головної сторінки зазвичай розташована так звана шапка, яку дублюють на інших сторінках сайту. Це роблять спеціально, адже ця частина відображається у вікні браузера першою і відвідувач насамперед звертає увагу на неї.

Щоб забезпечити швидкий перехід до основних тематичних розділів сайту, створюють меню сайту – список гіперпосилань на його розділи. Меню сайту може бути горизонтальним або вертикальним. Горизонтальне меню зазвичай розташовують у шапці, іноді дублюючи його в нижній частині сторінки, а вертикальне – переважно в лівій частині сторінки, у місці, звідки відвідувач починає її переглядати. Меню є одним із найважливіших компонентів сайту, користувач постійно звертає на нього увагу, і тому вимоги до нього високі. Меню має бути зручним, помітним і зрозумілим,

інакше користувач не знатиме, як перейти до потрібного розділу, і покине сайт. Пункти меню мають бути чітко відділені один від одного.

Гіперпосилання, розміщені в тексті чи у вигляді графічних об’єктів, дозволяють переходити на різні сторінки сайту або навіть на інші сайти. На сайтах із дуже великим обсягом інформації є сторінки третього рівня, а якщо необхідно –то й четвертого, п’ятого тощо.

Загалом виділяють три типи структур вебсайтів – лінійну, деревоподібну та довільну. Подорожуючи сайтом із лінійною структурою, з головної сторінки ви перейдете на другу сторінку, з неї – на третю тощо. На сайті з деревоподібною структурою з головної сторінки можна потрапити на одну зі сторінок другого рівня, звідти – на одну зі сторінок третього рівня тощо. Сайт із довільною структурою видається зовсім неорганізованим, але саме в цьому й полягає принцип його створення. Подорожуючи таким сайтом, ви можете переходити з однієї його сторінки на інші в різні способи, і ваш шлях назад не обов’язково має бути таким самим [7].

Вибір структури визначається особливостями завдань, що розв’язуються за допомогою вебсайту. Розглянемо чотири приклади структур сайту. Кожна із цих структур має свої недоліки та переваги у проєктуванні вебсайту.

1.3.2 Стандартна структура вебсайту

Стандартна структура вебсайту передбачає ієрархічну організацію веб-сторінок, де головна сторінка (home page) є центральним елементом і містить навігаційні посилання на інші розділи сайту. У свою чергу, ці розділи також мають посилання на головну сторінку, забезпечуючи зручне переміщення між основними інформаційними блоками [7].

Це найпростіший, найзрозуміліший і водночас найпоширеніший підхід до побудови вебресурсу, який особливо ефективний для невеликих або

середніх сайтів із фіксованою кількістю сторінок. Така структура добре сприймається користувачами, оскільки дозволяє швидко знайти потрібну інформацію.

Основні характеристики стандартної структури:

- центральна навігація через головну сторінку;
- взаємозв’язок сторінок за допомогою гіперпосилань;
- ієрархічне розміщення контенту (від загального до конкретного);
- семантична логічність (розділи сайту чітко відокремлені за темами);
- можливість масштабування (легко додати нові сторінки, не змінюючи базової логіки).

Переваги стандартної структури:

- зручність у навігації для користувача;
- легкість реалізації навіть при мінімальних знаннях веброзробки;
- логічна побудова, яка сприяє кращому індексуванню сайту пошуковими системами;
- адаптованість до різних типів контенту (текст, зображення, відео, аудіо).

Недоліки:

- обмежена функціональність при збільшенні обсягів сайту (при великій кількості сторінок навігація може ускладнитися);
- потреба в ручному оновленні структури при додаванні великої кількості матеріалів.

Це найпростіший і найпоширеніший спосіб організації вебсайту.

1.3.3 Каскадна структура

Каскадна структура передбачає лінійну послідовність сторінок, у якій існує лише один шлях навігації між документами сайту – від початкової сторінки до останньої. У такій моделі кожна сторінка містить посилання

лише на наступну (перехід «вперед») або попередню (перехід «назад») сторінку. Це створює замкнений каскадний ланцюг, за яким користувачі проходять інформацію поступово, крок за кроком [7].

Основні характеристики каскадної структури:

- лінійна послідовність сторінок без розгалужень;
- обмежена навігація, що унеможливлює швидкий перехід до довільного розділу;
- навігаційні елементи обмежуються кнопками «Назад» і «Далі»;
- суворий контроль над порядком подачі інформації.

Переваги каскадної структури:

- забезпечує поетапне ознайомлення з матеріалом у визначеній логічній послідовності;
- зручно для навчальних ресурсів, презентацій, покрокових інструкцій або опитувань;
- виключає випадкове «пропускання» важливої інформації.

Недоліки:

- обмежена гнучкість навігації – користувач не може швидко повернутися до конкретного розділу;
- незручність при великому обсязі інформації – багаторазове натискання кнопок для повернення назад;
- менш ефективно для сайтів із різноплановим або неконтекстним контентом.

Каскадна структура доцільна в тих випадках, коли важливо зберегти чіткий навчальний або хронологічний порядок подання матеріалу.

1.3.4 Структура типу «Хмарочос»

Структура типу «хмарочос» передбачає сувору ієрархію сторінок із чітко визначеними шляхами доступу до інформації. Користувачі можуть

потрапити на деякі сторінки тільки за умови, що пройшли певний попередній маршрут – подібно до того, як у реальному хмарочосі потрібно піднятися поверх за поверхом, щоб дістатися до потрібної кімнати [7].

Ця модель має структуровану, багаторівневу навігацію, де кожен рівень (або «поверх») відкриває доступ до наступного. Відвідувачі сайту зазвичай починають із головної сторінки, проходять через послідовні рівні розділів, і лише після цього можуть отримати доступ до глибших сторінок.

Особливості моделі:

- обмежений доступ до сторінок без проходження попередніх кроків;
- чітка ієрархія сторінок і підсторінок;
- контрольований досвід користувача, що виключає випадкове або хаотичне переміщення;
- застосування умов для доступу (наприклад, після прочитання або заповнення інформації).

Переваги:

- ідеально підходить для інтерактивних курсів, де доступ до наступного модуля відкривається лише після завершення попереднього;
- забезпечує поступове занурення в тему;
- сприяє логічному структуруванню складного або багаторівневого контенту.

Недоліки:

- може обмежувати гнучкість навігації для користувачів, які прагнуть швидко отримати доступ до конкретної інформації без проходження всіх попередніх етапів
- не підходить для довідкових сайтів, де потрібен вільний доступ до всіх розділів;
- ускладнює навігацію, якщо структура занадто глибока або розгалужена.

У цій моделі відвідувачі можуть опинитися на деяких сторінках, тільки якщо вони йдуть правильним шляхом.

1.3.5 Павутина

У цій моделі всі сторінки вебсайту взаємопов’язані між собою. Користувач має можливість вільно переходити з однієї сторінки на іншу за допомогою численних гіперпосилань. Такий підхід забезпечує гнучку навігацію та створює відчуття відкритого інформаційного простору, де кожен документ може бути точкою входу до змісту сайту [7].

Ця модель є особливо ефективною для сайтів із великою кількістю контенту, де користувачі самостійно обирають траєкторію перегляду, наприклад, у довідкових ресурсах, енциклопедіях, навчальних платформах тощо.

Однак за відсутності чіткої навігаційної структури або логічно продуманого розташування посилань така схема може викликати у користувача дезорієнтацію. Надмірна кількість переходів і відсутність ієрархії сторінок можуть перетворити користування сайтом на складний процес.

Переваги:

- гнучка навігація: можна легко перейти між пов’язаними сторінками.
- зручність для досвідчених користувачів, які шукають специфічну інформацію.
- добре підходить для інформаційно насичених або довідкових ресурсів.
- кожна сторінка може бути початковою точкою входу на сайт.

Недоліки:

- високий ризик дезорієнтації користувача через надмірну кількість посилань.
- складність підтримки та оновлення структури при зростанні кількості сторінок.
- відсутність чіткої логіки переходів може знижувати ефективність пошуку інформації.

- необхідність продуманої навігації для уникнення «ефекту лабіринту».

У цьому випадку всі сторінки вебсайту містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу. Ця схема може перетворитися на лабіринт, якщо вийде з-під контролю, але вона популярна в тих випадках, коли посиланнями на документи користуються не надто часто.

1.4 Основні принципи розробки макету сайта (wireframe)

Процес створення макету сайту, або wireframe, є важливою складовою підготовчого етапу веб-розробки. Він дає змогу наочно уявити структуру майбутнього ресурсу, оцінити розташування елементів на сторінках та продумати логіку користувацької взаємодії ще до початку реалізації дизайну й написання коду. Ретельне проєктування макету дозволяє уникнути багатьох помилок у подальшій роботі та значно підвищує ефективність командної взаємодії.

Створення сайту розпочинається з постановки цілей. Спершу визначаються загальні завдання, які повинен виконувати веб-ресурс, формулюється його призначення: чи буде це інформаційний сайт, освітній портал, онлайн-каталог або ресурс для комунікації. Паралельно здійснюється аналіз аналогічних або тематично суміжних сайтів, щоб оцінити їх сильні сторони й виявити недоліки. Важливим є й попереднє вивчення цільової аудиторії: хто користуватиметься ресурсом, які очікування й потреби мають потенційні відвідувачі. Після цього складається попередній опис майбутнього сайту, визначаються його основні розділи, можливі додаткові функції, такі як чат, форум, форма зворотного зв’язку чи підписка на новини. Також слід вирішити, чи буде інформація статичною, тобто незмінною, чи динамічною, яка автоматично оновлюється через базу даних або зовнішні

джерела. На цьому ж етапі враховуються технічні вимоги до роботи сайту, обирається програмна платформа й середовище розміщення.

Наступним кроком є створення схеми організації контенту та навігації. Визначається, як взаємодіятимуть між собою сторінки, якою буде ієрархія розділів, які кнопки або елементи керування забезпечуватимуть швидкий доступ до важливих матеріалів. Від цього залежить зручність використання сайту, а отже, й загальне враження користувачів.

Коли структура сайту вже чітко окреслена, переходять до побудови wireframe – схематичного макету сторінок. Це своєрідна «чорно-біла» версія сайту без кольорів, шрифтів та декоративних елементів, що зосереджується виключно на функціональності. Макет демонструє розташування заголовків, текстових блоків, зображень, форм, меню, кнопок тощо. Він допомагає оцінити, наскільки логічно розміщені елементи, чи зручною буде навігація, чи легко користувач орієнтуватиметься у вмісті.

Після створення макету важливо обговорити його з учасниками проєкту або представниками цільової аудиторії. Коментарі та зауваження дозволяють виявити слабкі місця, що потребують доопрацювання. Корекція структури на цьому етапі є значно простішою та менш витратною, ніж після запуску сайту.

Таким чином, грамотне проєктування wireframe є запорукою ефективної реалізації веб-ресурсу. Що дозволяє не лише сформувати чітке бачення майбутнього сайту, а й забезпечує його функціональність, доступність і зручність для кінцевого користувача.

1.5 Засоби розробки інформаційного ресурсу

JavaScript – це мова програмування, яка дозволяє робити веб-сторінки інтерактивними, тобто такими, що реагують на дії користувача. Послідовність інструкцій, яку називають програмою, скриптом або

сценарієм, виконується інтерпретатором, вбудованим у сучасний веб-браузер. Іншими словами, код JavaScript вбудовується в HTML-документ і виконується на боці клієнта (у браузері користувача), що дозволяє обробляти події без необхідності перезавантажувати сторінку.

Програми на JavaScript можуть виконуватися у відповідь на різноманітні події: наприклад, перед відправленням даних форми можна перевірити їх на коректність і у разі помилки заборонити відправлення, таким чином покращуючи взаємодію користувача з сайтом.

JavaScript є об’єктно-орієнтованою скриптовою мовою програмування та діалектом стандарту ECMAScript. Найчастіше вона використовується у веб-браузерах як мова сценаріїв для додання інтерактивності веб-сторінкам, проте може застосовуватись і поза браузером (наприклад, у середовищі Node.js).

1.5.1 Особливості мови програмування PHP

PHP – це скриптова мова програмування загального призначення, широко використовується для розробки веб-додатків. PHP виконується на сервері. Коли браузер надсилає запит на сторінку з PHP-кодом, сервер передає цей запит інтерпретатору PHP, який генерує HTML-код і повертає його серверу, а сервер надсилає готову сторінку клієнту. При цьому сам PHP-код ніколи не потрапляє у браузер, тому побачити вихідний код PHP-скрипта неможливо.

Для взаємодії користувача з PHP-скриптом необхідно відправити на сервер новий запит, наприклад, клікнувши на посилання або на кнопку у формі. PHP виконується лише на сервері і не працює безпосередньо у браузері. Скрипт, який обробляє запит і формує сторінку, виконується та завершується, після чого всі тимчасові дані, збережені в ньому, втрачаються.

Саме тому, якщо деякі дані або змінні потрібно зберігати між викликами скрипта, їх слід передавати знову або зберігати у сесії, базі даних чи файлах.

Як і будь-яка мова програмування, PHP має свої переваги та недоліки. До переваг належать популярність мови, підтримка більшістю хостинг-провайдерів, велика кількість навчальних матеріалів і безкоштовних модулів. PHP має зручні вбудовані функції для роботи з базами даних, що значно спрощує розробку вебсайтів [10].

Серед недоліків – те, що PHP не найкраще підходить для створення великих і складних програм. Спочатку мова була розроблена для написання невеликих скриптів, і хоча з кожною версією вона вдосконалюється, все ж для масштабних проєктів PHP іноді вважають менш ефективним, ніж компільовані мови, через особливості інтерпретованого виконання. Крім того, PHP довгий час не підтримував Unicode належним чином, хоча у нових версіях ця підтримка покращилась.

Ще один аспект – різні варіанти написання функцій у PHP, що може викликати плутанину у початківців. Розробники PHP прагнуть зробити мову більш універсальною, розширюючи її можливості за межі веб-програмування. Це робить PHP складнішим для вивчення і дещо впливає на його популярність серед новачків.

1.5.2 Огляд особливостей системи MySQL

Інтернет-програмування використовує широкий спектр інструментальних засобів для роботи з даними. Система управління базами даних MySQL – це не єдиний варіант для розв’язання інформаційних завдань, проте саме завдяки своїй надійності, простоті та ефективності вона вже давно стала стандартом у веб-розробці [1].

Програміст має змогу ефективно працювати з базою даних MySQL за допомогою популярних мов програмування. При дотриманні простих правил

формування запитів можна отримати безпечний, надійний і стабільний результат, який легко переноситься між різними платформами та не вимагає складної конфігурації чи обслуговування.

Для успішного застосування запитів MySQL важливі досвід і знання. Сервер MySQL орієнтований на швидку і ефективну обробку запитів, а не на детальне інформування розробника про всі внутрішні процеси. У цій системі діє неписане правило: правильний запит (питання) гарантує правильний результат (відповідь). Аналогічно до людського спілкування, де людина, поставивши запитання, очікує отримати відповідь, а не критику правильності формулювання, MySQL прагне надати максимально релевантний результат без зайвих повідомлень про помилки, якщо тільки це не суттєво для обробки запиту.

Оператор MySQL SELECT є одним із найпотужніших і гнучких елементів мови запитів. Він містить численні конструктивні елементи, які накопичувалися протягом багатьох років розвитку мови SQL. Однак швидкість розвитку та зміни структури даних часто випереджають стандарти мови запитів [1].

У реальних проєктах програмісти найчастіше використовують базові елементи оператора SELECT, які позначені синім кольором у документації. При роботі з MySQL через PHP слід орієнтуватися на семантику завдання, а не на синтаксис окремих конструкцій. Саме зміст і логіка задачі визначають, які поля і таблиці будуть використані у запитах, а також обмеження для вибірки (WHERE), групування (GROUP BY) та сортування (ORDER BY).

Чим більше програміст фокусується на логіці і семантиці даних, а не на використанні складних конструкцій мови, тим легше змінювати запити у майбутньому без потреби переписувати всю структуру.

Загалом, застосування запитів у межах жорсткого синтаксису – це лише один із способів реалізації логіки обробки даних. Набагато гнучкішим і практичнішим є підхід, коли обмеження на синтаксис пом’якшені, що дозволяє адаптувати запити до змін у задачах і структурах даних.

1.5.3 Опис застосунку phpMyAdmin

PhpMyAdmin – вебдодаток з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Ця програма користується великою популярністю у веб-розробників, оскільки дозволяє керувати базу даних MySQL без вводу SQL команд через дружній інтерфейс і з будь-якого комп'ютера під'єднаного до інтернету без необхідності встановлення додаткового програмного забезпечення.

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL. Переважна більшість українських провайдерів використовують цей застосунок як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

Програма розповсюджується під ліцензією GNU General Public License і тому деякі інші розробники інтегрують його у свої розробки, наприклад XAMPP, Denwer.

Ініціатором є німецький розробник Тобіас Ратшіллер (Tobias Ratschiller) і базується, так само як і phpPgAdmin, на аналогічному застосунку MySQL-Webadmin, що виник у 1997 році. Проєкт локалізовано 85-ма мовами.

1.5.4 Текстовий редактор Atom

Atom – безкоштовний текстовий редактор з відкритим вихідним кодом, який підтримує роботу на операційних системах Linux, macOS та Windows. Він має зручний інтерфейс, підтримує розширення (плагіни), написані на

Node.js, а також інтеграцію з системою контролю версій Git, що робить його популярним інструментом для програмування і розробки веб-додатків.

Редактор побудований на основі кросплатформового фреймворка Electron, який використовує Chromium і Node.js, що забезпечує сучасні можливості для розробників. Atom написаний на CoffeeScript і LESS. Перша стабільна версія була випущена у червні 2015 року [2].

Хоча офіційна підтримка редактора була припинена в грудні 2022 року, Atom і надалі залишається доступним для користувачів і використовується в багатьох навчальних та професійних середовищах. Його популярність пояснюється простотою використання, можливістю налаштування через плагіни, а також стабільністю роботи. Завдяки відкритому коду, спільнота користувачів продовжує підтримувати і оновлювати редактор.

1.5.5 Переваги системи Atom

Головна особливість Atom – це широкі можливості налаштування, які дозволяють адаптувати редактор під власні потреби. Спочатку він має вбудований файл-менеджер, розвинені функції пошуку і заміни, підтримку множинних курсорів, опції згортання коду, зручний та зрозумілий інтерфейс, а також можливість імпорту правил і тем із TextMate. Atom зазвичай вважають більш «важким» редактором у порівнянні з такими, як Sublime Text, через більші апаратні вимоги.

Раніше активна спільнота користувачів створювала численні плагіни та розширення, що дозволяло редактору постійно розвиватися і відповідати сучасним потребам програмістів. Проте з грудня 2022 року офіційна підтримка Atom припинена, і нові оновлення більше не виходять. Це означає, що подальший розвиток редактора обмежений, а спільнота користувачів значно зменшила свою активність.

Незважаючи на це, Atom досі залишається зручним редактором для багатьох програмістів, особливо для тих, хто цінує гнучкість налаштувань і знайомий інтерфейс.

Таким чином, основними перевагами системи Atom є її гнучкість у налаштуванні, зручний інтерфейс, багатий набір вбудованих функцій і велика кількість доступних розширень, що забезпечує високу адаптивність редактора під потреби розробника.

1.6 Аналіз предметної області

Розглянемо музичний сайт neformat.com.ua (див. рис. 1.1). Перевага сайту у тому, що він має приємний дизайн для створення якого використовували такі мови програмування як CSS і HTML. Створення даного дизайну не займає багато часу, а також дасть приємне оформлення для мого сайту. Сайт простий і складається тільки з навігації і новинної стрічки. З недоліків – навігація має не цікаве оформлення та незручне розташування.

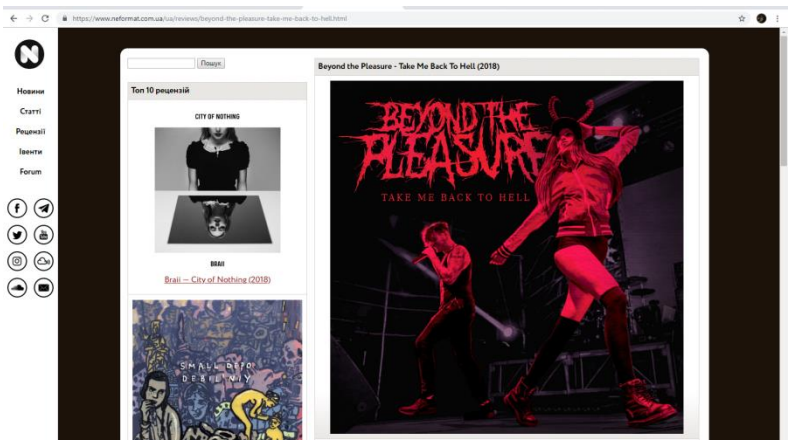


Рисунок 1.1 – Огляд сайту neformat.com.ua

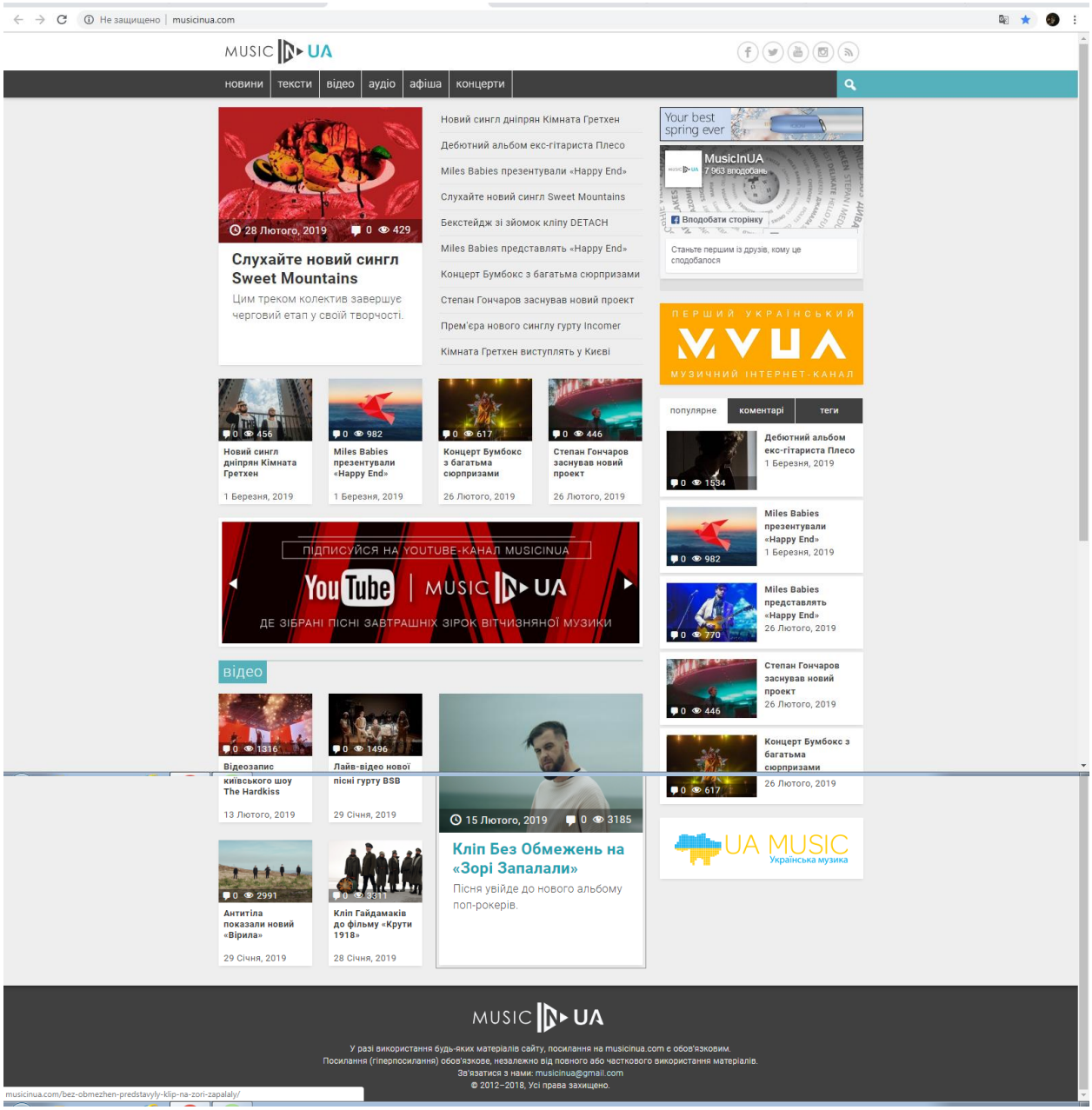


Рисунок 1.2 – Огляд сайту music.ua

Вебсайт music.ua має досить приємне оформлення і розташування навігації (див. рис. 1.2). Не займе багато часу на його створення. Меню розташовується вгорі, що є не великою перевагою ніж попереднє джерело. Оформлення стрічки новин і інших елементів має незручне розташування елементів, яке дає відчуття «Все в купі», коли всі елементи розташовані дуже близько один до одного. Дизайн, коли все здається просто, але не втрачає той функціонал, який властивий сучасним сайтам – це перевага, але на цьому ресурсі вона відсутня.

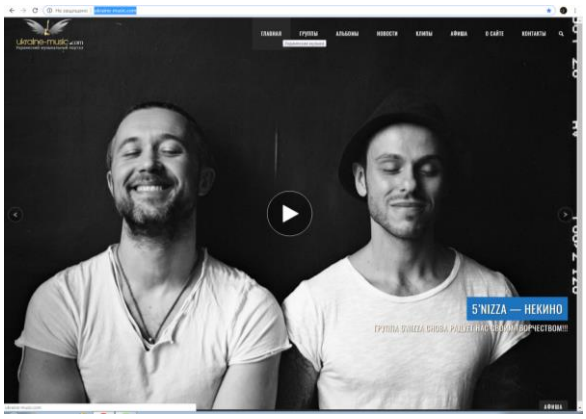


Рисунок 1.3 – Огляд сайту ukraine-music.com

Ще однин музичний сайт – ukraine-music.com (див. рис. 1.3). Даний сайт має безліч переваг: дизайн, розташування елементів, інформативність. Все підібрано грамотно. Немає нічого зайвого, нічого ніде не заважає. Простий інтерфейс, але дає повний доступ до всієї інформації в нашому напрямку.

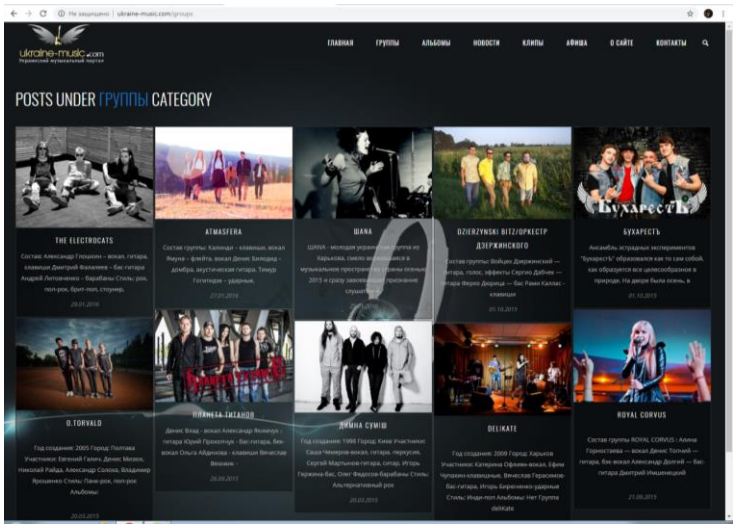


Рисунок 1.4 – Новинна стрічка сайту ukraine-music.com

На цьому рисунку можна помітити, що серед вибору груп української музики, є місця там де фотографія з’їжджає на ряд вище (див. рис. 1.4). Недоліком є недосконало зроблене розташування об’єктів.

2 ПРОЄКТУВАННЯ ВЕБСАЙТУ

2.1 Технічне завдання на розробку вебсайту

Технічне завдання (ТЗ) є ключовим документом, що визначає основні вимоги до майбутнього вебсайту, який розробляється в рамках проєкту на тему «Українська музика». Воно базується на розумінні мети створення ресурсу, обсягів та змістового наповнення, функціональних можливостей, цільової аудиторії та принципів зручності користування (usability).

Метою створення вебсайту є популяризація української музичної культури, ознайомлення користувачів із видатними композиторами, виконавцями, жанрами музики, а також надання якісного інформаційного ресурсу для навчання, самостійного вивчення та загального розвитку. Цільова аудиторія – студенти, викладачі та всі зацікавлені у вивченні українського музичного мистецтва.

Процес розробки ТЗ є складним і вимагає спеціальних знань у галузях веброзробки, UX/UI дизайну, інформаційної архітектури та інтернет-маркетингу. На початковому етапі формується загальна постановка завдання, яка слугує базою для подальшого уточнення вимог і функцій. Додаткову інформацію розробник отримує на основі власного досвіду і знань щодо створення зручних, інтуїтивно зрозумілих інтерфейсів.

Особливу увагу приділено інформаційній архітектурі – структурі розміщення даних, що забезпечує логічний та зручний доступ користувачів до вмісту сайту. Перед проєктуванням архітектури здійснюється повний аналіз діяльності організації та визначення ключових функцій ресурсу [11]. Важливо сформувати повний перелік інформації, що має бути представлена на сайті, оскільки додавання нового контенту у вже готовий вебресурс є складним процесом [4].

Зміст сайту згруповано у 7 логічних розділів, що відповідає стандартам ефективної навігації та запобігає інформаційному перевантаженню. Основні розділи сайту:

Головна сторінка

Про українську музику

Жанри

Виконавці

Музика за епохами

Медіатека (аудіо, відео, зображення)

Контакти або форма зворотного зв’язку

Для кожного з розділів передбачено інтуїтивно зрозумілу навігацію та можливість швидкого доступу до ключового контенту. Особливу увагу приділено пріоритетності інформації – найбільш важливі дані будуть візуально виділені, а користувачі матимуть постійний доступ до основних елементів меню.

Дизайн сайту розробляється у мінімалістичному стилі з використанням кольорової гами, що асоціюється з українською культурою, забезпечуючи читабельність, контрастність та зручність користування на різних пристроях [6].

З технічної точки зору, сайт буде створено із застосуванням сучасних вебтехнологій (HTML, CSS, JavaScript або CMS за потреби), з підтримкою адаптивного інтерфейсу для комфортного перегляду як на десктопах, так і на мобільних пристроях. Інтерфейс буде україномовним.

До технічного завдання входять також підготовчі роботи:

- вибір та реєстрація доменного імені;
- налаштування хостингу з підтримкою необхідних технологій (PHP, MySQL – за потреби);
- забезпечення захищеного з’єднання (HTTPS);
- базова SEO-оптимізація для покращення індексації пошуковими системами.

Розробка сайту включатиме такі основні етапи:

1. Збір і систематизація контенту.
2. Побудова логічної структури ресурсу.
3. Розробка дизайну інтерфейсу.
4. Вебпрограмування та верстка сторінок.
5. Наповнення сайту інформаційними матеріалами.
6. Тестування функціональності.
7. Публікація сайту в мережі Інтернет.
8. Затвердження кінцевого результату замовником.

2.2 Вимоги до вебсайту

- Тема сайту є довільною. Об’єм 4-5 сторінок.
- Головна сторінка сайту має назву index.*
- Наявність авторської сторінки за посиланням © Прізвище або нік. На авторській сторінці поставити аватар і написати коротку інформацію про себе.
- Верстання сайту блочне із застосуванням тегів HTML5.
- Кодування сторінки має підтримувати кирилицю (charset=utf-8).
- Застосувати таблицю стилів в окремому файлі. Опис стилів зі сторінки перенести в файл таблиці стилів *.css. Назви стилів – латиницею, змістовні (.red .big тощо).
- Наявність основних тегів у коді (<html><head><body> <h1>…<h6> тощо).
- Парні теги мають закриватися ( ). Зберігати вкладеність тегів (Текст).
- Основні модулі сторінки мають бути закоментовані.

- Назви файлів (сторінок, картинок, скриптів, стилів) та папок – латиницею, малий регістр, без пробілів (для візуального розмежування застосовувати символи «-» або «_»). Кирилицю для назв не застосовувати!

- Графіка, що відноситься до дизайну сторінок міститься в окремій директорії. Решта зображень, що відносяться до наповнення сторінок – в інших директоріях зі змістовними назвами.

- Єдиний стиль для всіх сторінок сайту. Однакове розміщення основних блоків сайту (заголовку, рубрик, логотипу).

- Наявність навігації на всіх сторінках сайту. Виключення лише для сторінок, що відкриваються у новому вікні і містять інформацію, яка може сприйматися і не в контексті сайту (збільшена картинка, інформація про розробника, стаття, тощо)

2.3 Проєктування вебсайта

Макет сайту має такі частини: Header (див. рис. 2.1), Footer (див. рис. 2.2), Блок інформації, Блок меню.

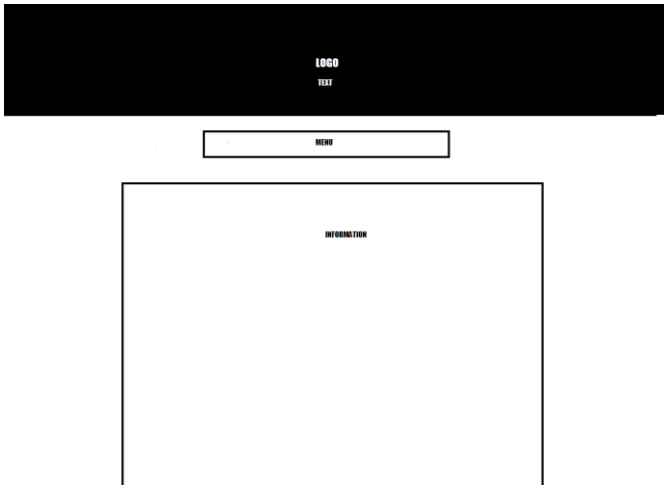


Рисунок 2.1 – Макет верхня частина макету

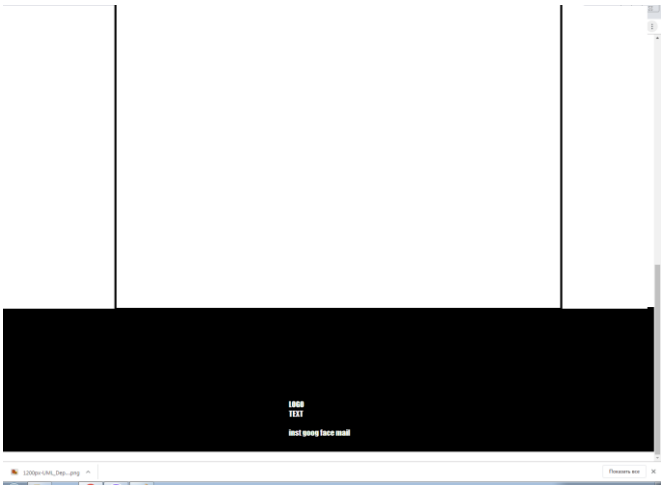


Рисунок 2.2 - Макет нижня частина макету

2.3.1 Діаграма варіантів використання

На діаграмі варіантів (див. рис.. 2.3) ми бачимо взаємозв’язок між користувачем та вебсайтом.

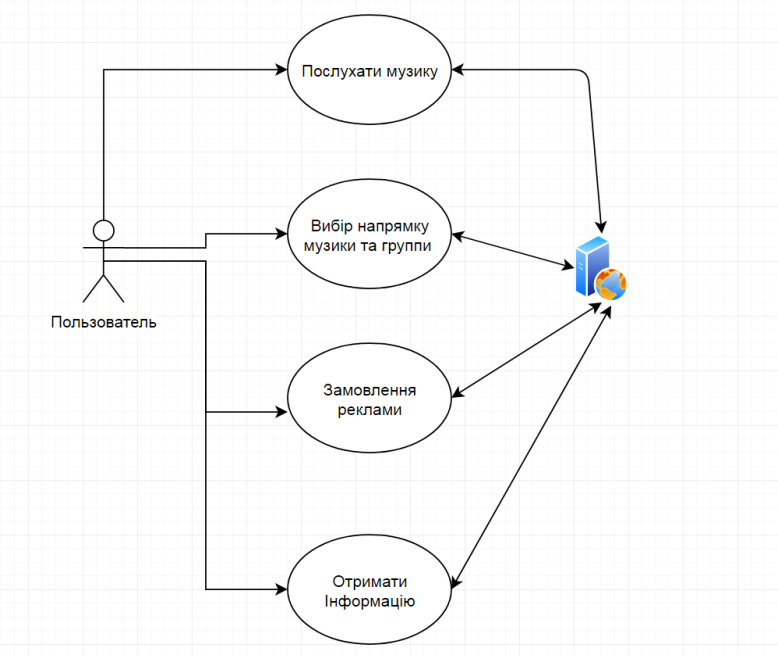


Рис. 2.3 – Діаграма варіантів використання

Між ними проходять такі дії:

1. Користувач робить вибір який напрямок та група його цікавить, запит потрапляє до вебсерверу і вебсервер відправляє до користувача список груп.

2. Користувач хоче отримати інформацію про цю групу, аналогічно з вибором груп, сайт дає інформацію користувачу.

3. Користувач хоче замовити рекламу, вебсайт надає html сторінку з контактами.

4. Користувач хоче слухати музику, вебсайт отримує запит та з архіву пісень відтворює необхідний трек.

2.3.2 Діаграма розгортання

На діаграмі розгортання (див. рис. 2.4) ми бачимо, як проходить процес отримання запиту від користувача до вебсайту.

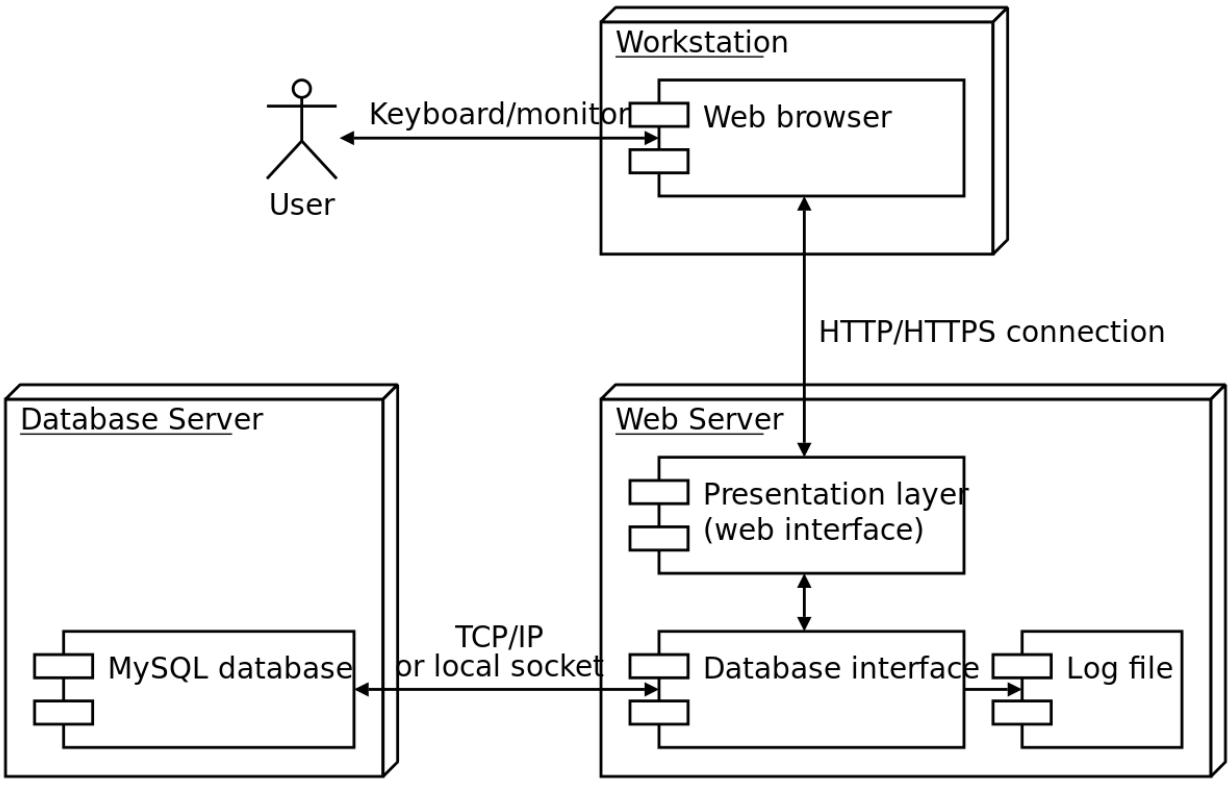


Рис.2.4 – Діаграма

Користувач через веббраузер з’єднується з нашим вебсервером та подає запит, який виконується в сервері бази даних та повертається на вебсайт, виводячи у браузер. Браузер передає запит користувачу.

3 РЕАЛІЗАЦІЯ МУЗИЧНОГО САЙТУ ПОСВ'ЯЧЕНОМУ УКРАЇНСЬКІЙ МУЗИЦІ

3.1 Створення головної сторінки та вторинних

Верхня частина сайту складається з хедера (header), який містить логотип та назву сайту, а також меню. У меню входять такі вкладки: «Нові релізи», «Головна», «Інформація про сайт», «Контакти», «Загальні проєкти», «Українські гурти» (див. рис. 3.1).

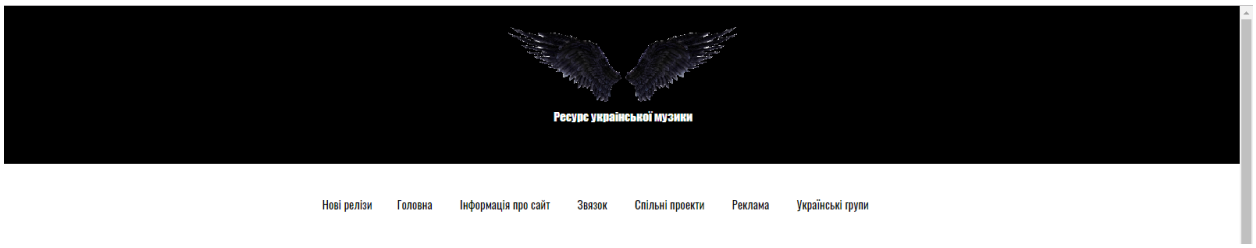


Рисунок 3.1 – Хедер та меню сайту

Нижня частина (див. рис. 3.2) складається так само, як і хедер – з логотипу та назви, але додані також іконки соцмереж для зв'язку через них.



Рисунок 3.2 – Створення футера



Рисунок 3.3 – Заповнення головної сторінки контентом

Головна сторінка була наповнена музичним контентом популярних гуртів (див. рис. 3.3), що включає фотографію гурту, інформацію про нього та останній випущений трек. Для оформлення та розташування об’єктів на сторінці використовувалися мови програмування HTML5 і CSS3. Створено файл JavaScript, який отримує інформацію про найпопулярніші гурти та додає її на сайт.

Приклад заповнення HTML-сторінки популярними українськими гуртами. У списку представлені такі популярні гурти, як: «Друга Ріка», «Бумбокс» та інші. При натисканні на зображення гурту відкривається сторінка з описом гурту, його історією та невеликим плейлистом, обмеженим п’ятьма треками задля економії пам’яті та зменшення навантаження на сервер, оскільки проєкт розрахований на ознайомлення з українськими гуртами, а не на надання всіх композицій виконавців. Приклад наведено нижче на рисунку 3.4.

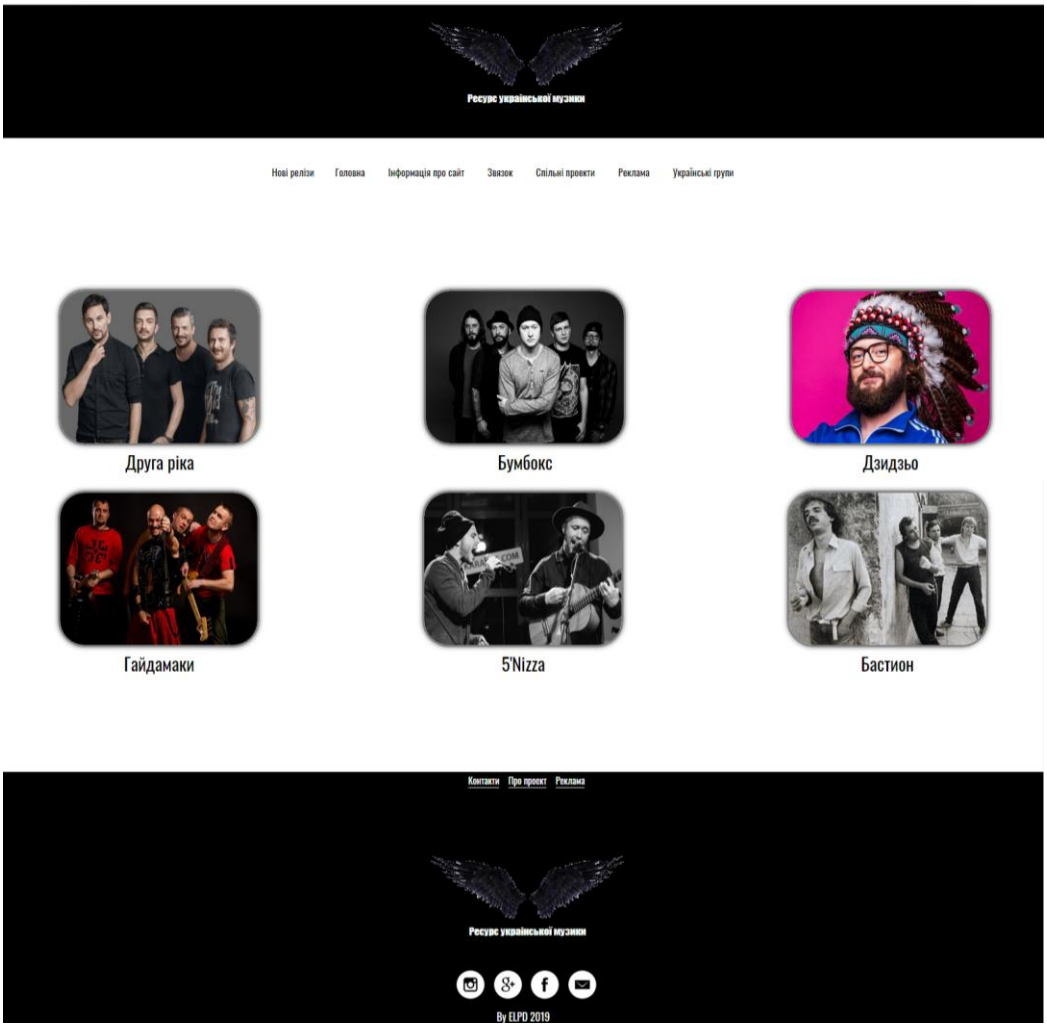


Рисунок 3.4 – Сторінка українських груп

Для опису була взята сторінка з гуртом «Бумбокс» (див. рис. 3.5). Сторінка містить необхідну, коротку інформацію, яка буде корисною для користувача, а також добірку композицій для ознайомлення з творчістю гурту.



Рисунок 3.5 – Вторинна сторінка

3.2 Створення адаптивності сайту та згортаючого меню

Меню розроблено за допомогою мов програмування JavaScript та CSS3 і працює таким чином: при зміні ширини сторінки менше ніж 768 пікселів меню згортається та з’являється значок (три смужки), у якому містяться всі пункти, крім найпопулярнішого (див. рис. 3.6).

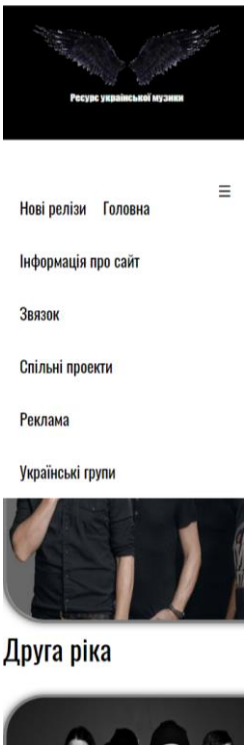


Рисунок 3.6 –Меню, що згортається

Завдяки цьому немає потреби зменшувати текст або прокручувати сторінку вбік, що забезпечує більший комфорт при користуванні сайтом.

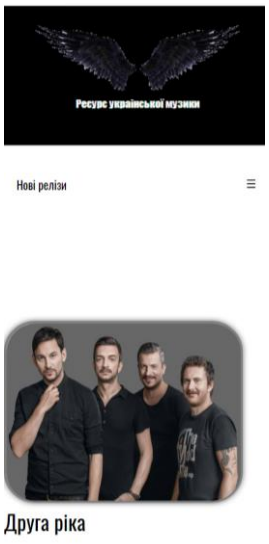


Рисунок 3.7 – Адаптивна головна сторінка

За допомогою мови програмування CSS3 була реалізована адаптивність сайту (див. рис. 3.7). Аналогічно до попередньої розробки, при зменшенні

ширини сайту менш ніж 728 пікселів зображення розташовуються в колонку разом із текстом та зменшуються в розмірах (див. рис. 3.8).

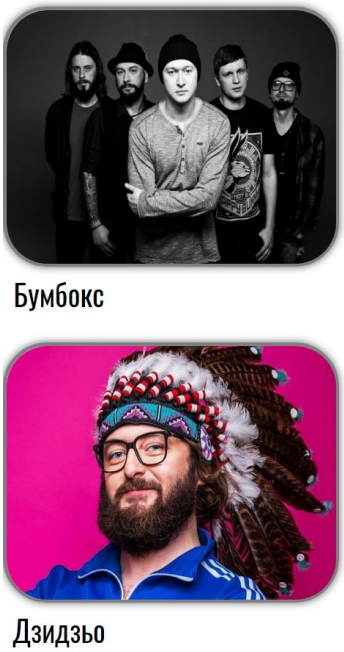


Рисунок 3.8 – Адаптивна головна сторінка

3.3 Викладення сайту на хостинг

В якості безкоштовного хостингу було обрано сервіс SmartApe, який надає два тижні безкоштовного користування (див. рис. 3.9). Після реєстрації було отримано персональний логін і пароль. На сайті можна створювати кілька доменів і виконувати багато дій, наприклад, редагування сторінок. Дуже корисні функції, особливо редагування, дозволяють змінювати сайт без необхідності щоразу завантажувати новий файл, що значно економить час [10].

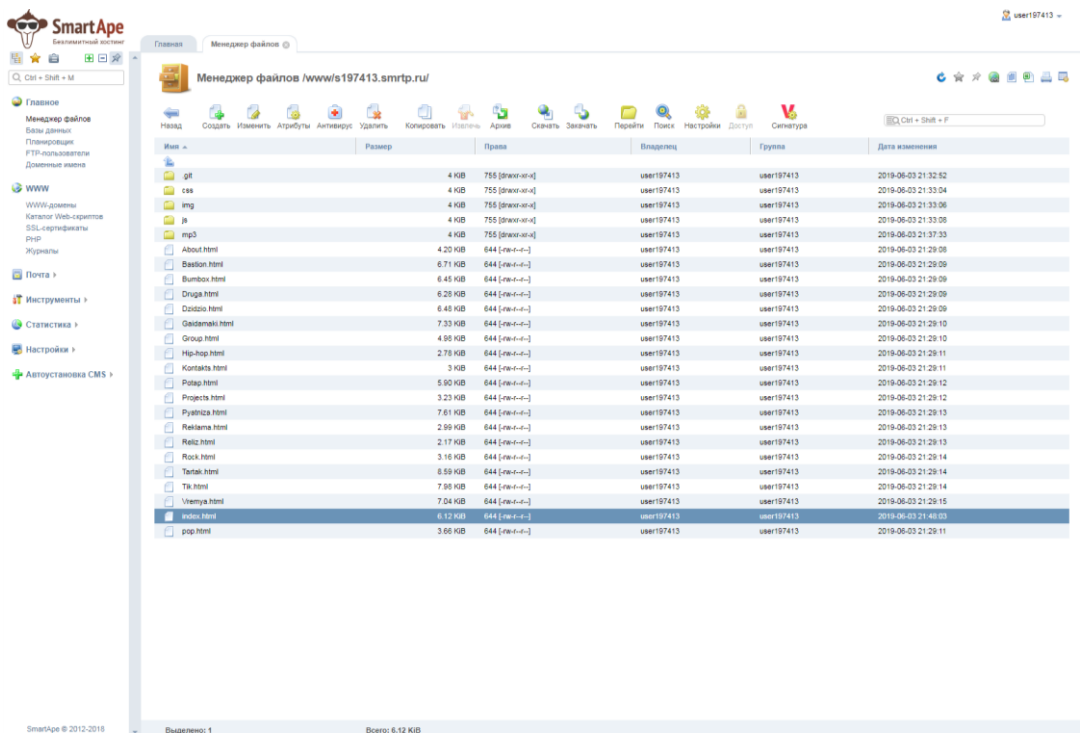


Рисунок 3.9 – Завантаження сайту на хостинг

Для завантаження сайту на хостинг була використана утиліта FileZilla. Завдяки реєстрації на сайті хостингу було отримано не лише логін і пароль для доступу до сайту, а також хост з окремими логіном і паролем. Для завантаження через утиліту потрібно було лише під’єднатися до серверу-хостингу, після чого з’являлися домени, серед яких необхідно було перейти до створюваного сайту й увійти в папку «WWW». Після переходу до цієї папки у правій частині вікна відкривається створений проєкт, який був перенесений до папки хостингу («WWW»). Після завантаження всіх файлів, переходячи за своїм доменом, можна було побачити створений проєкт (див. рис. 3.10).

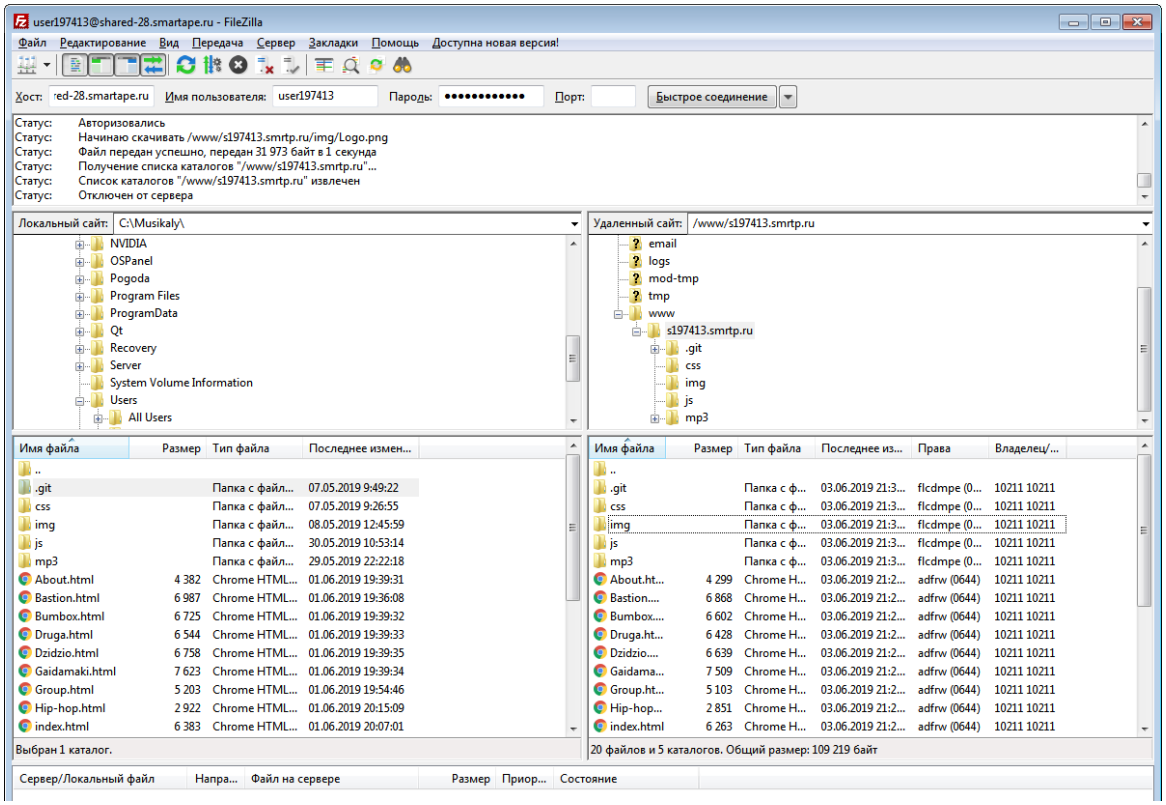


Рисунок 3.10 – Завантажений сайт на хостингу

3.4 Тестування музичного сайту на персональному комп’ютері

Після завантаження на хостинг було перевірено всі сторінки та внесено незначні виправлення. Завантаження на хостинг відбулося успішно (див. рис. 3.11). Зображення робочого сайту наведено в додатку А.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було поставлено низку практичних завдань, спрямованих на створення повноцінного вебресурсу, присвяченого українській музиці. Зокрема, передбачалося:

- створення головної та вторинних сторінок;
- створення адаптивності сайту у цілому;
- створення додатку для отримання інформації про сучасні групи;
- завантаження на хостинг;
- тестування сайту.

Було виконано огляд сучасних систем управління контентом для розробки вебсайта.

Здійснено проєктування вебсайта, а саме виконано опис вимог до вебсайту, побудовано діаграми варіантів використання, діаграму розгортання, а також розроблено технічне завдання на розробку вебсайту.

Вебсайт прихильників української музики був створений у безкоштовному варіанті середовища розробки Atom. Було виконано тестування вебсайту на комп’ютері та на мобільному пристрої.

Розроблений вебсайт слугуватиме інструментом інформування та залучення молоді до сучасної української музики, сприятиме підвищенню зацікавленості цільової аудиторії та формуванню спільноти прихильників.

Таким чином, були вирішені всі поставлені завдання. Сайт функціонує у повному обсязі, уся інформація доступна для користувача. Мету кваліфікаційного проєкту досягнуто в повному обсязі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bootstrap 5: підручник для початківців. URL: https://w3schoolsua.github.io/bootstrap5/index.html (дата звернення: 08.02.2025).

2. Bootstrap: основні поняття та початок роботи з фреймворком. URL: https://foxminded.ua/bootstrap (дата звернення: 08.02.2025).

3. DevZone. Візуальний гід по Bootstrap 4. URL: https://devzone.org.ua/post/vizualnyy-hid-po-bootstrap-4 (дата звернення: 14.03.2025).

4. Козак В.І. Архітектура вебсайтів: сучасні підходи. Комп’ютерні науки і інформаційні технології. 2022. Вип. 21. С. 34–45.

5. Мельник І.В. Юзабіліті вебресурсів: методи та практика. Київ: Видавництво КНУ, 2023. 176 с.

6. Мержевич В. Посібник з Bootstrap 4 URL: https://html.in.ua/blog/rukovodstvo-po-bootstrap-4 (дата звернення: 08.02.2025).

7. Петренко С.С. Сучасні підходи до структурування вебсайтів. Вісник КНУ. 2024. № 10. С. 78–85.

8. Типи сайтів URL: https://w3schoolsua.github.io/bootstrap/bootstrap_ver.html (дата звернення: 20.03.2025).

9. Ткачук О.І. Основи веб-дизайну: навч. посіб. Львів: Видавництво Львівської політехніки, 2021. 208 с.

10. Хостинг. URL: https://uk.wikipedia.org/wiki/Хостинг. – Дата звернення: 20.03.2025.

11. Шевченко М.М. Проєктування та організація інформації в Інтернет. Київ: Наукова думка, 2020. 142 с.

ДОДАТОК А



Рисунок А.1 –Сторінки музичної групи 5Nizza на вебсайті

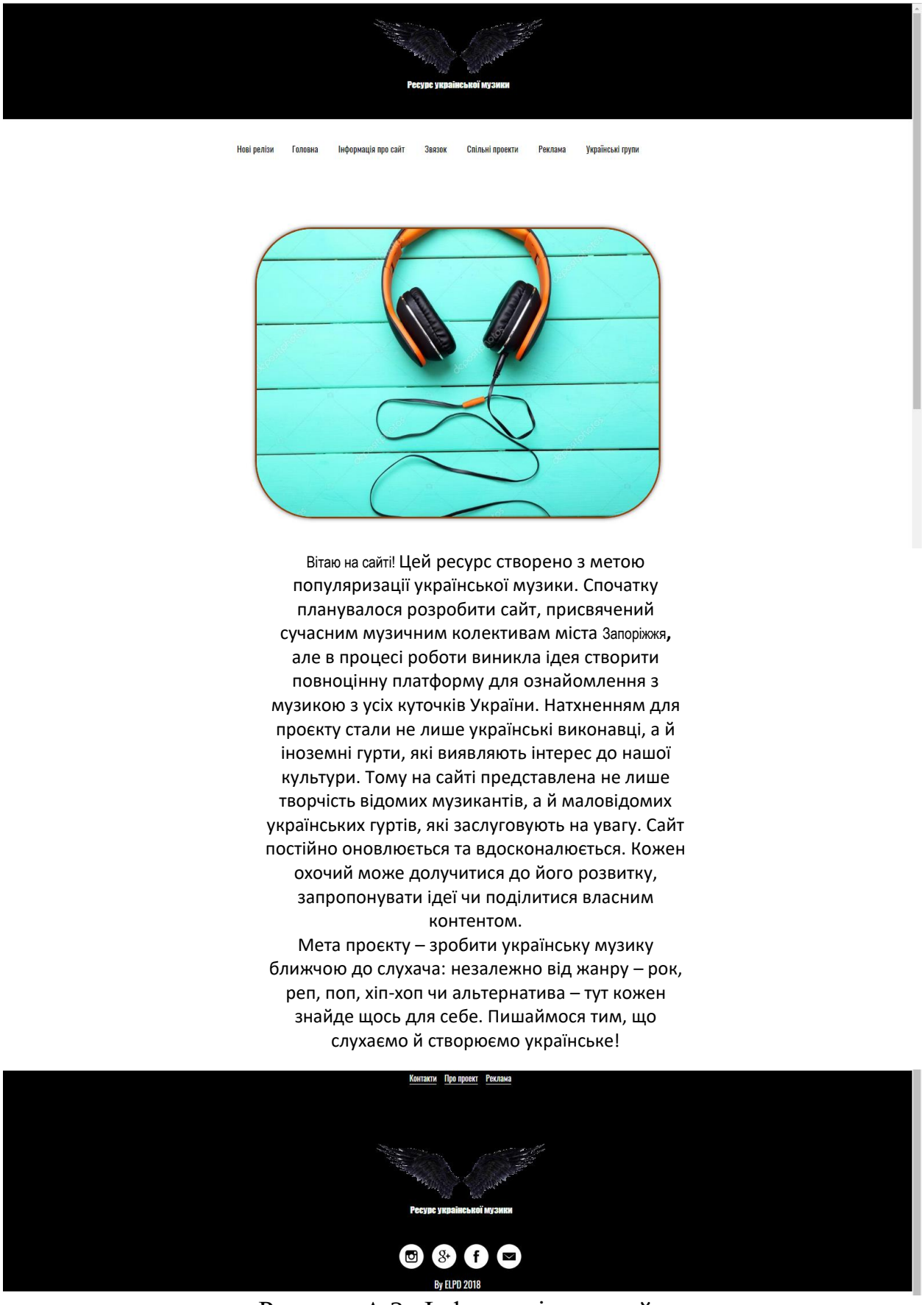


Рисунок А.2 –Інформація про сайт

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничий фаховий коледж ЗНУ»

Я, Гусєв Олександр Сергійович, здобувач освіти 4 курсу, денної форми здобуття освіти, спеціальності 121 Інженерія програмного забезпечення, освітньо-професійної програми/Розробка програмного забезпечення, групи К 121-21, адреса електронної пошти gusevaleksandr270@gmail.com;

– підтверджую, що написана мною кваліфікаційна робота на тему «Розробка вебсайту посвяченому українській музиці» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений;

– заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

– згоден на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата _____ Підпис _____ Олександр ГУСЄВ_____

Дата _____ Підпис _____ Альона ХОДАКОВСЬКА____