

БАКАЛАВРСЬКА РОБОТА

БР.КІ-27.00.00.000 ПЗ

Група КІ-21-2

Вірченко Олександр

2025

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Факультет інформаційних технологій
Кафедра комп'ютерних систем і мереж

Вірченко Олександр Сергійович

УДК 004.66

БАКАЛАВРСЬКА РОБОТА

**Розробка інформаційного web-сайту для фотостудії Cubic засобами HTML,
CSS, JavaScript, JQuery та PHP**

Комп'ютерна інженерія

(назва освітньої програми)

123 - Комп'ютерна інженерія

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня _____ Вірченко О. С. _____
(підпис, ініціали та прізвище здобувача)

Науковий керівник _____ Мануляк І. З., д.т.н., доцент _____
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри КСМ

д.т.н., професор _____ С.І. Мельничук _____
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025 рік

Івано-Франківський національний технічний університет нафти і газу

Факультет Інформаційних технологій

Кафедра Комп'ютерних систем і мереж

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:

Зав. кафедрою КСМ

С.І. Мельничук

« » травня 2025 року

**З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Вірченку Олександр Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка інформаційного web-сайту для фотостудії Cubic засобами HTML, CSS, JavaScript, JQuery та PHP
керівник проекту (роботи) Мануляк Ірина Зіновіївна, доцент.

затверджені наказом вищого навчального закладу від 05.05.2025 №275/7

2. Строк подання студентом проекту (роботи) 12 червня 2025р.

3. Вихідні дані до роботи Методичні вказівки, технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області та постановка завдання.

2. Проектування вебсайту фотостудії «Cubic».

3. Програмна реалізація вебсайту фотостудії «Cubic».

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)_____.

6. Консультанти розділів роботи

7. Дата видачі завдання 29 січня 2025 р.

№ з/п	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	<i>Збір інформації, вивчення літератури та пошук додаткової інформації</i>	<i>Лютий 2025р</i>	
2	<i>Аналіз предметної області та постановка завдання</i>	<i>Березень 2025р</i>	
3	<i>Проектування вебсайту фотостудії «Cubic».</i>	<i>Квітень 2025р</i>	
4	<i>Програмна реалізація вебсайту фотостудії «Cubic»</i>	<i>Травень 2025р</i>	
5	<i>Провести тестування вебсайту, оформити дипломну роботу та підготуватися до захисту.</i>	<i>Червень 2025р</i>	

Студент _____ Вірченко О.С.

Керівник роботи _____ Мануляк І.З.

АНОТАЦІЯ

В дипломній роботі було здійснено розробку інформаційного веб-сайту для фотостудії "Cubic" засобами HTML, CSS, JavaScript, JQuery та PHP.

Мета розробки – створення функціонального, зручного та візуально привабливого вебсайту, що дозволяє клієнтам ознайомитися з послугами фотостудії, переглянути портфоліо та локації, а також здійснити онлайн-бронювання. Для адміністраторів передбачено захищений інтерфейс управління контентом сайту та обробки бронювань.

Для реалізації вебсайту було обрано класичний та перевірений стек технологій: HTML для структурування контенту, CSS для стилізації, JavaScript та JQuery для забезпечення інтерактивності клієнтської частини. Серверна логіка, включаючи обробку запитів, взаємодію з базою даних та управління контентом, реалізована за допомогою PHP. Для зберігання даних про послуги, портфоліо, локації, бронювання та адміністраторів використано реляційну базу даних MySQL.

В результаті було створено повноцінний інформаційний вебсайт для фотостудії "Cubic", який забезпечує ефективну презентацію послуг, зручний механізм онлайн-бронювання для клієнтів та інструменти адміністрування для персоналу студії.

Ключові слова: вебсайт, фотостудія, онлайн-бронювання, адміністративна панель, HTML, CSS, JavaScript, JQuery, PHP, MySQL.

SUMMARY

In the thesis, the development of an informational website for the "Cubic" photo studio was carried out using HTML, CSS, JavaScript, JQuery, and PHP.

The purpose of the development is to create a functional, user-friendly, and visually appealing website that allows clients to get acquainted with the photo studio's services, view the portfolio and locations, and also make online bookings. A secure interface for content management and booking processing is provided for administrators.

A classic and proven technology stack was chosen for the website implementation: HTML for content structuring, CSS for styling, JavaScript and JQuery for ensuring client-side interactivity. Server-side logic, including request processing, database interaction, and content management, is implemented using PHP. A relational MySQL database is used to store data about services, portfolio, locations, bookings, and administrators.

As a result, a full-fledged informational website for the "Cubic" photo studio was created, which provides an effective presentation of services, a convenient online booking mechanism for clients, and administration tools for the studio staff.

Keywords: website, photo studio, online booking, admin panel, HTML, CSS, JavaScript, JQuery, PHP, MySQL.

Зміст

<u>ВСТУП</u>	4
<u>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ</u>	6
<u>1.1 Опис предметної області вебсайтів для фотостудій</u>	6
<u>1.2 Огляд існуючих вебсайтів фотостудій та аналіз їх функціоналу</u>	9
<u>1.3 Постановка завдання</u>	14
<u>2 ПРОЄКТУВАННЯ ВЕБСАЙТУ ФОТОСТУДІЇ "CUBIC"</u>	16
<u>2.1 Визначення функціональних та нефункціональних вимог до сайту</u>	16
<u>2.2 Розробка структури сайту та UML-діаграм</u>	18
<u>2.3 Архітектура вебзастосунку</u>	23
<u>2.4 Обґрунтування вибору засобів розробки</u>	28
<u>2.5 Проєктування структури бази даних MySQL</u>	32
<u>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБСАЙТУ ФОТОСТУДІЇ "CUBIC"</u>	39
<u>3.1 Розробка серверної частини та взаємодія з базою даних</u>	39
<u>3.2 Розробка клієнтської частини та користувацького інтерфейсу</u>	44
<u>3.3 Реалізація адміністративної панелі</u>	51
<u>3.4 Тестування ефективності вебсайту та аналіз результатів</u>	58
<u>ВИСНОВКИ</u>	61
<u>ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА</u>	63

					БР.КІ-27.00.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка інформаційного web-сайту для фотостудії Cubic засобами HTML, CSS, JavaScript, JQuery та PHP	<i>Літ.</i>	<i>Арк.</i>	<i>Архивів</i>
<i>Розроб.</i>		<i>Вірченко О.С.</i>					3	
<i>Перевір.</i>		<i>Мануляк І.З.</i>				ІФНТУНГ, КІ-21-2		
<i>Реценз.</i>		<i>Пашкевич О.П.</i>						
<i>Н. Контр.</i>		<i>Лазорів А.М.</i>						
<i>Затверд.</i>		<i>Мельничук С.І.</i>						

ВСТУП

У контексті стрімкого переходу підприємницької діяльності в онлайн-простір, потреба в сайтах для бізнесу обумовлена низкою факторів, серед яких підвищення впізнаваності бренду, залучення нових клієнтів, збільшення продажів та ефективне інформування про компанію та її пропозиції. Сайт є ключовим інструментом для онлайн-маркетингу, дозволяючи компаніям взаємодіяти з клієнтами, формувати позитивний імідж та конкурувати на ринку. [4]. Особливо це стосується фотостудій, для яких ключовими факторами конкурентоспроможності є візуальна привабливість та оперативність доступу до інформації про послуги й результати творчої діяльності.

Актуальність дослідження зумовлюється суттєвим зростанням кількості фотопослуг на ринку, де класичні методи просування (друкована реклама, «сарафанне радіо») дедалі більше поступаються цифровим каналам комунікації. Професійно спроектований та реалізований сайт не лише сприяє формуванню позитивного першого враження, а й автоматизує ключові бізнес-процеси: від демонстрації портфоліо й опису локацій до онлайн-бронювання фотосесій і збору відгуків [3]. З огляду на це, створення вебплатформи для фотостудії «Cubic» із застосуванням передових вебтехнологій є важливим кроком для посилення позицій на ринку та підвищення лояльності клієнтів.

Мета даної роботи полягає у комплексному проектуванні та реалізації вебсайту фотостудії «Cubic», який задовольнить вимоги кінцевих користувачів щодо зручності навігації, швидкості завантаження та візуальної естетики, а також забезпечить адміністраторам інструменти для ефективного управління контентом та бронюваннями. Для досягнення цієї мети було сформульовано такі завдання:

1. Аналіз предметної області та порівняння функціоналу існуючих вебресурсів фотостудій задля виявлення кращих практик і типових недоліків.
2. Формалізація функціональних і нефункціональних вимог, що визначають обсяг реалізації та критерії якості майбутньої системи.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

3. Обґрунтування вибору технологій для клієнтської частини (HTML5, CSS3, JavaScript, jQuery) та серверної логіки (PHP, MySQL), з урахуванням їхньої стабільності, продуктивності й простоти підтримки [2].

4. Проєктування архітектури сайту.

5. Створення моделі даних та проєктування бази даних для зберігання інформації.

6. Реалізація клієнтської та серверної частин.

7. Розробка адміністративної панелі, яка надає можливості повного циклу управління контентом: створення, редагування, видалення об'єктів і контроль статусів бронювань.

8. Тестування функціоналу та продуктивності системи.

Об'єкт дослідження – процес проєктування та розробки інформаційних вебсистем для підприємств сфери послуг, зокрема для фотостудій, з урахуванням сучасних тенденцій вебдизайну та вимог до функціональності.

Предмет дослідження – технології, методи та інструменти розробки вебсайту фотостудії "Cubic", включаючи використання мов розмітки та стилізації, мов програмування для клієнтської та серверної частин, системи управління базами даних, а також принципи побудови користувацького інтерфейсу та адміністративної панелі.

Практичне значення результатів полягає у створенні готового рішення, яке може бути впроваджене фотостудією «Cubic» для підвищення ефективності маркетингових комунікацій, оптимізації внутрішніх процесів бронювання та підвищення рівня сервісу.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Опис предметної області вебсайтів для фотостудій

Предметна область створення вебсайтів для фотостудій охоплює сукупність функціональних та інформаційних сервісів, що мають на меті забезпечити не лише презентацію підприємства, а й автоматизацію взаємодії з клієнтами й партнерами. Фотостудія, як організаційно-виробничий комплекс, поєднує обладнані приміщення з професійним обладнанням, команду фахівців та низку додаткових послуг, як-от: макіяж й стилізації до прокату одягу та реквізиту. У зв'язку з цим, вебресурс студії виконує багатоаспектні завдання. Вебсайт фотостудії, як ключовий інструмент комунікації та залучення клієнтів, має надавати вичерпну та структуровану інформацію щодо всіх аспектів її діяльності. Ці інформаційні ознаки критично важливі для прийняття клієнтом рішення про замовлення послуг. Грунтуючись на структурі даних, розробленій для фотостудії "Cubic" (див. п. 2.5), можна виділити наступні ключові інформаційні ознаки, характерні для вебсайту фотостудії:

1. Деталізована інформація про послуги. Кожна послуга, що надається студією, повинна бути представлена з максимальною повнотою, а саме:

- назва послуги повинна бути чітка та зрозуміла;
- розгорнутий опис того, що включає послуга, її особливості та переваги для клієнта;
- актуальна ціна послуги;
- орієнтовний час, необхідний для надання послуги;
- якісні фотографії, які ілюструють результат даної послуги.

2. Портфоліо та типи робіт. Демонстрація виконаних робіт є одним із найважливіших інформаційних аспектів. Портфоліо має відображати:

- приклади робіт у різних напрямках, які пропонує студія (наприклад, весільна, портретна, сімейна, предметна фотографія);

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		6

- візуальне підтвердження професійного рівня фотографів;
- для кожної роботи в портфоліо доцільно вказувати тему зйомки, власне зображення, та короткий опис, за потреби.

3. Інформація про локації (зали). Якщо фотостудія має власні інтер'єрні зали або пропонує зйомку в конкретних локаціях, вебсайт повинен містити:

- назви залів/локацій;
- деталі про стиль, доступний реквізит, світлове обладнання тощо;
- фотографії простору. Візуальне представлення локацій.
- умови оренди, а саме вартість та період часу, на який буде орендовано зал\локацію

4. Цінова політика. Клієнти очікують прозорості в питаннях ціноутворення. На сайті чітко має бути вказана вартість кожної послуги та, за наявності, ціни на оренду локацій чи додаткове обладнання. Будь-які пакетні пропозиції або знижки також мають бути чітко описані.

5. Інформація для бронювання, яку надає клієнт, є ключовою. Це одна з основних функцій, яку виконує вебсайт. На сайті, повинна бути інформація про доступні для вибору послуги, локації, а також можливість вибрати дату та час, яка залежить від вже наявних бронювань.

6. Загальна контактна та юридична інформація. Окрім специфічних для фотостудії даних, вебсайт повинен містити стандартні інформаційні блоки:

- контактні дані: телефони, email, адреса.
- інформація про студію та команду.
- відгуки клієнтів.
- часті питання (FAQ).

Отже, на основі цих функціональних блоків, що відображають інформаційні ознаки, здійснимо аналіз ключових аспектів функціонування вебсайту фотостудії.

По-перше, вебресурс має забезпечити докладну демонстрацію спектра фотопослуг, зокрема портретні та модельні тести, сімейні й дитячі зйомки,

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		7

весільні сесії (у студії або на виїзді), авторські проекти «Love Story» та фотозйомку вагітних.

По-друге, вебсайт має виступати в ролі каталогу локацій, де відображаються доступні студійні зали з професійними інтер'єрними фотографіями, технічними специфікаціями оснащення (габарити просторів, типи фонових систем, освітлювальні прилади) та умовами оренди. Така інформація є ключовою як для кінцевого споживача фотопослуги, так і для фотографів-фрилансерів, котрі орієнтуються на якісне технічне забезпечення.

По-третє, реалізація механізму онлайн-бронювання значно спрощує взаємодію з клієнтом: клієнт має змогу перевірити доступність обраної дати та часових слотів, подати заявку, залишити контактні дані та побажання до зйомки в єдиній багатокроковій формі. Такий підхід знижує навантаження на адміністрацію студії та мінімізує ризики виникнення двозначностей або подвійних бронювань.

По-четверте, інструмент зворотного зв'язку, який включає інтерактивні форми, адаптовані для мобільних пристроїв, та інтеграцію з месенджерами й соціальними мережами, забезпечує багатоканальне надходження запитів і запитів на консультацію. Це сприяє підвищенню рівня клієнтського обслуговування та створює додаткові точки взаємодії з аудиторією.

Крім того, ефективна адмін-панель, реалізована на серверній стороні, дає змогу керівникам і співробітникам студії оперувати контентом (додавати або оновлювати послуги, роботи портфолію й інформацію про локації). Цей аспект є невід'ємною частиною операційної діяльності й покликаний оптимізувати внутрішні бізнес-процеси.

Таким чином, наповнення вебсайту фотостудії переліченими інформаційними ознаками дозволяє не тільки повноцінно презентувати її діяльність, але й забезпечити клієнта всіма необхідними даними для прийняття рішення та здійснення замовлення. Вебсайт фотостудії є багатофункціональною платформою, що поєднує маркетингові, комунікативні та операційні складові, і

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

його ефективність прямо залежить від глибини інтеграції технічних рішень із потребами як клієнтів, так і персоналу студії.

1.2 Огляд існуючих вебсайтів фотостудій та аналіз їх функціоналу

Для визначення основних тенденцій, сильних та слабких сторін існуючих рішень було проведено аналіз кількох вебсайтів українських фотостудій. Розглянемо типові риси та функціонал таких сайтів.

Сайт-аналог 1: Artphoto Studio

Вебсайт: <https://www.artphoto.studio/>

1. Функціонал

- докладний опис кожної послуги з технічними параметрами (освітлення, використане обладнання), зразковими фото та зазначенням тривалості і вартості;
- портфоліо з можливістю фільтрації за тематичними категоріями (весілля, сімейна зйомка, портрет);
- розділ локацій, що включає фотоінтер'єри, перелік доступного реквізиту та тарифи оренди;
- інтегрований онлайн-календар з відміткою зайнятих інтервалів або спрощена форма запити на бронювання;
- контактна сторінка з інтерактивною картою, формою зворотного зв'язку та розгорнутими контактними даними;
- авторський блог із тематичними статтями та новинами студії.

2. Переваги

- витримана візуальна концепція, що сприяє швидкому формуванню довіри клієнта;
- чітка навігаційна структура;
- наявність інтегрованого інструмента бронювання.

3. Недоліки

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		9

– обмежена кількість фільтрувальних опцій у портфоліо (неможливість комбінованого пошуку за датою та ключовими словами);

– відсутність персональної секції «Наша команда» з деталізацією досвіду фотографів.

Застосований дизайн спирається на мінімалістичну естетику з пріоритетом великоформатного візуального контенту, що одночасно виконує роль і маркетингового інструменту, і засобу демонстрації якості зйомки. Інформаційна архітектура витримана в патерні «Ного – Послуги – Портфоліо – Локації – Контакти – Бронювання», що забезпечує швидкий доступ до основних розділів незалежно від глибини перегляду. Адаптивність реалізовано коректно: на мобільних пристроях елементи меню закладаються у «бургер-меню», а зображення автоматично підлаштовуються під ширину екрану.

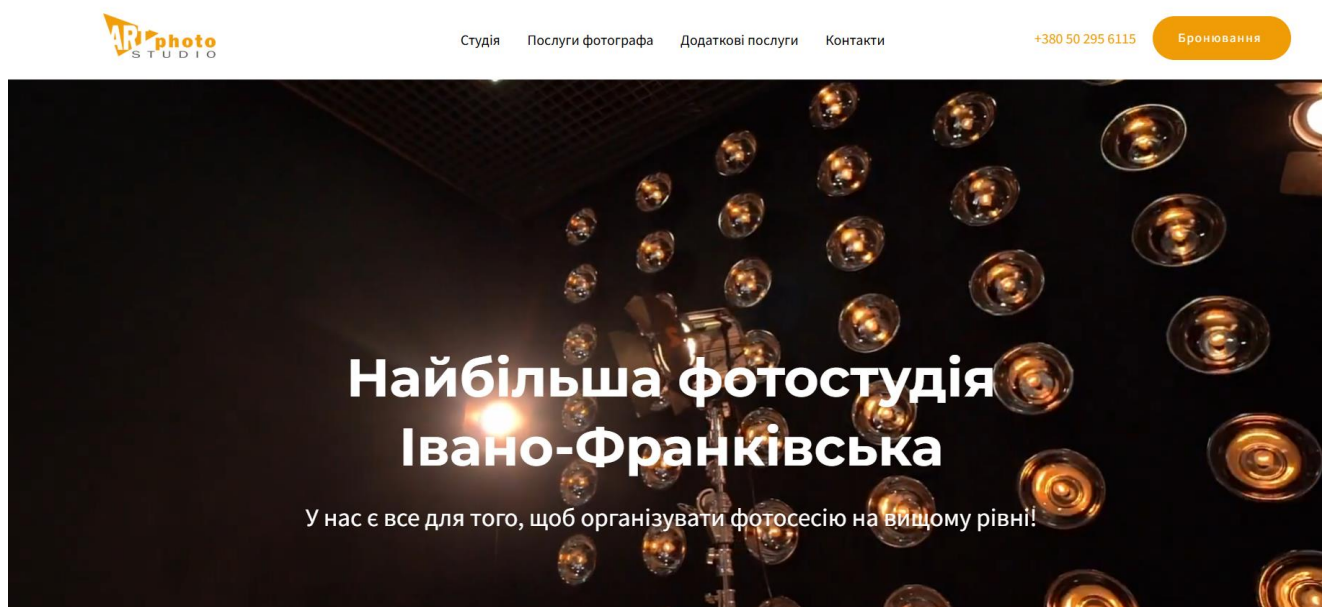


Рисунок 1.1 – Головна сторінка сайту-аналогу Artphoto Studio

Сайт-аналог 2: Wphotostudio

Вебсайт: <https://www.wphotostudio.com.ua/>

1. Функціонал

– галереї портфоліо з повноекранним переглядом і субтитруванням кожної серії знімків;

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

- розділ «Про команду» з профілями фотографів, їхніми спеціалізаціями та прикладами робіт;
- докладний опис локацій з інтегрованими віртуальними турами;
- форма запити на бронювання без прямої інтеграції календаря (контакт-менеджер обробляє заявки вручну);
- стрічка Instagram, вбудована у головну сторінку;
- блок з відгуками клієнтів з можливістю модерації.

2. Переваги

- яскравий, незабутній дизайн, який сприяє брендовій впізнаваності;
- глибоке представлення команди та портфолію;
- високий рівень інтерактивності.

3. Недоліки

- незручність у швидкому отриманні інформації про ціни (відсутній прайс-лист в одному місці);
- підвищені вимоги до ресурсів клієнтського пристрою, що може призводити до затримок через складні анімації.

Концептуально креативна реалізація інтерфейсу вирізняється нестандартними анімаційними ефектами та масштабними fullscreen-галереями, що підтверджують унікальний стиль бренду. Однак нетрадиційний підхід до інтерфейсної ієрархії може призвести до когнітивного навантаження під час первинного використання.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		11



Рисунок 1.2 – Сторінка портфоліо сайту-аналогу Wphotostudio

Сайт-аналог 3: INLIGHT Studio

Вебсайт: <https://inlight.com.ua/>

1. Функціонал

- зрозумілий перелік послуг з цінами та детальними описами;
- система бронювання з вибором дати, часу, послуги та локації;
- опція онлайн-оплати (часткова або повна передплата);
- розділ FAQ із відповідями на найпоширеніші запитання;
- блок акцій та спеціальних пропозицій.

2. Переваги

- швидка та прозора процедура бронювання;
- відкритість цінової політики;
- висока швидкість завантаження завдяки оптимізації ресурсів.

3. Недоліки

- знижена зручність навігації для нетехнічних користувачів;
- відносно «стриманий» візуальний ряд у порівнянні з конкурентами;
- відсутність блогових матеріалів для залучення аудиторії через контент-маркетинг;
- просте представлення портфоліо без інтерактивних елементів.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		12

Концепція сайту орієнтована на лаконічність і швидкий доступ до інформації. Інтерфейс вирізняється чіткою ієрархією та класичною структурою, що оптимізована для мобільних пристроїв із мінімальними затримками.

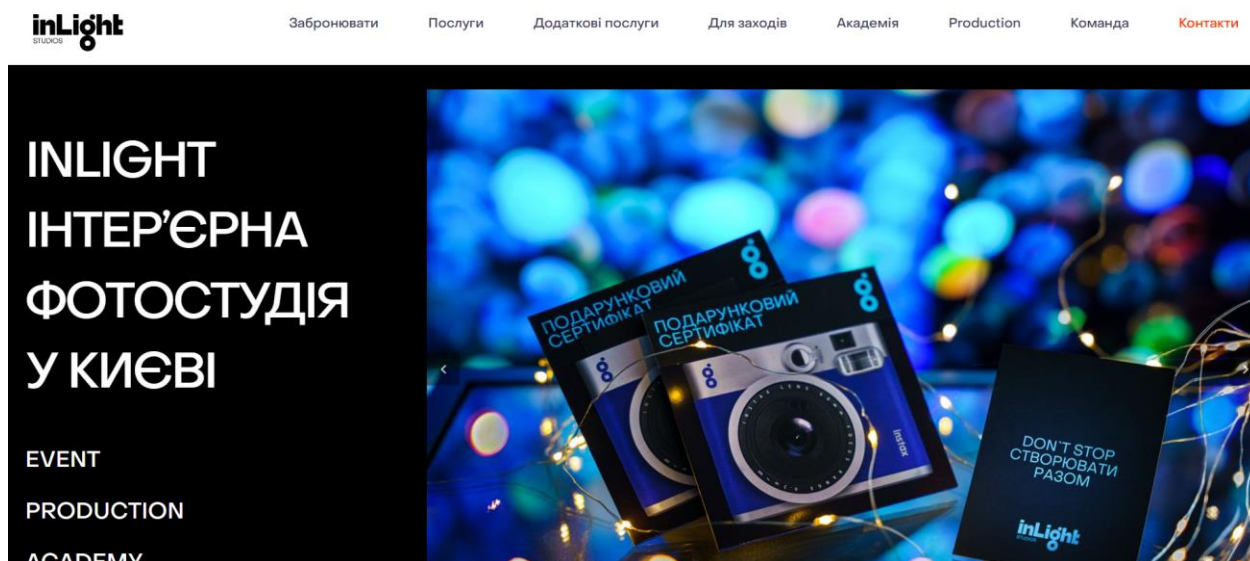


Рисунок 1.3 – Форма бронювання сайту-аналогу INLIGHT Studio

Узагальнення аналізу:

Більшість успішних сайтів фотостудій мають наступні спільні риси та функціональні блоки:

- візуальна привабливість: акцент на високоякісних фотографіях, сучасний дизайн;
- портфоліо: демонстрація робіт є обов'язковою, часто з можливістю фільтрації;
- послуги та ціни: чіткий опис послуг, тривалості, вартості;
- інформація про локації/студії: фотографії та опис залів;
- контактна інформація: легкодоступні контакти, карта;
- бронювання: від простої форми запиту до інтегрованого календаря;
- адаптивність: коректне відображення на всіх пристроях.

Проте, деякі сайти страждають від повільного завантаження, незручної навігації, відсутності прозорої цінової політики або недостатньо інформативного

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		13

опису послуг. Також не всі сайти пропонують зручну систему онлайн-бронювання, що є важливим для потенційного клієнта.

1.3 Постановка завдання

Метою даної роботи є створення інформаційного вебзастосунку для фотостудії "Cubic", який дозволяє потенційним клієнтам ознайомитися з послугами, переглянути портфоліо та локації, а також здійснювати онлайн-бронювання. Для адміністраторів має бути реалізовано захищений інтерфейс управління контентом сайту.

Проект буде реалізовано з використанням технологій HTML, CSS, JavaScript, jQuery для клієнтської частини, мови програмування PHP для серверної логіки та СУБД MySQL для зберігання даних [2].

Основні задачі, що необхідно виконати у рамках роботи:

- провести аналіз існуючих вебсайтів фотостудій з метою виявлення сильних та слабких сторін їх функціональності, структури та інтерфейсу;
- визначити перелік функціональних та нефункціональних вимог до майбутнього вебзастосунку;
- розробити макети інтерфейсу з урахуванням принципів адаптивності, зручності та привабливого дизайну;
- реалізувати клієнтську частину вебзастосунку, що включає створення сторінок з інформацією про послуги, портфоліо, локації, а також форму онлайн-бронювання;
- створити серверну частину для обробки запитів з клієнтської частини, збереження та обробки даних у базі MySQL;
- реалізувати захищену адмін-панель для керування контентом (додавання/редагування/видалення послуг, фотографій, локацій), а також перегляду й обробки бронювань;

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

– провести тестування вебзастосунку на різних пристроях і браузерах, зокрема оцінити його продуктивність, коректність роботи та зручність використання;

– підготувати висновки щодо ефективності реалізованого рішення та можливості його подальшого розвитку.

Розроблювана система повинна мати сучасний адаптивний дизайн, бути простою у використанні як для клієнтів, так і для адміністратора, забезпечувати швидку обробку запитів та захист даних. Детальні функціональні та нефункціональні вимоги будуть наведені у відповідному розділі цієї роботи.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		15

2 ПРОЄКТУВАННЯ ВЕБСАЙТУ ФОТОСТУДІЇ "CUBIC"

2.1 Визначення функціональних та нефункціональних вимог до сайту

Проаналізувавши предметну область та завдання, що постали під час розробки, необхідно спроектувати і реалізувати вебсайт фотостудії «Cubic». Система буде складатися з клієнтської частини, серверної частини та бази даних.

Клієнтська частина відповідає за взаємодію з користувачем, відображення інформаційних сторінок, інтерфейс бронювання та зворотний зв'язок. Інформація буде представлена у форматі HTML з відповідною стилізацією [4-5].

Серверна частина обробляє запити від клієнтської частини, взаємодіє з базою даних та виконує логіку обробки бронювань і контенту.

База даних відповідає за зберігання інформації про послуги, портфоліо, локації, користувацькі заявки та адміністративні дані.

Функціональні вимоги до вебсайту фотостудії «Cubic» включають наступне:

- відображення інформаційних сторінок: «Головна», «Про нас», «Послуги», «Портфоліо», «Локації», «Контакти»;
- «Головна» сторінка: коротка інформація про студію, перелік основних послуг, приклади останніх робіт;
- сторінка «Про нас»: подання історії студії, опис команди, переваги співпраці;
- розділ «Послуги»: детальний опис кожної послуги (назва, опис, ціна, тривалість, приклади робіт);
- галерея «Портфоліо»: перегляд виконаних робіт з можливістю збільшення зображень (лайтбокс) та фільтрацією за категоріями (наприклад, весільна, портретна, сімейна зйомка);
- сторінка «Локації»: відображення фото студійних залів, опис інтер'єру, обладнання, умов оренди;

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		16

– реалізація інтерактивної форми онлайн-бронювання фотосесії, що включає:

- 1) вибір послуги;
- 2) вибір студійного залу;
- 3) вибір дати в інтерактивному календарі;
- 4) динамічний вибір доступного часового слоту;
- 5) введення контактної інформації (ім'я, email, телефон);
- 6) можливість додавання коментаря або побажань;
- 7) повідомлення про статус відправки заявки (успішно/помилка).

– сторінка «Контакти»: контактні дані, інтерактивна мапа, форма зворотного зв'язку, посилання на соціальні мережі.

Для адміністративної частини вебзастосунку передбачено:

– авторизація адміністратора за логіном та паролем;

– управління контентом (CRUD-операції):

- 1) послуги: додавання, редагування, видалення, керування активністю;
- 2) портфоліо: завантаження нових фото, редагування описів,

видалення;

- 3) локації: додавання, редагування, видалення, керування

відображенням;

– управління бронюваннями:

- 1) перегляд списку заявок з деталями (контакти, дата, послуга, статус);
- 2) зміна статусу (очікує підтвердження / підтверджено / скасовано);
- 3) можливість залишати внутрішні коментарі.

– керування профілем адміністратора: зміна паролю.

Нефункціональні вимоги:

– дизайн: сучасний, візуально привабливий інтерфейс, що відповідає бренду фотостудії;

– адаптивність: коректне відображення на всіх типах пристроїв;

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		17

- швидкодія: швидке завантаження сторінок, динамічна взаємодія без затримок;
- надійність: стабільна робота системи 24/7, збереження достовірних даних;
- безпека: захист від SQL-ін'єкцій, XSS, шифрування паролів, використання HTTPS;
- масштабованість: можливість легкого розширення функціоналу у майбутньому;
- підтримуваність: структурований код з коментарями;
- зручність використання (юзабіліті): інтуїтивна навігація, простий процес бронювання, зручна адміністративна панель.

2.2 Розробка структури сайту та UML-діаграм

Проектування будь-якого вебресурсу починається з визначення його логічної структури та моделювання взаємодії користувачів з системою. Ці етапи є критично важливими для створення інтуїтивно зрозумілого, функціонального й ефективного сайту.

Розробка ієрархічної структури сайту

Для вебсайту фотостудії "Cubic" було розроблено ієрархічну структуру, що відображає логічне групування інформації та забезпечує зручну навігацію для користувачів. Карта сайту представляє основні розділи та їх взаємозв'язок:

Головна сторінка (index.php): є точкою входу на сайт та містить ключову інформацію для швидкого ознайомлення відвідувача зі студією, її основними пропозиціями та перевагами. Включає:

1. Презентаційний блок.
2. Короткий опис студії.
3. Вибірку основних послуг.
4. Демонстрацію останніх робіт з портфолію.
5. Блок з відгуками клієнтів.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		18

6. Заклик до дії (CTA).

Про нас (about.php): розділ, що детально розкриває інформацію про фотостудію, її цінності та команду.

1. Історія створення та розвитку студії.
2. Ключові переваги та особливості роботи.
3. Представлення команди фотографів та інших спеціалістів.
4. Віртуальна екскурсія студією (галерея інтер'єрів).

Послуги (services.php): містить повний перелік фотопослуг, що надаються студією.

1. Загальний опис напрямів фотозйомки.
2. Деталізований каталог послуг (назва, опис, вартість, тривалість, візуальний приклад).
3. Інформація про додаткові послуги (візаж, укладка тощо).
4. Відповіді на часті запитання (FAQ) стосовно послуг.

Портфоліо (portfolio.php): демонструє якість та стиль робіт фотостудії.

1. Галерея фотографій, згрупованих за категоріями (за потреби).
2. Реалізація пагінації для зручного перегляду великої кількості робіт.
3. Можливість перегляду зображень у збільшеному форматі (лайтбокс).

Локації (locations.php): надає інформацію про знімальні простори студії.

1. Опис та фотографії кожної доступної локації або студійного залу.
2. Інформація про умови та вартість оренди локацій.

Бронювання (booking.php): сторінка для здійснення онлайн-замовлення фотосесії.

1. Інструкції та опис процесу бронювання.
2. Інтерактивна форма для вибору послуги, локації, дати, часу та введення контактних даних.
3. FAQ стосовно процесу бронювання.

Контакти (contact.php): містить всю необхідну інформацію для зв'язку зі студією.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		19

1. Адреса, номери телефонів, адреса електронної пошти.
2. Графік роботи.
3. Інтерактивна карта місцезнаходження.
4. Посилання на профілі студії у соціальних мережах.

Окремим функціональним блоком є адміністративна панель, розташована в директорії /admin/, яка має власну структуру сторінок для управління контентом та операційною діяльністю сайту (див. п. 3.3).

Моделювання взаємодії користувачів із системою за допомогою UML-діаграм варіантів використання

Для візуалізації та аналізу взаємодії різних типів користувачів із функціональними можливостями вебсайту фотостудії "Cubic" були розроблені UML-діаграми варіантів використання (Use Case Diagrams). Ці діаграми дозволяють чітко визначити межі системи, основних акторів та завдання, котрі вони можуть виконувати.

У системі вебсайту "Cubic" виділено два ключових актори:

- відвідувач сайту (клієнт): будь-який користувач мережі Інтернет, що взаємодіє з публічною частиною сайту.
- адміністратор: авторизований користувач, що має розширені права для управління системою.

Для акторів було розроблено діаграму варіантів використання.

Діаграма варіантів використання для актора "Відвідувач сайту (Клієнт)"

Актор "Відвідувач сайту (клієнт)" взаємодіє з сайтом з метою отримання інформації та замовлення послуг. Основні варіанти використання для даного актора наведені в таблиці 2.1 та візуалізовані на рисунку 2.1.

Таблиця 2.1 – Основні варіанти використання для Відвідувача сайту

№	Варіант використання (Функція)	Короткий опис
1	Перегляд загальної інформації	Включає перегляд головної сторінки, сторінки "Про нас", контактної інформації.
2	Перегляд каталогу послуг	Дозволяє ознайомитися з переліком фотопослуг, їх детальним описом, тривалістю та вартістю.

3	Перегляд портфоліо	Надає можливість переглядати галерею виконаних робіт, з можливістю збільшення зображень для детального огляду.
4	Перегляд інформації про локації	Дозволяє переглянути опис та фотографії доступних студійних залів або знімальних зон.
5	Здійснення онлайн-бронювання	Комплексний процес, що включає: вибір послуги, вибір локації, вибір дати з календаря, вибір доступного часового слоту, введення контактних даних та надсилання запиту на бронювання.
6	Отримання динамічної інформації (напр., доступного часу)	Взаємодія з елементами сайту, які асинхронно оновлюють частину сторінки (наприклад, список доступних часів для бронювання при виборі дати).
7	Взаємодія з інтерактивними елементами	Використання слайдерів, лайтбоксу для фотографій, розгортання блоків FAQ тощо.

Діаграма варіантів використання для актора "Адміністратор".

Актор "Адміністратор" відповідає за управління контентом сайту, обробку замовлень та загальне адміністрування системи через захищену адміністративну панель. Основні варіанти використання для даного актора наведені в таблиці 2.2 та візуалізовані на рисунку 2.1.

Таблиця 2.2 – Основні варіанти використання для Адміністратора

№	Варіант використання (Функція)	Короткий опис
1	Авторизація в системі	Вхід до захищеної адміністративної панелі за допомогою унікального логіна та пароля.
2	Управління послугами	Включає можливість додавання нових послуг, редагування інформації про існуючі (назва, опис, ціна, тривалість, зображення), видалення послуг та зміну їх статусу активності (видимості на сайті).
3	Управління портфоліо	Включає додавання нових робіт (фотографій з назвами та описами) до портфоліо та видалення існуючих робіт.
4	Управління локаціями	Включає можливість додавання нових локацій, редагування їх описів та зображень, видалення локацій та зміну їх статусу активності.
5	Управління бронюваннями	Включає перегляд списку всіх отриманих бронювань, перегляд деталей кожного бронювання, зміну статусу бронювання (наприклад, "Очікує", "Підтверджено", "Скасовано") та додавання/редагування коментарів адміністратора.
6	Управління налаштуваннями профілю адміністратора	Надання можливості адміністратору змінювати власні облікові дані (логін, email, пароль).

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		21

7	Перегляд головної панелі адміністратора (Dashboard)	Відображення зведеної інформації та статистики (наприклад, кількість нових бронювань, загальна активність).
8	Вихід із системи	Безпечне завершення сесії роботи в адміністративній панелі.



Рисунок 2.1 – UML-діаграма варіантів використання для акторів вебсайту фотостудії "Cubic"

Розроблені UML-діаграми варіантів використання слугують основою для подальшого проектування архітектури системи, визначення функціональних модулів та розробки користувацьких сценаріїв, забезпечуючи чітке розуміння того, як різні користувачі будуть взаємодіяти з вебсайтом фотостудії "Cubic".

2.3 Архітектура вебзастосунку

Вебсайт фотостудії "Cubic" розроблений за класичною клієнт-серверною архітектурою, характерною для динамічних багатосторінкових вебзастосунків (MPA). Ця архітектура забезпечує чіткий поділ відповідальностей між компонентами, що відповідають за представлення інформації користувачеві, обробку бізнес-логіки та зберігання даних. Для покращення користувацького досвіду в певних інтерактивних частинах сайту, таких як форма бронювання, використовуються асинхронні запити (AJAX) для оновлення даних без повного перезавантаження сторінки [11].

Архітектура сайту включає наступні основні рівні та компоненти, взаємодія яких деталізована в таблиці 2.3:

Таблиця 2.3 – Етапи клієнт-серверної взаємодії на сайті "Cubic"

Етап	Дія на стороні клієнта (Браузер)	Дія на стороні сервера (PHP, MySQL)	Примітки
1. Початковий запит та завантаження сторінки	Користувач вводить URL / переходить за посиланням. Браузер надсилає HTTP GET-запит.	Вебсервер отримує запит. PHP-скрипт генерує HTML (можливо, з даними з MySQL), CSS, JS.	Стандартна робота MPA.
	Браузер отримує HTML, CSS, JS. Відображає сторінку. Виконує початкові JS-скрипти (напр., ініціалізація календаря).	Сервер відправляє сформовану сторінку та ресурси.	

Етап	Дія на стороні клієнта (Браузер)	Дія на стороні сервера (PHP, MySQL)	Примітки
2. Динамічна взаємодія з AJAX (приклад: вибір часу)	Користувач обирає дату/локацію. booking.js ініціює AJAX GET-запит до includes/get_available_time.php з параметрами.	get_available_times.php отримує параметри. Звертається до MySQL для аналізу бронювань та визначення вільних слотів. Формує JSON-відповідь зі списком доступних часів.	Оновлюється лише частина сторінки, без перезавантаження.
	booking.js отримує JSON-відповідь. Динамічно оновлює випадаючий список доступного часу на сторінці (маніпуляція DOM).	Сервер відправляє JSON-відповідь.	
3. Відправка даних форми (приклад: форма бронювання)	Користувач заповнює форму, натискає "Забронювати". booking.js виконує клієнтську валідацію. Відправляє AJAX POST-запит до includes/process_booking.php з даними форми.	process_booking.php отримує дані. Виконує серверну валідацію. Звертається до MySQL для фінальної перевірки доступності та збереження бронювання. Формує JSON-відповідь (успіх/помилка).	Захист від помилок введення та забезпечення цілісності даних. Також без повного перезавантаження сторінки.
	booking.js отримує JSON-відповідь. Відображає повідомлення користувачу. Може очистити форму.	Сервер відправляє JSON-відповідь.	

Таблиця 2.3 ілюструє основні сценарії взаємодії між клієнтською та серверною частинами сайту. Перший етап описує стандартний процес завантаження вебсторінки, характерний для МРА. Другий та третій етапи демонструють використання технології AJAX для підвищення інтерактивності та покращення користувацького досвіду, зокрема, при виборі доступного часу для бронювання та при відправці форми бронювання. Такі асинхронні операції

дозволяють оновлювати лише необхідні фрагменти сторінки, уникаючи повного її перезавантаження, що робить взаємодію з сайтом більш плавною та швидкою. Схема цієї взаємодії представлена на рисунку 2.3.

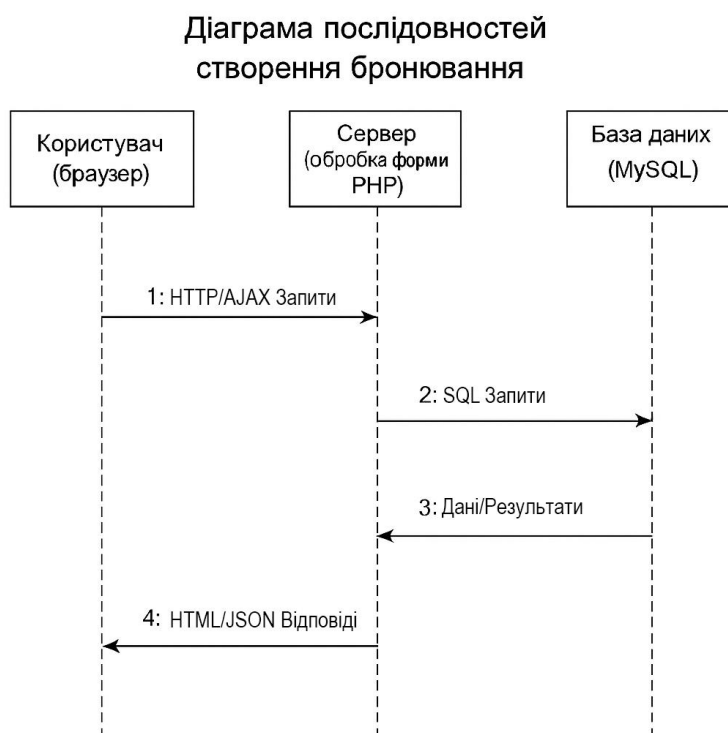


Рисунок 2.3 – Класична клієнт-серверна взаємодія з елементами AJAX для вебсайту "Cubic"

Враховуватимемо даний тип архітектури при виборі та описі засобів розробки.

У результаті проведеного проєктування було розроблено наступну структурну схему компонентів вебсайту фотостудії "Cubic", представлену на рисунку 2.4, що деталізує ключові технологічні шари та їх взаємодію.

Таблиця 2.4 – Компоненти архітектури вебсайту "Cubic" і технології

Шар архітектури	Компонент / Відповідальність	Використовувані технології/Протоколи
Клієнтська частина (Frontend)	Інтерфейс користувача: відображення сторінок, інтерактивні елементи, форми.	HTML5, CSS3, JavaScript, JQuery, JQuery UI (для DatePicker).

Шар архітектури	Компонент / Відповідальність	Використовувані технології/Протоколи
	Логіка на клієнті: обробка подій, валідація введених даних, динамічне оновлення DOM, AJAX-запити.	JavaScript (main.js, booking.js, gallery.js), JQuery
Канал зв'язку	Передача даних між клієнтом та сервером.	HTTP/HTTPS, AJAX (XHR)
Серверна частина (Backend)	Вебсервер XAMPP: прийом HTTP-запитів, виконання серверних скриптів.	Apache (або аналог)
	Обробка запитів та бізнес-логіка: генерація HTML, валідація даних, обробка бронювань, управління контентом, автентифікація.	PHP (скрипти в includes/, admin/)
	Взаємодія з БД: формування та виконання SQL-запитів.	PHP (розширення mysqli)
	Управління сесіями: підтримка стану авторизованого адміністратора.	PHP Sessions
Рівень зберігання даних (Data Tier)	Система управління базами даних (СУБД): зберігання, пошук та управління даними.	MySQL
	Структура даних: таблиці (users, services, portfolio, locations, bookings) з визначеними полями та зв'язками.	SQL (схема cubic_db)

Таблиця 2.4 деталізує технологічний стек та розподіл відповідальності між основними шарами архітектури. Клієнтська частина (Frontend), що виконується в браузері користувача, відповідає за представлення даних та первинну взаємодію. Технології HTML5 та CSS3 формують структуру та візуальне оформлення, тоді як JavaScript, разом з бібліотеками JQuery та JQuery UI, забезпечує інтерактивність, валідацію форм на стороні клієнта та асинхронну комунікацію з сервером. Канал зв'язку між клієнтом та сервером реалізований за допомогою протоколів HTTP/HTTPS. Серверна частина (Backend), що функціонує на вебсервері (у даному випадку Apache в складі XAMPP), використовує PHP для обробки запитів, реалізації бізнес-логіки (включаючи валідацію даних, управління бронюваннями, автентифікацію адміністраторів), взаємодії з базою даних через розширення mysqli та управління сесіями для адміністративної панелі. Рівень зберігання даних (Data Tier) представлений системою управління базами даних

									Арк.
									26
Зм.	Арк.	№ докум.	Підп.	Дата	БР.КІ-27.00.00.000 ПЗ				

MySQL, де зберігається вся структурована інформація проєкту в таблицях, визначених SQL-схемою. Ця схема взаємодії візуалізована на рисунку 2.4.

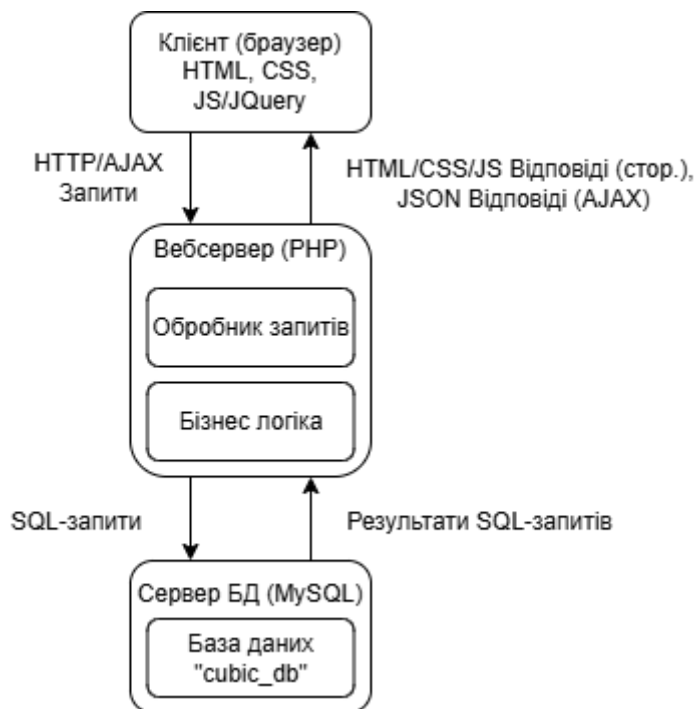


Рисунок 2.4 – Структурна схема компонентів вебсайту фотостудії "Cubic" та потоків даних

Дана структурна схема забезпечує чіткий поділ відповідальностей між клієнтською та серверною частинами, що сприяє кращій організації коду та полегшує розробку та підтримку. Клієнтська частина відповідає за візуальне представлення даних та первинну взаємодію з користувачем, ініціюючи запити до серверної частини. Серверна частина, в свою чергу, обробляє ці запити, виконує необхідну бізнес-логіку, взаємодіє з базою даних для отримання або збереження інформації, та повертає результат клієнту у вигляді готової HTML-сторінки або структурованих даних (JSON) для AJAX-запитів. База даних MySQL слугує надійним сховищем для всієї інформації, що використовується та генерується вебсайтом [2, 12].

Ефективний вебзастосунок, побудований за такою архітектурою, повинен мати оптимізований код на всіх рівнях. На клієнтській стороні це мінімізація CSS

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		27

та JS файлів, оптимізація зображень. На серверній – ефективні SQL-запити, правильна валідація даних та керування ресурсами.

Безпека вебсайту включає захист від поширених вебвразливостей. На серверній стороні це досягається шляхом валідації всіх вхідних даних від користувача та використання підготовлених SQL-запитів для взаємодії з базою даних, що запобігає SQL-ін'єкціям. На клієнтській стороні та при виводі даних на сторінку використовується екранування спеціальних символів для запобігання XSS-атакам. Адміністративна панель захищена системою автентифікації, а паролі користувачів зберігаються в базі даних у захешованому вигляді.

2.4 Обґрунтування вибору засобів розробки

У межах розробки вебсайту фотостудії «Cubic» вибір технологічного стеку обґрунтовується необхідністю досягнення балансу між продуктивністю, надійністю та простотою подальшої підтримки. З огляду на вимоги до функціональності та нефункціональні показники, такі як швидкодія, безпека та масштабованість, було прийнято рішення розділити систему на три основні компоненти: клієнтську частину, серверну логіку та сховище даних.

Серверна частина (PHP)

Обрана мова PHP (Hypertext Preprocessor) [7] забезпечує оптимальне співвідношення простоти використання та можливостей для реалізації бізнес-логіки:

- широка поширеність серед хостинг-провайдерів полегшує розгортання без потреби в додатковому програмному оточенні;
- численна спільнота та обширна документація прискорюють адаптацію до складних завдань і сприяють швидкому вирішенню нестандартних ситуацій;
- вбудована підтримка роботи з сесіями та формами дозволяє організувати безпечну авторизацію в адміністративній панелі й оброблювати дані користувача без додаткових бібліотек;

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		28

– гнучка інтеграція з MySQL-сервером забезпечує ефективне виконання запитів і підтримку транзакцій за потреби.

Як альтернативу розглядалися Node.js із JavaScript-фреймворками на сервері та Python (Django/Flask). Проте використання JavaScript-середовища потребувало б уніфікації подекуди різнорідних залежностей клієнтської та серверної частин, а впровадження Python-фреймворків суттєво ускладнило б конфігурацію типових хостинг-рішень, що не виправдовувалося обсягом функціоналу.

Система управління базою даних (MySQL).

Реляційна СУБД MySQL [8] обрана з огляду на природність моделювання зв'язків між сутностями (послуги, локації, бронювання):

- стабільність та висока доступність, підтвержені величезною кількістю успішних проєктів;
- достатня продуктивність для очікуваних обсягів даних і можливість індексації для швидкого пошуку;
- проста адміністрація через утиліти (phpMyAdmin, CLI-клієнт) та зрозумілий SQL-синтаксис;
- класична сумісність із PHP-модулем mysqlnd, що забезпечує мінімальні накладні витрати на з'єднання.

У якості альтернативи розглядалися документоорієнтовані СУБД (наприклад, MongoDB), які надали б більшу гнучкість у структурі даних. Однак було прийняте рішення, що для чіткого визначених сутностей із взаємопов'язаними записами, реляційна модель є природнішою, гарантуючи цілісність за допомогою зовнішніх ключів і знижуючи ризик розрізнених або неконсистентних даних.

Клієнтська частина (HTML5, CSS3, JavaScript + jQuery).

Використання класичних вебтехнологій забезпечує зручність розробки [14] та широке покриття пристроїв:

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

– HTML5 гарантує семантичну розмітку та поліпшену доступність інтерфейсу;

– CSS3 дає змогу реалізувати сучасний адаптивний дизайн із мінімальними витратами ресурсів;

– JavaScript забезпечує динамічну взаємодію з користувачем (валідація форм, анімаційні ефекти);

– jQuery спрощує маніпуляції з DOM і кросбраузерну реалізацію AJAX-запитів [11].

Хоча можна було б застосувати сучасні фронтенд-фреймворки (React, Vue.js), їхня інтеграція в проєкт з базовою інтерактивністю виявилася б надмірною через необхідність організації складнішого процесу збірки та налаштування керування станом.

Таким чином, обраний набір технологій дозволяє забезпечити високу швидкодію, безпеку та простоту подальшої підтримки й розширення вебсайту фотостудії «Cubic», одночасно мінімізуючи складність розгортання та експлуатації.

Розглянемо деякі популярні JavaScript-фреймворки, які могли б бути альтернативою для складніших проєктів, та порівняємо їх з обраним підходом.

Таблиця 2.5 – Порівняльна характеристика JavaScript-технологій

Критерії оцінки	JavaScript + JQuery (обраний підхід)	React.js	Angular	Vue.js
Ключові особливості	Простота, гнучкість, велика кількість плагінів, низький поріг входження.	Компонентний підхід, Virtual DOM, односпрямований потік даних, JSX.	Повноцінний MVC/MVVM фреймворк, TypeScript, двостороннє зв'язування даних.	Прогресивний фреймворк, простота інтеграції, Virtual DOM, легковаговість.
Складність вивчення	Низька	Середня (потрібно розуміти JSX)	Висока (багато концепцій, TypeScript)	Низька до середньої

Критерії оцінки	JavaScript + JQuery (обраний підхід)	React.js	Angular	Vue.js
Продуктивність	Залежить від оптимізації коду; може бути нижчою для складних UI.	Висока завдяки Virtual DOM.	Висока, але може бути важким для малих проєктів.	Висока, порівнянна з React.
Розмір бібліотеки	Малий (JQuery)	Середній	Великий	Малий
Екосистема	Дуже велика, безліч плагінів та рішень.	Велика, активна спільнота, багато бібліотек.	Велика, підтримується Google.	Зростаюча, активна спільнота.
Підхід до розробки	Імперативний, маніпуляція DOM.	Декларативний, компонентний.	Структурований, компонентний, з чіткими правилами.	Компонентний, гнучкий.
Придатність для проєкту "Cubic"	Оптимально. Достатньо для реалізації всієї необхідної інтерактивності, валідації, AJAX. Швидка розробка для сайту середньої складності.	Можливо надлишково для даного обсягу, але забезпечило б кращу структуру UI.	Явно надлишково, збільшило б час розробки.	Гарна альтернатива React, але JQuery вже використовується для календаря.

Для сайту фотостудії «Cubic», де основний акцент робиться на візуальному контенті та відносно нескладній користувацькій взаємодії (перегляд сторінок, портфоліо, заповнення форми бронювання), використання "чистого" JavaScript разом з бібліотекою JQuery є виправданим та ефективним рішенням. JQuery значно спрощує маніпуляції з DOM-елементами, обробку подій та виконання AJAX-запитів (наприклад, для динамічного завантаження доступного часу у формі бронювання). Використання повноцінних JavaScript-фреймворків, таких як React, Angular або Vue.js, для даного проєкту було б надлишковим, оскільки вони призначені для розробки більш складних застосунків і вимагали б більшого часу на розробку та налаштування, а також потенційно збільшили б розмір клієнтського коду без суттєвих переваг для кінцевого користувача в цьому

									Арк.
									31
Зм.	Арк.	№ докум.	Підп.	Дата	БР.КІ-27.00.00.000 ПЗ				

конкретному випадку. Також використовується JQuery UI для реалізації календаря (datepicker), що є стандартним та зручним рішенням [13].

jQuery UI Datepicker:

- Призначення: бібліотека компонентів інтерфейсу, що розширює JQuery.
- Обґрунтування: обрана для реалізації календаря вибору дати у формі бронювання через свою простоту інтеграції, надійність та широкі можливості кастомізації (локалізація, обмеження дат).

Отже, обраний стек технологій – HTML, CSS, JavaScript, JQuery для клієнтської частини, PHP для серверної логіки та MySQL для бази даних – є збалансованим, поширеним та добре документованим, що дозволяє ефективно реалізувати всі поставлені функціональні та нефункціональні вимоги до вебсайту фотостудії "Cubic". Такий підхід забезпечує достатню гнучкість, продуктивність та можливість подальшого розвитку проєкту.

2.5 Проєктування структури бази даних MySQL

Реляційна база даних MySQL є фундаментальною складовою вебресурсу фотостудії «Cubic», адже забезпечує зберігання, цілісність та ефективний доступ до всіх динамічних даних системи. Проєктування структури даних виконано з урахуванням принципів нормалізації, обмеження надлишковості та забезпечення масштабованості рішення.

Основні сутності та їх атрибути:

Таблиця 2.6 - Users (Користувачі - адміністратори):

Назва поля	Тип даних	Обмеження	Опис
id	INT	PK (Первинний ключ), AI (Автоінкремент)	Унікальний ідентифікатор користувача
username	VARCHAR(50)	NN (Не може бути NULL), UQ (Унікальний)	Логін адміністратора
password	VARCHAR(255)	NN	Захешований пароль адміністратора

Назва поля	Тип даних	Обмеження	Опис
Email	VARCHAR(100)	NN	Електронна пошта адміністратора
created_at	TIMESTAMP	NN, DEFAULT CURRENT_TIMESTAMP	Дата та час створення запису

Створення таблиці users в системі управління базами даних здійснюється за допомогою SQL-команди CREATE:

```
CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Поля таблиці users спроектовані для безпечної та ефективної автентифікації адміністраторів: id слугує унікальним первинним ключем з автоінкрементом; username є обов'язковим унікальним логіном достатньої довжини; password зберігає результат хешування для безпеки (довжина VARCHAR(255) враховує розмір хешу); email необхідний для комунікації та відновлення доступу; created_at фіксує час створення запису для аудиту.

Таблиця 2.7 – Services (Послуги)

Назва поля	Тип даних	Обмеження	Опис
id	INT	PK, AI	Унікальний ідентифікатор послуги
name	VARCHAR(100)	NN	Назва послуги
description	TEXT		Детальний опис послуги
price	DECIMAL(10,2)	NN	Ціна послуги
duration	INT	NN	Тривалість послуги в хвилинах
image	VARCHAR(255)		Шлях до зображення послуги
active	TINYINT(1)	NN, DEFAULT 1	Статус активності (1-активна, 0-ні)

Створення таблиці services в системі управління базами даних здійснюється за допомогою SQL-команди CREATE:

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		33

```

CREATE TABLE IF NOT EXISTS services (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10,2) NOT NULL,
    duration INT NOT NULL,
    image VARCHAR(255),
    active TINYINT(1) NOT NULL DEFAULT 1
);

```

Атрибути таблиці `services` забезпечують повне представлення послуг фотостудії: `name` є чітким ідентифікатором; `description` (тип `TEXT`) дозволяє детально описати послугу; `price` (тип `DECIMAL(10,2)`) гарантує точне зберігання вартості; `duration` (в хвиликах) використовується для планування; `image` містить шлях до візуального представлення; `active` (`TINYINT(1)`) дозволяє адміністратору динамічно керувати видимістю послуги на сайті.

Таблиця 2.8 – Portfolio (Портфоліо)

Назва поля	Тип даних	Обмеження	Опис
id	INT	PK, AI	Унікальний ідентифікатор роботи
title	VARCHAR(100)	NN	Назва роботи/фотосесії
image	VARCHAR(255)	NN	Шлях до зображення
description	TEXT		Короткий опис роботи (необов'язково)
created_at	TIMESTAMP	NN, DEFAULT CURRENT_TIMESTAMP	Дата та час додавання роботи

Створення таблиці `portfolio` в системі управління базами даних здійснюється за допомогою SQL-команди `CREATE`:

```

CREATE TABLE IF NOT EXISTS portfolio (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    image VARCHAR(255) NOT NULL,
    description TEXT,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);

```

Структура таблиці `portfolio` орієнтована на ефективну демонстрацію робіт: `title` та обов'язкове поле `image` є ключовими для ідентифікації та візуалізації;

description надає додатковий контекст; created_at дозволяє сортувати роботи, наприклад, для відображення найновіших, що є важливим для актуальності портфолію.

Таблиця 2.9 – Locations (Локації)

Назва поля	Тип даних	Обмеження	Опис
id	INT	PK, AI	Унікальний ідентифікатор локації
name	VARCHAR(100)	NN	Назва локації/залу
description	TEXT		Опис локації
image	VARCHAR(255)	NN	Шлях до основного зображення локації
active	TINYINT(1)	NN, DEFAULT 1	Статус активності (1-активна, 0-ні)

Створення таблиці locations в системі управління базами даних здійснюється за допомогою SQL-команди CREATE:

```
CREATE TABLE IF NOT EXISTS locations (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    image VARCHAR(255) NOT NULL,
    active TINYINT(1) NOT NULL DEFAULT 1
);
```

Атрибути таблиці locations дозволяють клієнтам та адміністраторам отримувати повну інформацію про доступні зали: name слугує для чіткої ідентифікації; description надає деталі про особливості локації; обов'язкове поле image забезпечує візуальне представлення; active дозволяє тимчасово вимикати локацію з бронювання без її фізичного видалення.

Таблиця 2.10 – Bookings (Бронювання)

Назва поля	Тип даних	Обмеження	Опис
id	INT	PK, AI	Унікальний ідентифікатор бронювання
client_name	VARCHAR(100)	NN	Ім'я клієнта

Таблиця bookings є ключовою для операційної діяльності фотостудії, тому її поля ретельно продумані: обов'язкові контактні дані клієнта (client_name, client_email, client_phone); зовнішні ключі service_id та location_id забезпечують посилальну цілісність з відповідними послугами та локаціями; окремі поля booking_date та booking_time спрощують логіку перевірки доступності; status (тип ENUM) стандартизує життєвий цикл бронювання; поля notes та admin_comment дозволяють зберігати додаткову інформацію; created_at фіксує час створення запиту.

Пояснення до позначень у стовпці "Обмеження":

1. PK: Primary Key (Первинний ключ) - унікально ідентифікує кожен запис у таблиці.

2. AI: Auto Increment (Автоінкремент) - значення автоматично збільшується при додаванні нового запису.

3. NN: Not NULL (Не може бути NULL) - поле обов'язкове для заповнення.

4. UQ: Unique (Унікальний) - значення в цьому полі має бути унікальним для всієї таблиці.

5. FK: Foreign Key (Зовнішній ключ) - посилається на первинний ключ в іншій таблиці, забезпечуючи зв'язок та цілісність даних.

6. DEFAULT value: значення за замовчуванням, якщо інше не вказано при додаванні запису.

Зв'язки між таблицями (ERD - Entity-Relationship Diagram):

– таблиця bookings пов'язана з таблицею services через поле service_id (одне бронювання відноситься до однієї послуги, одна послуга може бути у багатьох бронюваннях – зв'язок "один-до-багатьох");

– таблиця bookings пов'язана з таблицею locations через поле location_id (одне бронювання відноситься до однієї локації, одна локація може бути у багатьох бронюваннях – зв'язок "один-до-багатьох").

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		37

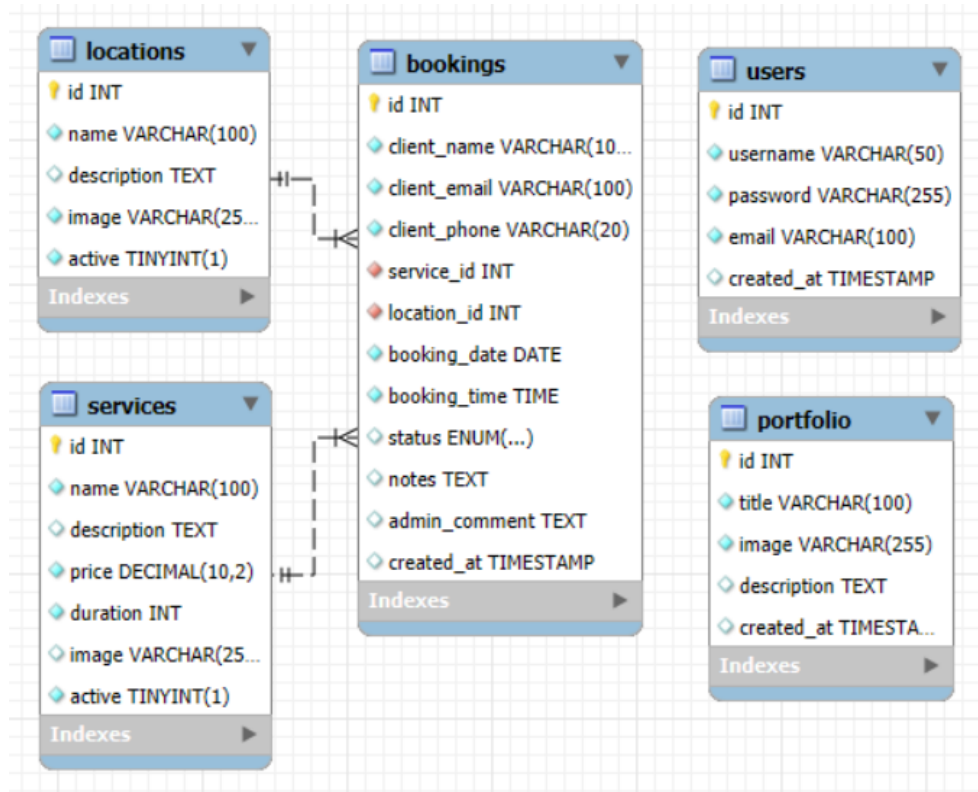


Рисунок 2.5 – ERD (діаграма "сутність-зв'язок")
бази даних фотостудії "Cubic"

Така структура бази даних дозволяє ефективно зберігати необхідну інформацію, забезпечує цілісність даних за допомогою зовнішніх ключів та є достатньо гнучкою для можливих майбутніх розширень.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБСАЙТУ ФОТОСТУДІЇ "CUBIC"

3.1 Розробка серверної частини та взаємодія з базою даних

Серверна частина вебсайту фотостудії "Cubic" розроблена мовою програмування PHP і є відповідальною за обробку HTTP-запитів від клієнта, реалізацію основної бізнес-логіки, взаємодію з базою даних MySQL для зберігання та отримання інформації, а також за функціонування адміністративної панелі.

Ключові серверні скрипти та функції організовані для забезпечення модульності та перевикористання коду.

Структура серверних файлів та підключення до бази даних:

Файл `include/db_connect.php` відповідає за встановлення з'єднання з базою даних MySQL. Він містить конфігураційні параметри (хост, ім'я користувача, пароль, назва бази даних) та ініціалізує об'єкт з'єднання `$conn`. Важливим аспектом є встановлення кодування UTF-8 для коректної обробки кирилических символів.

```
<?php
$host = 'localhost';
$username = 'root'; // Замініть на ваші дані
$password = 'root'; // Замініть на ваші дані
$database = 'cubic db';
$conn = mysqli_connect($host, $username, $password, $database);

if (!$conn) {
    // Для продуктивного середовища рекомендується логування
    помилки,
    // а не прямий вивід користувачу.
    error_log("Помилка підключення до БД: " .
mysqli_connect_error());
    // Можна відобразити загальну сторінку помилки
    die("Виникла технічна помилка на сервері. Будь ласка,
спробуйте пізніше.");
}
mysqli_set_charset($conn, "utf8");
?>
```

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		39

Цей скрипт забезпечує єдину точку конфігурації підключення до бази даних, що полегшує управління та можливі зміни налаштувань у майбутньому.

Основні функції для роботи з даними (includes/functions.php):

Для централізації логіки взаємодії з базою даних та уникнення дублювання коду було створено файл functions.php. Він містить набір функцій, що інкапсулюють типові операції.

Функції отримання даних:

– getAllServices(\$conn, \$limit = null, \$activeOnly = true): виконує SQL-запит SELECT до таблиці services для отримання списку послуг. Використовує підготовлені запити, якщо передано параметр \$limit. Параметр \$activeOnly дозволяє фільтрувати тільки активні послуги для відображення на клієнтській частині або всі послуги для адміністративної панелі.

```
function getAllServices($conn, $limit = null, $activeOnly = true) {
    $sql = "SELECT * FROM services";
    if ($activeOnly) {
        $sql .= " WHERE active = 1";
    }
    $sql .= " ORDER BY id ASC";
    if ($limit !== null && is_numeric($limit)) {
        // Використовуємо підготовлений запит для ліміту
        $sql .= " LIMIT ?";
        $stmt = mysqli_prepare($conn, $sql);
        mysqli_stmt_bind_param($stmt, "i", $limit);
        mysqli_stmt_execute($stmt);
        $result = mysqli_stmt_get_result($stmt);
        mysqli_stmt_close($stmt);
    } else {
        $result = mysqli_query($conn, $sql);
    }

    if (!$result) {
        error_log("Error fetching services: " .
mysqli_error($conn));
        return array();
    }
    return mysqli_fetch_all($result, MYSQLI_ASSOC);
}
```

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		40

Аналогічна логіка застосовується для функцій `getPortfolio($conn, $limit = null)` та `getLocations($conn, $activeOnly = true)`. Функції типу `getServiceById($conn, $id)` також використовують підготовлені запити з параметром ID.

Функція валідації вхідних даних:

– `validateInput()`. Здійснює базову обробку вхідних даних (видалення пробілів по краях, екранування HTML-спецсимволів за допомогою `htmlspecialchars()`) для запобігання XSS-атакам перед виведенням даних або їх збереженням [7, 10].

Функції логіки бронювання:

– `checkBookingOverlap()`: одна з ключових функцій системи. Вона отримує параметри нового бронювання (дату, час початку, тривалість послуги в хвилинах, ID локації) та опціонально ID бронювання, яке потрібно виключити з перевірки (для сценаріїв редагування). Функція виконує SQL-запит до таблиці `bookings` (з JOIN до `services` для отримання тривалості існуючих послуг), щоб знайти всі активні (`pending` або `confirmed`) бронювання на вказану дату та локацію. Потім, для кожного знайденого бронювання, розраховується його часовий інтервал (початок та кінець). Ці інтервали порівнюються з інтервалом нового бронювання. Якщо виявляється перетин ($(StartA < EndB) \text{ AND } (EndA > StartB)$), функція повертає `true`; в іншому випадку – `false`;

– `getAvailableHourlySlots()`: ця функція використовується для динамічного формування списку доступного часу. Вона генерує стандартний набір годинних слотів протягом робочого дня (з 10:00 до 19:00). Потім, аналогічно `checkBookingOverlap`, вона вибирає всі зайняті інтервали на вказану дату та локацію, враховуючи їх тривалості (тривалість послуг заокруглюється до годин вгору, щоб заблокувати весь годинний слот, якщо бронювання займає його частину). Ті годинні слоти, які перетинаються із зайнятими інтервалами, позначаються як заблоковані. Функція повертає масив лише тих годинних слотів, які залишилися вільними;

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		41

– createBooking(): відповідає за додавання нового запису про бронювання до таблиці bookings. Використовує підготовлений SQL-запит INSERT для безпечного збереження даних, отриманих з форми. Встановлює початковий статус бронювання як 'pending'.

Обробка запитів від клієнтської частини:

Обробка запиту на отримання доступного часу (/get_available_times.php): цей PHP-скрипт є ендпоінтом для AJAX-запиту з клієнтської сторони (з файлу /booking.js).

- 1) Приймає GET-параметри date та location_id.
- 2) Проводить валідацію отриманих параметрів.
- 3) Викликає функцію getAvailableHourlySlots(\$conn, \$date, \$location_id) для отримання масиву доступних годинних слотів.
- 4) Формує та відправляє відповідь у форматі JSON, яка містить прапорець успіху (success) та масив доступних часів (times).

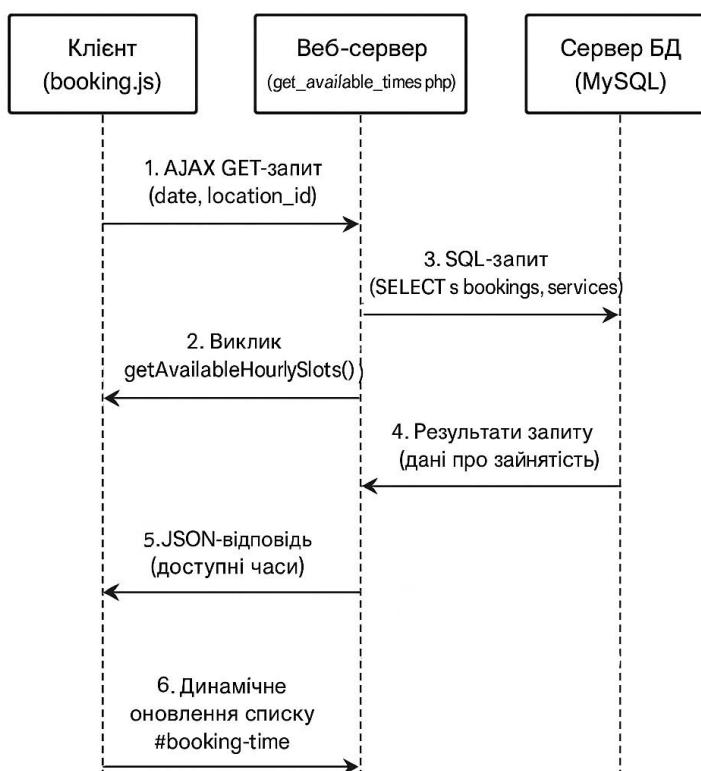


Рисунок 3.1 – Схема взаємодії при запиті доступного часу для бронювання

Обробка створення бронювання (includes/process_booking.php):

Цей скрипт обробляє POST-запит, що надходить від форми бронювання після клієнтської валідації.

1) Отримує дані з масиву \$_POST (ім'я клієнта, email, телефон, ID послуги, ID локації, дата, час, коментарі).

2) Виконує серверну валідацію отриманих даних (перевірка на заповнення, коректність форматів, існування послуги/локації в БД).

3) Якщо валідація не пройдена, формується JSON-відповідь з повідомленням про помилку.

4) Якщо дані коректні, отримується тривалість обраної послуги з таблиці services за її ID.

5) Викликається функція checkBookingOverlap() для фінальної перевірки доступності обраного часу, враховуючи тривалість послуги. Це важливо, оскільки між моментом завантаження доступних слотів на клієнті та відправкою форми хтось інший міг забронювати час.

6) Якщо час вільний, викликається функція createBooking() для збереження нового бронювання в базі даних зі статусом "pending".

7) Формується та відправляється JSON-відповідь клієнту про успішне створення бронювання або про помилку (наприклад, якщо час виявився зайнятим або сталася помилка при записі в БД).

Логіка адміністративної панелі:

Серверна частина адміністративної панелі (детальніше в п. 3.3) реалізує:

1) Автентифікацію адміністратора: перевірка логіна та пароля (з використанням password_verify()) з даними з таблиці users. Управління сесіями для підтримки стану авторизації.

2) CRUD-операції: для послуг, робіт у портфолію та локацій реалізовані операції створення (INSERT), читання (SELECT), оновлення (UPDATE) та видалення (DELETE) записів у відповідних таблицях бази даних. Усі операції, що змінюють дані, використовують підготовлені запити.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		43

3) Управління бронюваннями: перегляд списку бронювань, їх деталей, зміна статусу (pending, confirmed, cancelled) та додавання коментарів адміністратора.

4) Управління профілем адміністратора: зміна логіна, email та пароля.

Вся взаємодія з базою даних на серверній стороні здійснюється з урахуванням принципів безпеки для запобігання SQL-ін'єкціям шляхом використання підготовлених запитів та екранування даних, де це необхідно.

3.2 Розробка клієнтської частини та користувацького інтерфейсу

Клієнтська частина вебсайту "Cubic" відповідає за візуальне представлення інформації користувачеві, забезпечення інтерактивної взаємодії та збір даних для відправки на сервер. Вона розроблена з використанням HTML для структури, CSS для стилізації, та JavaScript (з бібліотекою JQuery) для динамічної поведінки [1].

Структура HTML-сторінок та шаблонізація:

Вебсайт складається з набору PHP-файлів, кожен з яких представляє окрему сторінку (index.php, about.php, services.php тощо). Для забезпечення консистентності та уникнення дублювання коду, загальні елементи інтерфейсу, такі як шапка (хедер) та підвал (футер), винесені в окремі файли:

– includes/header.php: містить початкову частину HTML-документа (<!DOCTYPE html>, <head> з мета-тегами, підключенням CSS-файлів, шрифтів, бібліотеки JQuery та JQuery UI), а також HTML-розмітку для шапки сайту, включаючи логотип та головне навігаційне меню.

– includes/footer.php: містить HTML-розмітку для підвалу сайту (копірайт, додаткова навігація, контактна інформація, посилання на соціальні мережі) та підключення основного JavaScript-файлу js/main.js.

PHP використовується всередині HTML-сторінок для динамічної генерації контенту. Наприклад, на сторінці послуг (services.php) PHP-код викликає функцію getAllServices() для отримання даних з бази, а потім в циклі foreach формує HTML-розмітку для кожної послуги.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		44

```

// (header.php, db_connect.php, functions.php)
<?php $services = getAllServices($conn); // Отримуємо всі
активні послуги ?>
<section class="services-list">
    <div class="container">
        <?php if (!empty($services)): ?>
            <?php foreach ($services as $service): ?>
                <div class="service-item" id="service-<?php
echo $service['id']; ?>">
                    <div class="service-image">
                        ">
                    </div>
                    <div class="service-details">
                        <h2><?php echo
htmlspecialchars($service['name']); ?></h2>
                        <div class="service-meta">
                            <div class="service-price">
                                <span>Вартість:</span> <?php
echo htmlspecialchars($service['price']); ?> грн
                            </div>
                            <div class="service-duration">
                                <span>Тривалість:</span> <?php
echo htmlspecialchars($service['duration']); ?> хв
                            </div>
                        </div>
                        <div class="service-description">
                            <p><?php echo
nl2br(htmlspecialchars($service['description'])); ?></p>
                        </div>
                        <a href="booking.php?service=<?php echo
$service['id']; ?>" class="btn">Забронювати</a>
                    </div>
                </div>
            <?php endforeach; ?>
            <?php else: ?>
                <p class="no-items">Наразі немає доступних
послуг.</p>
            <?php endif; ?>
        </div>
    </section>
// (footer.php)

```

Використання функції htmlspecialchars() при виведенні даних з бази даних є важливим для запобігання XSS-атакам. Функція nl2br() використовується для коректного відображення переносів рядків в описі послуг.

Стилізація (CSS):

Зовнішній вигляд сайту визначається правилами, описаними в CSS-файлах:

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		45

– `css/style.css`: містить основні стилі для всіх елементів сайту: типографіку, кольорову схему (визначену через CSS-змінні в `:root`), стилі для кнопок, форм, навігації, карток послуг, елементів портфоліо, секцій сторінок тощо.

– `css/responsive.css`: Забезпечує адаптивність сайту. Містить медіа-запити (`@media`), які застосовують різні стилі залежно від ширини екрану пристрою, дозволяючи сайту коректно відображатися на десктопах, планшетах та смартфонах. Це включає зміну макетів (наприклад, перехід від багатостовпкового до одностовпкового), розмірів шрифтів, приховування/показ мобільного меню.

– `css/jquery-ui.min.css`: надає стандартні стилі для календаря jQuery UI `Datericker`, який використовується на сторінці бронювання.

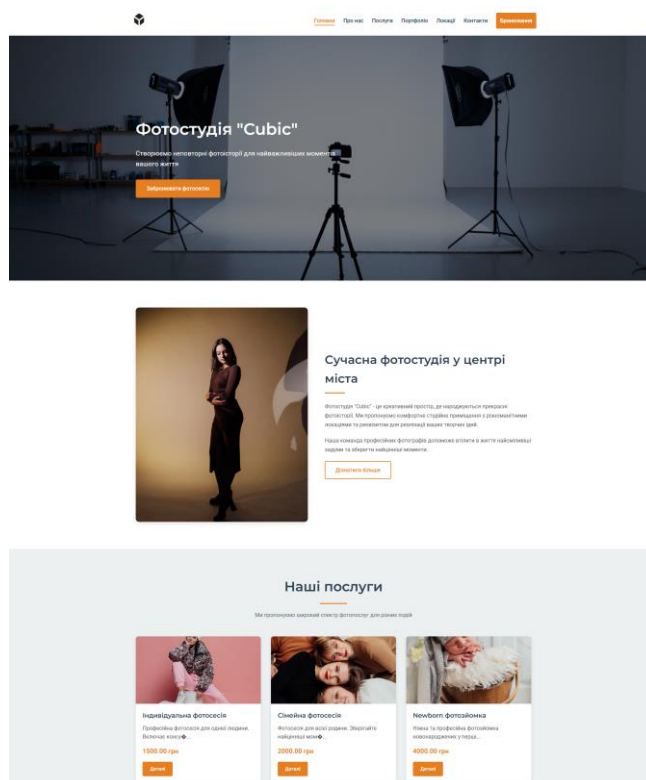


Рисунок 3.2 – Приклад відображення головної сторінки сайту "Cubic"

Інтерактивність (JavaScript та JQuery):

JavaScript, разом з бібліотекою JQuery, використовується для створення динамічної та інтерактивної поведінки на сторінці клієнта.

`js/main.js` (Загальний функціонал):

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		46

– мобільне меню: код реагує натискання по іконці "бургер" (.mobile-menu-toggle), додаючи/видаляючи клас active для навігаційного блоку (#main-nav), що, у свою чергу, керується CSS для анімації появи/зникнення меню.

– кнопка "Нагору" (#scroll-top): скрипт відстежує подію прокрутки вікна. Якщо сторінка прокручена нижче певного порогу, кнопці додається клас active (CSS робить її видимою). Клік по кнопці викликає анімоване плавне прокручування сторінки до верху.

– FAQ-аккордеон: натискання по елементу питання (.faq-question) викликає плавне розгортання/згортання відповідного блоку відповіді (.faq-answer) за допомогою методу slideToggle() з JQuery.

– ініціалізація простого слайдера: функція initSimpleSlider(sliderId) (якщо використовується для галереї студії на about.php) знаходить HTML-структуру слайдера та накладає обробники на кнопки "вперед" і "назад" для зміни CSS transform: translateX() у контейнера зі слайдами, забезпечуючи їх прокрутку.

js/gallery.js (Лайтбокс для портфоліо):

– при завантаженні сторінки скрипт збирає інформацію (шлях до зображення, назву, опис) з data-атрибутів кожного елемента портфоліо (.portfolio-trigger).

– клік по мініатюрі викликає функцію openLightbox(index), яка відображає модальне вікно лайтбоксу (#custom-lightbox).

– зображення (#lightbox-image) та підпис (#lightbox-caption) в лайтбоксі оновлюються відповідно до обраного елемента.

– реалізована навігація між зображеннями в лайтбоксі за допомогою кнопок "вперед" (.lightbox-next) та "назад" (.lightbox-prev), а також за допомогою клавіш зі стрілками на клавіатурі.

– закриття лайтбоксу можливе по кліку на кнопку закриття (.lightbox-close), на фон або по натисканню клавіші Escape.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		47

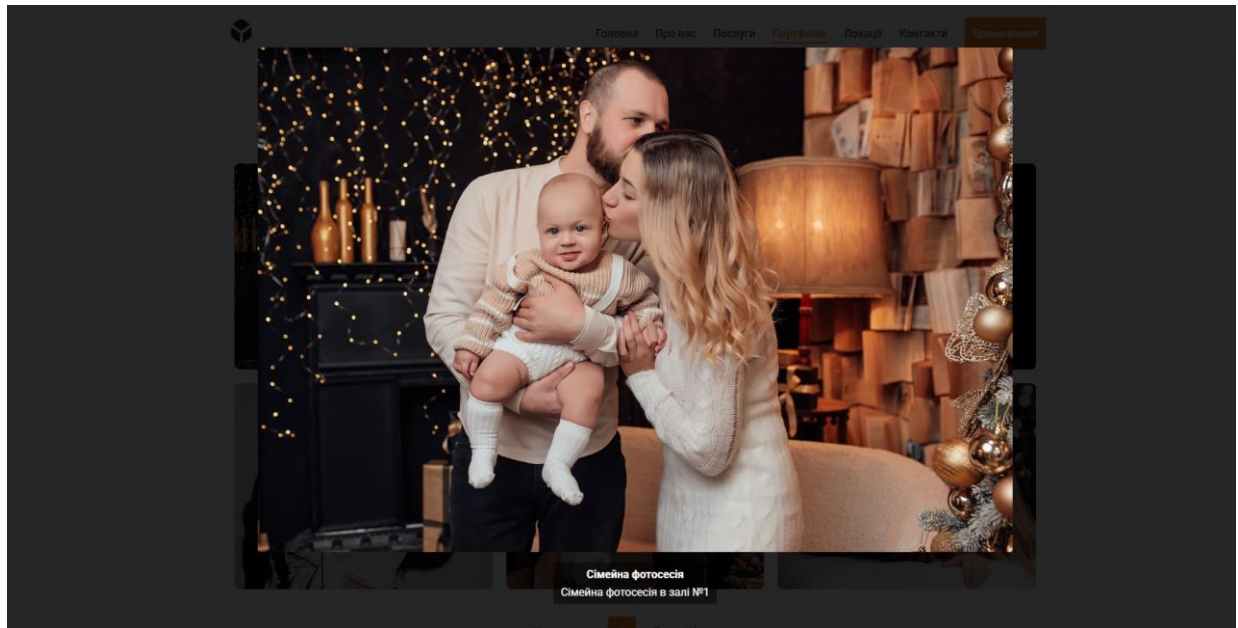


Рисунок 3.3 – Відображення роботи з портфоліо у лайтбоксі

js/booking.js (Логіка сторінки бронювання):

Ініціалізація календаря (jQuery UI DatePicker): для поля введення дати (#booking-date) активується віджет календаря. Налаштування включають:

- формат дати: yy-mm-dd;
- мінімальна доступна дата: minDate: 1 (завтрашній день);
- максимальна доступна дата: maxDate: "+3М" (три місяці наперед);
- локалізація: назви місяців та днів тижня українською;
- заборона вибору неділь: через функцію beforeShowDay;
- при виборі дати (onSelect) автоматично викликається функція loadAvailableTimeSlots() для завантаження доступного часу.

Динамічне завантаження доступного часу (loadAvailableTimeSlots):

- коли користувач обирає дату та локацію, ця функція відправляє асинхронний GET-запит (AJAX) до серверного скрипта includes/get_available_times.php;
- в запиті передаються обрана дата та ID локації;
- після отримання успішної JSON-відповіді від сервера (яка містить масив доступних годинних слотів), скрипт динамічно формує та вставляє опції

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		48

(<option>) у випадючий список вибору часу (#booking-time). Якщо доступних слотів немає, виводиться відповідне повідомлення.

The image shows a booking form with the following fields and elements:

- Ваше ім'я ***: Text input field.
- Електронна пошта ***: Text input field.
- Телефон ***: Text input field with a pre-filled value of +380.
- Тип фотосесії ***: Dropdown menu with the text "Оберіть тип фотосесії".
- Локація ***: Dropdown menu with the text "Оберіть локацію".
- Дата ***: Text input field with the text "Оберіть дату".
- Час ***: Dropdown menu with the text "Спочатку оберіть дату та локацію".
- Додаткові побажання**: Text area for additional comments.
- Забронювати**: Orange button at the bottom.

Рисунок 3.4 – Форма бронювання з активним календарем та динамічно завантаженим списком часу

Клієнтська валідація форми: перед відправкою форми на сервер, скрипт перевіряє заповнення всіх обов'язкових полів, коректність формату email та телефонного номера (відповідність шаблону +380XXXXXXXXXX). Якщо виявлені

помилки, під полями відображаються відповідні повідомлення, а поля підсвічуються класом error.

AJAX-відправка форми бронювання:

- при натисканні кнопки "Забронювати", стандартна відправка форми блокується (e.preventDefault());

- якщо клієнтська валідація пройдена, дані форми серіалізуються та відправляються асинхронним POST-запитом (AJAX) на серверний скрипт includes/process_booking.php;

- після отримання JSON-відповіді від сервера, користувачеві відображається повідомлення про успішне бронювання або про помилку. У разі успіху, форма автоматично очищається.

Обробка поля введення телефону: Спеціальний обробник події input для поля #client-phone забезпечує автоматичне збереження префіксу +380 та дозволяє вводити лише 9 цифр після нього, фільтруючи інші символи.

Адаптивний дизайн:

Ключовим аспектом розробки клієнтської частини є забезпечення адаптивності. Це досягається за допомогою:

- гнучких макетів (Flexible Grids/Layouts): використання відносних одиниць (% , vw , vh) та технік CSS Flexbox/Grid для побудови сіток, які адаптуються до ширини екрану;

- медіа-запитів (@media в css/responsive.css): застосування різних наборів CSS-правил для різних діапазонів ширини екрану. Це дозволяє, наприклад, змінювати кількість колонок, розміри шрифтів, приховувати або трансформувати елементи (наприклад, навігаційне меню перетворюється на "бургер");

- адаптивних зображень: забезпечення масштабування зображень без втрати пропорцій (max-width: 100%, height: auto);

- тестування на різних пристроях: перевірка відображення та функціональності на популярних смартфонах, планшетах та десктопах.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		50

3.3 Реалізація адміністративної панелі

Адміністративна панель вебсайту "Cubic" розроблена для надання авторизованим користувачам (адміністраторам) інструментів для управління контентом сайту та обробки бронювань. Вона розташована в окремій директорії /admin/ для логічного відокремлення від публічної частини сайту.

Структура та основні файли адміністративної панелі:

Перелік основних файлів та їх призначення:

- admin/index.php: сторінка входу до адміністративної панелі, містить форму для введення логіна та пароля;
- admin/dashboard.php: головна сторінка ("панель приладів") після успішної авторизації. Відображає ключову статистику (наприклад, кількість нових бронювань, загальну кількість послуг) та швидкі посилання на основні розділи управління;
- admin/manage-bookings.php: сторінка для перегляду списку всіх бронювань з можливістю сортування, фільтрації (якщо реалізовано) та пагінації;
- admin/booking-details.php: сторінка для перегляду детальної інформації про конкретне бронювання та зміни його статусу або додавання коментаря;
- admin/manage-services.php: інтерфейс для управління послугами (додавання, редагування, видалення, зміна статусу активності);
- admin/manage-portfolio.php: інтерфейс для управління роботами в портфоліо (додавання, видалення);
- admin/manage-locations.php: інтерфейс для управління локаціями студії;
- admin/settings.php: сторінка для налаштувань профілю адміністратора (зміна облікових даних);
- admin/logout.php: скрипт для завершення сесії адміністратора та виходу з системи;
- admin/includes/admin_header.php та admin/includes/admin_footer.php: шаблонні файли, що містять загальну HTML-розмітку для всіх сторінок

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		51

адміністративної панелі (наприклад, шапку, бічну навігаційну панель, підключення стилів та скриптів);

- `css/admin.css`: файл стилів, спеціально розроблений для оформлення адміністративної панелі.

Авторизація адміністратора:

Доступ до функціоналу адміністративної панелі суворо обмежений та вимагає автентифікації.

Сторінка входу (`admin/index.php`):

- при відправці форми входу, PHP-скрипт отримує введені `username` та `password`;

- виконується SQL-запит до таблиці `users` для пошуку користувача з відповідним `username`. Для безпеки використовується підготовлений запит;

- якщо користувач знайдений, введений пароль порівнюється із захешованим паролем, збереженим у базі даних, за допомогою PHP-функції `password_verify()`.

```
// ... (після отримання $user з бази даних) ...
if ($user && password_verify($_POST['password'],
$user['password'])) {
    // Пароль вірний, починаємо сесію
    session_start(); // Або session_regenerate_id(true); для
безпеки
    $_SESSION['admin_logged_in'] = true;
    $_SESSION['admin_user_id'] = $user['id'];
    $_SESSION['admin_username'] = $user['username'];
    header('Location: dashboard.php'); // Перенаправлення на
головну панель
    exit;
} else {
    $error = "Невірний логін або пароль."; // Повідомлення про
помилку
}
```

У разі успішної автентифікації, встановлюються сесійні змінні (`$_SESSION ['admin_logged_in'], $_SESSION ['admin_user_id'], $_SESSION ['admin_username']`), і адміністратор перенаправляється на `dashboard.php`.

Перевірка авторизації на сторінках (`checkAdminAuth()`):

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		52

– кожна сторінка в директорії /admin/ (окрім index.php) на початку свого виконання викликає функцію checkAdminAuth() з файлу includes/functions.php;

– ця функція перевіряє, чи встановлена та чи є дійсною сесійна змінна \$_SESSION ['admin_logged_in']. Якщо адміністратор не авторизований, відбувається автоматичне перенаправлення на сторінку входу admin/index.php.

Вихід з системи (admin/logout.php):

– цей скрипт знищує всі дані поточної сесії за допомогою session_destroy() та перенаправляє адміністратора на сторінку входу.

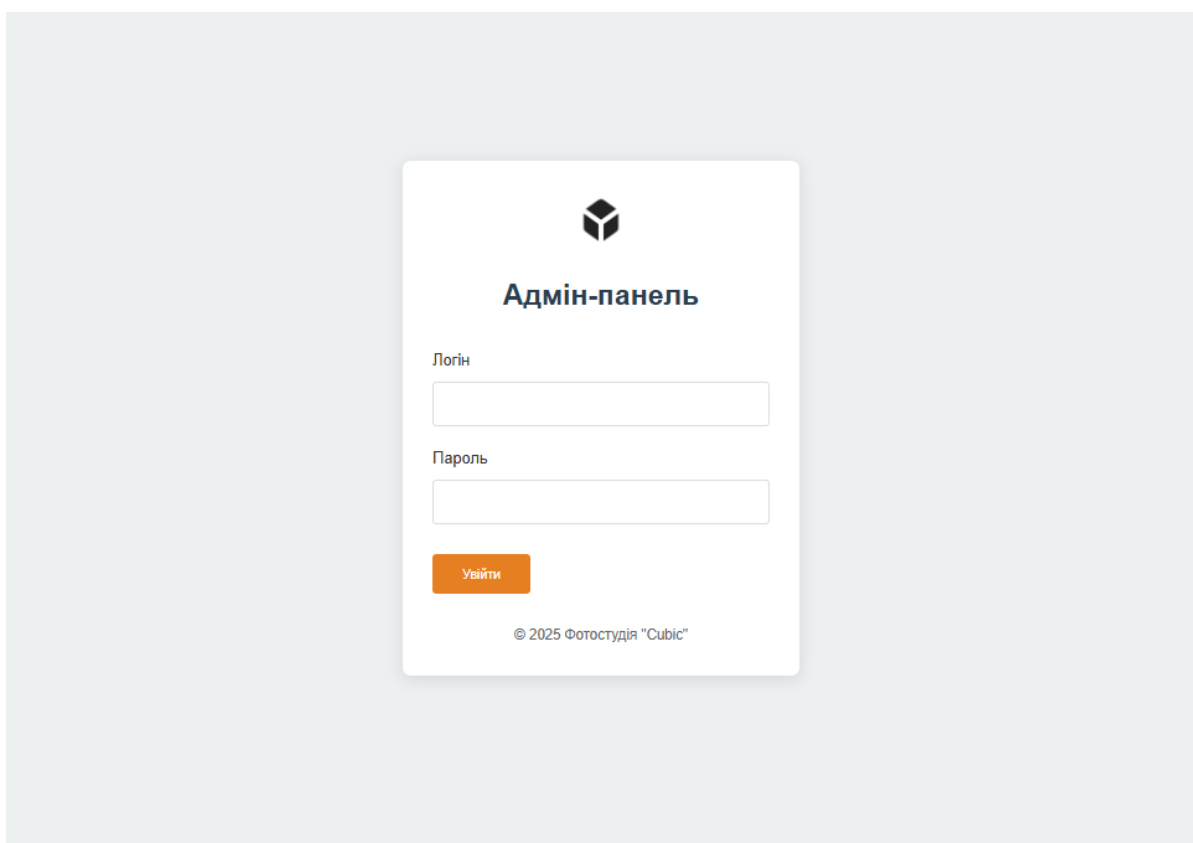


Рисунок 3.5 – Сторінка входу до адміністративної панелі сайту "Cubic"

Управління контентом (CRUD-операції):

Адміністративна панель надає інтерфейси для виконання операцій створення (Create), читання (Read), оновлення (Update) та видалення (Delete) для основних сутностей сайту.

Управління послугами (admin/manage-services.php):

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		53

1. Читання (Read): список усіх послуг (включаючи активні та неактивні) виводиться з бази даних (за допомогою функції `getAllServices($conn, null, false)`) у вигляді таблиці. Таблиця містить ID, мініатюру зображення, назву, ціну, тривалість та поточний статус активності.

2. Створення (Create): на сторінці присутня HTML-форма для додавання нової послуги. Форма містить поля для назви, опису, ціни, тривалості та завантаження файлу зображення (`<input type="file">`). При відправці форми (`isset($_POST['add_service'])`), PHP-скрипт отримує дані, проводить їх валідацію. Здійснюється обробка завантаженого файлу: перевірка типу та розміру, генерація унікального імені для файлу (наприклад, за допомогою `uniqid()`) та переміщення його до директорії `images/services/` за допомогою `move_uploaded_file()`. Якщо файл не завантажено, може використовуватися зображення за замовчуванням. Після успішної обробки даних та файлу, виконується підготовлений SQL-запит `INSERT` для додавання нової послуги до таблиці `services`.

3. Оновлення (Update): для кожної послуги у списку передбачена кнопка "Редагувати", яка зазвичай відкриває модальне вікно (або перенаправляє на окрему сторінку) з формою, попередньо заповненою поточними даними послуги. Адміністратор може змінити будь-які дані, включаючи заміну зображення. При збереженні змін (`isset($_POST['edit_service'])`), PHP-скрипт аналогічно проводить валідацію, обробляє можливе завантаження нового зображення (при цьому старе зображення, якщо воно не є дефолтним, може видалятися з сервера для економії місця) та виконує підготовлений SQL-запит `UPDATE` для таблиці `services` за ID послуги.

4. Видалення (Delete): кнопка "Видалити" для кожної послуги. Перед фактичним видаленням з бази даних (`isset($_GET['delete'])`), виконується перевірка, чи не пов'язана дана послуга з існуючими бронюваннями в таблиці `bookings`. Якщо зв'язки є, видалення блокується, і адміністратору виводиться відповідне повідомлення. Якщо зв'язків немає, виконується SQL-запит `DELETE`.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		54

Також видаляється файл зображення послуги з сервера (якщо воно не є дефолтним).

5. Зміна статусу активності: для кожної послуги є кнопка "Активувати"/"Деактивувати" (isset(\$_GET['toggle_active'])). При її натисканні, PHP-скрипт змінює значення поля active (0 або 1) для відповідного запису в таблиці services за допомогою SQL-запиту UPDATE. Це впливає на відображення послуги на клієнтській частині сайту.

Рисунок 3.6 – Сторінка управління послугами в адміністративній панелі "Cubic"

Управління портфоліо (admin/manage-portfolio.php):

1. Створення (Create): форма для завантаження нових робіт, що включає поля для назви, опису (необов'язково) та вибору файлу зображення. Реалізовано JavaScript для попереднього перегляду обраного зображення перед завантаженням, а також можливість завантаження через drag-and-drop для покращення юзабіліті. Серверна частина обробляє завантаження файлу аналогічно до зображень послуг та зберігає інформацію в таблиці portfolio.

2. Читання (Read): відображення всіх завантажених робіт у вигляді сітки (grid) з мініатюрами зображень та назвами.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		55

3. Видалення (Delete): можливість видалення обраної роботи з портфолію. При видаленні запису з бази даних також видаляється відповідний файл зображення з сервера.

Управління локаціями (admin/manage-locations.php):

– функціонал повністю аналогічний управлінню послугами: читання списку локацій, додавання нових локацій (з назвою, описом, завантаженням зображення), редагування існуючих локацій через модальні вікна, видалення локацій (з перевіркою на використання в бронюваннях) та зміна статусу їх активності.

Управління бронюваннями (admin/manage-bookings.php, admin/booking-details.php):

Цей блок є критично важливим для операційної діяльності студії.

Перегляд списку бронювань (admin/manage-bookings.php):

1. Відображає всі запити на бронювання у вигляді таблиці. Кожен рядок містить ключову інформацію: ID бронювання, ім'я клієнта, контакти (email, телефон), обрану послугу, локацію, дату та час, поточний статус (pending, confirmed, cancelled) та дату створення запиту.

2. Реалізовано функціонал сортування таблиці за різними стовпцями (наприклад, за датою бронювання, статусом, ID) та пагінація для зручного перегляду великої кількості записів.

3. Для кожного бронювання є можливість швидкої зміни статусу за допомогою випадаючого списку та кнопки "Оновити" безпосередньо в рядку таблиці, що відправляє POST-запит на цю ж сторінку для обробки.

4. Також є посилання для переходу на сторінку з детальною інформацією про бронювання.

Деталі бронювання (admin/booking-details.php):

1. При переході на цю сторінку (з передачею ID бронювання через GET-параметр), завантажуються повна інформація про конкретне бронювання, включаючи всі дані клієнта, деталі обраної послуги та локації, коментарі, залишені клієнтом при бронюванні, та коментар адміністратора.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		56

2. Адміністратор має можливість змінити статус бронювання (наприклад, з "pending" на "confirmed" після отримання передоплати, або на 'cancelled' у випадку якщо клієнт передумав, і відмінив бронювання).

3. Передбачено поле для додавання або редагування коментаря адміністратора до бронювання (наприклад, для внутрішніх нотаток або інформації для інших співробітників).

4. Також є кнопка "Написати клієнту", яка генерує посилання з темою та основними даними бронювання, і одразу можна написати клієнту на особисту пошту.

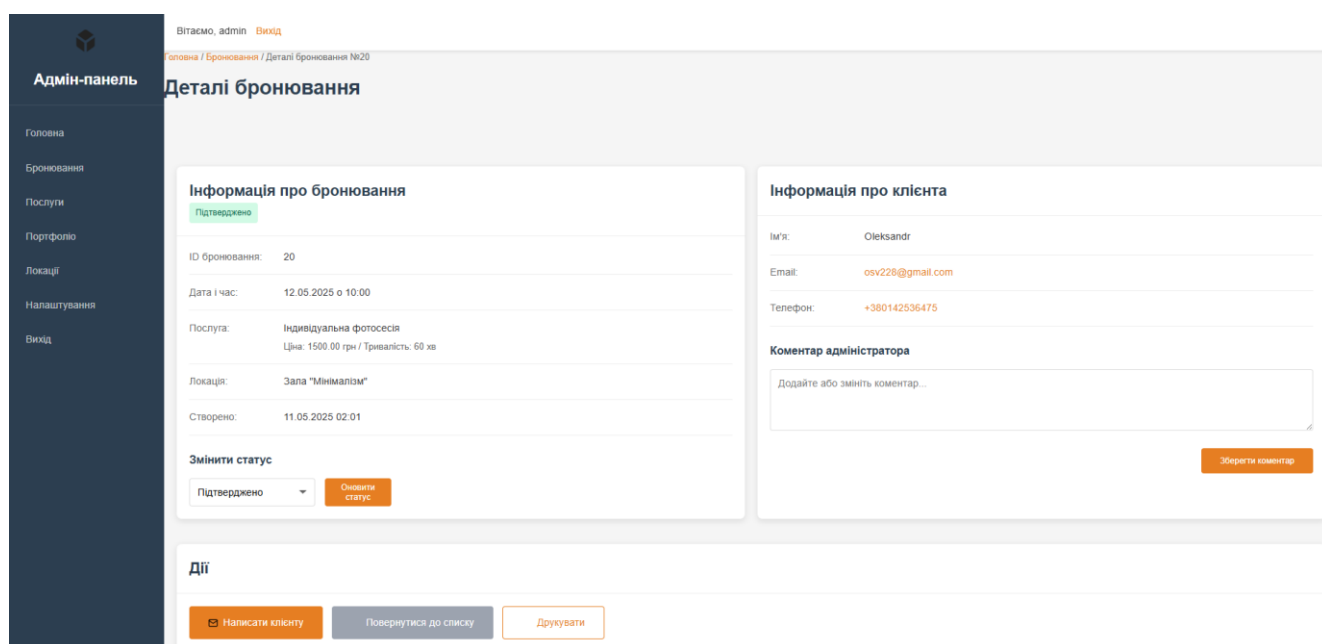


Рисунок 3.7 – Сторінка перегляду деталей бронювання в адміністративній панелі "Cubic"

Налаштування профілю адміністратора (admin/settings.php):

1. Ця сторінка дозволяє авторизованому адміністратору керувати власними обліковими даними.

2. Надається можливість змінити ім'я користувача (логін) та адресу електронної пошти. При цьому може виконуватися перевірка на унікальність нового логіна.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		57

3. Реалізовано функціонал зміни пароля. Для цього адміністратор повинен ввести свій поточний пароль (для підтвердження особи), а потім новий пароль та його підтвердження. Новий пароль перед збереженням до бази даних хешується за допомогою password_hash().

4. Також на цій сторінці може відображатися базова системна інформація (версія PHP, MySQL), що може бути корисною для діагностики.

Всі операції, що передбачають зміну даних в базі (додавання, редагування, видалення), виконуються через HTTP POST-запити для більшої безпеки (у порівнянні з GET для таких операцій). Після успішного виконання операції або у випадку помилки, адміністратору виводяться відповідні флеш-повідомлення (наприклад, "Послугу успішно додано" або "Помилка при видаленні локації"). Для взаємодії з базою даних використовуються підготовлені SQL-запити для запобігання SQL-ін'єкціям.

3.4 Тестування ефективності вебсайту та аналіз результатів

Після завершення створення вебресурсу для фотостудії "Cubic" було здійснено його перевірку з метою оцінки ефективності, зручності навігації та відповідності сучасним стандартам веброзробки. Для аналізу використовувався інструмент Lighthouse – спеціалізоване розширення для браузера Chrome, яке виконує автоматизовану діагностику вебсторінок.

Цей інструмент дозволяє комплексно оцінити сайт за кількома напрямками:

1) Продуктивність – визначає швидкість завантаження, реагування на дії користувача та ефективність використання ресурсів.

2) Доступність – перевіряє, наскільки сайт зручний для людей із різними обмеженнями, зокрема для користувачів, які застосовують допоміжні технології.

3) Найкращі практики – оцінює дотримання технічних стандартів та безпекових рекомендацій.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		58

4) Оптимізація для пошукових систем (SEO) – аналізує потенціал сайту бути правильно індексованим у пошукових системах.

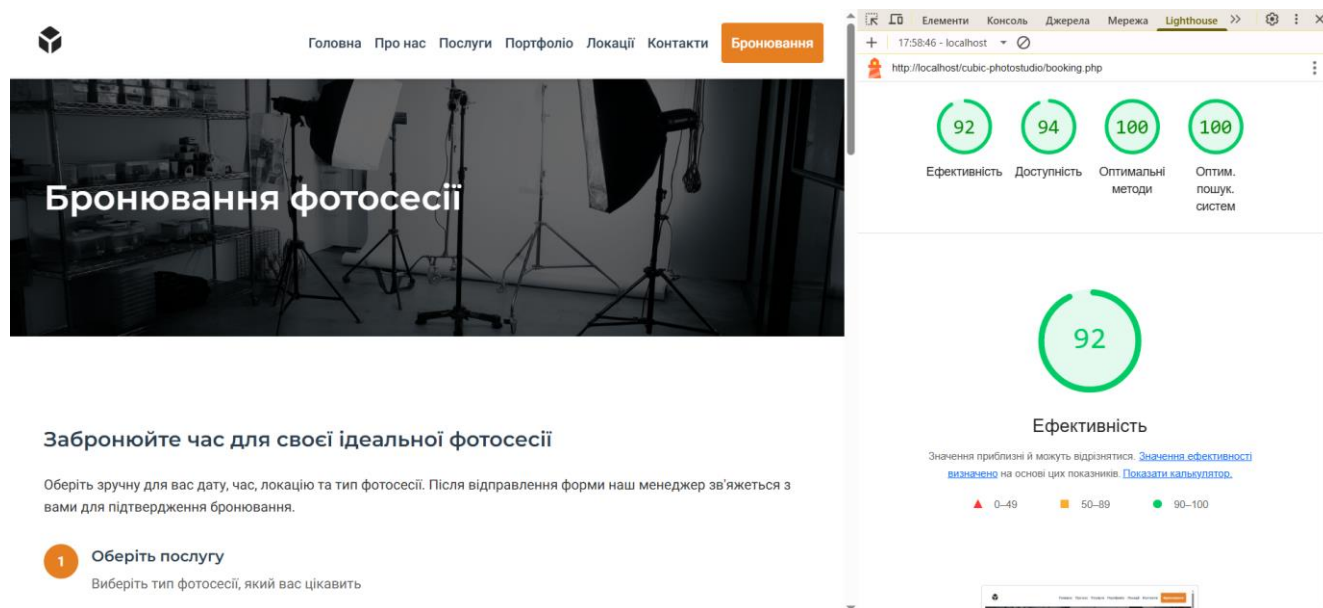


Рисунок 3.8 – Звіт Lighthouse для сторінки «Бронювання» вебсайту "Cubic"

Аналіз охопив усі основні сторінки ресурсу. Зведені результати наведено нижче:

Таблиця 3.1 – Продуктивність вебсайту

Сторінка	Ефективність	Доступність	Оптимальні методи	SEO
Головна (index.php)	81	89	100	100
Про нас (about.php)	86	93	100	100
Послуги (services.php)	90	92	100	100
Портфоліо (portfolio.php)	81	92	100	95
Локації (locations.php)	85	93	100	98
Контакти	90	93	100	100
Бронювання (booking.php)	92	94	100	100

Як видно з таблиці, всі сторінки продемонстрували високі показники. Особливо варто відзначити:

- високу доступність усіх сторінок, що свідчить про відповідність елементів інтерфейсу вимогам до інклюзивності;
- відмінну реалізацію технічних рішень (всюди 100 балів у категорії «Оптимальні методи»), що забезпечує стабільну та безпечну роботу сайту;
- гарну оптимізацію для пошукових систем (SEO), що покращує видимість ресурсу в результатах пошуку.

Найнижчий показник продуктивності (81 балів) було зафіксовано на сторінках портфоліо і головній. Це пояснюється наявністю великої кількості зображень, що збільшують час завантаження. Для підвищення ефективності рекомендовано реалізувати:

- поступове завантаження зображень (lazy loading);
- компресію та оптимізацію медіафайлів;
- кешування статичних ресурсів.

Проведене тестування вебсайту фотостудії "Cubic" засвідчило його відповідність сучасним стандартам у сферах продуктивності, доступності, безпеки та оптимізації. Це гарантує користувачам зручний та безпечний досвід взаємодії з ресурсом, а також сприяє покращенню його позицій у пошукових системах

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		60

ВИСНОВКИ

В межах дипломної роботи було створено функціональний вебресурс для фотостудії "Cubic", який реалізований із застосуванням HTML, CSS, JavaScript, JQuery, PHP і MySQL. Завдання з розробки ефективної онлайн-платформи для презентації студії успішно виконано.

Перший розділ присвячений вивченню тематики – розглянуто особливості діяльності фотостудій і роль сайту у взаємодії з клієнтами. Проведено аналіз сайтів-конкурентів, що допомогло сформуванню уявлення про найпоширеніші рішення, їх переваги та недоліки. Це дозволило визначити основні технічні й функціональні вимоги до майбутнього проєкту.

У другому розділі висвітлено процес проєктування: сформульовано вимоги до клієнтської частини і панелі адміністрування, запропоновано структуру сайту, створено UML-діаграми (діаграми варіантів використання та компонентів), які демонструють логіку системи та взаємодію користувачів із нею. Обґрунтовано вибір технологій, виходячи з їхньої доцільності у межах цього проєкту. Окрему увагу приділено розробці структури бази даних, що містить таблиці для адміністраторів, послуг, локацій, портфолію та бронювання.

Третій розділ описує реалізацію сайту на практиці. Докладно розглянуто створення бекенду на PHP: підключення до бази даних, функції для роботи з контентом, обробка запитів на бронювання та перевірка доступного часу. У частині фронтенду описано застосування HTML, CSS і JavaScript з JQuery для візуального оформлення та інтерактивності: динамічного вивантаження інформації, перевірки форм і реалізації галереї. Також представлено інструменти адміністративної частини, які забезпечують керування вмістом сайту та заявками на бронювання. На завершення проведено тестування функціональності, зручності використання, адаптивності під різні пристрої та коректної роботи в різних браузерах.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		61

Результатом є вебсайт, який:

- має сучасне оформлення, коректно відображається на різних екранах;
- містить всю необхідну інформацію про фотостудію, послуги, локації та приклади робіт;
- дозволяє клієнтам зручно бронювати послуги онлайн з перевіркою доступного часу;
- оснащений панеллю адміністрування для контролю за контентом і бронюваннями;
- розроблений з урахуванням вимог безпеки та стабільної роботи.

Розроблений вебсайт є готовим програмним продуктом, який може бути розгорнутий та використаний фотостудією "Cubic" для ведення бізнесу в Інтернеті. Подальший розвиток проєкту може включати розробку особистого кабінету для клієнтів, і, як наслідок, інтеграцію з платіжними системами для онлайн-оплати бронювань.

Виконана робота підтверджує можливість ефективного застосування обраного стеку технологій для створення сучасних та функціональних вебрішень для підприємств сфери послуг.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		62

ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Основи будування сайтів. В. Манако, Д.Манако, О.Данилова, О.Войченко, Видавництво: Шкільний світ , 2006.
2. Nixon R. Learning PHP, MySQL & JavaScript : with jQuery, CSS & HTML5 / Robin Nixon. – 5th ed. – Sebastopol : O’Reilly Media, 2018. – 832 p.
3. Інформаційні технології в бізнесі. Частина 1 : навч. посіб. І. Б. Шевчук [та ін.]; за заг. ред. І. Б. Шевчук; Львів. нац. ун-т ім. І. Франка. – Львів : ННБК «АТБ», 2020. – 455 с. – ISBN 978-966-2042-41-4.
4. Idea Digital Agency URL: <https://ideadigital.agency/blog/chomu-malomu-biznesu-potriben-sait/> (дата звернення: 17.04.2025).
5. HTML Tutorial URL: <https://www.w3schools.com/html/default.asp> (дата звернення: 11.04.2025).
6. CSS Tutorial URL: <https://www.w3schools.com/css/default.asp> (дата звернення: 11.04.2025).
7. JS Tutorial URL: <https://www.w3schools.com/js/default.asp> (дата звернення: 12.04.2025).
8. PHP Tutorial. URL: <https://www.w3schools.com/php/> (дата звернення: 12.04.2025).
9. SQL Tutorial URL: <https://www.w3schools.com/sql/default.asp> (дата звернення: 13.04.2025).
10. Mozilla Developer Network. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/uk/docs/Web/HTML> (дата звернення: 23.04.2025).
11. Strimpel J., Najim M. Building Isomorphic JavaScript Apps. O’Reilly Media, Inc. 2016. 210 с. URL: <https://learning.oreilly.com/library/view/building-isomorphic-javascript/9781491932926/ch01.html> (дата звернення: 17.04.2025).
12. Mesbah A., van Deursen A. Migrating Multi-page Web Applications to Single-page Ajax Interfaces. 2006. 14 с. URL: <https://arxiv.org/ftp/cs/papers/0610/0610094.pdf> (дата звернення: 25.04.2025).

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		63

13. Robson E., Freeman E. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages / E. Robson, E. Freeman. – 2nd ed. – Sebastopol : O'Reilly Media, 2012. – 762 с. – ISBN 978-0596159900

14. Benedetti R., Cranley R. Head First jQuery: A Brain-Friendly Guide / R. Benedetti, R. Cranley. – 1st ed. – Sebastopol : O'Reilly Media, 2011. – 544 p. – ISBN 978-1-4493-9321-2.

15. Duckett J. HTML and CSS: Design and Build Websites. Indianapolis: «John Wiley & Sons, Inc.», 2011. 490 с.

					БР.КІ-27.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		64

ДОДАТКИ

Лістинг структури головної сторінки (index.php)

```

<?php
$page_title = "Головна";
include 'includes/header.php'; // Підключення шапки сайту
require_once 'includes/db_connect.php'; // Підключення до БД
require_once 'includes/functions.php'; // Функції для роботи з
даними

// Отримання даних для відображення
$featured_services = getAllServices($conn, 3); // 3 основні
послуги
$latest_portfolio = getPortfolio($conn, null, 6); // Останні 6
робіт
?>

<!-- Головний слайдер/герой-секція -->
<section class="hero" style="background-image:
url('images/slider/main.jpg');">
    <div class="container">
        <div class="hero-content">
            <h1>Фотостудія "Cubic"</h1>
            <p>Створюємо неповторні фотоісторії для найважливіших
моментів вашого життя</p>
            <a href="booking.php" class="btn">Забронювати
фотосесію</a>
        </div>
    </div>
</section>

<!-- Секція про студію (короткий опис) -->
<section class="about-section">
    <div class="container">
        <div class="about-content">
            <div class="about-image">
                
            </div>
            <div class="about-text">
                <h2>Сучасна фотостудія у центрі міста</h2>
                <p>Фотостудія "Cubic" - це креативний
простір...</p>
                <a href="about.php" class="btn btn-
outline">Дізнатися більше</a>
            </div>
        </div>
    </div>
</section>

```

```

<!-- Секція послуг -->
<section class="services">
  <div class="container">
    <div class="section-title">
      <h2>Наші послуги</h2>
      <p>Ми пропонуємо широкий спектр фотопослуг для різних
подій</p>
    </div>
    <div class="services-grid">
      <?php foreach ($featured_services as $service): ?>
        <div class="service-card">
          <div class="service-img">
            ">
          </div>
          <div class="service-content">
            <h3><?php echo
htmlspecialchars($service['name']); ?></h3>
            <p><?php echo
htmlspecialchars(substr($service['description'], 0, 100)) . '...';
?></p>
            <div class="service-price"><?php echo
htmlspecialchars($service['price']); ?> грн</div>
            <a href="services.php#service-<?php echo
$service['id']; ?>" class="btn btn-sm">Деталі</a>
          </div>
        </div>
      <?php endforeach; ?>
    </div>
    <div class="section-footer">
      <a href="services.php" class="btn">Всі послуги</a>
    </div>
  </div>
</section>

<!-- Секція портфоліо (останні роботи) -->
<section class="portfolio">
  <div class="container">
    <div class="section-title">
      <h2>Наші роботи</h2>
      <p>Перегляньте наше портфоліо та оцініть якість наших
фотографій</p>
    </div>
    <div class="portfolio-grid">
      <?php foreach ($latest_portfolio as $item): ?>
        <div class="portfolio-item js-lightbox-trigger"
data-index="<?php echo $item['id']; /* Або індекс
в масиві, якщо потрібно */ ?>"
data-src="<?php echo
htmlspecialchars($item['image']); ?>"

```

```

        data-title="<?php echo
htmlspecialchars($item['title']); ?>"
        data-description="<?php echo
htmlspecialchars($item['description'] ?? ''); ?>"
        "
        <div class="portfolio-overlay">
            <h3><?php echo
htmlspecialchars($item['title']); ?></h3>
            <span class="zoom-icon">+</span>
        </div>
    </div>
    <?php endforeach; ?>
</div>
<div class="section-footer">
    <a href="portfolio.php" class="btn">Все портфоліо</a>
</div>
</div>
</section>

<!-- Секція відгуків (приклад статичних) -->
<section class="testimonials">
    <!-- ... HTML для відгуків ... -->
</section>

<!-- Секція СТА (Call to Action) -->
<section class="cta-section">
    <!-- ... HTML для СТА ... -->
</section>

<?php include 'includes/footer.php'; // Підключення підвалу сайту
?>

```

Лістинг файлу підключення до бази даних (includes/db_connect.php)

```

<?php
// Параметри підключення до бази даних
$host = 'localhost';
$username = 'root'; // Замініть на ваше ім'я користувача БД
$password = 'root'; // Замініть на ваш пароль БД
$database = 'cubic_db'; // Назва вашої бази даних

// Створення з'єднання
$conn = mysqli_connect($host, $username, $password, $database);

// Перевірка з'єднання
if (!$conn) {

```

```

    // У робочому середовищі краще логувати помилку, а не виводити
    її користувачу
    error_log("Помилка підключення до БД: " .
mysqli_connect_error());
    die("Вибачте, виникла технічна проблема. Спробуйте пізніше.");
}

// Встановлення кодування для з'єднання
if (!mysqli_set_charset($conn, "utf8")) {
    error_log("Помилка встановлення кодування utf8: " .
mysqli_error($conn));
}
?>

```

Лістинг ключових функцій для роботи з даними (includes/functions.php)

```

<?php
// Функція для отримання всіх послуг
function getAllServices($conn, $limit = null, $activeOnly = true)
{
    $sql = "SELECT * FROM services";
    if ($activeOnly) {
        $sql .= " WHERE active = 1";
    }
    $sql .= " ORDER BY id ASC";
    if ($limit !== null && is_numeric($limit)) {
        $sql .= " LIMIT " . (int)$limit;
    }
    $result = mysqli_query($conn, $sql);
    return $result ? mysqli_fetch_all($result, MYSQLI_ASSOC) : [];
}

// Функція для отримання однієї послуги за ID
function getServiceById($conn, $id) {
    $id = (int)$id;
    $stmt = mysqli_prepare($conn, "SELECT * FROM services WHERE id
= ?");
    mysqli_stmt_bind_param($stmt, "i", $id);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);
    $service = $result ? mysqli_fetch_assoc($result) : null;
    mysqli_stmt_close($stmt);
    return $service;
}

// Функція для отримання портфоліо
function getPortfolio($conn, $category = null, $limit = null) {
    $sql = "SELECT * FROM portfolio";

```

```

// Фільтрація за категорією, якщо вона використовується
// if ($category !== null) {
//     $sql .= " WHERE category = '" .
mysqli_real_escape_string($conn, $category) . "'";
// }
$sql .= " ORDER BY created_at DESC";
if ($limit !== null && is_numeric($limit)) {
    $sql .= " LIMIT " . (int)$limit;
}
$result = mysqli_query($conn, $sql);
return $result ? mysqli_fetch_all($result, MYSQLI_ASSOC) : [];
}

// Функція для отримання локацій
function getLocations($conn, $activeOnly = true) {
    $sql = "SELECT * FROM locations";
    if ($activeOnly) {
        $sql .= " WHERE active = 1";
    }
    $sql .= " ORDER BY id ASC";
    $result = mysqli_query($conn, $sql);
    return $result ? mysqli_fetch_all($result, MYSQLI_ASSOC) : [];
}

// Функція для перевірки перетину бронювань
function checkBookingOverlap($conn, $date, $startTime, $duration,
    $locationId, $excludeBookingId = null) {
    // ... (код функції з попередньої відповіді) ...
    // Важливо: ця функція має точно розраховувати часові
інтервали
    // $newStartTimestamp = strtotime($date . ' ' . $startTime);
    // $newEndTimestamp = $newStartTimestamp + ($duration * 60);
    // ... (цикл по існуючим бронюванням з аналогічним
розрахунком) ...
    // if ($newStartTimestamp < $existingEndTimestamp &&
    $newEndTimestamp > $existingStartTimestamp) return true;
    // return false;
}

// Функція для отримання доступних годинних слотів
function getAvailableHourlySlots($conn, $date, $location_id) {
    // ... (код функції з попередньої відповіді, що блокує зайняті
годинні слоти) ...
}

// Функція для створення бронювання
function createBooking($conn, $name, $email, $phone, $service_id,
    $location_id, $date, $time, $notes = '') {
    $stmt = mysqli_prepare($conn, "INSERT INTO bookings
(client_name, client_email, client_phone, service_id, location_id,

```

```

booking_date, booking_time, notes) VALUES (?, ?, ?, ?, ?, ?, ?,
?)"");

    mysqli_stmt_bind_param($stmt, "sssiiss", $name, $email,
$phone, $service_id, $location_id, $date, $time, $notes);
    $success = mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
    return $success;
}

// Функція для валідації вводу
function validateInput($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data, ENT_QUOTES, 'UTF-8');
    return $data;
}

// Функція перевірки авторизації адміністратора
function checkAdminAuth() {
    if (session_status() == PHP_SESSION_NONE) {
        session_start();
    }
    if (!isset($_SESSION['admin_logged_in']) ||
$_SESSION['admin_logged_in'] !== true) {
        header('Location: index.php'); // Або login.php, якщо є
окрема сторінка
        exit;
    }
}
?>

```

Лістинг обробки форми бронювання (includes/process_booking.php)

```

<?php
$page_title = "Бронювання";
include 'includes/header.php';
require_once 'includes/db_connect.php';
require_once 'includes/functions.php';

// Отримуємо список послуг та локацій (тільки активні)
$services = getAllServices($conn, null, true);
$locations = getLocations($conn, true);

// Перевірка GET-параметрів для попереднього вибору
$selected_service = isset($_GET['service']) ?
(int)$_GET['service'] : '';

```

```

$selected_location = isset($_GET['location']) ?
(int)$_GET['location'] : '';
?>

<section class="page-header" style="background-image:
url('images/headers/booking-header.jpg');">
  <div class="container">
    <h1>Бронювання фотосесії</h1>
  </div>
</section>

<section class="booking-section">
  <div class="container">
    <div class="booking-content">
      <div class="booking-info">
        <h2>Забронюйте час для своєї ідеальної
фотосесії</h2>
        <p>Оберіть зручну для вас дату, час, локацію та
тип фотосесії...</p>
        <!-- ... booking-steps ... -->
      </div>
      <div class="booking-form-container">
        <form id="booking-form"
action="includes/process_booking.php" method="post">
          <div class="form-group">
            <label for="client-name">Ваше ім'я
*</label>
            <input type="text" id="client-name"
name="client_name" required>
          </div>
          <div class="form-group">
            <label for="client-email">Електронна пошта
*</label>
            <input type="email" id="client-email"
name="client_email" required>
          </div>
          <div class="form-group">
            <label for="client-phone">Телефон
*</label>
            <input type="tel" id="client-phone"
name="client_phone" required>
          </div>
          <div class="form-group">
            <label for="service">Тип фотосесії
*</label>
            <select id="service" name="service_id"
required>
              <option value="">Оберіть тип
фотосесії</option>
              <?php foreach ($services as $service):
?>

```

```

        <option value="<?php echo
$service['id']; ?>" <?php if($selected_service == $service['id'])
echo 'selected'; ?>>
                <?php echo
htmlspecialchars($service['name']); ?> - <?php echo
htmlspecialchars($service['price']); ?> грн.
                </option>
        <?php endforeach; ?>
    </select>
</div>
<div class="form-group">
    <label for="location">Локація *</label>
    <select id="location" name="location_id"
required>
        <option value="">Оберіть
локацію</option>
        <?php foreach ($locations as
$location): ?>
            <option value="<?php echo
$location['id']; ?>" <?php if($selected_location ==
$location['id']) echo 'selected'; ?>>
                <?php echo
htmlspecialchars($location['name']); ?>
            </option>
        <?php endforeach; ?>
    </select>
</div>
<div class="form-group">
    <label for="booking-date">Дата *</label>
    <input type="text" id="booking-date"
name="booking_date" readonly required placeholder="Оберіть дату">
</div>
<div class="form-group">
    <label for="booking-time">Час *</label>
    <select id="booking-time"
name="booking_time" required disabled>
        <option value="">Спочатку оберіть дату
та локацію</option>
    </select>
</div>
<div class="form-group">
    <label for="notes">Додаткові
побажання</label>
    <textarea id="notes" name="notes"
rows="4"></textarea>
</div>
<div class="form-submit">
    <button type="submit" class="btn btn-
primary">Забронювати</button>
</div>
</form>

```

```

        </div>
    </div>
</div>
</section>

<!-- ... секція booking-faq ... -->
<!-- jQuery UI вже має бути підключений в header.php або тут -->
<!-- <script src="js/libs/jquery-ui.min.js"></script> -->
<script src="js/booking.js"></script> <!-- Скрипт для динаміки
форми -->

<?php include 'includes/footer.php'; ?>

```

Лістинг обробки форми бронювання (includes/process_booking.php)

```

<?php
require_once 'db_connect.php';
require_once 'functions.php';

header('Content-Type: application/json'); // Важливо для AJAX

if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
    echo json_encode(['success' => false, 'message' =>
'Неправильний метод запиту']);
    exit;
}

// Отримання та валідація даних
$client_name = isset($_POST['client_name']) ?
validateInput($_POST['client_name']) : '';
// ... (аналогічно для всіх полів) ...
$service_id = isset($_POST['service_id']) ?
(int)$_POST['service_id'] : 0;
$location_id = isset($_POST['location_id']) ?
(int)$_POST['location_id'] : 0;
$booking_date = isset($_POST['booking_date']) ?
validateInput($_POST['booking_date']) : '';
$booking_time = isset($_POST['booking_time']) ?
validateInput($_POST['booking_time']) : '';
$notes = isset($_POST['notes']) ? validateInput($_POST['notes']) :
'';

// Базова перевірка на заповненість
if (empty($client_name) || empty($client_email) || /* ...інші
поля... */ empty($booking_time)) {
    echo json_encode(['success' => false, 'message' => 'Будь
ласка, заповніть всі обов\язкові поля']);
}

```

```

    exit;
}

// Перевірка email
if (!filter_var($client_email, FILTER_VALIDATE_EMAIL)) {
    echo json_encode(['success' => false, 'message' => 'Введіть
коректну електронну пошту']);
    exit;
}
// Отримання тривалості послуги
$service = getServiceById($conn, $service_id);
if (!$service || !isset($service['duration'])) {
    echo json_encode(['success' => false, 'message' => 'Обрана
послуга недійсна або не має тривалості']);
    exit;
}
$duration = (int)$service['duration'];

// Точна перевірка на перетин інтервалів
if (checkBookingOverlap($conn, $booking_date, $booking_time,
$duration, $location_id)) {
    echo json_encode([
        'success' => false,
        'message' => 'На жаль, обраний час (з урахуванням
тривалості послуги ' . $duration . ' хв) вже зайнятий. Будь ласка,
оновіть сторінку та виберіть інший час.'
    ]);
    exit;
}

// Створення бронювання
if (createBooking($conn, $client_name, $client_email,
$client_phone, $service_id, $location_id, $booking_date,
$booking_time, $notes)) {
    echo json_encode([
        'success' => true,
        'message' => 'Бронювання успішно створено! Ми зв\'яжемося
з вами для підтвердження. Сторінка зараз оновиться.'
    ]);
    // Тут можна додати логіку відправки email-повідомлень
} else {
    error_log("Помилка створення бронювання в БД: " .
mysqli_error($conn));
    echo json_encode(['success' => false, 'message' => 'Сталася
помилка під час створення бронювання.']);
}

mysqli_close($conn);
?>

```

Лістинг JavaScript для динамічної форми бронювання (js/booking.js)

```

$(document).ready(function() {
    'use strict';

    // Ініціалізація Datepicker
    if ($.fn.datepicker && $('#booking-date').length) {
        $('#booking-date').datepicker({
            dateFormat: 'yy-mm-dd',
            minDate: 1, // Завтра
            maxDate: '+3M', // На 3 місяці вперед
            firstDay: 1, // Понеділок
            // ... (локалізація для datepicker) ...
            beforeShowDay: function(date) {
                return [date.getDay() !== 0, '']; // Неділя -
вихідний
            },
            onSelect: function(dateText) {
                loadAvailableTimeSlots(dateText);
            }
        });
    }

    // Завантаження доступних часових слотів
    function loadAvailableTimeSlots(date) {
        const locationId = $('#location').val();
        const $bookingTimeSelect = $('#booking-time');

        if (!locationId || !date) {
            $bookingTimeSelect.html('<option value="">Спочатку
оберіть дату та локацію</option>').prop('disabled', true);
            return;
        }

        $bookingTimeSelect.html('<option
value="">Завантаження...</option>').prop('disabled', true);

        $.ajax({
            url: 'includes/get_available_times.php', // Скрипт для
отримання часу
            method: 'GET',
            data: { date: date, location_id: locationId },
            dataType: 'json',
            success: function(response) {
                let timesHtml = '<option value="">Оберіть
час</option>';
                if (response.success && response.times &&
response.times.length > 0) {
                    response.times.forEach(function(time) {

```

```

        timesHtml += `<option
value="${time}">${time.substring(0,5)}</option>`;
        });
        $bookingTimeSelect.prop('disabled', false);
    } else {
        timesHtml = '<option value="">Немає доступного
часу</option>';
        $bookingTimeSelect.prop('disabled', true);
    }
    $bookingTimeSelect.html(timesHtml);
},
error: function() {
    $bookingTimeSelect.html('<option value="">Помилка
завантаження</option>').prop('disabled', true);
}
});
}

// Обробник зміни локації
$('#location').change(function() {
    const date = $('#booking-date').val();
    if (date) {
        loadAvailableTimeSlots(date);
    }
});

// Обробка відправки форми бронювання
$('#booking-form').submit(function(e) {
    e.preventDefault();
    const form = $(this);
    const submitButton = form.find('button[type="submit"]');
    // ... (клієнтська валідація полів) ...

    submitButton.prop('disabled', true).text('Бронювання...');

    $.ajax({
        url: form.attr('action'),
        method: form.attr('method'),
        data: form.serialize(),
        dataType: 'json',
        success: function(response) {
            alert(response.message);
            if (response.success) {
                location.reload(); // Перезавантаження
сторінки
            }
        },
        error: function() {
            alert('Помилка з\'єднання. Спробуйте пізніше.');
```

```

        submitButton.prop('disabled',
false).text('Забронювати');
    }
    });
});
});

```

Лістинг JavaScript для лайтбоксу/галереї (js/gallery.js)

```

document.addEventListener('DOMContentLoaded', function() {
    'use strict';

    const lightboxTriggers = document.querySelectorAll('.js-
lightbox-trigger');
    const lightbox = document.getElementById('custom-lightbox');

    if (!lightbox || lightboxTriggers.length === 0) return; //
Якщо немає чого показувати

    const lightboxImage = document.getElementById('lightbox-
image');
    const lightboxCaption = document.getElementById('lightbox-
caption');
    const closeButton = lightbox.querySelector('.lightbox-close');
    const prevButton = lightbox.querySelector('.lightbox-prev');
    const nextButton = lightbox.querySelector('.lightbox-next');

    let currentIndex = 0;
    let imageData = [];

    lightboxTriggers.forEach((item, index) => {
        imageData.push({
            src: item.dataset.src,
            title: item.dataset.title,
            description: item.dataset.description
        });
        item.dataset.galleryIndex = index; // Зберігаємо індекс
для кліку
    });

    function openLightbox(index) {
        if (index < 0 || index >= imageData.length) return;
        currentIndex = index;
        const imgData = imageData[currentIndex];

        lightboxImage.src = imgData.src;
        lightboxImage.alt = imgData.title || '';
    }
}

```

```

        lightboxCaption.innerHTML = `${imgData.title ||
''}</strong>` +
                                (imgData.description ?
`<br>${imgData.description}` : '');
        updateNavigationButtons();
        lightbox.style.display = 'flex';
        document.body.classList.add('lightbox-open');
    }

    function closeLightbox() {
        lightbox.style.display = 'none';
        document.body.classList.remove('lightbox-open');
    }

    function updateNavigationButtons() {
        prevButton.style.display = currentIndex > 0 ? 'block'
: 'none';
        nextButton.style.display = currentIndex <
imagesData.length - 1 ? 'block' : 'none';
    }

    function showNextImage() {
        if (currentIndex < imagesData.length - 1)
openLightbox(currentIndex + 1);
    }
    function showPrevImage() {
        if (currentIndex > 0) openLightbox(currentIndex
- 1);
    }

    lightboxTriggers.forEach(item => {
        item.addEventListener('click', function(e) {
            if (e.target.closest('a')) e.preventDefault();
            openLightbox(parseInt(this.dataset.galleryIndex, 10));
        });
    });

    closeButton.addEventListener('click', closeLightbox);
    prevButton.addEventListener('click', showPrevImage);
    nextButton.addEventListener('click', showNextImage);
    lightbox.addEventListener('click', (e) => { if (e.target ===
lightbox) closeLightbox(); });

    document.addEventListener('keydown', (e) => {
        if (lightbox.style.display === 'flex') {
            if (e.key === 'Escape') closeLightbox();
            if (e.key === 'ArrowRight') showNextImage();
            if (e.key === 'ArrowLeft') showPrevImage();
        }
    }); });

```

Лістинг структури адмін-панелі (admin/dashboard.php)

```

<?php
session_start();
require_once '../includes/functions.php';
checkAdminAuth(); // Перевірка авторизації
require_once '../includes/db_connect.php';

// Отримання статистики для дашборду
$total_bookings = mysqli_fetch_assoc(mysqli_query($conn, "SELECT
COUNT(*) as total FROM bookings"))['total'];
// ... (інша статистика) ...
$recent_bookings = mysqli_fetch_all(mysqli_query($conn, "SELECT
b.*, s.name as service_name FROM bookings b JOIN services s ON

b.service_id = s.id ORDER BY b.created_at DESC LIMIT 5"),
MYSQLI_ASSOC);

include 'includes/admin_header.php'; // Шапка адмін-панелі
?>

<div class="content-header">
    <h1>Панель керування</h1>
</div>

<div class="content">
    <div class="dashboard-stats">
        <div class="stat-card">
            <h3>Всього бронювань</h3>
            <div class="stat-number"><?php echo $total_bookings;
?></div>
        </div>
        <!-- ... інші картки статистики ... -->
    </div>

    <div class="card mt-4">
        <div class="card-header"><h2>Останні бронювання</h2></div>
        <div class="card-body">
            <div class="table-responsive">
                <table class="table">
                    <thead>

<tr><th>Клієнт</th><th>Послуга</th><th>Дата</th><th>Статус</th><th>Дії</th></tr>
                    </thead>
                    <tbody>
                        <?php if (count($recent_bookings) > 0): ?>

```

```

<?php foreach ($recent_bookings as $booking): ?>
    <tr>
        <td><?php echo
htmlspecialchars($booking['client_name']); ?></td>
        <td><?php echo
htmlspecialchars($booking['service_name']); ?></td>
        <td><?php echo date('d.m.Y
H:i', strtotime($booking['booking_date'] . ' ' .
$booking['booking_time'])); ?></td>
        <td><span class="status
status-<?php echo $booking['status']; ?>"><?php echo
$booking['status']; ?></span></td>
        <td>
            <a href="booking-
details.php?id=<?php echo $booking['id']; ?>" class="btn-sm btn-
view">Детали</a>
        </td>
    </tr>
    <?php endforeach; ?>
    <?php else: ?>
        <tr><td colspan="5" class="text-
center">Немає бронювань</td></tr>
    <?php endif; ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<?php include 'includes/admin_footer.php'; // Підвал адмін-панелі

```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: *Розробка інформаційного web-сайту для фотостудії Cubic засобами HTML, CSS, JavaScript, JQuery та PHP*

Обсяг пояснювальної записки 64 аркушів:

11 таблиць;

15 рисунків;

1 додаток.

Дата завершення роботи: *02 червня 2025р.*

Підпис студента- _____*Вірченко О.С.*