

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Сайт студії для навчання діджеїнгу»

Виконав студент групи ІПЗ-21-5
ГОРОШКО Дмитрій Юрійович

Керівник роботи:
ВЛАСЕНКО Олег Васильович

Рецензент:
ГРОМСЬКИЙ Олександр Олександрович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2025р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ГОРОШКО Дмитрій Юрійович**

Керівник роботи: **ВЛАСЕНКО Олег Васильович**

Тема роботи: **«Сайт студії для навчання діджеїнгу»**,

затверджена Наказом закладу вищої освіти від **«14» лютого 2025р., №61/с**

Вихідні дані для роботи:

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів,
що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Власенко О.В.	27.02.2025	27.02.2025
2	Власенко О.В.	27.02.2025	27.02.2025
3	Власенко О.В.	27.02.2025	27.02.2025

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з сайту студії для навчання діджеїнгу та пояснювальної записки. Пояснювальна записка до випускної роботи загалом містить 68 сторінки, з яких 49 сторінок займає основний текст і 19 сторінок — додатки. Документ включає 35 ілюстрації, 12 таблиць та 30 найменування використаних джерел.

Метою роботи є розробка зручного, адаптивного вебзастосунку для музичної школи, який дозволяє користувачам переглядати інформацію про заняття, бронювати доступні дати через інтерактивний календар та здійснювати оплату за допомогою інтегрованої платіжної системи LiqPay. Вебсайт також забезпечує зручний доступ до контактної інформації викладачів та загальної інформації про школу.

Архітектура застосунку реалізована з урахуванням принципів розділення відповідальностей: фронтенд частина побудована за допомогою React, а бекенд частина реалізована як REST API на базі WordPress. Такий підхід дозволяє легко масштабувати функціонал та забезпечує ефективну інтеграцію з зовнішніми сервісами.

Інтерфейс сайту інтуїтивно зрозумілий, адаптований для різних типів пристроїв і орієнтований на кінцевого користувача. Реалізована система бронювання враховує можливі конфлікти доступності дат, а модуль оплати гарантує безпечне та швидке здійснення транзакцій.

КЛЮЧОВІ СЛОВА: REACT, WORDPRESS, REST API, LIQPAY, ВЕБЗАСТОСУНОК, DJ-ШКОЛА, БРОНЮВАННЯ, ОПЛАТА ОНЛАЙН, АДАПТИВНИЙ ІНТЕРФЕЙС.

					ІПЗ.КР.Б-121.25 - ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Горошко Д.Ю.			Сайт студії для навчання діджеїнгу			Лім.		Арк.	Аркушів	
Керівник		Власенко О.В.								4	46	
Н. контр.								ФІКТ, гр. ІПЗ-21-5				
Затвердив												

Зміст

РЕФЕРАТ	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ САЙТУ СТУДІЇ ДІДЖЕЇНГУ	8
1.1. Постановка задачі	8
1.2. Аналіз аналогів програмного продукту	9
1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	16
Висновки до першого розділу.	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ САЙТУ СТУДІЇ НАВЧАННЯ ДІДЖЕЇНГУ ...	19
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	19
2.2. Розробка системи збереження даних	24
2.3. Проектування та реалізація алгоритмів роботи системи.....	25
2.4 Реалізація функціоналу сайту студії навчання діджеїнгу	28
Висновки до другого розділу	32
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ДОДАТКУ	34
3.1 Порядок встановлення та налаштування системи.....	34
3.2 Структура інтерфейсу та порядок взаємодії з веб застосунком	35
Висновки до третього розділу	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
Додатки.....	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface – інтерфейс прикладного програмування

CMS – Content Management System – система керування вмістом

DJ – Disc Jockey – диск-жокей

HTML – HyperText Markup Language – мова розмітки гіпертексту

HTTP – HyperText Transfer Protocol – протокол передавання гіпертексту

JSON – JavaScript Object Notation – формат обміну даними

REST – Representational State Transfer – архітектурний стиль для вебсервісів

UI – User Interface – інтерфейс користувача

URL – Uniform Resource Locator – уніфікований локатор ресурсу

UX – User Experience – досвід користувача

DOM – Document Object Model – об'єктна модель документа

JS – JavaScript – мова програмування для динамічного контенту на вебсторінках

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ВСТУП

Актуальність теми. У сучасних умовах цифрової трансформації створення вебсайтів стає ключовим інструментом для забезпечення онлайн-присутності та автоматизації бізнес-процесів у різних сферах, зокрема в освіті та культурі. Для студій, що займаються навчанням діджеїнгу, важливо мати сучасний, інформативний і функціональний вебсайт, який дозволяє не лише представляти інформацію про курси, викладачів та розклад занять, а й забезпечує можливість онлайн-запису, зворотного зв'язку та адміністрування контенту.

З позиції написання програмних систем, розробка такого вебресурсу є актуальним завданням, що вимагає застосування сучасних технологій веб-програмування, баз даних, принципів клієнт-серверної архітектури, адаптивного дизайну та забезпечення захисту персональних даних користувачів. Створення сайту включає повний цикл життєвого циклу програмного забезпечення — від аналізу вимог до тестування та впровадження, що сприяє формуванню практичних навичок майбутніх фахівців у галузі програмної інженерії.

Таким чином, проєкт вебсайту навчальної студії діджеїнгу є не лише прикладом впровадження ІТ-рішень у творчу індустрію, а й актуальним завданням яке дозволяє реалізувати комплекс знань і навичок у реальному прикладному середовищі.

Сайт студії для навчання діджеїнгу покликаний стати інформативною платформою, що сприятиме популяризації діджеїнгу серед широкої аудиторії, залученню нових учнів і підвищенню рівня організованості навчального процесу.

Метою випускної кваліфікаційної роботи є розробка функціональної, зручної та безпечної веб-платформи для навчання діджеїнгу, яка забезпечить ефективну взаємодію між учнями та викладачами, дозволить здійснювати запис на заняття, переглядати розклад, отримувати навчальні матеріали тощо. Крім того, платформа має на меті сприяти автоматизації внутрішніх процесів студії діджеїнгу, зокрема адміністрування груп, оплат, зворотного зв'язку та аналітики.

Для реалізації поставленої мети потрібно виконати наступні завдання:

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

1. Провести аналіз існуючих аналогів подібних платформ.
2. Здійснити вибір засобів проектування.
3. Розробити архітектуру платформи (сайту), а саме: розробити серверну частину та клієнтську частину.
4. Провести тестування сайту для виявлення та усунення помилок, а також оцінити якість створеного програмного продукту.

Об’єкт дослідження це процес створення, розгортання та впровадження веб-орієнтованого навчального додатку, призначеного для підтримки освітнього процесу у сфері діджеїнгу. До об’єкта дослідження входить розробка інтерфейсів взаємодії, структурування даних, а також інтеграція із зовнішніми сервісами для покращення користувацького досвіду.

Предмет дослідження це методи, технології та інструменти проектування, розробки та тестування сучасних веб-платформ навчального призначення. Особлива увага приділяється підходам до реалізації клієнт-серверної архітектури, адаптивного дизайну, систем бронювання та оплати, засобам автентифікації, а також забезпеченню зручності користування та безпеки персональних даних користувачів.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ САЙТУ СТУДІЇ ДІДЖЕЇНГУ

1.1. Постановка задачі

Метою кваліфікаційної роботи бакалавра є розробка вебсайту для студії навчання діджеїнгу з використанням сучасних підходів та технологій веброзробки, що забезпечить можливість зручного оновлення інформації, розширення функціоналу та подальшого вдосконалення проєкту з боку адміністратора студії. З боку користувача сайт має забезпечити зручний доступ до актуальної інформації про навчальні програми, викладачів, розклад занять, а також комфортний інтерфейс для запису на курси та комунікації зі студією.

Сайти, що пропонують офлайн-навчання діджеїнгу, слугують як віртуальні вітрини реальних шкіл, студій або академій, де охочі можуть записатися на курси, пройти живі майстер-класи або повноцінні програми з навчання діджеїнгу. Такі ресурси зазвичай містять детальний опис програм, графік занять, відомості про викладачів, умови вступу та вартість. Однією з перших і найвідоміших установ такого типу є Point Blank Music School, заснована у Лондоні в 1994 році. Її сайт дозволяє не лише дізнатися про курси, а й зареєструватися на очне навчання у філіях у Лондоні, Лос-Анджелесі, Ібіці та інших містах.

Схожі можливості пропонують і сайти інших діджейських академій, як-от DJ Lab (Барселона), Dubspot (Нью-Йорк), SAE Institute (глобальна мережа шкіл звукорежисури), а також локальні ініціативи в багатьох країнах, зокрема й в Україні — наприклад, True School DJ Academy у Києві або Mixmaster DJ School. Завдяки сайтам цих закладів користувачі можуть ознайомитися з офлайн-програмами, переглянути фотогалереї навчальних просторів, отримати консультацію онлайн та одразу записатися на курс з діджеїнгу у своєму місті або найближчому регіоні.

Сайт має представляти собою не лише сторінку з наданою інформацією про студію та викладачів, а і надавати можливість обрати дату та записатись на заняття онлайн, не виходячи з дому.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

Основними етапами розробки платформи є створення дизайну вебсайту, побудова архітектури клієнтської та серверної частин, реалізація основного функціоналу, а також тестування додатку.

Для реалізації поставленої мети необхідно виконати низку послідовних завдань. Насамперед слід провести аналіз існуючих аналогів подібних платформ, що дозволить визначити їхні функціональні особливості, переваги та недоліки, а також сформулювати вимоги до майбутнього вебсайту. Далі потрібно здійснити вибір засобів проєктування, включаючи мови програмування, фреймворки, системи керування базами даних та інші інструменти, які будуть використовуватися під час розробки.

Далі, проєктування архітектури самої платформи, зокрема побудова логічної структури клієнтської та серверної частин, визначення принципів їх взаємодії та організація обробки даних. Завершується процес розробки тестуванням створеного сайту з метою виявлення та усунення помилок, а також перевірки якості, стабільності й адаптивності функціоналу на різних пристроях.

Результатом виконання плану дій є створення платформи, за допомогою якої користувачі можуть записатись на фізичні курси з діджеїнгу, отримати інформацію про викладачів та студію.

1.2. Аналіз аналогів програмного продукту

Розробка сайту подібного наповнення та жанру потребує глибокого аналізу та дослідження аналогів на ринку. Оскільки подібних сайтів досить багато, серед них потрібно виділитись унікальністю, зручністю та індивідуальним підходом до задач. Тому почнемо розглядати схожі за тематикою програмні рішення.

Почнемо з сайту *djademy*: DJ Academy — це навчальний центр у Києві, який пропонує курси діджеїнгу та музичного продакшну для початківців і тих, хто бажає поглибити свої навички. Вони надають індивідуальні заняття, сучасне обладнання та можливість практики на реальних івентах.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

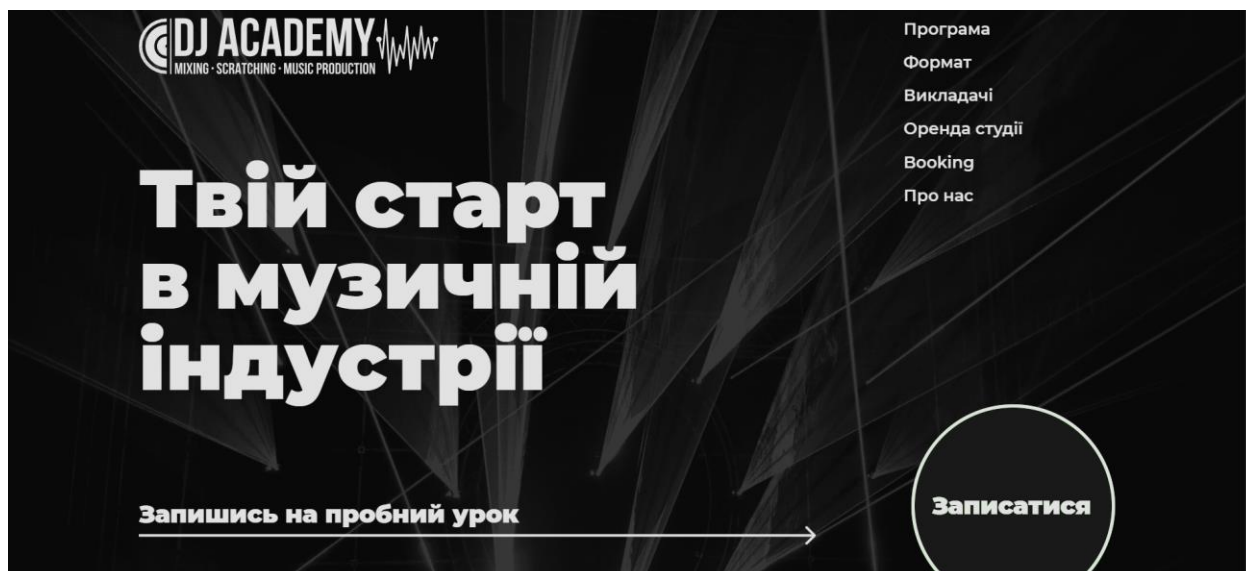


Рис. 1.1. DJ Academy

До переваг платформи можна віднести її інформативність — сайт є зручним у користуванні, а основна інформація майже повністю зосереджена на одній сторінці, що дозволяє користувачам швидко ознайомитися з ключовими аспектами діяльності студії. Крім того, позитивною стороною є привабливий та індивідуально розроблений дизайн, який створює позитивне враження та сприяє формуванню впізнаваного візуального стилю.

Серед недоліків слід зазначити, що інтерфейс сайту місцями виглядає перевантаженим: інформація хоч і подана на головній сторінці, однак структура і розміщення елементів можуть створювати враження "інформаційної каші", а візуальний акцент не падає на ключові кнопки, зокрема кнопку запису на заняття. Ще одним суттєвим недоліком є відсутність функції онлайн-оплати. Користувач має змогу лише залишити свій номер телефону та спосіб зв'язку, проте не може здійснити запис на конкретну дату або оплатити заняття безпосередньо через платформу.

Наступним розглянемо kulturazvuka — це українська музична формація, яка об'єднує спільноту діджеїв, продюсерів та слухачів електронної музики. Заснована у Харкові, вона розвинулася у багатофункціональний культурний хаб, що включає клуб, студію звукозапису, музичну школу, букінг-агентство, лейбл, магазин вінілу та Hi-Fi техніки, а також сервіс оренди професійного звукового обладнання:

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

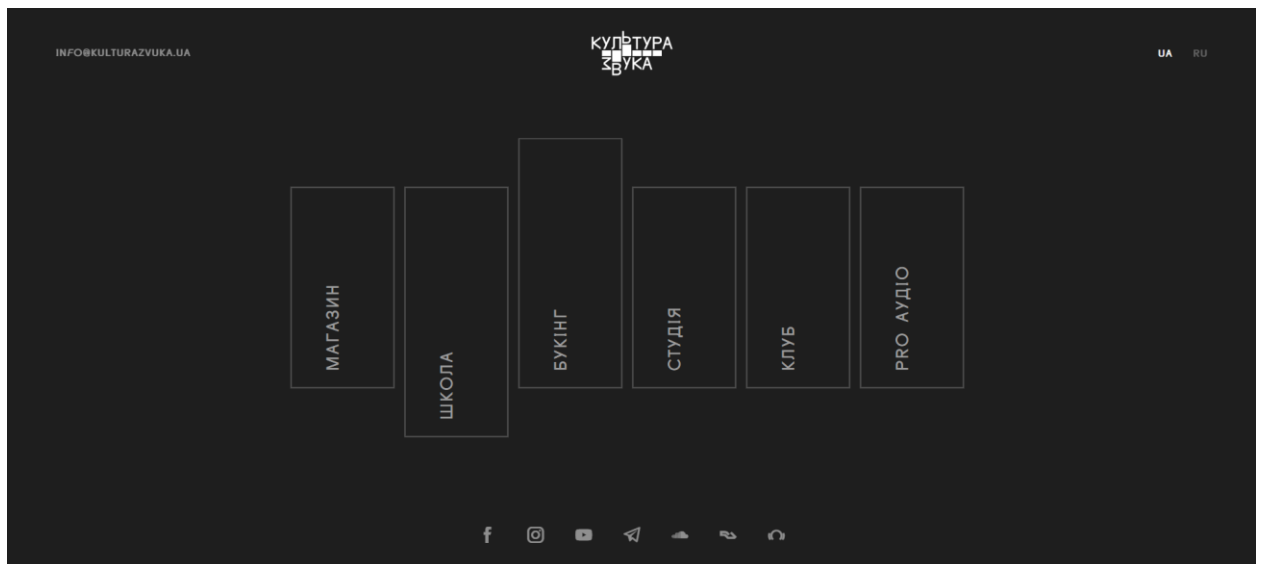


Рис. 1.2. kulturazvuka

Перевагами сайту є його приємний дизайн та зручна навігація, що дозволяє користувачам швидко орієнтуватися на сторінках і знаходити потрібну інформацію. Весь контент подано у зрозумілому та структурованому вигляді, що покращує сприйняття та спрощує ознайомлення з діяльністю студії. Додатковою перевагою є підтримка кількох мов, що розширює потенційну аудиторію платформи та робить її доступнішою для користувачів з різних регіонів.

Серед недоліків можна відзначити незручну форму запису на заняття. Усі поля форми автоматично форматують введений текст так, що він починається з великої літери. Це створює труднощі під час введення капчі, оскільки користувач не завжди розуміє, чи активовано Caps Lock, що особливо заважає при сліпому друці. Відсутність повідомлення про ввімкнений Caps Lock лише ускладнює ситуацію, змушуючи користувача перевіряти розкладку вручну та витратити зайвий час на повторне введення.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11



Рис. 1.3. PRODJ School

PRODJ School пропонує понад 12 курсів з діджеїнгу та саундпродюсування для музикантів будь-якого рівня. Новачки можуть освоїти різні музичні стилі, базові техніки зведення, мікшування та скретчингу. Більш досвідчені музиканти мають можливість розширити свої знання та отримати корисні навички. Особливістю курсів для діджеїв є велика кількість практики. У школі працюють досвідчені викладачі-практики та відомі українські діджеї.

Серед переваг сайту варто відзначити привабливий сучасний дизайн, який створює позитивне враження та відповідає сучасним трендам у веброзробці. Також платформа має зручну форму запису на заняття, що спрощує взаємодію користувача з сайтом та сприяє швидкому оформленню заявки. Інформація представлена чітко та логічно, що дозволяє відвідувачам легко орієнтуватися на сторінках і знаходити потрібні дані без зайвих зусиль.

Водночас існують і деякі недоліки. По-перше, неоптимізований код може спричиняти повільне завантаження сторінок або неправильне відображення на окремих браузерях. По-друге, хоча сайт і є адаптивним, деякі елементи, зокрема блок із відгуками учнів, відображаються некоректно на мобільних пристроях — зображення накладаються на текст, що погіршує зручність користування. Крім

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

того, є проблеми з SEO-оптимізацією: неправильна структура URL-адрес, відсутність мета-описів або неякісний контент можуть негативно впливати на ранжування сайту в пошукових системах.



Рис. 1.4. Resonance Music School

Resonance Music School є навчальним закладом, який спеціалізується на підготовці майбутніх діджеїв та саундпродюсерів. Школа пропонує різноманітні навчальні програми, розраховані на студентів з різним рівнем підготовки – від початківців, які тільки починають свій шлях у світі музики, до більш досвідчених музикантів, які прагнуть поглибити свої знання та вдосконалити навички. Однією з ключових особливостей школи є індивідуальний підхід до навчання, що дозволяє викладачам приділяти максимум уваги кожному студенту, враховуючи його темп навчання та індивідуальні потреби.

Серед переваг сайту можна виділити зручну форму запису на заняття, яка дозволяє швидко та без зайвих труднощів залишити заявку. Дизайн сайту виглядає стильно та сучасно, що позитивно впливає на візуальне сприйняття й формує професійний імідж студії. Важливою перевагою є адаптивний дизайн,

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

який коректно працює на більшості розмірів екранів, забезпечуючи комфортне користування як на комп'ютерах, так і на мобільних пристроях. Додаткову естетичну цінність додає той факт, що всі елементи, включно з картою розташування студії, витримані в єдиному стилі й гармонійно поєднуються з кольоровою гамою сайту.

Проте основним недоліком є те, що сайт не має власного унікального коду — він функціонує на базі CRM-платформи EasyWeek. Це обмежує можливості кастомізації, розширення функціоналу та оптимізації, що в перспективі може стримувати розвиток сайту або його адаптацію під специфічні потреби студії.



Рис. 1.5. Vedel School

Vedel School позиціонує себе як музична школа нового покоління, що пропонує цілісний та гармонійний підхід до музичної освіти без кордонів. Вони наголошують на розвитку музичного мислення студентів незалежно від обраного жанру. Засновником школи є український джазовий саксофоніст та музикант Михайло Балог.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

Серед переваг сайту варто відзначити інтеграцію з відеоматеріалами, зокрема з YouTube. Активна присутність школи на цій платформі та розміщення відео на сайті сприяють підвищенню залученості користувачів і дозволяють їм отримати більше інформації про курси, переглянувши демонстраційні або ознайомчі ролики. Також позитивним моментом є наявність форм зворотного зв'язку, включно з контактними формами або чатами, що полегшує комунікацію з адміністрацією та дає змогу оперативно отримувати відповіді на запитання або уточнення.

Разом із тим сайт має і певні недоліки. Він може здаватися перевантаженим - велика кількість спливаючих вікон і вбудованих відео уповільнює його роботу, особливо при слабкому інтернет-з'єднанні або на малопродуктивних пристроях. Крім того, хоча сайт має адаптивний дизайн, навігація на мобільних пристроях залишається складною: окремі елементи можуть бути розташовані незручно або некоректно масштабуватися на малих екранах. Ще одним недоліком є відсутність інтеграції з системами онлайн-платежів, що ускладнює користувачам можливість оплатити курси безпосередньо через сайт, змушуючи їх використовувати альтернативні або ручні способи оплати.

Таблиця 1.1

Порівняльна таблиця аналогів

Назва сайту	Зручна форма запису	Онлайн-оплата	Адаптивний дизайн	Багатомовність	Унікальний код
DJ Academ y	+-	-	+	-	+
kulturaz vuka	-	-	+	+	+
PRODJ School	+	-	-	-	+
Resonan ce	+	-	+	-	- (на базі EasyWeek)
Vedel School	+	-	+-	-	+

Загальний аналіз існуючих аналогів дає змогу визначити ключові напрями, у яких варто рухатися під час створення власного вебсайту для навчальної DJ-

студії. Більшість наявних рішень мають сильні сторони в дизайні, інформативності та базовому функціоналі запису на заняття, проте майже всі з них мають спільні недоліки, які можна усунути у власному продукті.

Передусім варто зробити зручну та функціональну форму запису на заняття, яка дозволяє користувачеві самостійно обирати дату, курс, викладача та одразу підтверджувати запис. Це має бути інтуїтивно зрозумілий процес із мінімальною кількістю кроків.

Також, важливим кроком є реалізація можливості онлайн-оплати. Жоден із розглянутих сайтів не надає такої функції, хоча вона є очікуваною в сучасних цифрових продуктах. Додавання онлайн-оплати напряму через сайт (наприклад, за допомогою інтеграції з платіжними системами) значно підвищить зручність для клієнтів і конкурентоспроможність платформи.

Ще одним ключовим елементом є унікальність реалізації — варто уникати шаблонних рішень або платформ-основ (як у випадку з EasyWeek), і створити власний код, який дозволить гнучко змінювати, доповнювати або масштабувати сайт відповідно до потреб студії.

1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Опис вибору технологій для реалізації нашої платформи був сформований на основі порівняння декількох популярних рішень, які активно використовуються в сучасній веброзробці. Основна увага приділялася гнучкості, швидкості розробки, можливості масштабування, а також простоті підтримки та розвитку проекту в майбутньому.

Для реалізації бекенд-частини платформи було обрано мову PHP у поєднанні з фреймворком WordPress. Хоча на ринку існують численні альтернативи — наприклад, Node.js (з використанням Express.js) або Python (з фреймворками Django чи Flask) — кожне з цих рішень має свої переваги та недоліки. Node.js часто обирають за швидкодію та можливість написання і серверного, і клієнтського коду на одній мові — JavaScript. Django, у свою чергу,

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

пропонує високий рівень безпеки й структурування коду, що добре підходить для складних корпоративних систем.

Однак у нашому випадку було обрано саме WordPress, оскільки ця система керування вмістом (CMS), створена на PHP, є оптимальною для проєктів із фокусом на контент, швидку розробку, просте адміністрування та активне використання API. WordPress дозволяє легко реалізувати необхідний функціонал через плагіни, підтримує REST API «з коробки» й має вбудовану адміністративну панель, яка значно полегшує роботу з контентом. Це стало вирішальним фактором у контексті проєкту, де важливо швидко реалізувати MVP і при цьому зберегти можливість подальшого масштабування.

Фронтенд-частина реалізується з використанням React — сучасної JavaScript-бібліотеки для створення інтерфейсів користувача. Хоча існують інші популярні рішення — наприклад, Vue.js (відомий своєю простотою інтеграції та швидким стартом) або Angular (фреймворк з потужною структурою, підходящий для великих enterprise-проєктів) — React було обрано завдяки його гнучкості, масштабованості, великій спільноті та підтримці сучасних концепцій, таких як компонентний підхід, віртуальний DOM та односпрямований потік даних. Це дозволяє створювати динамічні, швидкі й зручні у використанні вебінтерфейси.

Крім того, React добре інтегрується з API, що є критично важливим для проєкту. Взаємодія між клієнтською та серверною частинами реалізується через API, який надається сервером (WordPress) та використовується на фронтенді. Для реалізації цього API використовується стандартний REST-підхід, реалізований у WordPress через спеціальні плагіни. Це забезпечує ефективну взаємодію між двома частинами системи.

API (Application Programming Interface) — це набір правил і механізмів, який дозволяє різним програмним компонентам взаємодіяти між собою. У нашому випадку це дозволяє React-застосунку звертатися до серверної частини на WordPress для отримання або надсилання даних, таких як інформація про заняття, викладачів чи заявки користувачів.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

Таким чином, зваживши переваги кожної з технологій, ми зупинилися саме на зв'язці WordPress (PHP) для бекенду та React (JavaScript) для фронтенду як найбільш раціональному рішенні для нашої платформи — з урахуванням цілей, бюджету, часу розробки та необхідного функціоналу.

Висновки до першого розділу.

У межах підготовчого етапу проєкту було проведено детальний аналіз предметної області, що дозволило краще зрозуміти специфіку функціонування освітніх платформ, орієнтованих на сферу діджеїнгу. Це дало змогу визначити ключові функціональні та нефункціональні вимоги до майбутньої системи, враховуючи очікування цільової аудиторії, сучасні тенденції в UX/UI-дизайні, а також технічні можливості вебплатформ.

Під час дослідження ринку аналогічних продуктів було розглянуто низку популярних та функціональних сайтів, що мають схожу тематику. Для кожного з них були виявлені сильні та слабкі сторони, а також проведено порівняння за критеріями зручності, технічної реалізації, підтримки онлайн-сервісів та загального рівня доступності для користувача. Це дозволило виділити низку недоліків, на яких варто зосередитися при створенні власного продукту.

На основі зібраної інформації сформовано загальні вимоги до функціоналу сайту: обов'язкова наявність зручної форми запису на заняття, можливість онлайн-оплати, адаптивність для мобільних пристроїв, а також підтримка багатомовності. Крім того, було визначено важливість створення унікального продукту без використання шаблонних рішень.

У результаті дослідження було сформовано набір відповідних технологій, інструментів та середовищ розробки, що найбільш повно відповідають поставленим цілям і дозволять реалізувати платформу згідно з визначеними вимогами.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

РОЗДІЛ 2. ПРОЕКТУВАННЯ САЙТУ СТРУДІЇ НАВЧАННЯ ДІДЖЕЇНГУ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Наша платформа для студії навчання діджеїнгу надає зручний вебінтерфейс, за допомогою якого користувачі можуть ознайомитися з інформацією про студію, викладачів та адміністрацію, а також записатися на заняття, обравши доступну дату з вільних у розкладі. Крім того, передбачена можливість здійснити оплату за заняття онлайн через платіжну систему LiqPay у кілька простих дій.

Адміністративна частина сайту дозволяє уповноваженим особам змінювати контактну інформацію, яка відображається у футері сайту та на сторінці "Про нас", додавати або видаляти новини, а також вносити зміни до інформації про викладачів та інших учасників проєкту.

Для наочного відображення взаємодії між елементами системи та її користувачами була розроблена діаграма варіантів використання (рис. 2.1).



Рис. 2.1. Діаграма використання

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Опис діаграми варіантів використання наведено в таблицях 2.1 – 2.9.

Таблиця 2.1

Опис прецеденту “Додавання діджею”

Назва варіанту використання	Додавання діджею
Основні актори	Адмін
Короткий опис	Додавання нових діджеїв (викладачів на сайт через меню

Таблиця 2.2

Опис прецеденту “Додавання/редагування/видалення контактної інформації сайту”

Назва варіанту використання	Додавання/редагування/видалення контактної інформації сайту
Основні актори	Адмін
Короткий опис	Додавання, редагування та видалення контактної інформації, а саме: соціальних мереж та пошти через меню адміністрації

Таблиця 2.3

Опис прецеденту “Додавання/редагування/видалення новин”

Назва варіанту використання	Додавання/редагування/видалення новин
Основні актори	Адмін
Короткий опис	Додавання, редагування та видалення новин на головній сторінці сайту через меню адміністрації

Таблиця 2.4

Опис прецеденту “Змінення інформації сторінок сайту”

Назва варіанту використання	Змінення інформації сторінок сайту
Основні актори	Адмін
Короткий опис	Змінення вмісту сторінок, а саме: про нас, головної та інших через меню адміністрації

Таблиця 2.5

Опис прецеденту “Перегляд сторінок”

Назва варіанту використання	Перегляд сторінок
Основні актори	Звичайний користувач
Короткий опис	Можливість переглядати вміст всіх сторінок клієнтської частини сайту

Таблиця 2.6

Опис прецеденту “Запис на заняття”

Назва варіанту використання	Запис на заняття
Основні актори	Звичайний користувач
Короткий опис	Можливість обрати дату заняття із доступних та записатись на неї

Таблиця 2.7

Опис прецеденту “Оплата заняття через LiqPay”

Назва варіанту використання	Оплата заняття через LiqPay
Основні актори	Звичайний користувач
Короткий опис	Можливість оплатити заняття через сервіс LiqPay

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Таблиця 2.8

Опис прецеденту “Навігація до студії”

Назва варіанту використання	Навігація до студії
Основні актори	Звичайний користувач
Короткий опис	Допомагає знайти студії для людей які ні разу там не були

Таблиця 2.9

Опис прецеденту “Діджейська панель”

Назва варіанту використання	Діджейська панель
Основні актори	Звичайний користувач
Короткий опис	Самий безовий діджейський пульти в якому можна поробити маніпуляції з обраним треком

Основна ціль розробки сайту студії з навчання діджеїнгу полягає у створенні середовища яке дозволить користувачам зручно записуватись на заняття та зрозуміло і в приємному вигляді отримувати потрібну йому інформацію, таку як наприклад розташування студії (Так як у нас запланований певний «навігатор»).

Основна мета розробки полягає у сприянні становлення нового артисту шляхом залучення до спільноти діджеїв за допомогою новітніх підходів до цієї області.

Користувачами сайту є як внутрішні, так і зовнішні особи. Внутрішній користувач-адміністратор має можливість редагувати інформацію на сторінках сайту, додавати контактні дані, а також вносити нових викладачів разом із супровідною інформацією про них. Зі сторони клієнта функціонал передбачає перегляд сторінок сайту, ознайомлення з вільними датами занять, можливість запису на обрану дату та подальшу оплату заняття через платіжну систему LiqPay.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Функціональні вимоги до сайту формуються, перш за все, з позиції кінцевого користувача. Основна мета — забезпечити просту, логічну і зручну взаємодію з платформою для запису на заняття з діджеїнгу. Користувач має мати змогу без зусиль переглянути доступну інформацію про школу, ознайомитись із викладачами, обрати бажану дату з вільного розкладу та оперативно забронювати місце. Для завершення реєстрації має бути передбачена можливість онлайн-оплати — швидко, без зайвих переходів, у декілька кліків через інтеграцію з сервісом LiqPay.

Для адміністраторів система повинна надавати зручний інтерфейс керування вмістом сайту: можливість редагувати текстову та контактну інформацію, оновлювати новини, додавати або змінювати дані викладачів. Це має відбуватися без складних технічних процедур — у зрозумілій, інтуїтивно побудованій адміністративній панелі.

Що стосується нефункціональних вимог, сайт повинен бути швидким, адаптивним і стабільним у роботі. Всі елементи мають завантажуватись без помітних затримок — реакція на дії користувача не повинна перевищувати 3 секунд, а сторінка з оплатою повинна відкриватися не довше ніж за 6 секунд. Інтерфейс сайту має бути максимально зрозумілим навіть для нових користувачів — на знайомство з платформою має вистачати 3–5 хвилин, у той час як досвідчений користувач повинен орієнтуватися менш ніж за 2 хвилини. Крім того, вебсайт має коректно відображатися на всіх типах пристроїв — від смартфонів до великих моніторів, з підтримкою різних браузерів і платформ.

Щодо надійності, сайт повинен бути доступний цілодобово без простоїв. У сфері продуктивності важливо забезпечити можливість одночасного доступу до ресурсу великої кількості користувачів без зниження швидкодії.

Системні вимоги передбачають роботу сайту на пристроях із мінімальною апаратною конфігурацією: щонайменше 2 ГБ оперативної пам'яті, процесор із тактовою частотою не менше 1.4 GHz, а також стабільне підключення до Інтернету. Серверна частина повинна підтримувати технології React для клієнтського інтерфейсу та базу даних MySQL для збереження інформації.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

2.2. Розробка системи збереження даних

Для збереження інформації у додатку використовуються MySQL, оскільки ця система керування базами даних забезпечує надійне, масштабоване та ефективне зберігання даних. MySQL дозволяє організувати збереження користувацької інформації, результатів ігрового процесу, налаштувань та іншої важливої інформації в структурованому вигляді. Завдяки підтримці SQL-запитів, розробник має змогу легко здійснювати пошук, оновлення та видалення даних. Крім того, MySQL добре інтегрується з багатьма мовами програмування, включаючи PHP, Python та JavaScript, що робить її зручною для використання у веб- та десктопних застосунках.

В системі наявно дві власноруч створених таблиці, решта даних зберігається в зарезервованих Wordpress таблицях. Перейдемо до власноруч створених та розглянемо таблиці.

Таблиця 2.10

Опис таблиці booking_locks

Поле	Тип	Призначення
id	bigint(20)	Ідентифікатор
date_time	date	Дата бронювання
locked_until	datetime	Час до якого слот буде заблоковано

Дана таблиця відповідає за блокування слоту для запису після натискання на нього, блокування відбувається на 10 хвилин, це зроблено з ціллю уникнути подвійних натискань, а саме, коли два юзера майже одночасно натискають на дату та хочуть провести оплату, вони попадають на форму оплати та проводять її, але при цьому во ни записались на одну й ту саму дату, щоб цього уникнути і реалізовано систему букінг локу.

Таблиця 2.11

Опис таблиці booking_orders

Поле	Тип	Призначення
order_id	varchar(55)	Ідентифікатор
first_name	varchar(55)	Імя особи що бронює дату

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

last_name	varchar(55)	Фамілія особи що бронює дату
patronymic	varchar(55)	Прізвище особи що бронює дату
phone_number	varchar(55)	Номер телефону особи що бронює дату
email	varchar(55)	Емейл особи що бронює дату
day	date	День бронювання

Ця таблиця містить інформацію про успішно здійснені бронювання, включаючи особисті дані користувача та дату, на яку було заброньовано слот. Вона є основною для ведення обліку замовлень.

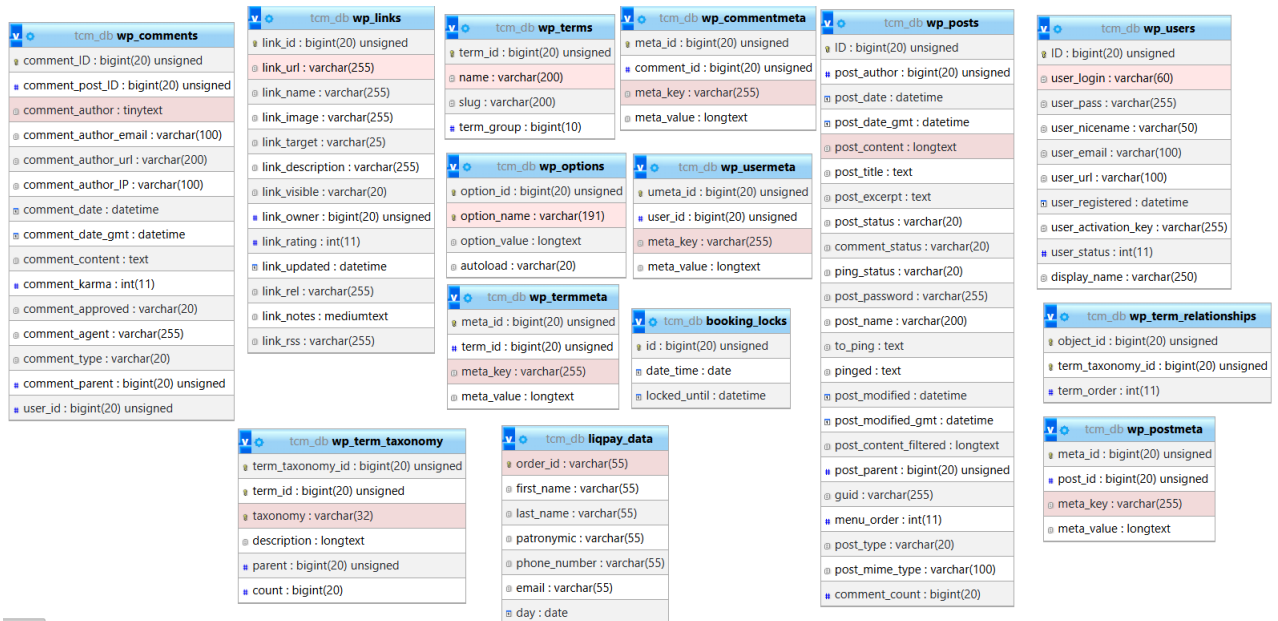


Рис. 2.2. Діаграма бази даних

2.3. Проектування та реалізація алгоритмів роботи системи

Додаток Додаток є простим у використанні та має всі основні функції притаманні подібній платформі, клієнтська частина спроектована для зручного використання юзером, в свою чергу адмін панель реалізована для коректного та комфортного додавання, редагування та видалення потрібного контенту. Алгоритм роботи програми описано на рисунку, також детально описано його пункти нижче.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

Для звичайного користувача передбачено інтуїтивний шлях взаємодії з сайтом. Спершу користувач потрапляє на головну сторінку, звідки може перейти до інших розділів ресурсу. Обравши сторінку з заняттями, він переходить на календар із відображенням найближчого місяця, де має змогу вибрати потрібну дату для заняття. Після натискання на конкретну дату вона автоматично резервується на 10 хвилин, протягом яких користувач повинен здійснити оплату. Якщо оплата пройшла успішно, ця дата стає недоступною для інших користувачів. У разі одночасного бронювання однієї дати кількома користувачами, система підтвердить бронювання лише тому, хто встиг зробити це раніше, а іншим буде повідомлено про недоступність обраної дати.

Оплата заняття здійснюється через сервіс LiqPay. Окрім цього, користувач має можливість перейти на сторінку «Про нас», де розміщена інформація про студію та школу, а також на сторінку «Контакти», де наведена контактна інформація про всіх викладачів.

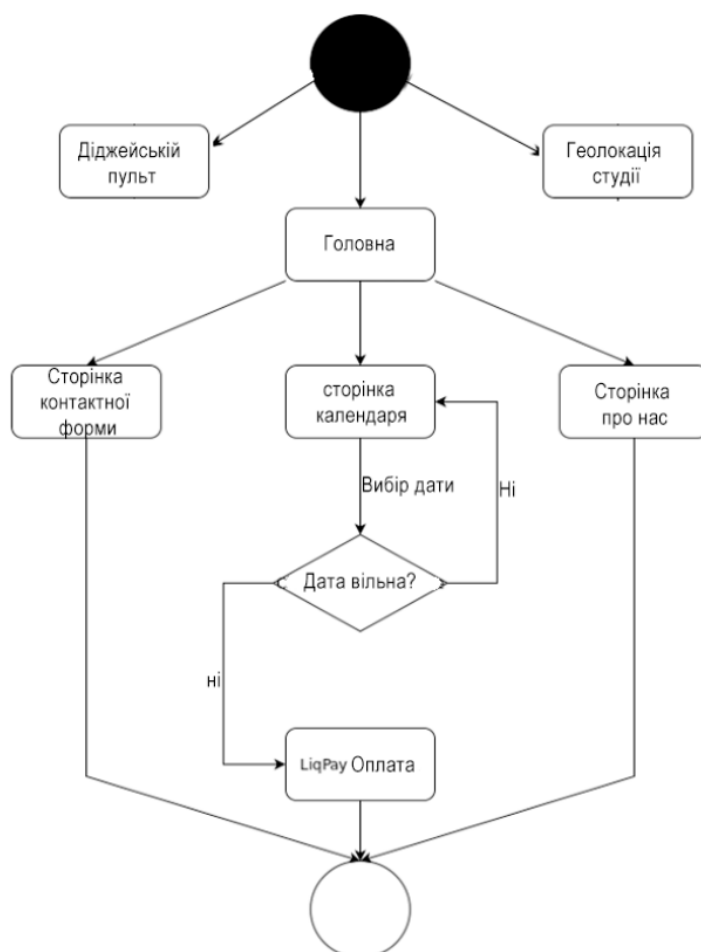


Рис. 2.3. Діаграма активності

Адміністративна панель, що реалізована в рамках платформи, є повноцінним інструментом для динамічного управління вмістом сайту без потреби в ручному втручанні в код чи базу даних. З технічної точки зору, вона створена як частина серверної логіки на базі WordPress, що дозволяє ефективно взаємодіяти з базою даних через API-запити. Цей підхід забезпечує зручність керування і водночас дозволяє розширювати функціонал без переробки основного ядра сайту.

У процесі розробки адмін-панель була інтегрована з системою авторизації та розмежування доступу, що дозволяє надавати адміністративні права лише перевіреним користувачам. Через неї адміністратор може створювати, змінювати або видаляти новини, що дозволяє оперативно оновлювати публічну інформацію для відвідувачів. Управління викладачами також реалізоване у вигляді окремого модуля — адмін має змогу додавати нових викладачів, змінювати їх дані, прикріплювати фотографії та опис, а також видаляти неактуальні профілі.

Окрему увагу було приділено редагуванню контактної інформації як для самих викладачів, так і для сайту загалом. Весь контент зберігається у внутрішній базі даних, і його можна змінювати через адміністративну форму, що автоматично оновлює відповідні блоки на клієнтській стороні через API. Це дозволяє зберігати структуру сайту сталою, водночас забезпечуючи повну контрольованість вмісту.

Для ефективного управління студійним процесом було реалізовано функціонал перегляду заброньованих і оплачених занять. Адміністратор бачить всі заплановані зустрічі в структурованому вигляді, що допомагає оперативно реагувати на зміни та слідкувати за завантаженістю розкладу. Уся інформація про бронювання синхронізується з платіжною системою LiqPay, що забезпечує актуальність та достовірність даних про оплату.

Таким чином, адмін-панель була розроблена як гнучкий, розширюваний інструмент керування сайтом, який дозволяє керувати всіма ключовими аспектами роботи платформи без необхідності втручання в технічну частину.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

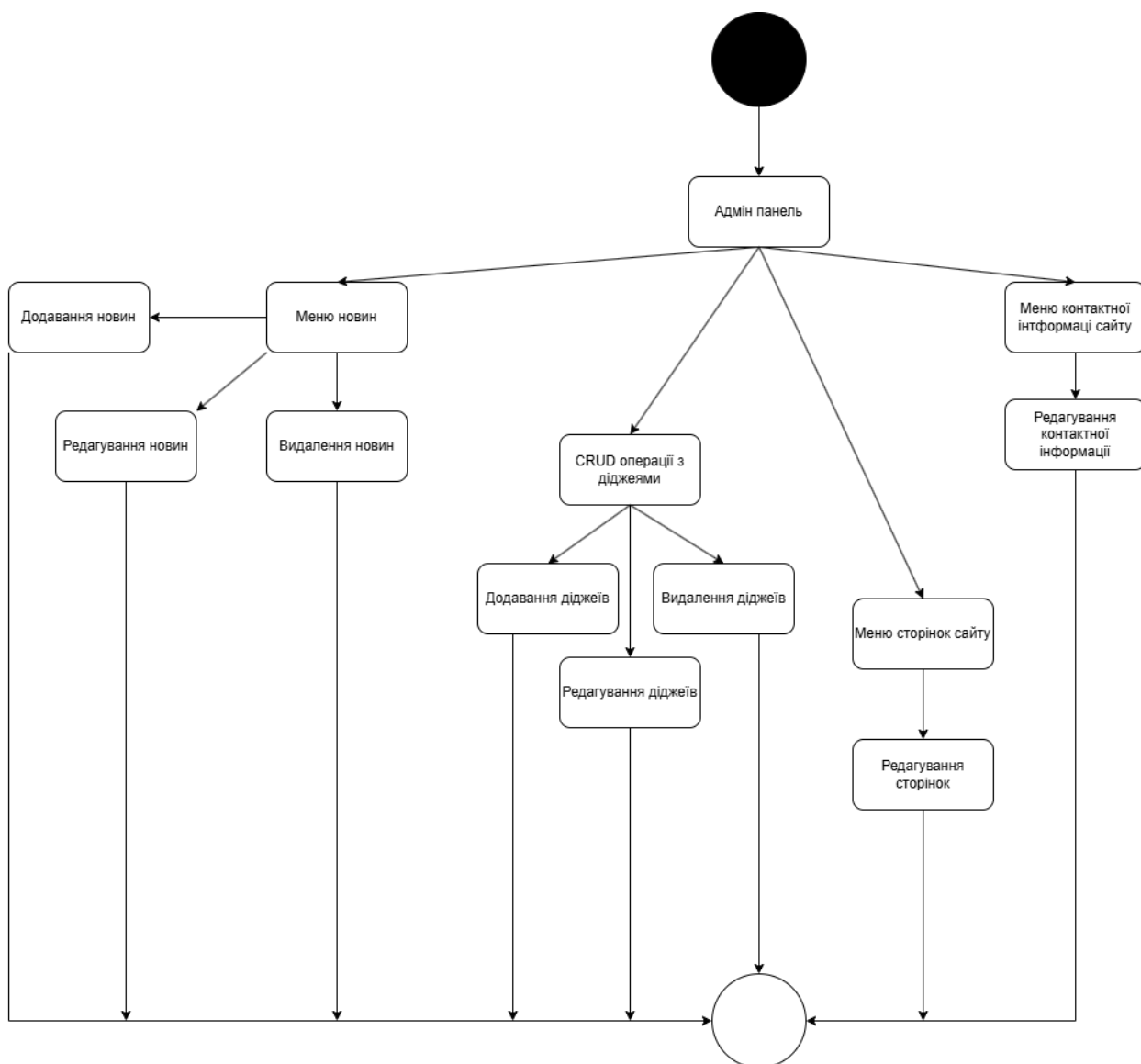


Рис. 2.4. Діаграма активності

2.4 Реалізація функціоналу сайту студії навчання діджеїнгу

Розгляд фронтенду починається з опису сервісу, який відповідає за звернення до API бекенду (Додаток А). Це набір асинхронних функцій, що працюють із платіжною системою LiqPay та дозволяють здійснювати запити до сервера. Всі функції використовують базову URL-адресу, яка імпортується з середовища за допомогою `import.meta.env.VITE_BASE_URL`.

Перейдемо до загальної структури додатку щоб краще розуміти описані в тексті моменти, саму діаграму можна побачити на рисунку 2.5.

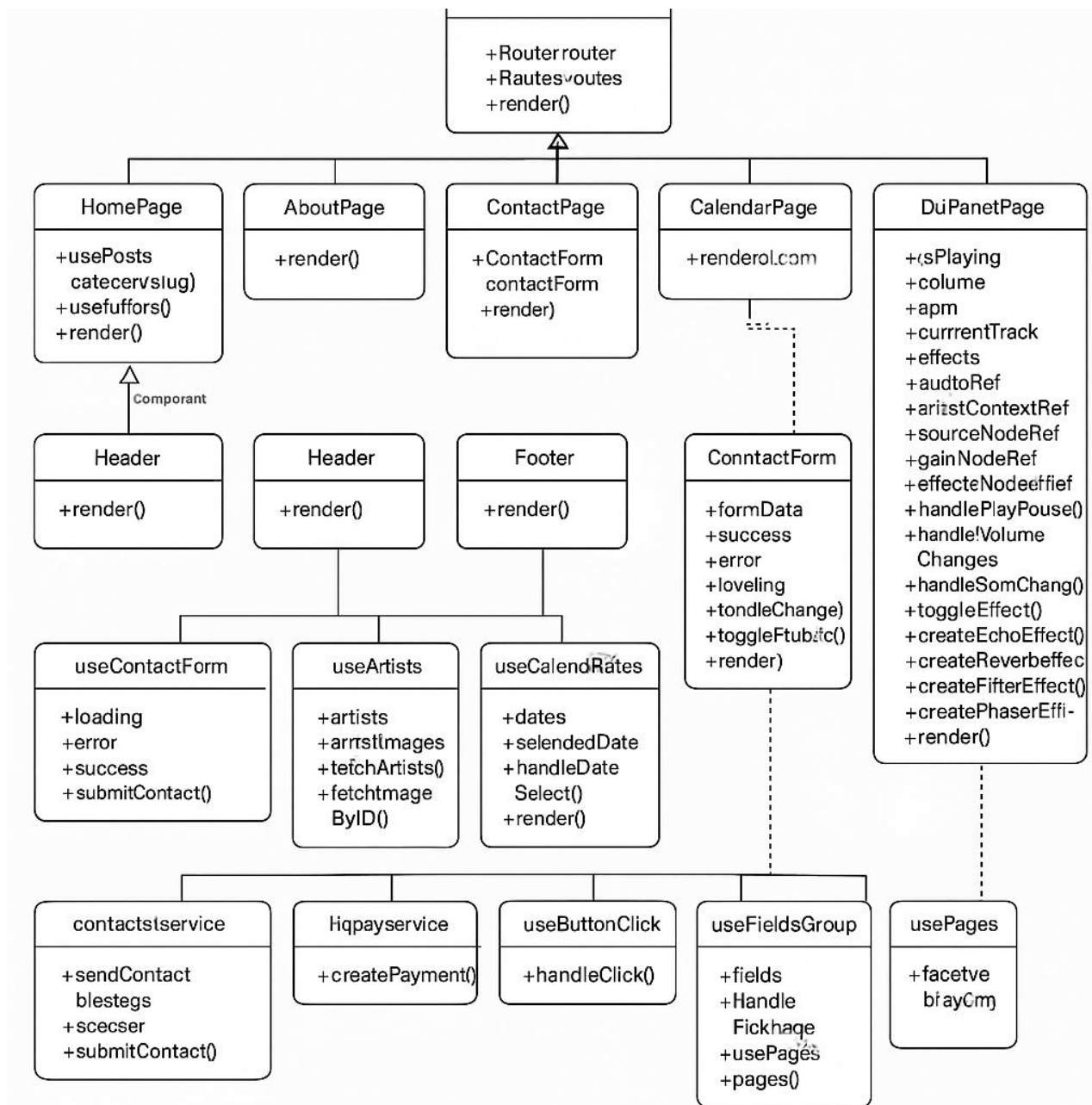


Рис. 2.5. Діаграма компонентів

Самі роути зберігаються тут: `VITE_BASE_URL='http://tcm/wp-json'`, це статична змінна яка зберігає в собі інформацію про URL бекенду нашого сайту.

Функція `fetchLiqPayForm` використовується для отримання HTML-форми оплати з LiqPay. Вона приймає параметри, зокрема суму, опис, унікальний номер замовлення (`order_id`) та інші дані. У результаті ця функція повертає HTML-код форми з кнопкою для здійснення платежу. У випадку помилки функція повертає `null`.

Функція `checkPaymentStatus` виконує перевірку статусу платежу на основі унікального ідентифікатора замовлення. Вона повертає відповідь зі статусом

транзакції — наприклад, успішно (success), не вдалося (failure), або в обробці (pending). Якщо під час перевірки виникає помилка, функція повертає статус "error".

Функція `sendTransactionData` відповідає за надсилання інформації про транзакцію на сервер. Вона приймає об'єкт, що містить дані транзакції (такі як сума, обрана дата, ім'я користувача тощо) і повертає відповідь від сервера у форматі JSON. У разі виникнення помилки, функція викидає її.

Остання функція — `isDatePicked` — використовується для перевірки, чи обрана або вже заброньована певна дата. Вона приймає дату у форматі, наприклад, "2025-05-01", і виконує перевірку на її доступність. Якщо дата вже заброньована або заблокована, функція повертає `true`, якщо ж дата ще доступна для вибору — повертає `false`.

Також в системі є хуки, наприклад – `useLiqPayForm` (Додаток В), який відповідає за завантаження HTML-форми `LiqPay` на основі заданих параметрів платежу. Всі хуки системи описані подібним чином, детальніше з ними як і з самим `useLiqPayForm` можна буде ознайомитись в додатку В.

Календар для бронювання дати заняття реалізований за допомогою `React` та спеціального хуку `useCalendarDates`, який відповідає за генерацію списку дат і визначення доступності кожної з них. Основна утиліта, яка використовується в цьому процесі — `generateDates`, створює масив із 28 послідовних дат, починаючи з третього дня після поточної дати. Кожна дата в цьому списку містить інформацію про те, чи вона доступна для бронювання. Для перевірки наявності дати використовується функція `checkDateAvailability`, яка визначає, чи дата вже заброньована або тимчасово недоступна для вибору.

Відображення кожної окремої дати в інтерфейсі реалізовано через компонент `CalendarDay`, який показує статус дати (доступна/недоступна) та дозволяє взаємодіяти з нею. Головним елементом календаря є компонент `Calendar`, який відповідає за побудову всього інтерфейсу вибору дати. Користувач за допомогою цього компоненту може переглянути календар на найближчі тижні, обрати доступну дату й перейти до оплати через `LiqPay`. Цей

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

компонент забезпечує зручну взаємодію між користувачем і системою бронювання. Компоненти календаря можна переглянути в додатку С, також хук useCalendarDates, як і решта хуків, описана в додатку В.

Наступний модуль у проєкті відповідає за взаємодію з WordPress REST API і містить набір універсальних функцій-запитів, що забезпечують зручну інтеграцію React-застосунку з бекендом на WordPress. Він охоплює як стандартні ресурси, такі як пости, сторінки, медіафайли та категорії, так і кастомні сутності, наприклад, «artist» або «custom-fields». Це дозволяє централізовано та ефективно отримувати необхідні дані для фронтенду.

Серед основних функцій, які виконує цей модуль, є запит категорії за її slug, отримання постів певної категорії, завантаження конкретної сторінки за slug, отримання користувацьких полів (групованих), список художників (у разі використання такого типу даних), а також зображення за його ID. Крім того, окремо реалізовано функцію для отримання загальної інформації про сайт — наприклад, його назви або опису — через API WordPress. Детальніше про цей сервіс можна дізнатись в додатку Е.

Також з боку фронтенду реалізовано мапу навігації до студії де власне і відбуватиметься заняття, зроблено це для того щоб користувач міг знайти шлях від свого дому до студії прямо на самому сайті, без переходу на інші джерела. Сама навігація релізована за допомогою API openrouteservice, на стороні фронтенду.

Також на фронтенд частині релізовано базову діджейську панель, яка релізована на базову JS. І містить в собі зміну BPM, та різні інші фільтри на звук треку.

З боку бекенду основна логіка реалізована у вигляді власних плагінів для WordPress. Центральним із них є плагін основної логіки сайту, що інтегрує платіжну систему LiqPay разом із механізмом блокування дат для бронювання та відправкою листів про запис на заняття. У його основі лежить клас LiqPay, узятий з офіційної документації API LiqPay. Основна функція цього класу — створення HTML-форми для оплати — дозволяє генерувати кнопку транзакції на

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

стороні клієнта з урахуванням суми, опису, валюти та ідентифікатора замовлення.

Допоміжні функції у плагіні відповідають за обробку інформації про платіж. Зокрема, функція `liqpay_payment_button` створює HTML-форму з усіма необхідними параметрами, `liqpay_payment_status` перевіряє статус оплати через API, а `parseRawPostData` обробляє необроблені дані, отримані з LiqPay, щоб перетворити їх у зручний асоціативний масив. Функція `check_liqpay_payment_and_add_transaction` перевіряє платіж і додає транзакцію в базу, якщо вона успішна, або логує помилки в разі невдачі.

Як було описано вище в системі присутній `PhpMailer`, який відправляє лист про запис на заняття на вказану при записі поштову скриньку, також в плагіні міститься `API Route`, який відповідає за відправку на бек та подальшу відправку адміну (або адмінам), повідомлення з контактної форми.

Також у плагіні реалізована логіка бронювання через функцію `LockSlot`, яка перевіряє, чи дата вже заблокована, і, якщо ні — додає її до списку заблокованих. Окремий файл відповідає за API-інтерфейс цього функціоналу, забезпечуючи доступ до бронювань через REST-запити. Крім того, у плагін вбудований окремий файл для відображення і налаштування всіх можливостей через меню адміністратора `WordPress`, що дозволяє керувати платежами, заблокованими датами й інформацією з інтерфейсу панелі управління. З центральними файлами логіки даного плагіну можна ознайомитись в додатку Н.

Висновки до другого розділу

В процесі проектування сайту студії діджеїнгу були проаналізовані усі варіанти використання програмного продукту відповідно до вимог програмного забезпечення. На основі отриманого результату була спроектована структура сайту. Проектування включало в себе моделювання структури проекту та взаємодії між їх складовими частинами.

Також було спроектовано базу даних, для збереження інформації.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Було описано алгоритм роботи з сайтом та описані послідовні пункти варіантів його використання. Також були наведені основні класи та методи які були реалізовані для роботи додатку.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ДОДАТКУ

3.1 Порядок встановлення та налаштування системи

Проект має архітектуру – «Decoupled Architecture», її принцип в розділення на фронтенд та бекенд частину які пов’язані між собою за допомогою API. Детальніше з принципом роботи даної архітектури можна ознайомитись на рисунку 3.1.

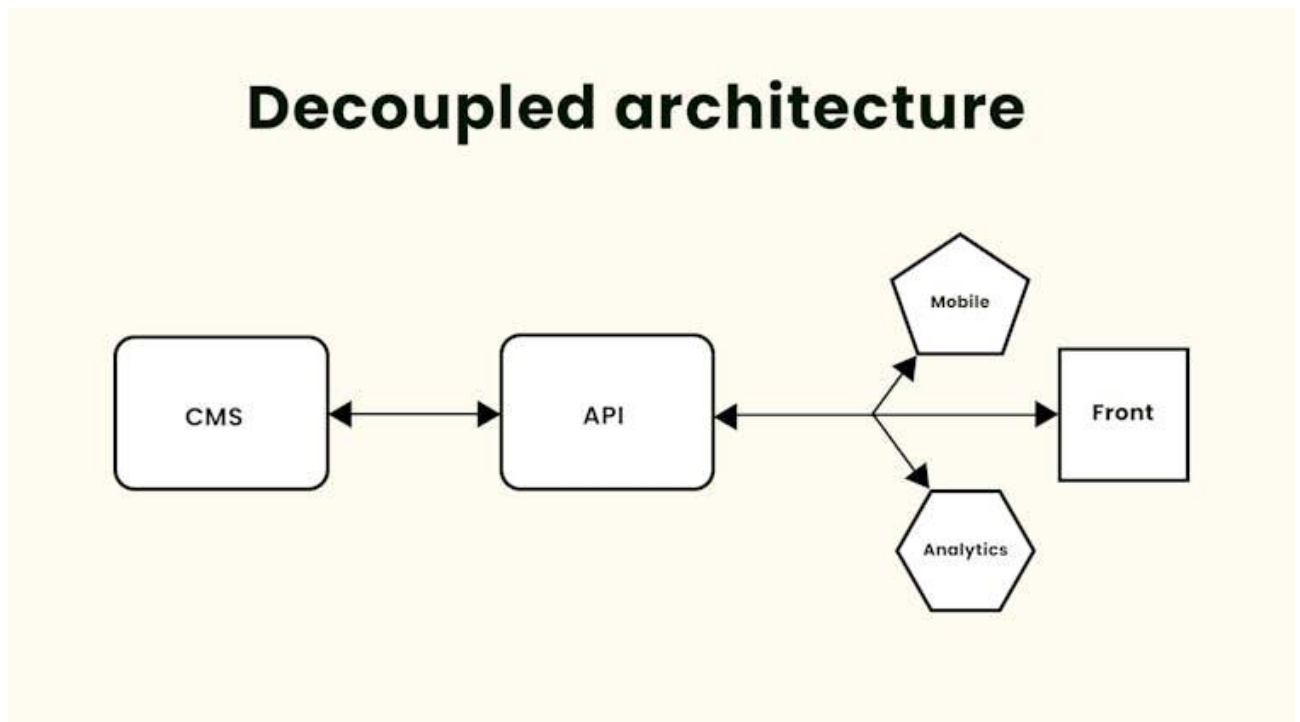


Рис. 3.1. Діаграма Decoupled Architecture

У цьому розділі описується порядок розгортання готового вебсайту, що складається з бекенду на WordPress (у вигляді REST API) та фронтенду на React, на хостинг-сервері. Процес починається з підготовки серверного середовища, яке має підтримувати PHP, MySQL і бути сумісним з WordPress. Зазвичай для цього використовуються популярні хостинг-провайдери з панеллю керування (наприклад, cPanel або ISPmanager).

Спочатку розгортається WordPress. Для цього завантажуються всі файли CMS, включаючи створені користувачем плагіни, в директорію сайту на хостингу через FTP або файловий менеджер у панелі керування. Після цього імпортується база даних з локального середовища через phpMyAdmin або

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк. 34
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		

аналогічний інструмент. У файлі `wp-config.php` вказуються коректні параметри підключення до БД хостингу: назва бази, логін, пароль та адреса сервера бази даних. Також переконуємось, що встановлено SSL-сертифікат, і WordPress працює через HTTPS. При потребі оновлюються посилання на сайт у базі даних (через SQL-запит або плагін Better Search Replace), щоб фронт та бекенд коректно взаємодіяли з новим доменом.

Далі переходять до фронтенду. У React-проєкті відкривається файл `.env` і змінюється змінна `VITE_BASE_URL` на адресу WordPress-сайту, що розгорнутий на хості. Потім застосунок компілюється командою `npm run build`, у результаті чого створюється папка `dist`, яка містить всі готові статичні файли сайту. Цю папку завантажують на хостинг – або в окрему директорію сайту, або в корінь, якщо це односторінковий застосунок, наприклад у випадку, коли фронт і бек розгорнуті на окремих субдоменах. У випадку, якщо фронтенд завантажується у WordPress-директорію, переконуються, що не конфліктують шляхи доступу до REST API.

Після розміщення обох частин проєкту перевіряється працездатність API, доступність React-застосунку, правильність зв'язку між клієнтом і сервером, а також робота платіжної системи LiqPay. Якщо все налаштовано вірно, сайт вважається повністю готовим до роботи в продуктивному середовищі.

3.2 Структура інтерфейсу та порядок взаємодії з веб застосунком

Продемонструємо інтерфейс системи на прикладі основного потоку використання, на початку ми попадаємо на стартову сторінку.

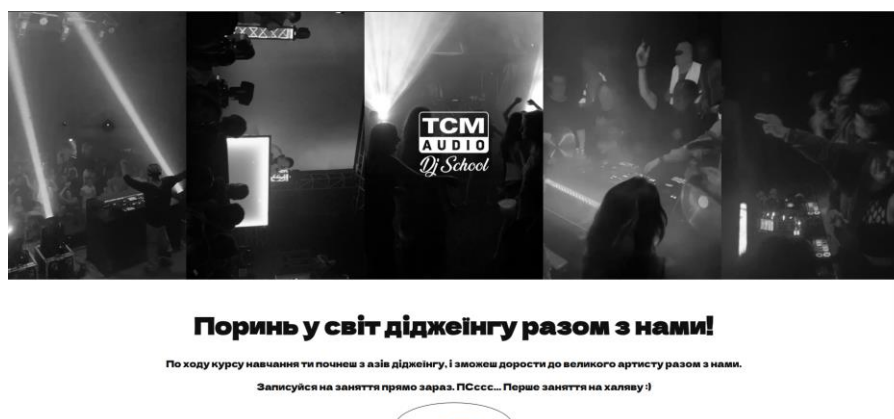


Рис. 3.2. Головна сторінка сайту

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

На ній розміщено кнопку для переходу на сторінку запису та панель з останніми новинами які видно на рисунку 3.3, панель про нас з кнопкою переходу на відповідну сторінку (рисунок 3.4), далі ми можемо бачити контактну форму (рисунок 3.5) після чого є кнопка яка перенесе нас на діджейську панель (рисунок 3.8).

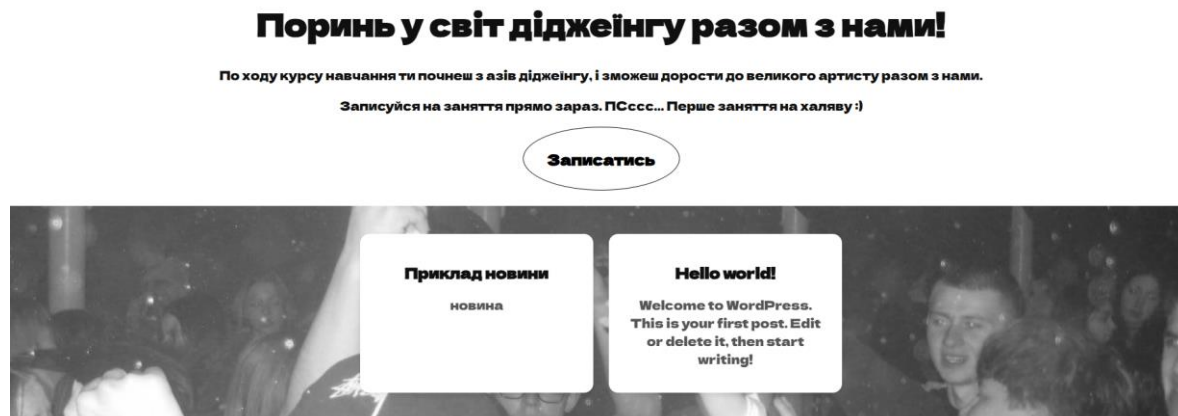


Рис. 3.3. Панель новин та запису на заняття

При натисканні кнопки записатись нас перенесе на відповідну сторінку з календарем для запису, яку буде розглянуто нижче, панель новин містить невелику картку з новинами, яка містить: заголовок та новину.

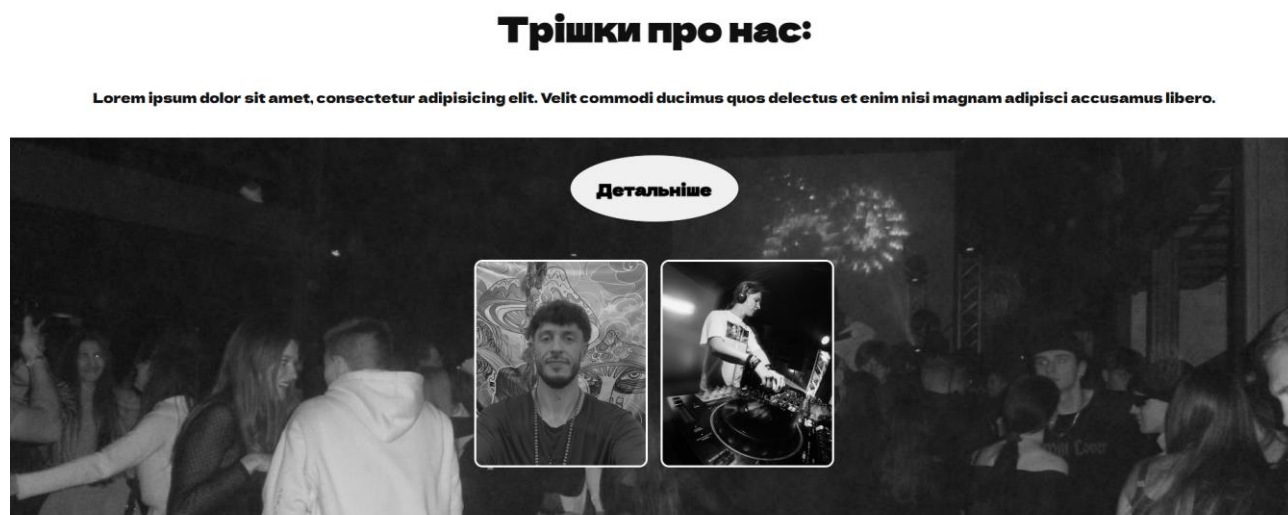


Рис. 3.4. Панель про нас з карточками керівників проекту

На рисунку видно базове поле інформації про нас, та панель на якій розміщенні основні учасники (керівники), проекту (школи), на цій панелі також

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

міститься кнопка для переходу до сторінки детальної інформації про керівників та основної відомості про проект.

Get in Touch

Name:

Email:

Message:

Send Message

Рис. 3.5. Форма для тримання контакту з керівниками проекту

Після заповнення та відправки форми користувачу виведеться повідомлення про успішну або не успішну відправку, яке можна побачити на рисунку 3.6.

Повідомлення надіслано!

Рис. 3.6. Текст інформації стосовно відправки повідомлення

Після того як користувач заповнив та відправив форму, адміну надійде повідомлення з відповідною інформацією, що можна побачити на рисунку 3.7.

Ім'я: 123123
Email: qw1@gmail.com
Повідомлення:
qwewqeqwe

Рис. 3.7. Лист відправлений з форми «Зв’жіться з нами»

Далі на сторінці розміщена вище описана кнопка для переходу на сторінку діджейської панелі, яку можна роздивитись на рисунку 3.8.



Рис. 3.8. Кнопка переходу до джіджейського пульта

Перейдемо до сторінки запису, на якій розташований календарь з датами (рисунок 3.10) червоні дати – це зайняті дати, при натисканні на них виведе відповідну помилку, яку видно на рисунку 3.9.



Рис. 3.9. Повідомлення про те що обрана дата зайнята

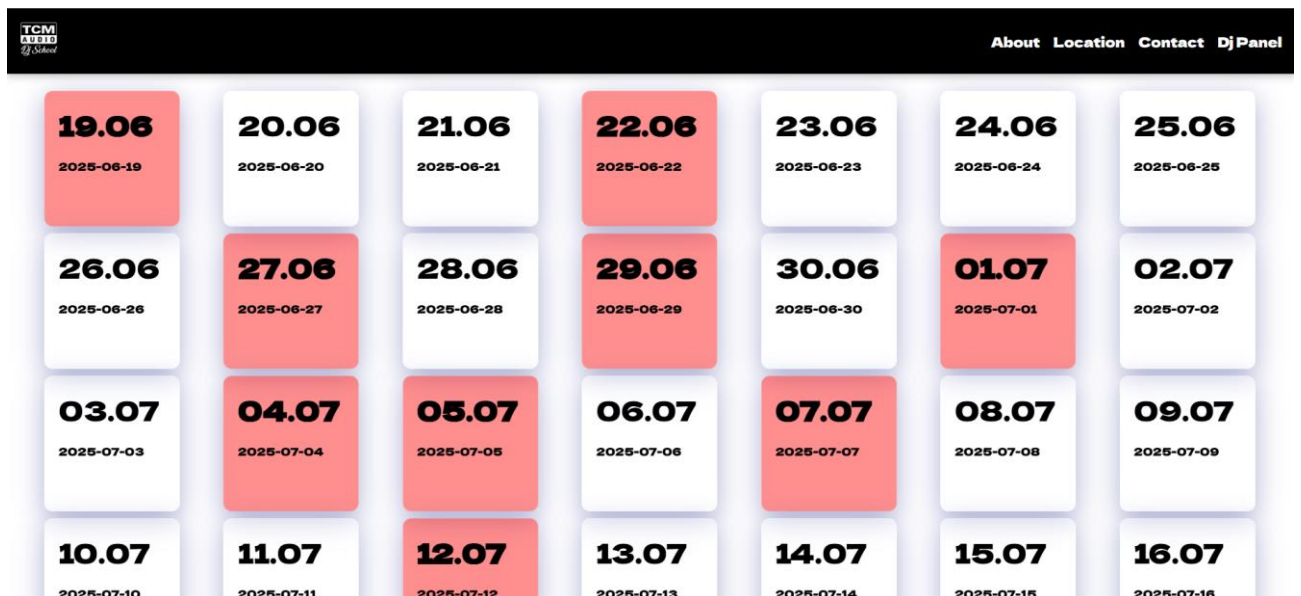


Рис. 3.10. Сторінка календаря для запису

Також на сторінці реалізовано декілька систем безпека, якщо паралельно на сайті записуються однакові користувачі, реалізовано ряд перевірок як на стороні бекенду так і фронтенду, при натисканні на кнопку дати, сайт запитає у сервера додаткову перевірку чи дата не зайнята що буде запобігати подвійному натисканню від двох користувачів.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

Оплатити через LiqPay

First Name: Аліса

Last Name: Цігирьова

Patronymic: Антонівна

Phone Number: +380977777777

Email: qwe123@gmail.com

Сплатити

Рис. 3.11. Модальне вікно для заповнення інформації про людину яка записується

Зверху на рисунку 3.11. видно модальне вікно для заповнення інформації про людину яка записується, в ньому також реалізована певна система безпеки яка при натисканні на кнопку оплати, відразу закидає потрібні дані на сервер, який вже автономно очікує оплату від користувача, це створено для запобігання того що користувач випадково закриє вікно сайту, і його оплата не додасться в базу, тому що саме оплата LiqPay, в нашому випадку, відкривається в новому вікні. Оплату видно на рисунку 3.12.

QR-код для оплати

LIQPAY

Дані про оплату

Оплата за клас

До сплати: **100.00 UAH**

Використовуйте Privat24

тестовий режим

24 Pay

G Pay | VISA •••• 0365

або

Картка | Приват24 | Рахунок

Рис. 3.12. Оплата LiqPay

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Після оплати користувачу надійде лист на поштову скриньку вказану в формі про успішну оплату та новий запис.

Новий запис!

Ім'я: 123312

Прізвище: 123123321

По батькові: 12321231

Телефон: +380977983333

Рис. 3.13. Лист який надходить людині яка записалась на заняття

На решті сторінко крім головної, розміщений інших хедер, в якому є посилання на всі сторінки сайту, це можна побачити на рисунку 3.14.



Рис. 3.14. Альтернативний хедер

Перемістимось на сторінку «про нас», де розміщена повна інформація про нас та карточки керівників проекту, на карточках розміщено ім'я людини та його соціальні мережі, всю інформацію на цій сторінці можна редагувати на адмін сторінці, саму сторінку «про нас» показано на рисунку 3.15.



Приклад тексту на сторінці про нас, він редагується за допомогою адмін панелі



Рис. 3.15. Сторінка «про нас»

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

Далі розглянемо сторінку локації показану на рисунку 3.16. Основним її компонентом є мапа з навігацією на якій розміщено студію, на ній користувач може вказати потрібну йому точку та побачити маршрут.

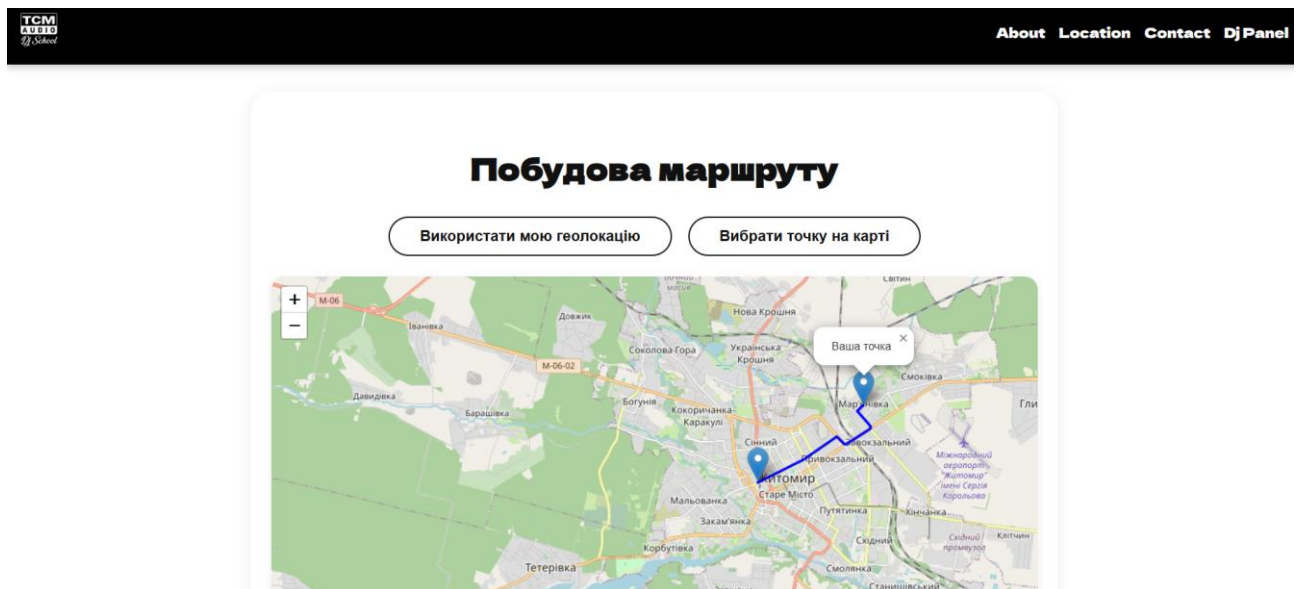


Рис. 3.16. Сторінка з мапою

Сторінка «Contact» - містить ту саму форму для зв'язку з керівниками проєкту, вона перевикористовує компонент форми на окремій сторінці, саму сторінку можна переглянути на рисунку 3.17.

Рис. 3.17. Сторінка «Contact»

Далі перейдемо до детального розгляду сторінки з діджейським пультом, а точніше – його базовим макетом, вигляд сторінки – рисунок 3.18. На цій сторінці користувач може вибрати будь який аудіофайл на своєму ПК та спробувати використати базові функції стандартного діджейського пульта.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

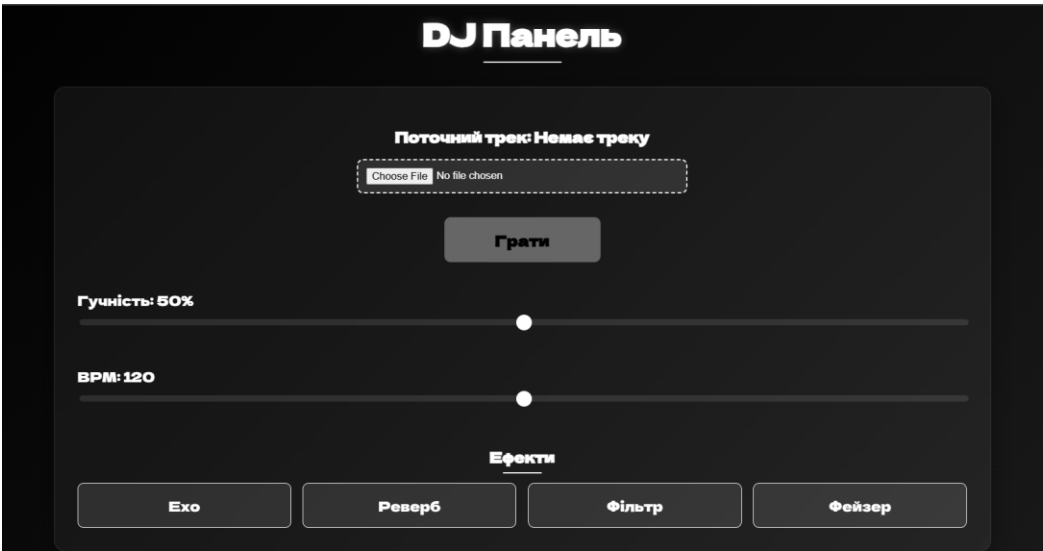


Рис. 3.18. Сторінка з діджейським пультом

Далі розглянемо бекенд частину веб ресурсу, а саме адмін панель яка реалізована на Wordpress, за допомогою неї можна редагувати майже всю інформацію показану в вище описаній частині сайту.

Редагування, додавання та видалення новин, показано на рисунках 3.19; 3.20.

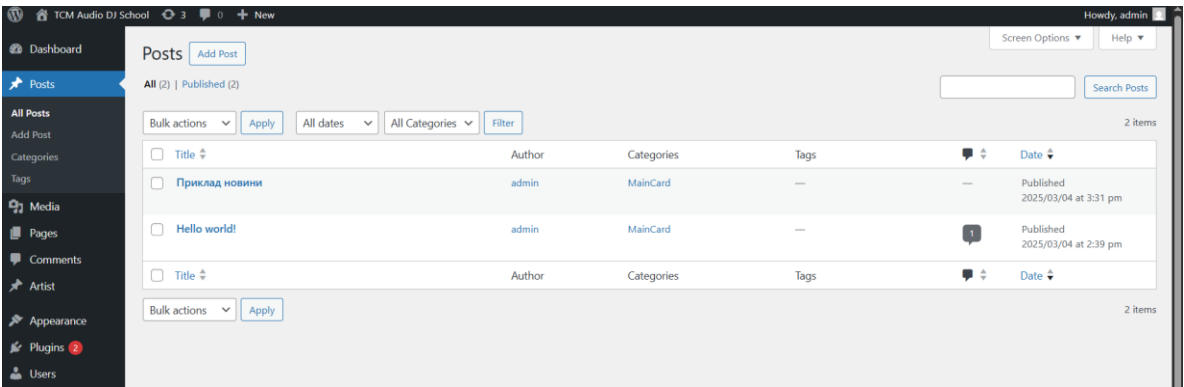


Рис. 3.19. Перегляд всіх постів

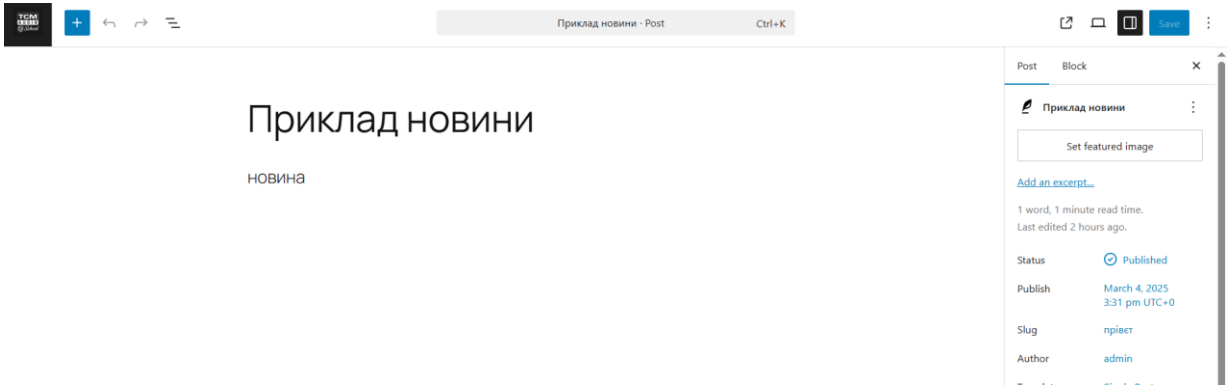


Рис. 3.20. Сторінка редагування поста

Далі в адмін панелі створено окремих розділ з артистам (рисунк 3.21), на якому можна виконувати операції редагування, додавання та видалення, також «артисти» мають додаткові катомні поля по типу їх зображення (фото, картинка, арт тощо), їх ім'я, соціальні мережі та трішки інформації про них та їх діяльність, всі ці дані також можна редагувати, детально роздивитись цей фрагмент на сайті можна буде на рисунку 3.22.

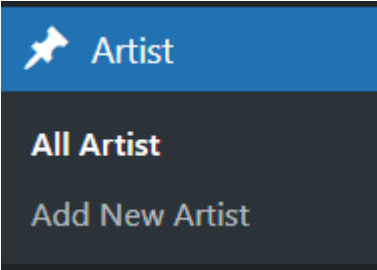


Рис. 3.21. Окрема вкладка «артистів»

Artist Name

Artist Bio

Social

Instagram

Telegram

X/Twitter

Facebook

Soundcloud

Рис. 3.22.

Також в адмін панелі реалізовано додавання власних полів (рисунок 3.24), для різних типів задач, наприклад на сайті присутня реалізація додавання соціальних мереж, які в результаті відображаються в футері фронтенд частини сайту (рисунок 3.23).



Рис. 3.23.

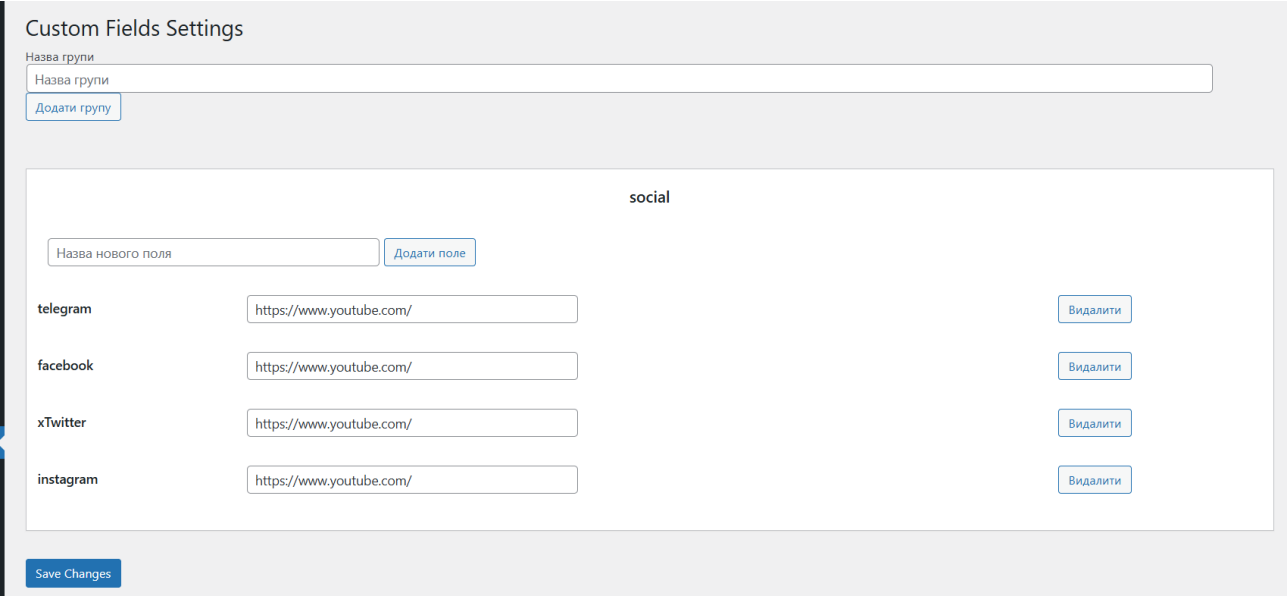


Рис. 3.24. Сторінка створення власних полів

Також в адмін сторінці можна переглянути список майбутніх занять на які записались користувачі, цей список відсортований за зростанням дат, від найближчої, також дати які вже пройшли не будуть відображатись в списку дат, переглянути вигляд сторінки можна на рисунку 3.25.

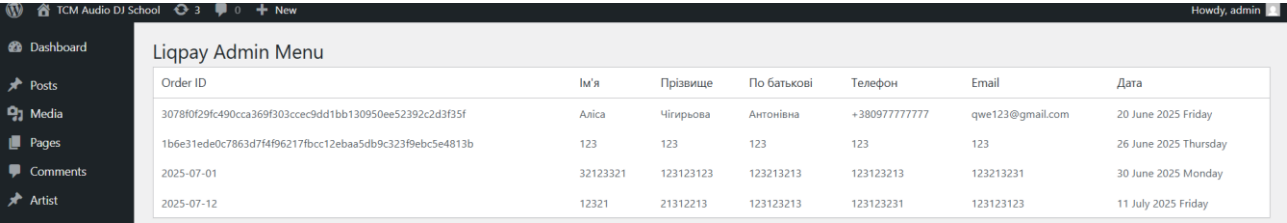


Рис. 3.25. Сторінка перегляду записів

Висновки до третього розділу

У третьому розділі було розглянуто порядок встановлення та налаштування вебзастосунку, що складається з клієнтської частини, розробленої за допомогою React, та серверної частини, реалізованої як REST API на базі CMS

WordPress з власними плагінами. Описано особливості взаємодії користувача з інтерфейсом сайту, включаючи вибір дати заняття через інтерактивний календар, процес бронювання, оплату через платіжну систему LiqPay та перегляд інформаційних сторінок.

Було детально описано модулі клієнтської частини, відповідальні за запити до API, обробку платежів та відображення динамічного контенту. Також висвітлено функціональність адміністративної панелі, яка дозволяє керувати викладачами, новинами та контактною інформацією. Проведене тестування підтвердило правильну взаємодію між компонентами фронтенду та бекенду, стабільність процесу бронювання і оплати, а також відповідність функціоналу вимогам користувачів. Це дозволяє зробити висновок, що система є технічно стабільною, зручною для використання та готовою до розгортання і експлуатації.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

Висновки

У ході виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано вебзастосунок для навчальної DJ-школи, який складається з клієнтської частини, розробленої за допомогою React, та серверної частини, реалізованої у вигляді REST API на базі CMS WordPress із власними плагінами. Було проведено аналіз вимог до майбутнього продукту, вивчено сучасні підходи до реалізації інтерактивних функцій бронювання та онлайн-оплати, що дозволило сформулювати чіткі функціональні та нефункціональні вимоги до системи.

У процесі розробки було створено архітектуру взаємодії між фронтендом та бекендом, реалізовано обробку динамічних даних за допомогою API, а також забезпечено інтеграцію з платіжною системою LiqPay для онлайн-оплати занять. Особливу увагу приділено системі бронювання, яка забезпечує тимчасове блокування вибраної дати на 10 хвилин до моменту підтвердження оплати, з урахуванням конкурентного доступу до одного й того самого слоту кількома користувачами.

Було реалізовано зручну адміністративну панель для керування контентом сайту, включаючи додавання та редагування новин, інформації про викладачів і контактів. Завдяки цьому адміністратор має повний контроль над інформаційним наповненням ресурсу без необхідності втручання у програмний код.

У рамках роботи описано логіку користувацької взаємодії: від вибору дати заняття, перегляду сторінок про школу та контактів викладачів — до здійснення платежу. Особливу роль відіграє календар з інтерактивною можливістю бронювання, який є ключовим елементом з точки зору зручності користувача.

Було проведено тестування функціональності застосунку, зокрема коректність бронювання, обробку платежів, роботу з API, а також відображення контенту з бази даних WordPress. Система показала стабільну та надійну роботу у типовому середовищі розгортання, відповідає очікуванням користувачів і технічним вимогам.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

Результатом є повноцінний вебзастосунок з інтуїтивно зрозумілим інтерфейсом, ефективною взаємодією з бекендом, підтримкою онлайн-оплати та зручною адмін-панеллю. Розроблений продукт демонструє принципи побудови сучасних інтегрованих вебрішень і може бути легко масштабований або доповнений новим функціоналом у майбутньому.

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. WordPress REST API Handbook [Електронний ресурс]. – Режим доступу: <https://developer.wordpress.org/rest-api/>
2. WordPress Plugin Developer Handbook [Електронний ресурс]. – Режим доступу: <https://developer.wordpress.org/plugins/>
3. LiqPay – API Documentation [Електронний ресурс]. – Режим доступу: <https://www.liqpay.ua/documentation/en>
4. React Documentation [Електронний ресурс]. – Режим доступу: <https://react.dev/>
5. React Router – Declarative Routing for React [Електронний ресурс]. – Режим доступу: <https://reactrouter.com/>
6. Create React App Documentation [Електронний ресурс]. – Режим доступу: <https://create-react-app.dev/>
7. Axios – Promise based HTTP client [Електронний ресурс]. – Режим доступу: <https://axios-http.com/>
8. PHP Manual [Електронний ресурс]. – Режим доступу: <https://www.php.net/manual/en/>
9. JavaScript MDN Documentation [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
10. HTML Living Standard [Електронний ресурс]. – Режим доступу: <https://html.spec.whatwg.org/>
11. CSS: Cascading Style Sheets – MDN Web Docs [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/CSS>
12. MySQL 8.0 Reference Manual [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/>
13. Gutenberg – Block Editor Handbook [Електронний ресурс]. – Режим доступу: <https://developer.wordpress.org/block-editor/>
14. REST API Concepts and Examples [Електронний ресурс]. – Режим доступу: <https://restfulapi.net/>

15. Duckett, J. JavaScript and jQuery: Interactive Front-End Web Development. – Wiley, 2014. – 640 p.
16. Freeman, E., Freeman, E. Head First HTML and CSS. – O'Reilly Media, 2012. – 768 p.
17. Flanagan, D. JavaScript: The Definitive Guide. – 7th ed. – O'Reilly Media, 2020. – 706 p.
18. Keith, J. DOM Scripting: Web Design with JavaScript and the Document Object Model. – Friends of ED, 2010. – 432 p.
19. Banks, A. Learning React. – 3rd ed. – O'Reilly Media, 2022. – 464 p.
20. Wieruch, R. The Road to React. – 2023. – Self-published. – 250 p.
21. Krasner, G. Modern PHP: New Features and Good Practices. – O'Reilly Media, 2015. – 270 p.
22. Gackenhimer, C. Introduction to Web Development using HTML 5. – Cengage Learning, 2020. – 480 p.
23. Smashing Magazine – Articles on UX and Web Design [Електронний ресурс]. – Режим доступу: <https://www.smashingmagazine.com/>
24. CSS-Tricks – Tips, Tricks, and Techniques on using CSS [Електронний ресурс]. – Режим доступу: <https://css-tricks.com/>
25. GitHub – Git Documentation [Електронний ресурс]. – Режим доступу: <https://git-scm.com/doc>
26. WebDevSimplified – React, JavaScript and Frontend Tutorials [Електронний ресурс]. – Режим доступу: <https://www.webdevsimplified.com/>
27. Netlify Docs – Deploy modern web projects [Електронний ресурс]. – Режим доступу: <https://docs.netlify.com/>
28. Vercel Docs – Deploy React and Next.js Apps [Електронний ресурс]. – Режим доступу: <https://vercel.com/docs>
29. Cloudflare Docs – DNS, CDN, and Page Rules [Електронний ресурс]. – Режим доступу: <https://developers.cloudflare.com/>
30. kulturazvuka [Електронний ресурс]. – Режим доступу: <https://kulturazvuka.ua/ua/>

		Горошко Д.Ю.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Власенко О.В.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

Додатки

```

const BASE_URL = import.meta.env.VITE_BASE_URL;

export const fetchLiqPayForm = async (params) => {
  try {
    const queryString = new URLSearchParams(params).toString();
    const response = await fetch(`${BASE_URL}/liqpay/v1/generate-
button?${queryString}`);
    const data = await response.json();
    return data.form;
  } catch (error) {
    console.error('Помилка при отриманні форми LiqPay:', error);
    return null;
  }
};

export const checkPaymentStatus = async (orderId) => {
  try {
    const response = await fetch(`${BASE_URL}/liqpay/v1/payment-
status?order_id=${orderId}`);
    const data = await response.json();
    return data.status
  } catch (error) {
    console.error('Помилка при перевірці статусу платежу:', error);
    return 'error';
  }
};

export const sendTransactionData = async (data) => {
  try {
    const serializedData = {};
    for (const [key, value] of Object.entries(data)) {
      serializedData[key] = typeof value === 'object' ? JSON.stringify(value)
: String(value);
    }

    const response = await
fetch(`${import.meta.env.VITE_BASE_URL}/liqpay/v1/add-transaction`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded',
      },
      body: new URLSearchParams(serializedData).toString(),
    });

    const responseData = await response.text();

    if (!response.ok) {
      throw new Error(`Не вдалося відправити дані. Код: ${response.status}`);
    }
  }
};

```

```

        return JSON.parse(responseData);
    } catch (error) {
        console.error('Помилка при відправці даних:', error);
        throw error;
    }
};

export const isDatePicked = async (date) => {
    try {
        const response = await
fetch(`${BASE_URL}/liqpay/v1/locked?day=${encodeURIComponent(date)}`);
        const result = await response.json();

        if (response.ok) {
            return result.locked;
        } else {
            throw new Error(result.message || 'Не вдалося перевірити статус
дати.');
```

Додаток В

```

import { useState, useEffect } from 'react';
import { fetchLiqPayForm } from '../services/liqpayService';

const useLiqPayForm = (order_id, amount, currency, description) => {
    const [liqPayForm, setLiqPayForm] = useState('');

    useEffect(() => {
        const loadForm = async () => {
            const form = await fetchLiqPayForm({
                order_id: order_id?.iso || order_id,
                amount,
                currency,
                description,
            });
            setLiqPayForm(form);
        };

        loadForm();
    }, [order_id, amount, currency, description]);

    return liqPayForm;
};
```

```
export default useLiqPayForm;
```

```
import { useState, useEffect } from "react";
import { fetchPageBySlug } from "../utils/Api";

const usePages = (slug) => {
  const [pages, setPages] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      const pagesData = await fetchPageBySlug(slug);
      setPages(pagesData[0]);
    };

    fetchData();
  }, [slug]);

  return pages;
};

export default usePages;
```

```
import { useState, useEffect } from "react";

import { generateDates } from "../utils/GenDays";
import { checkDateAvailability } from "../utils/CheckDateAvailability";

export function useCalendarDates() {
  const [dates, setDates] = useState([]);

  useEffect(() => {
    const fetchDates = async () => {
      const generatedDates = await generateDates();

      const updatedDates = await Promise.all(
        generatedDates.map(async (date) => {
          const isUnavailable = await checkDateAvailability(date.iso);
          return { ...date, locked: isUnavailable };
        })
      );

      setDates(updatedDates);
    };

    fetchDates();
  }, []);
```

```

    return dates;
}

```

```

import { useState, useEffect } from "react";
import { fetchArtists, fetchImageByID } from "../utils/Api";

const useArtists = () => {
  const [artists, setArtists] = useState([]);
  const [artistImages, setArtistImages] = useState({});

  useEffect(() => {
    const fetchData = async () => {
      const artistsData = await fetchArtists();
      setArtists(artistsData);
    };

    fetchData();
  }, []);

  useEffect(() => {
    const fetchImages = async () => {
      const imagesMap = {};
      for (const artist of artists) {
        if (artist.acf?.image) {
          const imageUrl = await fetchImageByID(artist.acf.image);
          imagesMap[artist.id] = imageUrl;
        }
      }
      setArtistImages(imagesMap);
    };

    if (artists.length > 0) {
      fetchImages();
    }
  }, [artists]);

  return { artists, artistImages };
};

export default useArtists;

```

```

import { useEffect } from 'react';

const useButtonClick = (buttonId, onClick) => {
  useEffect(() => {
    const button = document.getElementById(buttonId);

```

```

    if (button) {
      button.addEventListener('click', onClick);
    }

    return () => {
      if (button) {
        button.removeEventListener('click', onClick);
      }
    };
  }, [buttonId, onClick]);
};

export default useButtonClick;

import { useState } from 'react';
import { sendContactMessage } from '../services/contactService';

export default function useContactForm() {
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState('');
  const [success, setSuccess] = useState('');

  const submitContact = async (formData) => {
    setLoading(true);
    setError('');
    setSuccess('');
    try {
      const res = await sendContactMessage(formData);
      setSuccess(res.message || 'Повідомлення надіслано!');
    } catch (err) {
      setError(err.message || 'Сталася помилка');
    } finally {
      setLoading(false);
    }
  };

  return { submitContact, loading, error, success };
}

import { useState, useEffect } from "react";
import { fetchFieldsByGroup } from "../utils/Api";

const useFieldsGroup = (title) => {
  const [fields, setFields] = useState([]);

  useEffect(() => {
    const fetchData = async () => {

```

```

        const fieldsData = await fetchFieldsByGroup(title);
        setFields(fieldsData);
    };

    fetchData();
}, [title]);

return fields;
};

export default useFieldsGroup;

```

```

import { useState, useEffect } from "react";
import { fetchPostsByCategorySlug } from "../utils/Api";

const usePosts = (slug) => {
    const [posts, setPosts] = useState([]);

    useEffect(() => {
        const fetchData = async () => {
            const postsData = await fetchPostsByCategorySlug(slug);
            setPosts(postsData);
        };

        fetchData();
    }, [slug]);

    return posts;
};

export default usePosts;

```

Додаток С

```

import Header from '../partials/Header';
import Calendar from '../partials/Calendar/Calendar';
import '../assets/scss/calendar.scss';

function CalendarPage() {
    return (
        <>
            <Header />
            <Calendar/>
        </>
    );
}

export default CalendarPage;

```

```

import React, { useState } from 'react';
import CalendarDay from '../CalendarDay';
import LiqPayForm from '../forms/LiqPayForm';
import { useCalendarDates } from '../../hooks/useCalendarDates';
import { lockDate } from '../../services/bookingService';
import { checkDateAvailability } from '../../utils/CheckDateAvailability';

function Calendar() {
  const dates = useCalendarDates();
  const [selectedDate, setSelectedDate] = useState(null);
  const [isModalOpen, setIsModalOpen] = useState(false);
  const [orderId, setOrderId] = useState(null);

  const generateOrderId = () => {
    return crypto.subtle.digest('SHA-256', new
TextEncoder().encode(Date.now().toString()))
      .then(hashBuffer => {
        const hashArray = Array.from(new Uint8Array(hashBuffer));
        const fullHash = hashArray.map(b => b.toString(16).padStart(2,
'0')).join('');
        return fullHash.slice(0, 55);
      });
  };

  const handleDayClick = async (date) => {
    const isUnavailable = await checkDateAvailability(date.iso);
    if (isUnavailable) {
      alert("Ця дата вже зайнята!");
      setTimeout(() => {
        window.location.reload();
      }, 500);
      return;
    }

    const newOrderId = await generateOrderId();
    setOrderId(newOrderId);
    setSelectedDate(date);
    setIsModalOpen(true);
    lockDate(date.iso);
  };

  return (
    <>
      <div className="calendar">
        <ul>
          {dates.length > 0 ? (
            dates.map((date) => (

```

```

                                <CalendarDay key={date.iso} date={date}
onClick={handleDayClick} />
                                ))
                                ) : (
                                <p>Завантаження...</p>
                                )}
                                </ul>
                                </div>

                                {isModalOpen && selectedDate && orderId && (
                                <LiqPayForm selectedDate={selectedDate} id={orderId} closeModal={()
=> setIsModalOpen(false)} />
                                )}
                                </>
                                );
                                }

export default Calendar;

```

Додаток Е

```

import { useEffect, useState } from "react";
import { fetchSiteMetaData } from "../../utils/Api";

const MetaData = () => {
  const [siteName, setSiteName] = useState("");

  useEffect(() => {
    const fetchMetaData = async () => {
      const { siteName } = await fetchSiteMetaData();
      setSiteName(siteName);
      document.title = siteName;
    };

    fetchMetaData();
  }, []);

  return <></>;
};

export default MetaData;

```

Додаток Н

```

<?php
/*
Plugin Name: LiqPay Payment
Description: Генерація кнопки LiqPay для оплати та перевірки статусу платежу.
Version: 1.2

```

Author: Горошко Дмитрій

**/*

```
define('MAILER_HOST', '');
define('MAILER_SMTPAuth', true);
define('MAILER_EMAIL', '');
define('MAILER_KEY', '');
define('MAILER_SMTPSecure', '');
define('MAILER_PORT', );

define('LIQPAY_PUBLIC_KEY', '');
define('LIQPAY_PRIVATE_KEY', '');
define('DB_LIQPAY_TABLE_NAME', 'liqpay_data');
define('DB_BOOKING_TABLE_NAME', 'booking_locks');

date_default_timezone_set('Europe/Kiev');

if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

require_once plugin_dir_path(__FILE__) . 'admin/liqpay-admin-page.php';

require_once plugin_dir_path(__FILE__) . 'lib/LiqPay.php';
require_once plugin_dir_path(__FILE__) . 'lib/PHPMailer/src/PHPMailer.php';
require_once plugin_dir_path(__FILE__) . 'lib/PHPMailer/src/Exception.php';
require_once plugin_dir_path(__FILE__) . 'lib/PHPMailer/src/SMTP.php';
require_once plugin_dir_path(__FILE__) . 'includes/liqpay-functions.php';
require_once plugin_dir_path(__FILE__) . 'includes/booking-api.php';
require_once plugin_dir_path(__FILE__) . 'includes/liqpay-api.php';
require_once plugin_dir_path(__FILE__) . 'includes/liqpay-db.php';
require_once plugin_dir_path(__FILE__) . 'includes/booking-db.php';
require_once plugin_dir_path(__FILE__) . 'includes/mailer-functions.php';
require_once plugin_dir_path(__FILE__) . 'includes/mailer-api.php';

add_action( 'rest_api_init', 'register_liqpay_payment_api' );
add_action( 'rest_api_init', 'register_booking_api' );
add_action('wp_ajax_check_liqpay_payment_and_add_transaction',
'check_liqpay_payment_and_add_transaction');
add_action('wp_ajax_nopriv_check_liqpay_payment_and_add_transaction',
'check_liqpay_payment_and_add_transaction');
add_action('rest_api_init', 'register_mailer_api');

add_action('wp_scheduled_delete_locks', function() {
    global $wpdb;
    $wpdb->query("DELETE FROM DB_BOOKING_TABLE_NAME WHERE locked_until < NOW()");
});
if (!wp_next_scheduled('wp_scheduled_delete_locks')) {
    wp_schedule_event(time(), 'hourly', 'wp_scheduled_delete_locks');
}
```

```
new LiqpayAdmin();
?>
```

```
<?php
if (!defined('ABSPATH')) {
    exit;
}

require_once plugin_dir_path(__FILE__) . '/mailer-functions.php';

const CONTACT_RECEIVER_EMAIL = '';

function mailer_api_send_message($request) {
    $params = $request->get_json_params();
    $name = sanitize_text_field($params['name'] ?? '');
    $email = sanitize_email($params['email'] ?? '');
    $message = sanitize_textarea_field($params['message'] ?? '');

    if (empty($name) || empty($email) || empty($message)) {
        return new WP_Error('missing_fields', 'Всі поля обов'язкові.',
array('status' => 400));
    }
    if (!is_email($email)) {
        return new WP_Error('invalid_email', 'Некоректний email.', array('status'
=> 400));
    }

    $subject = 'Нове повідомлення з контактної форми';
    $body = "Ім'я: $name<br>Email: $email<br>Повідомлення:<br>$message";

    Mailer::SendEmail(CONTACT_RECEIVER_EMAIL, $subject, $body);

    return array('success' => true, 'message' => 'Повідомлення надіслано!');
}

function register_mailer_api() {
    register_rest_route('mailer/v1', '/send', array(
        'methods' => 'POST',
        'callback' => 'mailer_api_send_message',
        'permission_callback' => '__return_true',
    ));
}

<?php

use PHPMailer\PHPMailer\PHPMailer;
```

```

class Mailer
{
    private static $mail;

    public static function init()
    {
        self::$mail = new PHPMailer(true);
        self::$mail->isSMTP();
        self::$mail->Host = MAILER_HOST;
        self::$mail->SMTPAuth = MAILER_SMTPAuth;
        self::$mail->Username = MAILER_EMAIL;
        self::$mail->Password = MAILER_KEY;
        self::$mail->SMTPSecure = MAILER_SMTPSecure;
        self::$mail->Port = MAILER_PORT;
    }

    public static function SendEmail($email, $subject, $message)
    {
        try {
            self::init();

            self::$mail->setFrom(self::$mail->Username, 'Запис на послугу');
            self::$mail->addAddress($email);
            self::$mail->isHTML(true);
            self::$mail->Subject = $subject;
            self::$mail->Body = $message;

            self::$mail->send();
            error_log("Лист успішно надіслано до $email");
        } catch (Exception $e) {
            error_log("Помилка надсилання листа: {$e->getMessage()}");
        }
    }
}

```

<?php

```

function GetLiqPayOrderByDate($date){
    global $wpdb;

    $tablename = DB_LIQPAY_TABLE_NAME;

    $sql = "SELECT * FROM $tablename WHERE day = '$date'";
    $existing_lock = $wpdb->get_row($sql);

    if($existing_lock){
        return true;
    }
}

```

```

        return false;
    }

function GetLiqPayTransaction()
{
    global $wpdb;

    $tablename = DB_LIQPAY_TABLE_NAME;
    $results = $wpdb->get_results("SELECT * FROM $tablename WHERE day > CURDATE()
ORDER BY day ASC", ARRAY_A);

    return $results;
}

function AddLiqPayTransaction($order_id, $first_name, $last_name, $patronymic,
$phone_number, $email, $day)
{
    global $wpdb;
    $tablename = DB_LIQPAY_TABLE_NAME;

    $data = array(
        'order_id' => $order_id,
        'first_name' => $first_name,
        'last_name' => $last_name,
        'patronymic' => $patronymic,
        'phone_number' => $phone_number,
        'email' => $email,
        'day' => $day
    );

    $format = array('%s', '%s', '%s', '%s', '%s', '%s', '%s');

    $result = $wpdb->insert($tablename, $data, $format);

    if ($result === false) {
        error_log('Помилка вставки в БД: ' . $wpdb->last_error);
        return false;
    }

    $subject = "We are waiting for you: $day";
    $message = "
        <h3>Новий запис!</h3>
        <p><strong>Ім'я:</strong> $first_name</p>
        <p><strong>Прізвище:</strong> $last_name</p>
        <p><strong>По батькові:</strong> $patronymic</p>
        <p><strong>Телефон:</strong> $phone_number</p>
        <p><strong>Email:</strong> $email</p>
        <p><strong>Дата запису:</strong> $day</p>
        <p><strong>Order ID:</strong> $order_id</p>
    ";

```

```

        Mailer::SendEmail($email, $subject, $message);

        error_log("Успішно додано транзакцію order_id = $order_id");
        return true;
    }

<?php

if (!defined('ABSPATH')) {
    exit;
}

function liqpay_payment_button($order_id, $amount, $currency, $description)
{
    $liqpay = new LiqPay(LIQPAY_PUBLIC_KEY, LIQPAY_PRIVATE_KEY);

    return $liqpay->cnb_form(array(
        'action' => 'pay',
        'version' => '3',
        'amount' => $amount,
        'currency' => $currency,
        'description' => $description,
        'order_id' => $order_id,
    ));
}

function liqpay_payment_status($order_id)
{
    $liqpay = new LiqPay(LIQPAY_PUBLIC_KEY, LIQPAY_PRIVATE_KEY);

    error_log("Перевірка статусу платежу для order_id: " . $order_id);

    $response = $liqpay->api("request", array(
        'action' => 'status',
        'version' => '3',
        'order_id' => $order_id
    ));

    return $response;
}

function parseRawPostData($rawData) {
    $result = [];

    $start = strpos($rawData, '{') + 1;
    $end = strrpos($rawData, '}');
    if ($start === false || $end === false || $start >= $end) {
        return [];
    }
}

```

```

    }

    $body = substr($rawData, $start, $end - $start);

    $parts = explode(';', $body);

    $key = null;
    foreach ($parts as $part) {
        $part = trim($part);

        if (strpos($part, 's:') === 0) {
            $quoteStart = strpos($part, '"');
            $quoteEnd = strrpos($part, '"');

            if ($quoteStart === false || $quoteEnd === false || $quoteStart ==
$quoteEnd) {
                continue;
            }

            $value = substr($part, $quoteStart + 1, $quoteEnd - $quoteStart - 1);
            $value = stripslashes($value);

            if ($key === null) {
                $key = $value;
            } else {
                $result[$key] = $value;
                $key = null;
            }
        }
    }

    foreach ($result as $k => $v) {
    }

    return $result;
}

function check_liqpay_payment_and_add_transaction()
{
    set_time_limit(0);

    if (isset($_POST['data']) && !empty($_POST['data'])) {
        $result = parseRawPostData($_POST['data']);
        if (is_object($result)) $result = (array)$result;

        if (!isset($result['order_id'])) {
            error_log('Missing order_id in received data');
            wp_die();
        }

        $order_id = $result['order_id'];
    }
}

```

```

$start_time = time();
$max_wait_time = 600;
$check_interval = 20;

while ((time() - $start_time) < $max_wait_time) {
    $response = liqpay_payment_status($order_id);
    error_log('Поточний статус платежу: ' . print_r($response, true));

    if ($response === false) {
        error_log("Failed to fetch payment status for order ID:
$order_id");
        break;
    }

    if ((is_array($response) && isset($response['status']) &&
$response['status'] == 'success') ||
        (is_object($response) && isset($response->status) && $response-
>status == 'success')) {
        AddLiqPayTransaction(
            $result['order_id'],
            $result['first_name'],
            $result['last_name'],
            $result['patronymic'],
            $result['phone_number'],
            $result['email'],
            $result['day']
        );
        break;
    }

    sleep($check_interval);
}

wp_die();
} else {
    error_log('No data received or data is empty');
    wp_die();
}
}

function LockSlot($date)
{
    $date = date('Y-m-d', strtotime($date));

    if (GetLockedSlotByDate($date)) {
        return false;
    }

    AddLockedSlot($date);
}

```

```

        return true;
    }

?>

<?php

if (!defined('ABSPATH')) {
    exit;
}

function add_liqpay_transaction(WP_REST_Request $request)
{
    $params = array_map('sanitize_text_field', $request->get_params());

    if (empty($params['order_id']) || empty($params['first_name']) ||
        empty($params['last_name']) || empty($params['patronymic']) ||
        empty($params['phone_number']) || empty($params['email']) || empty($params['day']))
    {
        return new WP_REST_Response(array('error' => 'Всі поля є обов'язковими'),
400);
    }

    error_log('Дата отримана з форми: ' . $params['day']);

    $async_data = array(
        'order_id' => $params['order_id'],
        'first_name' => $params['first_name'],
        'last_name' => $params['last_name'],
        'patronymic' => $params['patronymic'],
        'phone_number' => $params['phone_number'],
        'email' => $params['email'],
        'day' => $params['day'],
    );

    wp_remote_post(
        admin_url('admin-ajax.php'),
        array(
            'method' => 'POST',
            'timeout' => 0.01,
            'blocking' => false,
            'body' => array(
                'action' => 'check_liqpay_payment_and_add_transaction',
                'data' => serialize($async_data),
            ),
        )
    );

    return new WP_REST_Response([
        'success' => 'Запит на транзакцію успішно надіслано для обробки',
    ]

```

```

        'отримана_дата' => $params['day'],
    ], 200);
}

function generate_button_api(WP_REST_Request $request)
{
    $params = array_map('sanitize_text_field', $request->get_params());

    if (empty($params['order_id']) || empty($params['amount'])) {
        return new WP_REST_Response(array('error' => 'order_id та amount є обов'язковими'), 400);
    }

    $form = liqpay_payment_button(
        $params['order_id'],
        $params['amount'],
        $params['currency'],
        $params['description'],
    );

    return new WP_REST_Response(array('form' => $form), 200);
}

function liqpay_payment_status_api(WP_REST_Request $request)
{
    $order_id = sanitize_text_field($request->get_param('order_id'));

    if (empty($order_id)) {
        return new WP_REST_Response(array('error' => 'order_id є обов'язковим'),
400);
    }

    $status = liqpay_payment_status($order_id);

    return new WP_REST_Response($status, 200);
}

function get_picked_slot(WP_REST_Request $request)
{
    $date_time = $request->get_param('day');

    if (!$date_time) {
        return new WP_Error('missing_date', 'Дата не вказана.', array('status' =>
400));
    }

    $isLocked = GetLiqPayOrderByDate($date_time);

    return rest_ensure_response(['locked' => (bool) $isLocked]);
}

```

```
function register_liqpay_payment_api()
{
    register_rest_route('liqpay/v1', '/generate-button', array(
        'methods' => 'GET',
        'callback' => 'generate_button_api',
    ));

    register_rest_route('liqpay/v1', '/payment-status', array(
        'methods' => 'GET',
        'callback' => 'liqpay_payment_status_api',
    ));

    register_rest_route('liqpay/v1', '/add-transaction', array(
        'methods' => 'POST',
        'callback' => 'add_liqpay_transaction',
    ));

    register_rest_route('liqpay/v1', '/locked', array(
        'methods' => 'GET',
        'callback' => 'get_picked_slot',
    ));
}
?>
```