

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

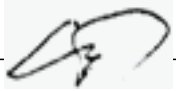
Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

—  — проф. Приходько С.Б.

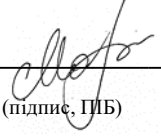
«___» _____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

**на тему: Розробка вебзастосунку для представництва та взаємодії з
клієнтами дизайн-студії «Creative_DS»**

Виконав: студент групи 4151

—  — Марчук В.С.
(підпис, ПІБ)

Керівник роботи:

Доцент кафедри ПЗАС, к. ф.-м. н., доцент

(посада, науковий ступень вчене звання)

—  — Латанська Л. О.
(підпис, ПІБ)

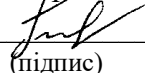
Миколаїв – 2026 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

 доц. Макарова Л.М.
 (підпис)

« 07 » 04 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту _____ Марчуку Віктору Сергійовичу _____
 (Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Керівник роботи Латанська Людмила Олексіївна

Затверджені наказом ректора №275-уч від 07.04.2026 р.

2. Термін подання роботи: 01.06.2026 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
 Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1 Титульний слайд, 2 Мета, завдання та об'єкт роботи, 3 Існуючі рішення, 4 Функціональні вимоги до вебзастосунку дизайн-студії «Creative_DS», 5 Діаграма варіантів використання вебзастосунку дизайн-студії «Creative_DS», 6 Архітектура вебзастосунку дизайн-студії «Creative_DS», 7 Діаграми діяльності вебзастосунку дизайн-студії «Creative_DS», 8 Інформаційна модель вебзастосунку дизайн-студії «Creative_DS», 9 Діаграма класів вебзастосунку дизайн-студії «Creative_DS», 10 Діаграми послідовності вебзастосунку дизайн-студії «Creative_DS», 11 Логічна модель даних вебзастосунку дизайн-студії «Creative_DS», 12 Засоби розробки вебзастосунку дизайн-студії «Creative_DS», 13 Результати розробки вебзастосунку дизайн-студії «Creative_DS», 14 Висновки, 15 Заклучний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

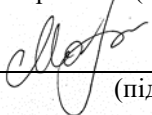
7. Дата видачі завдання 07.04.2026 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	30.03.2026	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	06.04.2026	ПП*
3.	Аналіз вимог до ПЗ	13.04.2026	ПП*
4.	Розробка архітектури ПЗ	20.04.2026	
5.	Розробка проекту ПЗ	04.05.2026	
6.	Реалізація та тестування ПЗ	11.05.2026	
7.	Підготовка розділу Результати розробки ПЗ	15.05.2026	
8.	Підготовка розділу з охорони праці	18.05.2026	ПП*
9.	Підготовка розділу ВИСНОВКИ	22.05.2026	
10.	Оформлення списку використаних джерел та додатків	25.05.2026	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	01.06.2026	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	05.06.2026	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	08.06.2026	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	15.06.2026	

* - за результатами переддипломної практики (ПП), яка була з 02.02.2026 до 22.03.2026 р.

Студент


 (підпис)

Марчук В.С.

(ПБ)

Керівник роботи


 (підпис)

Латанська Л. О.

(ПБ)

АНОТАЦІЯ

Дана кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS». Комплексність задачі полягає у необхідності реалізації взаємопов'язаних функціональних компонентів, забезпеченні ефективної взаємодії користувачів із системою, різноманітності вимог до програмного забезпечення та потребі у створенні сучасного адаптивного інтерфейсу. Невизначеність умов обумовлюється можливими змінами вимог користувачів, динамічним розвитком вебтехнологій та необхідністю забезпечення подальшого розширення функціональних можливостей системи.

Розроблюваний вебзастосунок призначений для представлення діяльності дизайн-студії «Creative_DS», демонстрації портфоліо виконаних робіт, забезпечення взаємодії з потенційними клієнтами, організації зворотного зв'язку та спрощення процесу комунікації між студією та користувачами вебресурсу.

Робота включає в себе: аналіз предметної галузі дизайн-студії «Creative_DS», дослідження існуючих програм-аналогів та обґрунтування необхідності розробки вебзастосунку, постановку задачі на розробку програмного забезпечення, аналіз вимог, проектування та реалізацію вебзастосунку, результати розробки, розділ з охорони праці та додатки.

Робота виконана на 90 сторінці друкованого тексту, містить 15 рисунків, 4 таблиці, 5 додатків та список використаних джерел із 15 найменувань.

ABSTRACT

This qualification work is dedicated to solving a practical problem of software engineering, characterized by complexity and uncertainty of conditions, namely, the development of a web application for representation and interaction with clients of the design studio "Creative_DS". The complexity of the problem lies in the need to implement interconnected functional components, ensuring effective user interaction with the system, the diversity of software requirements and the need to create a modern adaptive interface. The uncertainty of the conditions is due to possible changes in user requirements, the dynamic development of web technologies and the need to ensure further expansion of the system's functionality.

The developed web application is intended to present the activities of the design studio "Creative_DS", demonstrate a portfolio of completed works, ensure interaction with potential clients, organize feedback and simplify the communication process between the studio and users of the web resource.

The work includes: analysis of the subject area of the design studio "Creative_DS", research of existing similar programs and justification of the need to develop a web application, formulation of a task for software development, analysis of requirements, design and implementation of a web application, development results, section on labor protection and appendices.

The work is completed on 90 pages of printed text, contains 15 figures, 4 tables, 5 appendices and a list of sources used from 15 names.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ ПРЕДСТАВНИЦТВА ТА ВЗАЄМОДІЇ З КЛІЄНТАМИ ДИЗАЙН-СТУДІЇ «CREATIVE_DS»	10
1.1 Аналіз практичної задачі інженерії програмного забезпечення.....	10
1.2 Аналіз існуючих програмних рішень.....	12
1.3 Постановка задачі на розробку вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»	16
2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ПРЕДСТАВНИЦТВА ТА ВЗАЄМОДІЇ З КЛІЄНТАМИ ДИЗАЙН-СТУДІЇ «CREATIVE_DS»	20
2.1 Аналіз вимог до вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».....	20
2.1.1 Побудова моделі варіантів використання	20
2.1.2 Специфікації варіантів використання	22
2.2 Архітектура вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».....	25
2.3 Проєкт вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»	30
2.3.1 Ескізний проєкт вебзастосунку	30
2.3.2 Технічний проєкт вебсервісу	36
2.3.3 Робочий проєкт програмного забезпечення.....	45
3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ПРЕДСТАВНИЦТВА ТА ВЗАЄМОДІЇ З КЛІЄНТАМИ ДИЗАЙН-СТУДІЇ «CREATIVE_DS»	52
4 ОХОРОНА ПРАЦІ	54
4.1 Загальні принципи охорони праці	54
4.2. Аналіз небезпечних і шкідливих факторів при роботі з ПК	54

4.3. Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів	55
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
Додаток А Технічне завдання на розробку вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»	62
Додаток Б Код вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».....	67
Додаток В Опис вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».....	81
Додаток Г Інструкція користувача вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»	85
Додаток Д Програма та методика випробування вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».....	89

ВСТУП

У сучасних умовах активного розвитку цифрових технологій та мережі Інтернет важливого значення набуває представництво компаній у вебпросторі. Особливо це стосується дизайн-студій, діяльність яких безпосередньо пов'язана з візуальним представленням робіт, взаємодією з клієнтами та формуванням власного іміджу. Для ефективної організації комунікації з потенційними замовниками, демонстрації портфоліо та надання інформації про послуги виникає необхідність у використанні сучасних вебзастосунків.

Використання вебзастосунку для представництва дизайн-студії дозволяє автоматизувати процес взаємодії з клієнтами, забезпечити зручний доступ до інформації про діяльність студії, реалізовані проєкти та перелік послуг. Крім того, вебзастосунок сприяє покращенню комунікації між студією та користувачами, спрощує процес подання заявок і забезпечує швидкий зворотний зв'язок.

Розуміння потреб сучасних користувачів та необхідності ефективного представлення дизайн-студії у мережі Інтернет є важливим етапом при розробці вебзастосунку «Creative_DS». Даний вебзастосунок має забезпечити демонстрацію портфоліо дизайн-студії, надання інформації про послуги, підтримку взаємодії з клієнтами та створення зручного середовища для комунікації між користувачами та представниками студії.

Задача розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» характеризується комплексністю та невизначеністю умов. Комплексність задачі полягає у необхідності реалізації взаємопов'язаних функціональних компонентів, забезпеченні ефективної взаємодії користувачів із системою, різноманітності вимог до програмного забезпечення та потребі у створенні сучасного адаптивного інтерфейсу. Невизначеність умов обумовлюється можливими змінами вимог користувачів,

динамічним розвитком вебтехнологій та необхідністю забезпечення подальшого розширення функціональних можливостей системи.

Аналіз існуючих вебресурсів дизайн-студій показав, що значна частина аналогічних систем має надлишковий або, навпаки, обмежений функціонал, складний інтерфейс користувача чи недостатньо зручні засоби взаємодії з клієнтами. Крім того, деякі вебресурси не забезпечують достатньо ефективного представлення портфоліо або не підтримують сучасні засоби комунікації з користувачами.

Після аналізу існуючих аналогів та дослідження практичної задачі інженерії програмного забезпечення можна зробити висновок про доцільність створення власного вебзастосунку для дизайн-студії «Creative_DS», який буде орієнтований на забезпечення зручної взаємодії з клієнтами, сучасне представлення діяльності студії та ефективне керування контентом.

Метою даної кваліфікаційної роботи є розв'язання практичної задачі інженерії програмного забезпечення, а саме розробка сучасного, зручного та функціонального вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

Для досягнення поставленої мети необхідно виконати такі завдання:

- здійснити аналіз практичної задачі інженерії програмного забезпечення;
- дослідити існуючі програмні аналоги та обґрунтувати доцільність розробки власного вебзастосунку;
- визначити вимоги до програмного продукту та сформулювати технічне завдання;
- спроектувати архітектуру вебзастосунку;
- реалізувати програмне забезпечення відповідно до поставлених вимог;
- виконати перевірку та тестування роботи вебзастосунку;
- виконати документування розробки.

Об'єктом даної кваліфікаційної роботи є процес розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ ПРЕДСТАВНИЦТВА ТА ВЗАЄМОДІЇ З КЛІЄНТАМИ ДИЗАЙН-СТУДІЇ «CREATIVE_DS»

1.1 Аналіз практичної задачі інженерії програмного забезпечення

У сучасних умовах значна кількість дизайн-студій використовує вебзастосунки для представлення власної діяльності, демонстрації портфоліо, надання інформації про послуги та взаємодії з клієнтами. Особливо актуальними такі вебресурси є для студій, що спеціалізуються на дизайні інтер'єру, ландшафтному дизайні, проєктуванні меблів та оформленні фасадів будівель. Вебзастосунок дозволяє не лише презентувати власні роботи, а й забезпечувати ефективну комунікацію із замовниками, формувати імідж студії та залучати нових клієнтів.

Дизайн-студія «Creative_DS» знаходиться на етапі становлення та розвитку власного бренду. Основною проблемою на початковому етапі діяльності є недостатній рівень представленості студії в інформаційному просторі та відсутність єдиного цифрового середовища для демонстрації послуг і комунікації з клієнтами. Використання лише соціальних мереж або месенджерів не забезпечує достатнього рівня структурованості інформації, професійного подання портфоліо та ефективного управління зверненнями користувачів. Саме тому виникає задача створення спеціалізованого вебзастосунку, який стане основним інструментом онлайн-присутності дизайн-студії «Creative_DS».

Дана практична задача інженерії програмного забезпечення характеризується комплексністю та невизначеністю умов. Комплексність полягає у необхідності реалізації взаємопов'язаних функціональних компонентів, забезпеченні ефективної взаємодії користувачів із системою,

різноманітності вимог до програмного забезпечення та потребі у створенні сучасного інтерфейсу. Невизначеність умов обумовлюється можливими змінами вимог користувачів, динамічним розвитком вебтехнологій та необхідністю забезпечення подальшого розширення функціональних можливостей системи.

Для презентації своєї діяльності та пошуку потенційних клієнтів на початковому етапі необхідний презентаційний вебзастосунок, що забезпечить представлення діяльності дизайн-студії, демонстрацію дизайнерських робіт, надання контактної інформації та організацію зворотного зв'язку з потенційними клієнтами. Вебзастосунок повинен бути зручним у використанні та мати привабливий інтерфейс, оскільки саме через онлайн-платформи потенційні замовники отримують перше враження про рівень професіоналізму та творчі можливості компанії. Також вебзастосунок повинен мати гнучку архітектуру, що забезпечить простоту його подальшого розширення.

Під час аналізу практичної задачі були визначені основні категорії користувачів системи:

- адміністратор студії;
- потенційні клієнти.

Адміністратор відповідає за наповнення вебзастосунку контентом, оновлення портфолію, редагування інформації про послуги та обробку звернень користувачів. Потенційні клієнти використовують вебзастосунок для ознайомлення з діяльністю студії, перегляду прикладів робіт, отримання контактної інформації, підтримання комунікації зі студією та отримання актуальної інформації про нові послуги або проєкти.

Для обґрунтування необхідності власної розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» необхідно провести аналіз існуючих вебресурсів і програмних рішень, що використовуються у сфері дизайну інтер'єру, ландшафтного дизайну, проєктування меблів та оформлення фасадів.

1.2 Аналіз існуючих програмних рішень

На сьогодні значна кількість дизайн-студій використовує власні вебресурси та програмні рішення для представлення діяльності й взаємодії з клієнтами. Розглянемо деякі з них:

– Дизайн-студія «Гуменюк Design» [1]

На рисунку 1.1 представлена головна сторінка вебзастосунку дизайн-студії «Гуменюк Design».

Основні функціональні можливості:

- представлення інформації про студію;
- демонстрація портфоліо дизайн-проектів;
- опис послуг;
- форма зворотного зв'язку;
- контактна інформація;
- адаптивний інтерфейс користувача.

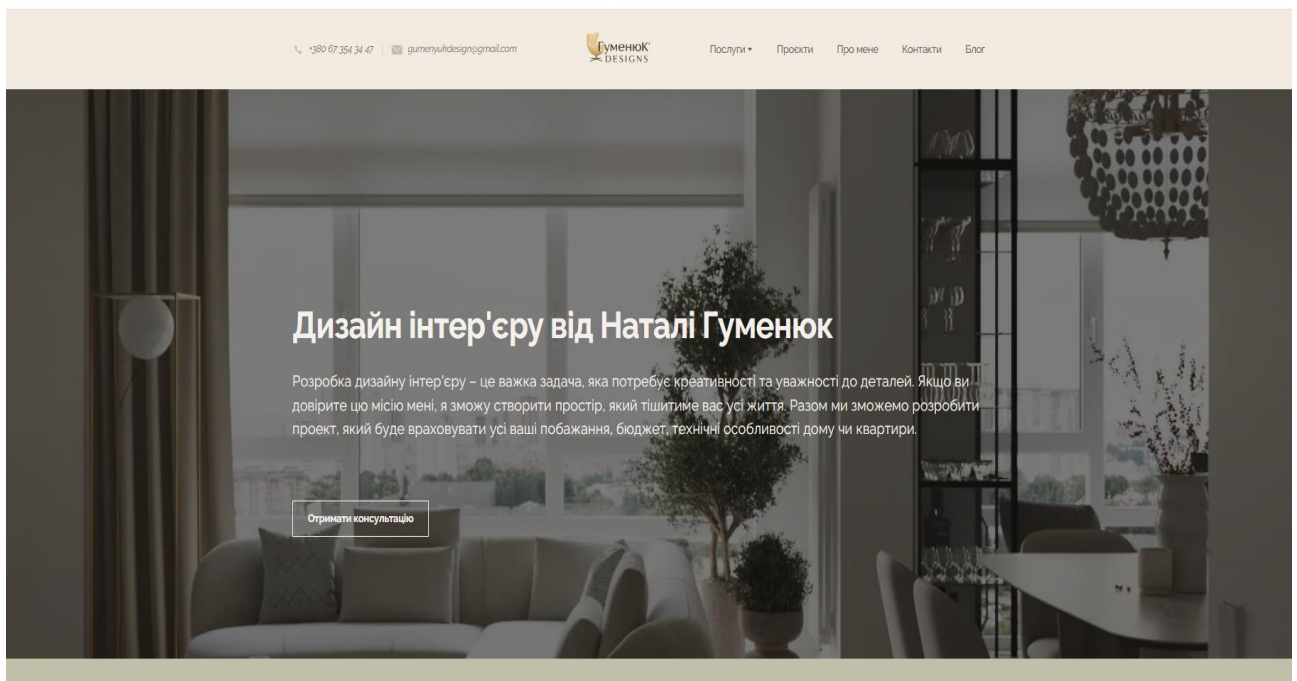


Рисунок 1.1 – Головна сторінка вебзастосунку дизайн-студії «Гуменюк Design»

Переваги:

- сучасний дизайн вебресурсу;

- зручна навігація;
- якісне представлення графічного контенту;
- адаптивність для мобільних пристроїв.

Недоліки:

- відсутність особистого кабінету клієнта;
- обмежені можливості взаємодії з користувачами;
- відсутність системи керування заявками.

– Дизайн-студія "Alex Deco" [2]

На рисунку 1.2 представлена головна сторінка вебзастосунку дизайн-студії "Alex Deco".

Основні функціональні можливості:

- демонстрація реалізованих проєктів;
- представлення послуг студії;
- контактна форма;
- презентація портфоліо;
- адаптивний інтерфейс;
- відгуки.

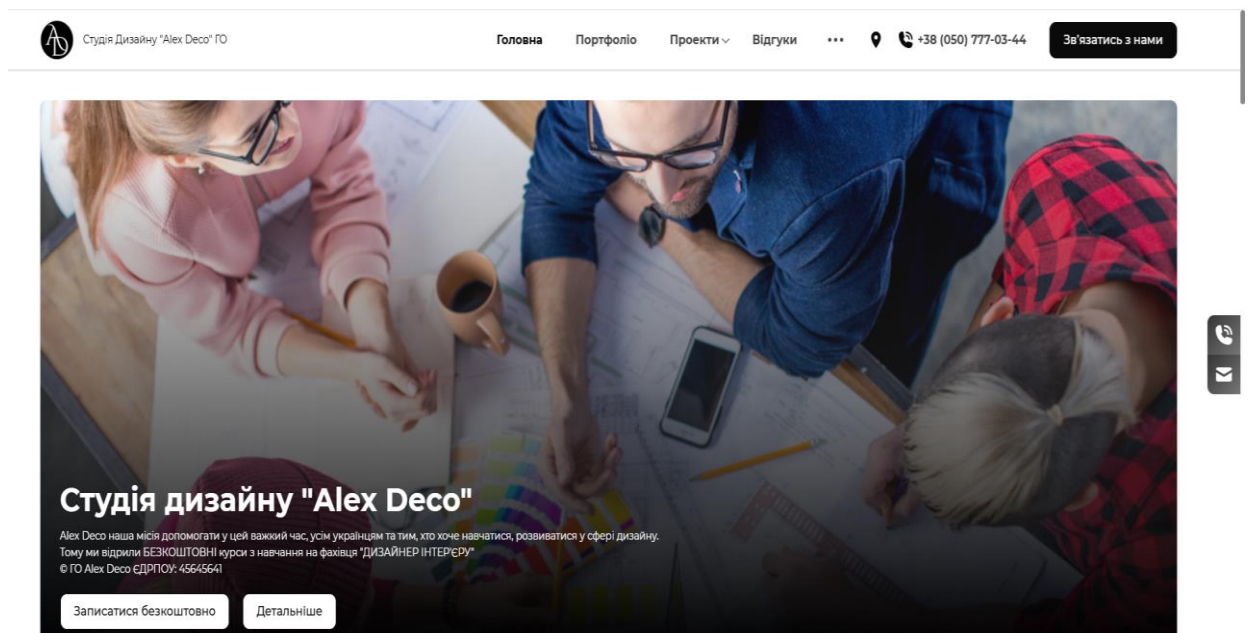


Рисунок 1.2 – Головна сторінка вебзастосунку дизайн-студії "Alex Deco"

Переваги:

- сучасне оформлення сторінок;
- структуроване подання інформації;
- зручне представлення портфоліо;
- адаптивність інтерфейсу.

Недоліки:

- відсутність системи реєстрації користувачів;
- немає персоналізованої взаємодії з клієнтами;
- обмежені можливості комунікації;
- відсутність функціоналу керування зверненнями.

– Дизайн-студія «Design» Романа Москаленка [3]

На рисунку 1.3 представлена головна сторінка вебзастосунку дизайн-студії «Design».



Рисунок 1.3 – Головна сторінка вебзастосунку дизайн-студії «Design»

Основні функціональні можливості:

- представлення дизайн-проектів;
- демонстрація портфоліо;
- опис послуг студії;
- форма зворотного зв'язку;

- контактна інформація;
- адаптивний вебінтерфейс.

Переваги:

- сучасний стиль оформлення;
- якісна візуалізація проєктів;
- адаптивність для різних пристроїв;
- зручна структура навігації.

Недоліки:

- відсутність особистих кабінетів користувачів;
- немає інтерактивної взаємодії з клієнтами;
- відсутній функціонал керування заявками;
- обмежені можливості адміністрування контенту.

Проведений аналіз показав, що на сьогодні існує значна кількість вебсайтів та вебзастосунків для дизайн-студій, які спеціалізуються на розробці дизайну інтер'єру, ландшафтного дизайну, проєктуванні меблів, оформленні фасадів будівель. Більшість таких вебресурсів забезпечує представлення інформації про студію, демонстрацію портфоліо виконаних робіт, опис послуг та можливість зворотного зв'язку з клієнтами.

Проте існуючі аналоги переважно орієнтовані лише на інформаційне представлення діяльності студії та мають обмежені можливості взаємодії з користувачами. Значна частина вебресурсів не підтримує персоналізовану роботу клієнтів, не містить системи обробки заявок, а також не забезпечує повноцінного функціоналу для організації комунікації між студією та замовниками.

Крім того, окремі вебресурси містять надлишковий функціонал або складну структуру інтерфейсу, що може ускладнювати використання системи користувачами. Також деякі аналоги мають обмежені можливості керування контентом та подальшого розширення функціональних можливостей системи. Крім того, більшість вебресурсів створюється індивідуально для конкретних дизайн-студій з урахуванням їхніх потреб і специфіки діяльності.

У зв'язку з цим виникає необхідність розробки власного вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS». Розроблювана система дозволить врахувати особливості діяльності студії, забезпечити ефективне представлення послуг у сфері дизайну інтер'єру, ландшафтного дизайну, проєктування меблів та оформлення фасадів будівель, а також реалізувати необхідний функціонал взаємодії з клієнтами.

Власний вебзастосунок забезпечить сучасний інтерфейс, зручне керування контентом, можливість подання та обробки заявок, підтримку зворотного зв'язку з клієнтами та можливість подальшого розширення функціональних можливостей системи відповідно до потреб дизайн-студії «Creative_DS»

1.3 Постановка задачі на розробку вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Проведений аналіз наявних вебресурсів та програмних рішень показав, що найбільш доцільним є власне розроблення вебзастосунок для дизайн-студії «Creative_DS». Такий підхід дозволить врахувати особливості діяльності студії, реалізувати необхідний функціонал відповідно до поставлених вимог та уникнути додаткових витрат на придбання й адаптацію готових систем, які часто потребують значного доопрацювання.

На основі проведеного аналізу сформулюємо постановку практичної задачі. У межах даної роботи необхідно розробити вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS», який забезпечуватиме ефективне представлення діяльності студії, демонстрацію дизайнерських робіт, підтримку комунікації з користувачами та можливість подальшого розширення функціоналу. Розроблюваний застосунок повинен задовольняти наступним вимогам.

Вимоги до складу виконуваних функцій

З даним вебзастосунком повинні мати можливість працювати: адміністратор та клієнт.

Функції адміністратора:

- керування інформацією про студію;
- керування портфоліо робіт;
- керування послугами;
- перегляд та обробка звернень від клієнтів;
- керування контентом вебзастосунку;
- підтримка актуальності інформації на сайті.

Функції клієнта:

- перегляд інформації про дизайн-студію «Creative_DS»;
- перегляд послуг студії;
- перегляд портфоліо виконаних дизайнерських робіт;
- перегляд контактів;
- надсилання звернень через форму зворотного зв'язку;

Нефункціональні вимоги включають:

- швидкодію вебзастосунку;
- зручність користування;
- адаптивність інтерфейсу;
- надійність роботи системи;
- масштабованість програмної архітектури;
- безпечне опрацювання даних користувачів.

Вимоги до організації вхідних та вихідних даних

Вимоги до організації вхідних даних

Введення інформації до вебзастосунку здійснюється користувачами системи (клієнтами та адміністратором) через графічний вебінтерфейс із використанням стандартних засобів взаємодії, таких як клавіатура, комп'ютерна миша або сенсорний екран мобільних пристроїв.

Інформація може надходити до системи такими способами:

- ручне введення даних у текстові поля;
- вибір значень із доступних випадаючих списків;
- автоматичне отримання даних із бази даних вебзастосунку.

Вимоги до організації вихідних даних

Відображення результатів роботи системи здійснюється у режимі реального часу на екранах користувачів через вебінтерфейс. Користувачам надається актуальна інформація про діяльність дизайн-студії, портфоліо, послуги, контакти та результати взаємодії із системою.

Основу інформаційного забезпечення вебзастосунку будуть складати відомості про користувачів системи, відомості про дизайн-студію, дані про виконані дизайнерські роботи, включаючи зображення та опис, інформація про звернення користувачів через форму зворотного зв'язку

Вимоги до складу та параметрів технічних засобів

Для забезпечення роботи вебзастосунку необхідно використовувати технічні засоби з наступними мінімальними характеристиками:

- Сервер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 20 ГБ;
 - мережеве підключення зі швидкістю 100 Мбіт/с.
- Користувацький комп'ютер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 500 МБ;
 - мережеве підключення зі швидкістю 10 Мбіт/с.

Вимоги до інформаційної та програмної сумісності

Для забезпечення роботи вебзастосунку необхідна наявність встановленого програмного забезпечення з наступними мінімальними характеристиками:

– Сервер:

– операційна система: Windows Server 2022 або Ubuntu Server 22.04 LTS;

– вебсервер: Apache HTTP Server 2.4 або Nginx 1.20;

– PHP версії 8.3;

– система керування базами даних MySQL 8.0.

– Користувацький комп'ютер:

– операційна система: Windows 10 або Ubuntu 22.04 LTS;

– сучасний веббраузер із підтримкою HTML5, CSS3 та JavaScript, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

Розширений опис постановки задачі наведено у технічному завданні, поданому в додатку А.

2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ПРЕДСТАВНИЦТВА ТА ВЗАЄМОДІЇ З КЛІЄНТАМИ ДИЗАЙН-СТУДІЇ «CREATIVE_DS»

2.1 Аналіз вимог до вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

2.1.1 Побудова моделі варіантів використання

Для моделювання функціональних вимог вебзастосунку дизайн-студії «Creative_DS» було розроблено діаграму варіантів використання (Use Case Diagram). Дана діаграма відображає взаємодію між акторами системи та основними функціональними можливостями програмного забезпечення, що дозволяє наочно представити поведінку системи з точки зору користувачів [4].

Діаграма варіантів використання вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» наведена на рисунку 2.1.

У межах розроблюваного вебзастосунку визначено два основних актори: «Клієнт» та «Адміністратор». Клієнт взаємодіє із системою як зовнішній користувач і має можливість переглядати інформацію про дизайн-студію, ознайомлюватися з переліком послуг, переглядати портфоліо виконаних робіт, отримувати контактну інформацію, а також надсилати звернення через форму зворотного зв'язку.

Адміністратор є розширеним користувачем системи та додатково виконує функції керування контентом вебзастосунку. До його повноважень належить додавання, редагування та видалення інформації про студію, управління переліком послуг, керування портфоліо робіт, а також перегляд і обробка повідомлень, отриманих від клієнтів. Крім того, адміністратор здійснює підтримку актуальності інформації на вебсайті.

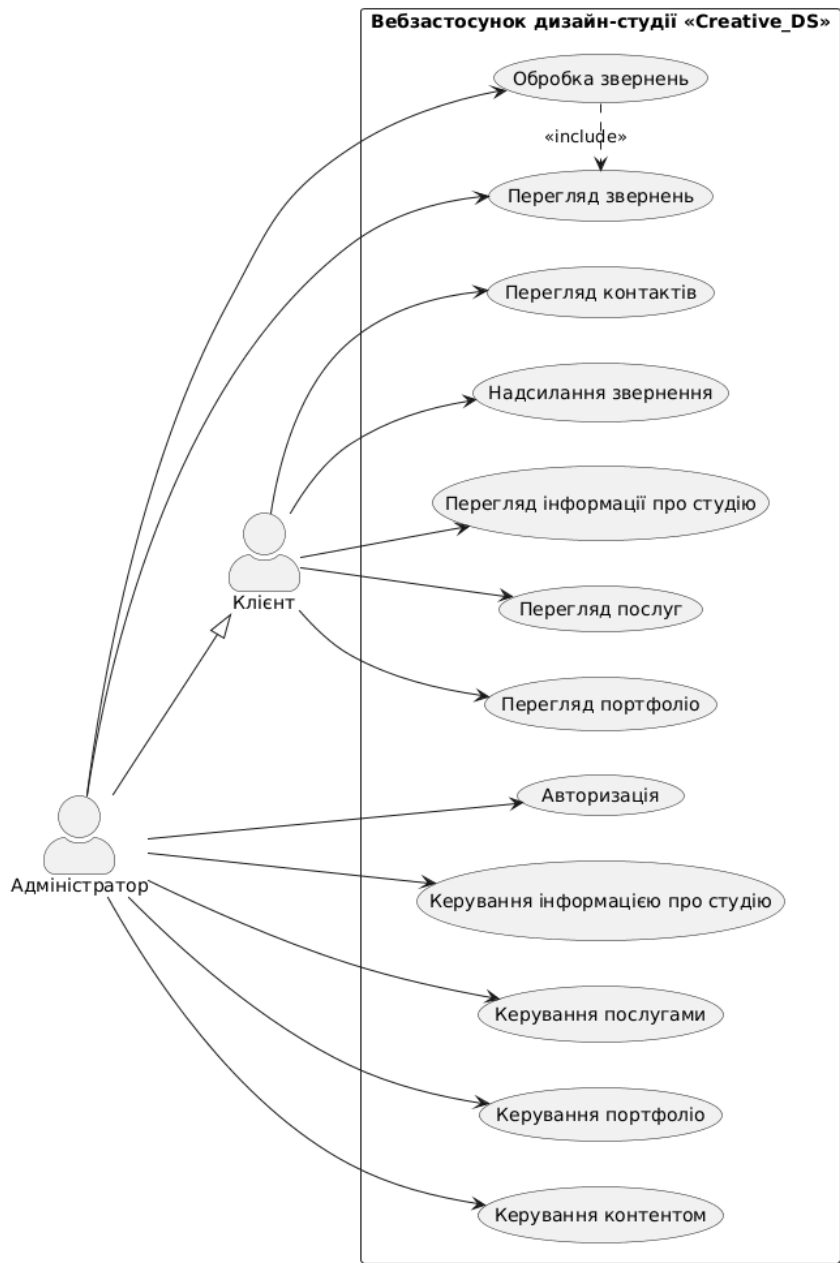


Рисунок 2.1 – Діаграма варіантів використання вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

На діаграмі також передбачено варіант використання «Авторизація», який забезпечує захищений доступ адміністратора до функцій керування системою. Частина функцій адміністратора пов’язана з базовими можливостями клієнта, що відображено через відношення узагальнення між акторами.

Таким чином, діаграма варіантів використання дозволяє визначити межі системи, структуру її функціональних можливостей та ролі користувачів, що є

основою для подальшого етапу технічного проєктування програмного забезпечення.

2.1.2 Специфікації варіантів використання

Для більшої деталізації діаграми варіантів використання для кожного варіанту використання розробимо специфікації у наступному вигляді:

Назва варіанту використання

1. Короткий опис
2. Актор
3. Передумови
4. Основний потік подій
5. Альтернативні потоки подій
6. Постумови

Далі наведено основні специфікації варіантів використання.

Назва: Перегляд інформації про студію

1. Короткий опис: Варіант використання дозволяє клієнту переглядати загальну інформацію про дизайн-студію «Creative_DS».

2. Актор: Клієнт

3. Передумови: Користувач відкрив вебзастосунок та має доступ до головної сторінки.

4. Основний потік подій:

Користувач переходить на головну сторінку.

Система завантажує інформацію про студію.

Система відображає контент користувачу.

Користувач переглядає інформацію.

5. Альтернативні потоки подій:

Якщо сервер недоступний, система відображає повідомлення про помилку завантаження.

6. Постумови: Користувач отримує доступ до актуальної інформації про студію.

Назва: Перегляд портфоліо

1. Короткий опис: Варіант використання забезпечує перегляд дизайнерських робіт студії у вигляді галереї.

2. Актор: Клієнт

3. Передумови: Користувач відкрив розділ «Портфоліо».

4. Основний потік подій:

Користувач відкриває розділ портфоліо.

Система отримує дані з бази даних.

Система формує список робіт.

Користувач переглядає роботи.

5. Альтернативні потоки подій:

Якщо портфоліо порожнє, система відображає повідомлення «Немає доступних робіт».

6. Постумови: Користувач ознайомлений із роботами студії.

Назва: Надсилання звернення

1. Короткий опис: Варіант використання дозволяє клієнту надіслати повідомлення до адміністратора через форму зворотного зв'язку.

2. Актор: Клієнт

3. Передумови: Користувач відкрив сторінку контактів та форму зворотного зв'язку.

4. Основний потік подій:

Користувач заповнює форму (ім'я, email, повідомлення).

Система виконує перевірку введених даних.

Користувач підтверджує відправку.

Система зберігає повідомлення в базі даних.

Система відображає повідомлення про успішну відправку.

5. Альтернативні потоки подій:

Якщо дані некоректні, система відображає повідомлення про помилки валідації.

Якщо сталася помилка збереження, повідомлення не відправляється.

6. Постумови: Повідомлення збережено та доступне адміністратору.

Назва: Авторизація адміністратора

1. Короткий опис: Варіант використання забезпечує доступ адміністратора до панелі керування системою.

2. Актор: Адміністратор

3. Передумови: Адміністратор відкрив сторінку входу в систему.

4. Основний потік подій:

Адміністратор вводить логін і пароль.

Система перевіряє облікові дані.

Система створює сесію користувача.

Адміністратор отримує доступ до панелі керування.

5. Альтернативні потоки подій:

Якщо дані невірні, система відхиляє спробу входу.

Якщо акаунт заблоковано, доступ заборонено.

6. Постумови: Адміністратор авторизований у системі.

Назва: Керування портфоліо

1. Короткий опис: Варіант використання дозволяє адміністратору додавати, редагувати та видаляти роботи портфоліо.

2. Актор: Адміністратор

3. Передумови: Адміністратор авторизований у системі.

4. Основний потік подій:

Адміністратор відкриває розділ портфоліо.

Система відображає список робіт.

Адміністратор обирає дію (додати / редагувати / видалити).

Система виконує операцію в базі даних.

Система оновлює список робіт.

5. Альтернативні потоки подій:

Якщо сталася помилка бази даних, зміни не зберігаються.

6. Постумови: Дані портфоліо актуалізовані.

2.2 Архітектура вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Загальна характеристика архітектури

Розроблюваний вебзастосунок для дизайн-студії «Creative_DS» побудовано на основі клієнт-серверної архітектури з використанням шаблону проектування MVC (Model–View–Controller). Такий підхід забезпечує розділення відповідальності між рівнями системи, спрощує супровід програмного забезпечення та підвищує його масштабованість.

Архітектура системи передбачає взаємодію трьох основних рівнів: рівня представлення, рівня бізнес-логіки та рівня доступу до даних.

Рівні архітектури системи

Рівень представлення (Presentation Layer)

Даний рівень відповідає за взаємодію з користувачем та відображення даних. Він реалізує користувацький інтерфейс вебзастосунку та забезпечує навігацію між сторінками.

До його складу входять: HTML5, CSS3, JavaScript, Bootstrap 5, шаблони відображення даних.

Основні функції рівня:

- відображення інформації про студію;
- перегляд портфоліо та послуг;
- введення даних користувачем;
- відправка запитів до серверної частини.

Рівень бізнес-логіки (Application Layer)

Рівень бізнес-логіки реалізує основну функціональність системи та обробку користувацьких запитів.

Він включає: контролери, модулі авторизації, модулі обробки звернень, логіку керування контентом.

Основні функції:

- обробка HTTP-запитів;
- перевірка та валідація даних;
- реалізація правил доступу;
- взаємодія з рівнем даних.

Рівень даних (Data Layer)

Рівень даних забезпечує збереження та отримання інформації.

Він включає: базу даних, механізми доступу до даних, сутності системи.

Основні функції:

- збереження інформації про студію;
- збереження портфоліо робіт;
- збереження звернень користувачів;
- виконання запитів до бази даних.

Основні модулі системи

Вебзастосунок складається з наступних функціональних модулів:

- модуль управління інформацією про студію;
- модуль управління послугами;
- модуль портфоліо;
- модуль зворотного зв'язку;
- модуль авторизації адміністратора;
- публічний інформаційний модуль.

Переваги обраної архітектури

Використання клієнт-серверної архітектури та MVC-підходу забезпечує:

- чітке розділення логіки та представлення;
- можливість масштабування системи;
- простоту супроводу та модифікації;
- повторне використання компонентів;
- підвищену надійність програмного забезпечення.

Діаграма компонентів вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» представлена на рисунку 2.2.

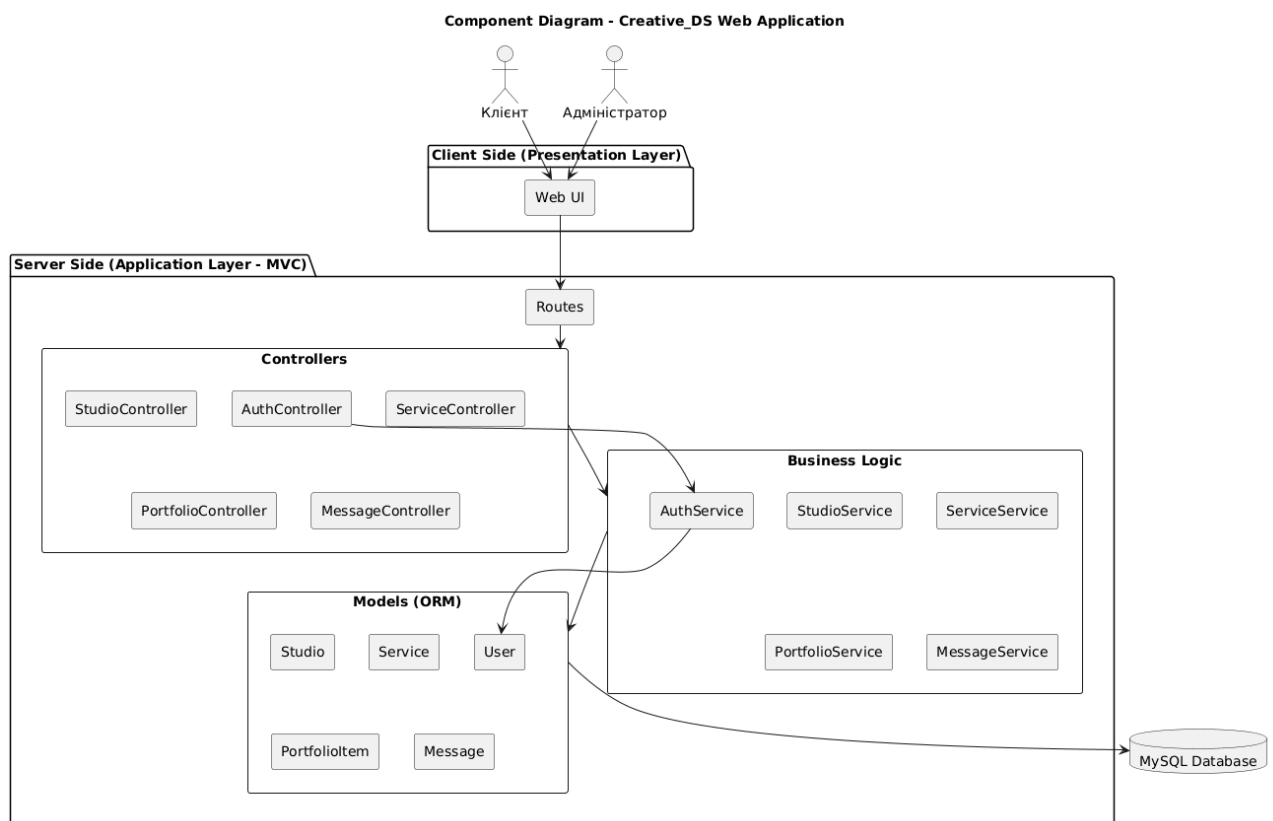


Рисунок 2.2 – Діаграма компонентів вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Наведена діаграма компонентів відображає структурну організацію вебзастосунку «Creative_DS» та взаємодію між його основними програмними модулями. Система побудована на основі багаторівневої архітектури, яка включає рівень представлення, рівень бізнес-логіки та рівень доступу до даних.

На рівні представлення знаходиться вебінтерфейс користувача, через який клієнти та адміністратор взаємодіють із системою. Рівень бізнес-логіки реалізовано за допомогою контролерів та сервісів, які забезпечують обробку запитів, реалізацію бізнес-правил та взаємодію з моделями даних. Рівень доступу до даних представлений моделями ORM, які забезпечують взаємодію з реляційною базою даних MySQL.

Діаграма також відображає основні компоненти підсистеми авторизації, яка забезпечує безпечний доступ адміністратора до функціоналу керування системою. Таким чином, компонентна діаграма демонструє модульну структуру системи та взаємодію між її основними частинами.

Діаграма розгортання (Deployment Diagram) відображає фізичну структуру вебзастосунку «Creative_DS» та описує взаємодію між апаратними вузлами і програмними компонентами системи.

Розроблюваний вебзастосунок реалізовано за клієнт-серверною архітектурою. На стороні клієнта використовується веббраузер, через який користувач (клієнт або адміністратор) здійснює доступ до системи.

На серверній стороні розміщено вебсервер (Apache або Nginx), який обробляє HTTP-запити та передає їх до серверного застосунку, реалізованого на PHP-фреймворку Laravel.

Бізнес-логіка системи виконується в межах Laravel-застосунку, який забезпечує обробку запитів, реалізацію функціональних модулів та взаємодію з базою даних.

Для збереження інформації використовується реляційна база даних MySQL, яка розміщується на окремому сервері баз даних або на тому ж фізичному сервері в межах навчального середовища.

Діаграма розгортання вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» представлена на рисунку 2.3.

Deployment Diagram - Creative_DS Web Application

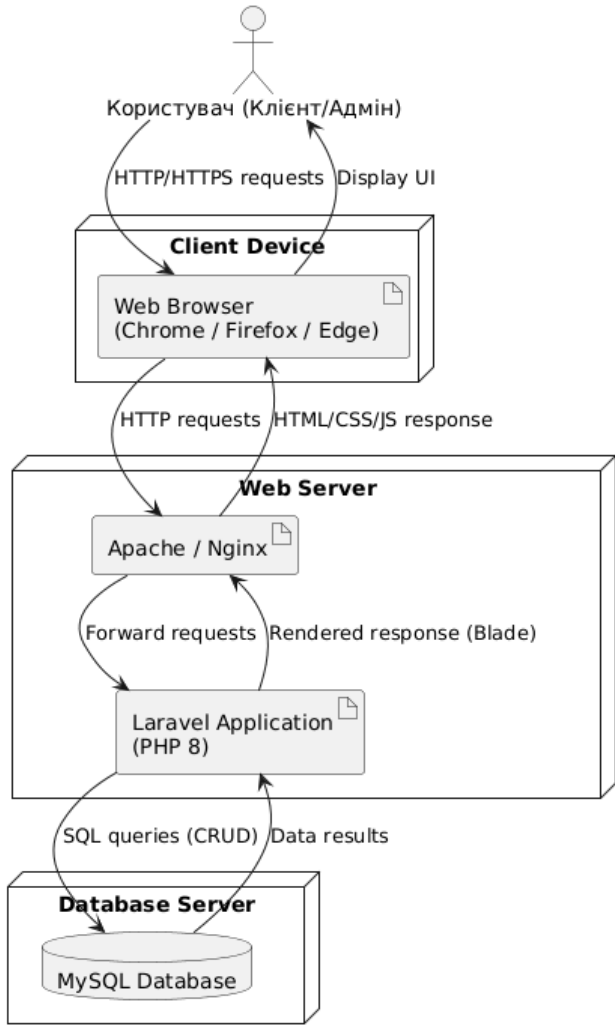


Рисунок 2.3 – Діаграма розгортання вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Таким чином, діаграма розгортання демонструє фізичну структуру системи, а також взаємодію між клієнтською частиною, серверним застосунком і базою даних.

Запропонована архітектура вебзастосунку «Creative_DS» є структурованою, логічно розділеною на рівні та відповідає вимогам сучасних вебсистем, що забезпечує ефективну реалізацію функціональних та нефункціональних вимог проєкту.

2.3 Проєкт вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

2.3.1 Ескізний проєкт вебзастосунку

Розробка концептуальної моделі даних

Для зберігання даних вебзастосунку дизайн-студії «Creative_DS» буде розроблено реляційну базу даних MySQL. Структура бази даних спроектована відповідно до вимог предметної області та забезпечує збереження інформації про користувачів, послуги, портфоліо та звернення клієнтів [5].

Концептуальна модель даних вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» представлена на рисунку 2.4.

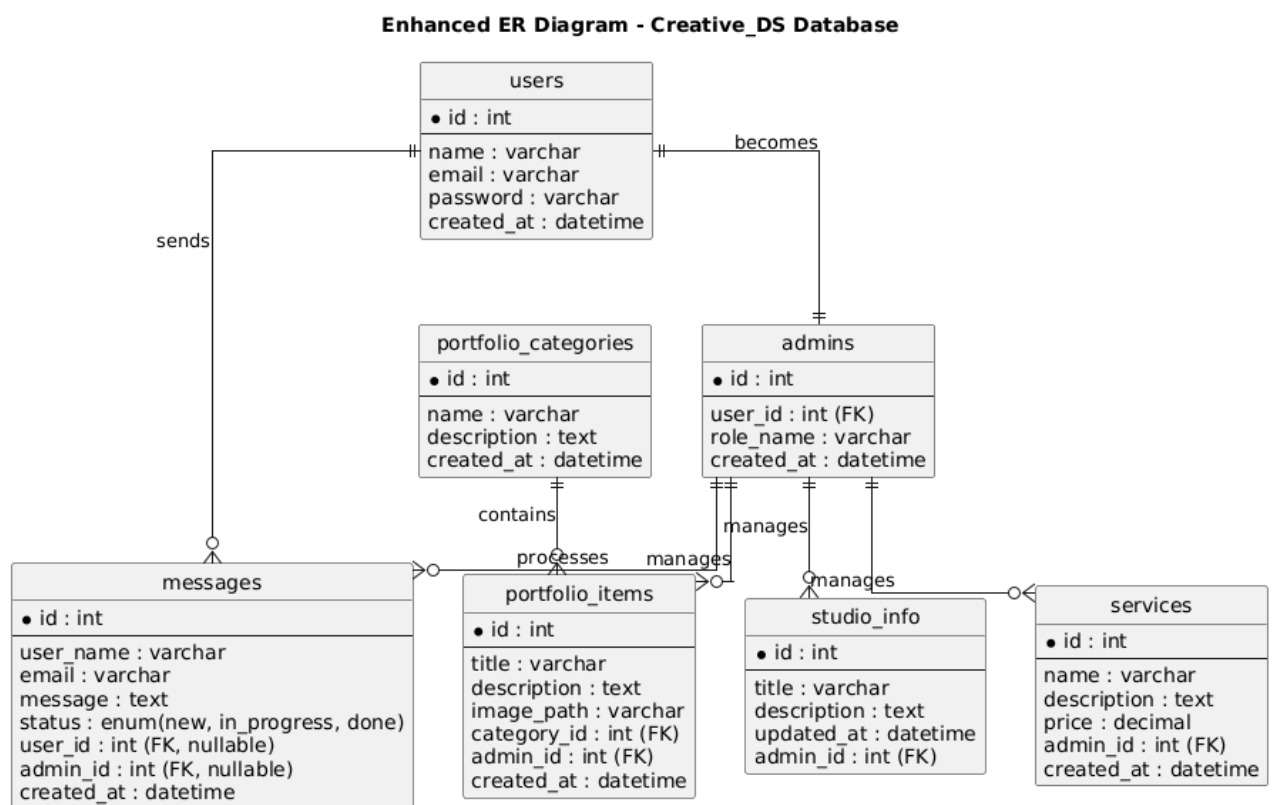


Рисунок 2.4 – Концептуальна модель даних вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Концептуальна модель даних вебзастосунку відображає основні сутності

системи та зв'язки між ними. Основними сутностями є:

- users – таблиця користувачів системи, яка зберігає дані адміністратора та клієнтів, а також їх ролі;
- studio_info – містить інформацію про дизайн-студію;
- services – зберігає перелік послуг, що надаються студією;
- portfolio_items – містить дані про виконані дизайнерські роботи, включаючи зображення та опис;
- messages – використовується для збереження звернень користувачів через форму зворотного зв'язку.

Між сутностями встановлено логічні зв'язки. Зокрема, користувачі можуть надсилати повідомлення, які зберігаються в таблиці messages. Портфоліо робіт може бути пов'язане з категоріями або послугами, що дозволяє структурувати представлені роботи.

Запропонована структура бази даних забезпечує цілісність даних, підтримує нормалізацію та дозволяє ефективно виконувати CRUD-операції, необхідні для функціонування вебзастосунку.

Для забезпечення коректного зберігання та обробки даних далі у вебзастосунку було розширено структуру бази даних шляхом введення додаткових сутностей та встановлення більш детальних зв'язків між ними.

У моделі даних виділено окремі сутності користувачів (users) та адміністраторів (admins), що дозволяє реалізувати рольову модель доступу та забезпечити безпеку системи. Сутність admins відповідає за управління контентом системи, включаючи інформацію про студію, послуги, портфоліо та обробку повідомлень користувачів.

Для структурування портфоліо введено сутність portfolio_categories, яка забезпечує класифікацію робіт та встановлює зв'язок типу «один-до-багатьох» з таблицею portfolio_items.

Таким чином, усі сутності бази даних взаємопов'язані, що забезпечує цілісність даних, нормалізацію структури та ефективну взаємодію між компонентами системи.

Сценарії варіантів використання

Далі розробимо сценарії варіантів використання вебзастосунку. Для прикладу у кваліфікаційній роботі наведено два з них: Надсилання звернення через форму зворотного зв'язку та Керування портфоліо адміністратором

Надсилання звернення через форму зворотного зв'язку

Даний сценарій описує процес взаємодії клієнта з вебзастосунком під час надсилання звернення до дизайн-студії «Creative_DS».

Сценарій варіанту використання «Надсилання звернення» представлено на рисунку 2.5.



Рисунок 2.5 – Сценарій варіанту використання «Надсилання звернення»

Клієнт переходить до сторінки контактів та відкриває форму зворотного зв'язку. Після цього він вводить необхідні дані, зокрема ім'я, адресу електронної пошти та текст повідомлення. Система виконує перевірку коректності введених даних. У випадку успішного проходження валідації звернення зберігається в базі

даних та отримує статус «Нове». Після успішного збереження система відображає користувачеві повідомлення про успішне надсилання звернення.

Якщо під час перевірки буде виявлено помилки введення, система повідомляє користувача про необхідність їх виправлення та повертає його до форми введення даних.

Керування портфоліо

Даний сценарій описує процес керування портфоліо дизайн-студії адміністратором системи.

Сценарій варіанту використання «Керування портфоліо» представлено на рисунку 2.6.

Після успішної авторизації адміністратор відкриває модуль управління портфоліо. Система відображає список наявних робіт. Адміністратор обирає необхідну операцію: додавання нової роботи, редагування існуючої або видалення роботи.

У випадку додавання або редагування роботи адміністратор заповнює необхідні поля, завантажує зображення та підтверджує зміни. Система перевіряє введені дані та зберігає інформацію в базі даних. Після успішного виконання операції оновлений перелік робіт відображається на екрані.

У випадку видалення система запитує підтвердження операції, після чого видаляє відповідний запис із бази даних та оновлює список робіт.

Activity Diagram - Керування портфоліо

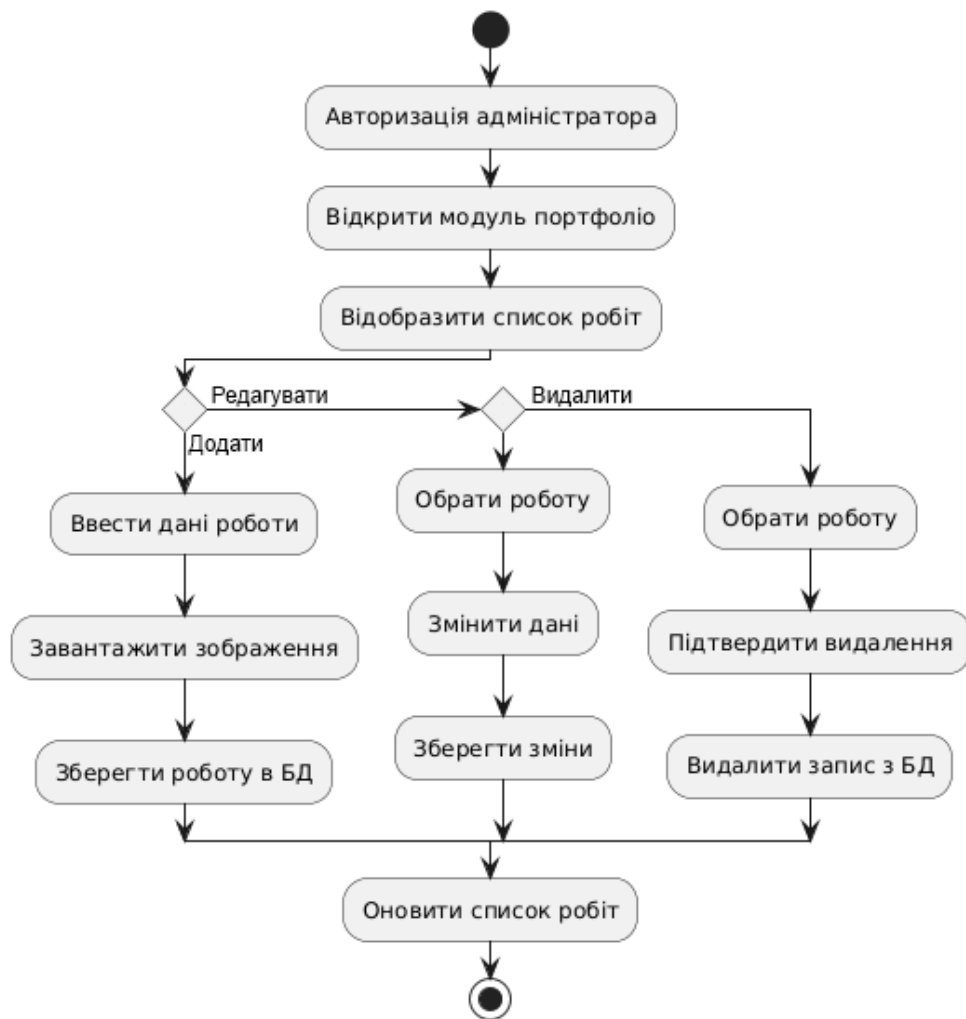


Рисунок 2.6 – Сценарій варіанту використання «Керування портфоліо»

Розробка прототипу інтерфейсу користувача

В якості прототипу інтерфейсу користувача розробимо прототипи двох сторінок: Портфоліо дизайн-студії (публічна частина) та Панель адміністратора – керування портфоліо. Розроблений прототип сторінки «Портфоліо дизайн-студії» представлено на рисунку 2.7.

Сторінка «Портфоліо дизайн-студії» буде мати: структуру Bootstrap 5, адаптивне меню, Неро-блок, фільтр категорій, галерею робіт, модальне вікно перегляду роботи, футер.

Розроблений прототип сторінки «Панель адміністратора – керування портфоліо» представлено на рисунку 2.8.

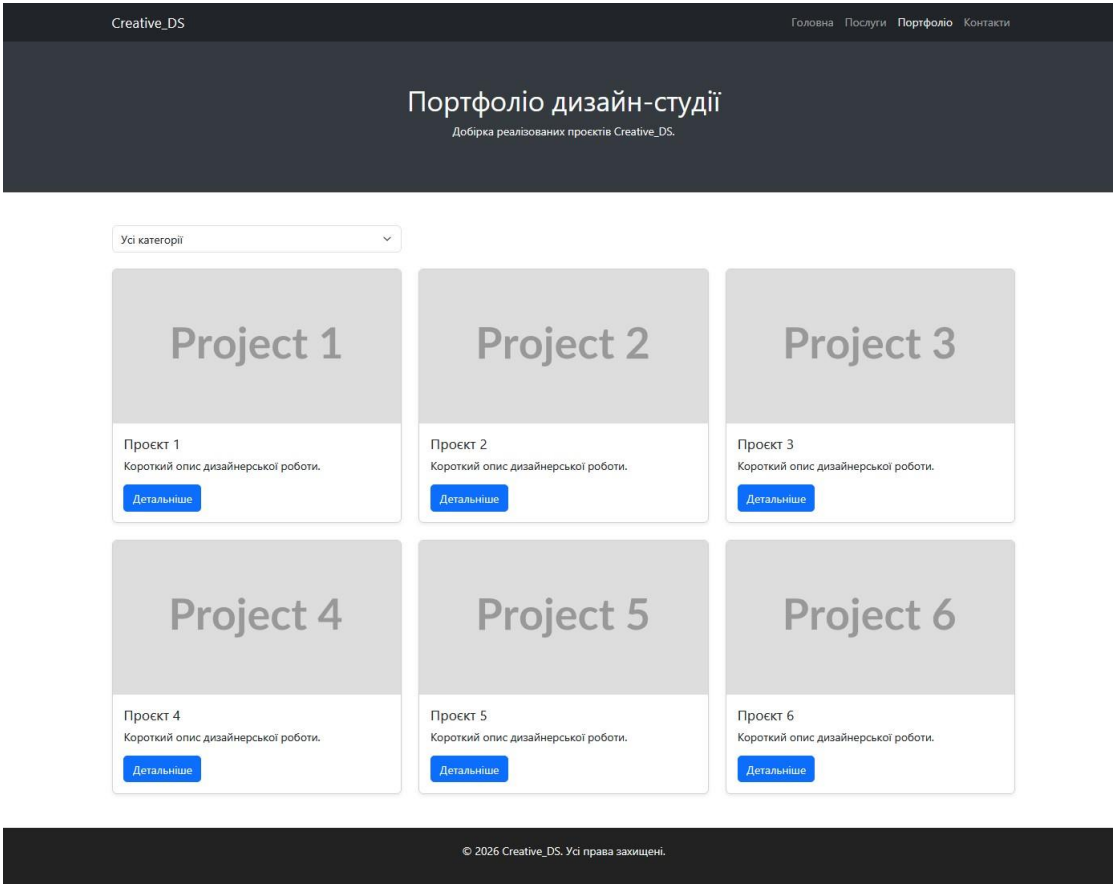


Рисунок 2.7 – Прототип сторінки «Портфоліо дизайн-студії»

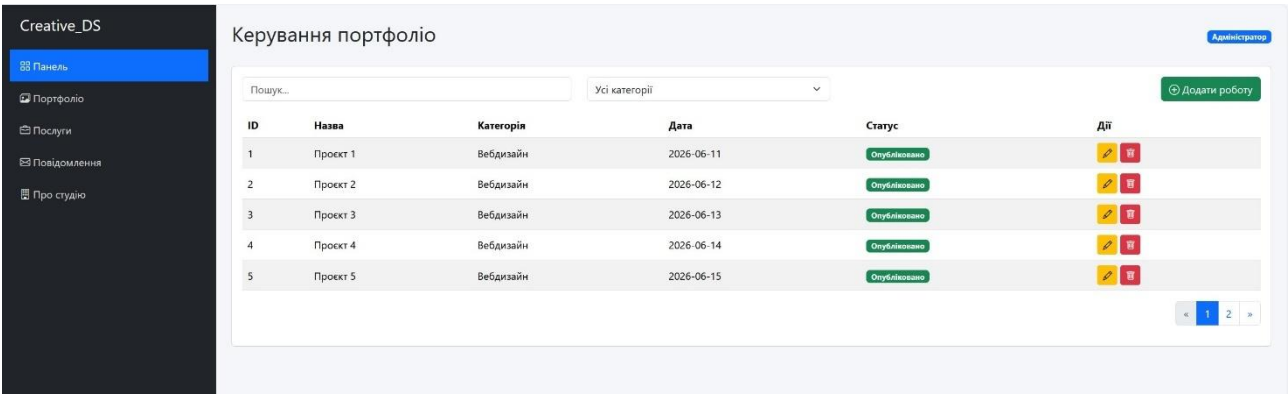


Рисунок 2.8 – Прототип сторінки «Панель адміністратора – керування портфоліо»

Сторінка «Панель адміністратора – керування портфоліо» буде мати: сучасну панель адміністратора, бічне меню, верхню панель, таблицю портфоліо, пошук, пагінацію, кнопки CRUD, модальне вікно "Додати роботу".

hasOne, hasMany та belongsTo.

Розроблена діаграма класів дозволяє чітко відобразити архітектуру вебзастосунку, визначити основні компоненти, їхні обов'язки та взаємозв'язки. Використання патерну MVC у поєднанні з ORM Eloquent забезпечує високу структурованість коду, легкість супроводу та масштабованість системи.

Далі розробимо специфікація класів вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

Клас User

Призначення: Зберігання даних користувачів системи та забезпечення їх взаємодії з функціоналом повідомлень.

Атрибути:

- id: int – унікальний ідентифікатор користувача;
- name: string – ім'я користувача;
- email: string – електронна пошта;
- password: string – пароль (хешований);
- created_at: datetime – дата створення облікового запису .

Методи:

- admin() – зв'язок з моделлю Admin (1:1);
- messages() – отримання всіх повідомлень користувача (1:N).

Клас Admin

Призначення: Управління системою та контентом вебзастосунку.

Атрибути:

- id: int – ідентифікатор адміністратора;
- user_id: int – зв'язок із користувачем;
- role_name: string – роль адміністратора;
- created_at: datetime – дата створення.

Методи:

- user() – зв'язок із користувачем (belongsTo);
- studioInfo() – доступ до інформації про студію;

- services() – керування послугами;
- portfolioItems() – керування портфоліо;
- messages() – обробка повідомлень.

Клас StudioInfo

Призначення: використовується для зберігання загальної інформації про студію.

Атрибути:

- id: int – унікальний ідентифікатор запису;
- title: string – заголовок або назва студії;
- description: text – детальний опис діяльності студії;
- updated_at: datetime – дата останнього оновлення інформації;
- admin_id: int – зовнішній ключ адміністратора, який керує записом.

Методи:

- admin() – повертає адміністратора, відповідального за оновлення інформації про студію (belongsTo).

Клас Service

Призначення: описує послуги, які надає студія.

Атрибути:

- id: int – унікальний ідентифікатор послуги;
- name: string – назва послуги;
- description: text – детальний опис послуги;
- price: decimal – вартість послуги;
- admin_id: int – адміністратор, який створив або керує послугою;
- created_at: datetime – дата створення послуги.

Методи:

- admin() – повертає адміністратора, який відповідає за послугу (belongsTo).

Клас PortfolioCategory

Призначення: використовується для групування елементів портфоліо за категоріями.

Атрибути:

- id: int – унікальний ідентифікатор категорії;
- name: string – назва категорії;
- description: text – опис категорії;
- created_at: datetime – дата створення категорії.

Методи:

- portfolioItems() – повертає всі елементи портфоліо, що належать даній категорії (hasMany).

Клас PortfolioItem

Призначення: зберігає інформацію про окремі роботи портфоліо студії.

Атрибути:

- id: int – унікальний ідентифікатор роботи;
- title: string – назва роботи;
- description: text – опис виконаної роботи;
- image_path: string – шлях до зображення роботи;
- category_id: int – категорія, до якої належить робота;
- admin_id: int – адміністратор, який додав роботу;
- created_at: datetime – дата створення запису.

Методи:

- category() – повертає категорію, до якої належить елемент портфоліо (belongsTo);
- Iadmin() – повертає адміністратора, який створив або керує елементом.

Клас Message

Призначення: відповідає за обробку повідомлень, отриманих від користувачів або адміністраторів.

Атрибути:

- id: int – унікальний ідентифікатор повідомлення;
- user_name: string – ім'я відправника повідомлення;
- email: string – електронна адреса відправника;
- message: text – текст повідомлення;

- status: enum(new, in_progress, done) – статус опрацювання повідомлення;
- user_id: int (nullable) – зв'язок із користувачем (якщо існує);
- admin_id: int (nullable) – адміністратор, який обробляє повідомлення;
- created_at: datetime – дата створення повідомлення.

Методи:

- user() – повертає користувача, який надіслав повідомлення (belongsTo);
- admin() – повертає адміністратора, який опрацьовує повідомлення (belongsTo).

Деталізовані специфікації класів відображають логічну структуру програмної системи та деталізують її об'єктно-орієнтовану модель. Кожен клас відповідає окремій сутності предметної області та реалізує набір атрибутів і методів, що забезпечують взаємодію між компонентами системи відповідно до архітектури Laravel MVC та ORM Eloquent.

Динамічна модель програмного забезпечення

Діаграми послідовності використовуються для відображення динамічної взаємодії між об'єктами системи в часі під час виконання конкретного варіанту використання. Вони дозволяють наочно представити порядок обміну повідомленнями між користувачем, інтерфейсом, контролерами, моделями та базою даних, що забезпечує глибше розуміння логіки роботи програмної системи. Завдяки таким діаграмам можна детально проаналізувати послідовність викликів, виявити можливі помилки у проєктуванні взаємодій та оптимізувати архітектуру системи ще на етапі розробки.

Представимо дві діаграми послідовності для варіантів використання «Надсилення звернення» та «Керування портфоліо».

Діаграма послідовності «Надсилення звернення» відображає взаємодію між користувачем, інтерфейсом системи, контролером, моделлю та базою даних у процесі створення нового звернення.

Процес починається з того, що користувач відкриває форму зворотного зв'язку у вебінтерфейсі та вводить необхідні дані, зокрема ім'я, електронну

адресу та текст повідомлення. Після заповнення форми користувач ініціює відправку даних до системи.

Отримані дані передаються до `MessageController`, який виконує їх валідацію. На цьому етапі перевіряється коректність введеної інформації, зокрема наявність обов'язкових полів та відповідність формату електронної пошти.

У випадку успішної валідації контролер створює новий запис через модель `Message`. Модель взаємодіє з базою даних, виконуючи операцію вставки нового повідомлення в таблицю `messages`. Після успішного збереження система повертає підтвердження до контролера.

Далі користувач отримує повідомлення про успішне надсилання звернення. У разі виникнення помилок валідації система повертає список помилок для відображення у інтерфейсі користувача.

Таким чином, діаграма демонструє повний цикл обробки звернення користувача в архітектурі MVC, включаючи перевірку даних, взаємодію з моделлю та збереження інформації в базі даних.

Динамічна модель варіанту використання «Надсилання звернення» представлена у вигляді діаграми послідовності на рисунку 2.10.

Діаграма послідовності «Керування портфоліо» відображає процеси створення, оновлення та видалення елементів портфоліо, які виконуються адміністратором у системі.

На початку адміністратор відкриває адміністративну панель та переходить до розділу керування портфоліо. Система відображає інтерфейс для виконання операцій над елементами портфоліо.

У випадку створення нового елемента адміністратор заповнює форму, яка містить назву роботи, опис, зображення та категорію. Після відправки даних вони передаються до `PortfolioController`, де виконується їх валідація.

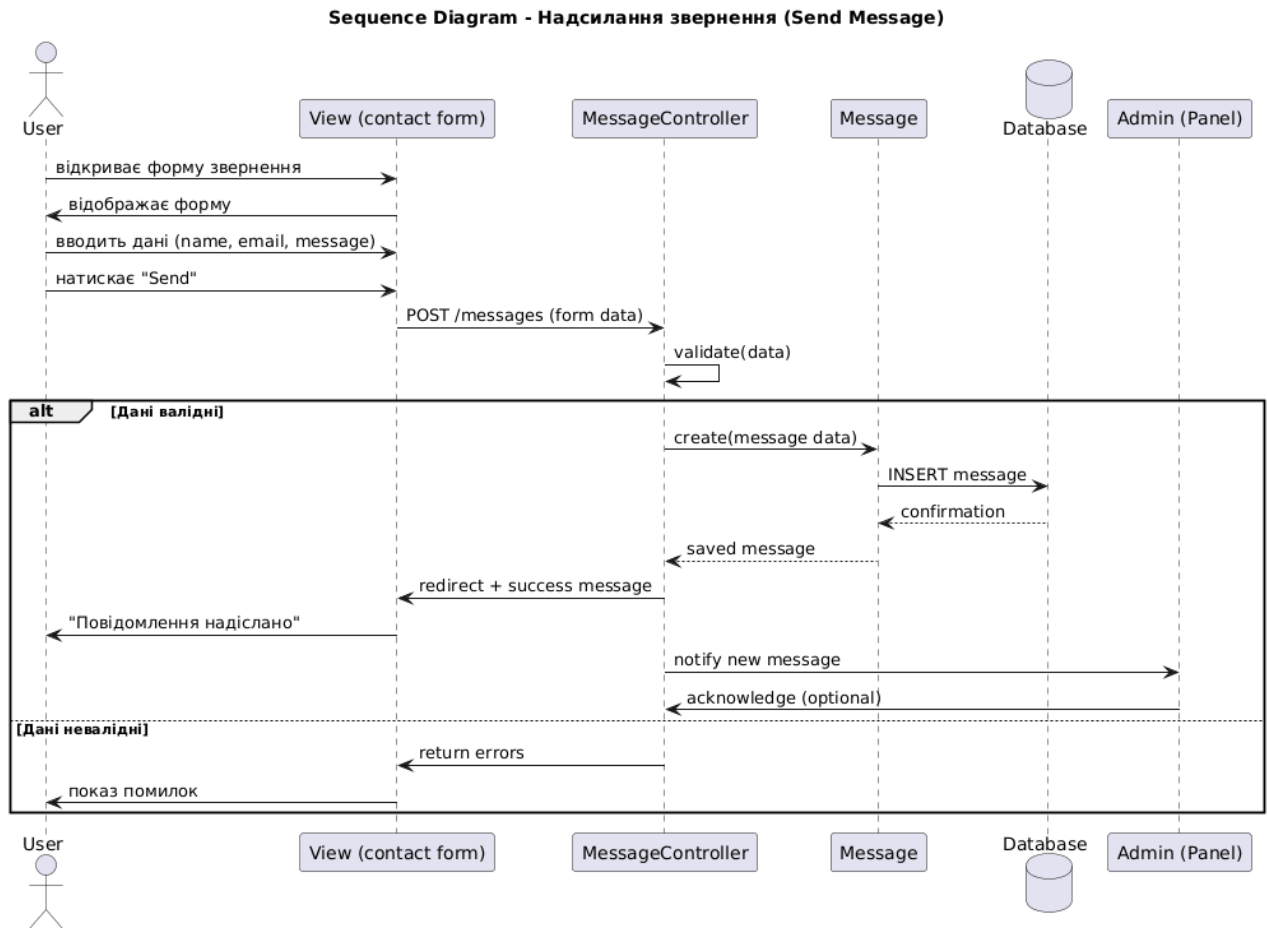


Рисунок 2.10 – Діаграма послідовності варіанту використання «Надсилання звернення»

Якщо дані є коректними, контролер створює новий запис через модель PortfolioItem, яка виконує вставку даних у таблицю portfolio_items бази даних. Після успішного збереження система повертає повідомлення про успішне додавання елемента.

У разі оновлення запису контролер спочатку здійснює пошук відповідного елемента за ідентифікатором, після чого виконує оновлення даних у базі. Якщо запис не знайдено, користувачу повертається помилка 404.

Процес видалення передбачає отримання запиту від адміністратора, після чого контролер викликає метод видалення моделі PortfolioItem, що призводить до видалення запису з бази даних.

Таким чином, діаграма відображає повний цикл управління портфоліо відповідно до принципів MVC-архітектури та демонструє взаємодію між

адміністративною частиною системи, контролером, моделями та базою даних.

Динамічна модель варіанту використання «Керування портфоліо» представлена у вигляді діаграми послідовностей на рисунку 2.11.

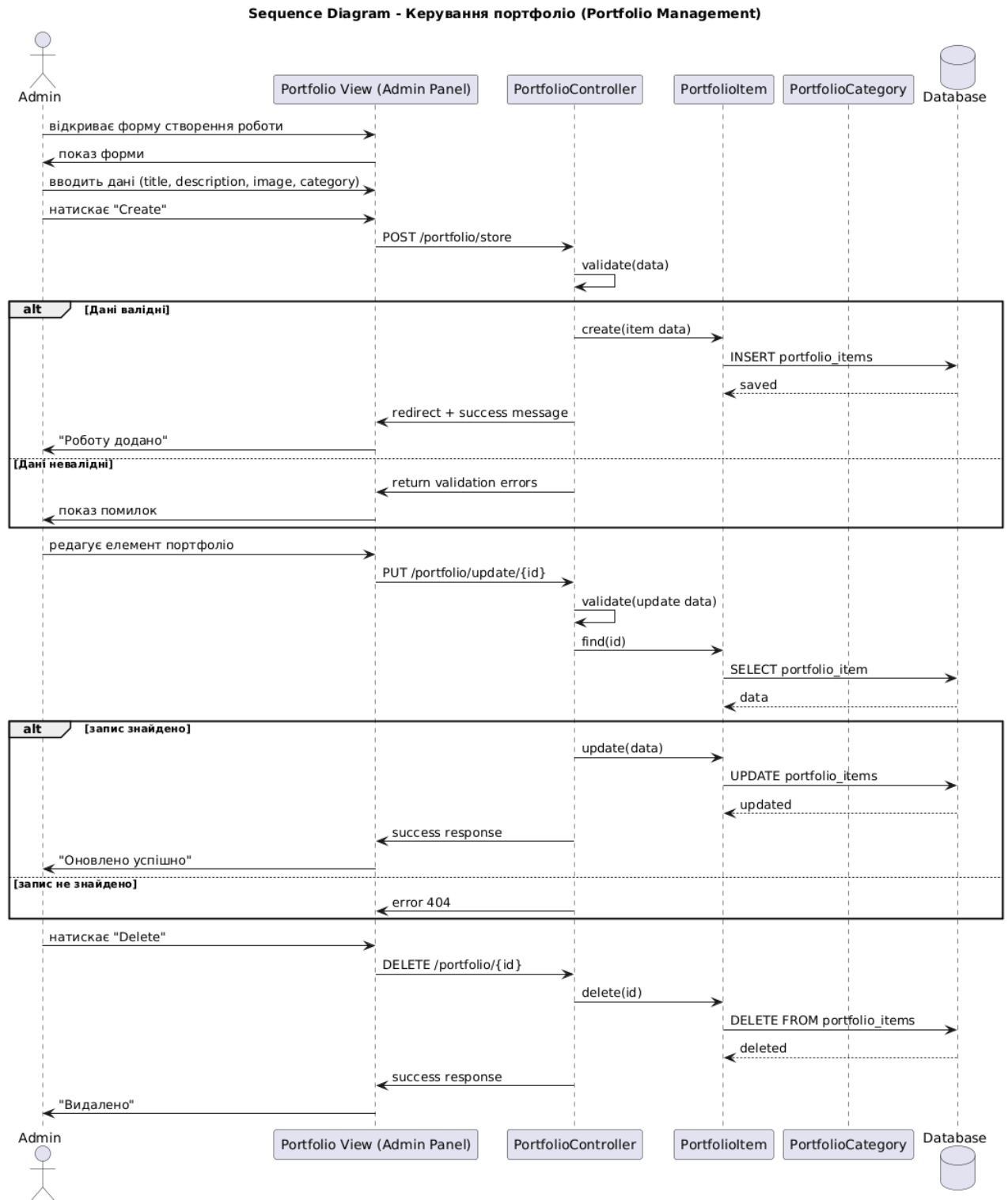


Рисунок 2.11 – Діаграма послідовностей варіанту використання «Керування портфоліо»

Побудовані діаграми послідовності дозволяють детально проаналізувати процеси взаємодії між компонентами системи. Вони підтверджують коректність реалізації архітектури Model–View–Controller у вебзастосунку та демонструють логіку обробки ключових сценаріїв використання: надсилання звернення та керування портфоліо.

Розробка логічної моделі даних

Після побудови ER-діаграми було розроблено логічну модель даних, яка визначає структуру бази даних та взаємозв'язки між її таблицями. Логічна модель відображає основні сутності програмної системи дизайн-студії «Creative_DS»: користувачів, адміністраторів, інформацію про студію, послуги, категорії портфоліо, роботи портфоліо та повідомлення користувачів.

Логічна модель даних вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» наведена на рисунку 2.12.

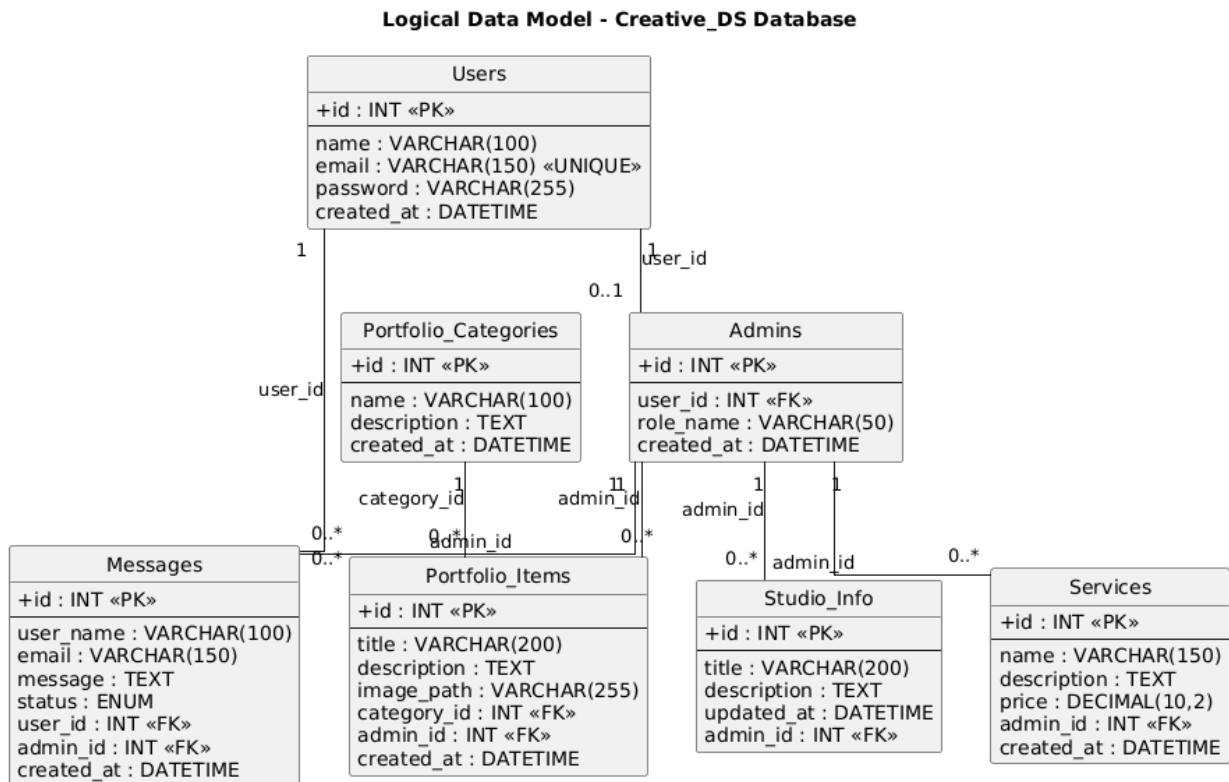


Рисунок 2.12 – Логічна модель даних вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Для кожної таблиці визначено набір атрибутів, первинний ключ (Primary Key) та зовнішні ключі (Foreign Key), що забезпечують цілісність даних. Між таблицями встановлено зв'язки типу «один до одного» та «один до багатьох». Зокрема, кожен адміністратор пов'язаний з одним користувачем, тоді як адміністратор може керувати декількома послугами, роботами портфоліо, інформацією про студію та опрацьовувати повідомлення користувачів. Роботи портфоліо належать до певної категорії, а повідомлення можуть бути пов'язані як із зареєстрованими користувачами, так і з адміністраторами.

Розроблена логічна модель є основою для створення фізичної моделі бази даних та подальшої реалізації програмної системи, забезпечуючи узгодженість структури даних, мінімізацію надлишковості та підтримку цілісності інформації.

2.3.3 Робочий проєкт програмного забезпечення

Технологічний стек розробки

Вибір технологічного стеку є одним із ключових етапів проєктування програмної системи, оскільки саме він визначає функціональні можливості програмного продукту, його продуктивність, масштабованість, безпеку та зручність подальшого супроводу. Для розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» було обрано сучасний стек технологій, який включає HTML5, CSS3, JavaScript, Bootstrap 5, PHP 8.3, Laravel та систему керування базами даних MySQL. Зазначені технології є взаємосумісними, широко використовуються у веброзробці та мають значну спільноту розробників, що спрощує їх використання та підтримку.

Основою інтерфейсу користувача було обрано HTML5, який є сучасним стандартом розмітки вебсторінок. HTML5 забезпечує логічну структуру документа, підтримує семантичні елементи, що позитивно впливає на

доступність вебресурсу та оптимізацію для пошукових систем. Крім того, стандарт підтримує інтеграцію мультимедійного контенту без необхідності використання сторонніх модулів, що є важливим для сайту творчої студії, який містить фотографії, портфоліо та інформаційні матеріали [6].

Для оформлення сторінок використовується CSS3, який надає широкі можливості для створення сучасного дизайну та адаптивного інтерфейсу. Використання каскадних таблиць стилів дозволяє розділити структуру документа та його оформлення, що спрощує супровід програмного коду та забезпечує повторне використання стилів. CSS3 також підтримує анімацію, плавні переходи, тіні, градієнти та інші сучасні елементи інтерфейсу без застосування додаткових графічних засобів [7].

Для реалізації інтерактивної поведінки вебзастосунку використовується JavaScript. Ця мова програмування забезпечує взаємодію користувача зі сторінками без необхідності їх повторного завантаження, що покращує швидкодію та зручність використання вебзастосунку. За допомогою JavaScript реалізуються перевірка введених даних, відкриття модальних вікон, динамічне оновлення окремих елементів інтерфейсу, а також інші інтерактивні функції, необхідні для сучасного вебзастосунку [8].

Для прискорення розробки інтерфейсу було використано Bootstrap 5 – один із найпоширеніших CSS-фреймворків. Bootstrap містить готові компоненти користувацького інтерфейсу, систему адаптивної сітки, набір утилітарних класів та засоби для забезпечення коректного відображення сторінок на різних пристроях. Використання Bootstrap дозволяє значно скоротити час розробки, забезпечити єдиний стиль оформлення та реалізувати адаптивний дизайн без необхідності створення великої кількості власних стилів [9].

Для реалізації серверної логіки обрано PHP 8.3. Незважаючи на появу нових мов програмування для веброзробки, PHP залишається однією з найпоширеніших технологій створення вебзастосунків. Основними перевагами PHP є простота інтеграції з вебсервером, підтримка великої кількості систем керування базами даних, висока продуктивність сучасних версій та широкі

можливості для створення динамічних вебресурсів [10].

Версія PHP 8.3 містить суттєві покращення продуктивності, оптимізації роботи з пам'яттю та механізмів типізації, що позитивно впливає на швидкодію програмного забезпечення. Крім того, сучасні можливості мови дозволяють створювати більш структурований, безпечний та легкий у підтримці програмний код.

Для організації серверної частини використовується фреймворк Laravel, який реалізує архітектурний шаблон Model–View–Controller (MVC). Використання MVC забезпечує розподіл відповідальності між компонентами системи, що підвищує читабельність програмного коду та полегшує його подальшу модифікацію.

Laravel містить широкий набір готових механізмів, необхідних для створення сучасних вебзастосунків. Зокрема, він забезпечує маршрутизацію запитів, автентифікацію користувачів, систему авторизації, механізм міграцій бази даних, шаблонізатор Blade, ORM Eloquent для роботи з базою даних, систему валідації введених даних та засоби захисту від поширених вебзагроз, таких як SQL-ін'єкції, міжсайтовий скриптинг (XSS) та підробка міжсайтових запитів (CSRF) [11].

Важливою перевагою Laravel є використання ORM Eloquent, що дозволяє працювати з даними у вигляді об'єктів замість написання великої кількості SQL-запитів. Це значно спрощує розробку, робить програмний код більш зрозумілим та зменшує ймовірність виникнення помилок/

Крім цього, Laravel підтримує механізм міграцій, який дозволяє автоматизувати створення та модифікацію структури бази даних. Такий підхід забезпечує контроль версій структури даних та значно спрощує розгортання програмного забезпечення на інших серверах.

Для зберігання інформації використовується реляційна система керування базами даних MySQL. Вибір саме цієї СКБД обумовлений її високою продуктивністю, надійністю та широким використанням у веброзробці. MySQL підтримує стандартну мову SQL, забезпечує ефективну роботу з великими

обсягами інформації та підтримує механізми забезпечення цілісності даних [12].

У межах розробленої програмної системи MySQL використовується для зберігання даних про користувачів, адміністраторів, послуги, портфоліо, категорії робіт, інформацію про студію та повідомлення користувачів. Використання зовнішніх ключів дозволяє підтримувати зв'язки між таблицями та забезпечує референтну цілісність даних.

MySQL повністю підтримується Laravel через ORM Eloquent та систему міграцій, що значно спрощує взаємодію між прикладною програмою та базою даних. Завдяки цьому зменшується кількість низькорівневого програмного коду та підвищується продуктивність процесу розробки.

Обраний технологічний стек є збалансованим рішенням для реалізації вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS». Поєднання HTML5, CSS3, JavaScript і Bootstrap 5 дозволяє створити сучасний, адаптивний та зручний користувацький інтерфейс. Використання PHP 8.3 та Laravel забезпечує реалізацію надійної серверної логіки, високий рівень безпеки, структурованість програмного коду та простоту його подальшого супроводу. Система керування базами даних MySQL забезпечує ефективне зберігання інформації та підтримує всі необхідні механізми роботи з реляційними даними.

Фрагменти коду вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» наведено в додатку Б. Наведений інтерфейс та база даних.

Тестування програмного забезпечення

Тестування є одним із завершальних етапів розробки вебзастосунку та спрямоване на перевірку відповідності його функціональних можливостей встановленим вимогам. Метою тестування є виявлення можливих помилок, перевірка коректності роботи функціональних модулів та підтвердження готовності програмного продукту до експлуатації.

Для перевірки працездатності вебзастосунку для представництва та

взаємодії з клієнтами дизайн-студії «Creative_DS» було використано метод функціонального тестування «чорної скриньки». Цей метод передбачає перевірку функціонування програмного забезпечення без аналізу його внутрішньої структури або програмного коду. Під час тестування оцінюються лише відповідність отриманих результатів очікуваній поведінці системи.

Для побудови тестових сценаріїв було застосовано метод розбиття на класи еквівалентності, який дозволяє зменшити кількість тестових випадків без втрати якості перевірки. Сутність методу полягає у поділі множини можливих вхідних даних на групи (класи), представники яких мають однакову поведінку під час обробки системою. Для кожного класу достатньо виконати перевірку одного або декількох типових значень.

Об'єктом тестування було обрано функцію «Надсилання звернення», оскільки вона є однією з основних функцій взаємодії користувача із вебзастосунком та містить перевірку коректності введених даних. Форма звернення включає поля для введення імені, адреси електронної пошти, номера телефону та тексту повідомлення.

На першому етапі було визначено правильні та неправильні класи еквівалентності для кожного поля форми. До правильних класів віднесено значення, що відповідають встановленим вимогам щодо формату та довжини даних, а до неправильних - порожні значення, некоректний формат, перевищення максимально допустимої довжини та введення недопустимих символів. Правильні та неправильні класи еквівалентності для кожного поля форми функції «Надсилання звернення» наведено у таблиці 2.1.

Таблиця 2.1 – Класи еквівалентності функції «Надсилання звернення»

Поле	Правильні класи	Неправильні класи
Ім'я	2–50 символів, літери	порожнє; <2 символів; >50 символів; цифри
Email	коректний email	порожній; без @; без домену
Телефон	+380XXXXXXXXX	порожній; менше цифр; літери
Повідомлення	10–1000 символів	порожнє; <10; >1000

На другому етапі було сформовано тестові випадки для правильних класів еквівалентності, які підтверджують коректне приймання даних системою та успішне створення звернення. Тести для правильних класів еквівалентності функції «Надсилання звернення» наведено у таблиці 2.2.

Таблиця 2.2 – Тести для правильних класів еквівалентності функції «Надсилання звернення»

№	Вхідні дані	Очікуваний результат
1	Коректне ім'я	Дані прийняті
2	Коректний email	Email прийнято
3	Коректний телефон	Телефон прийнято
4	Повідомлення 100 символів	Повідомлення прийнято
5	Усі поля коректні	Звернення успішно надіслано

На третьому етапі було виконано тестування неправильних класів еквівалентності. Для кожного тестового випадку перевірялося відображення повідомлень про помилки та недопущення збереження некоректних даних у системі. Тести для неправильних класів еквівалентності функції «Надсилання звернення» наведено у таблиці 2.3.

Таблиця 2.3 – Тести для неправильних класів еквівалентності функції «Надсилання звернення»

№	Вхідні дані	Очікуваний результат
1	Порожнє ім'я	Помилка "Заповніть поле"
2	Email "ivan.gmail.com"	Помилка формату email
3	Телефон "12345"	Помилка формату телефону
4	Порожнє повідомлення	Помилка "Введіть повідомлення"
5	Повідомлення >1000 символів	Помилка про перевищення довжини

За результатами проведеного тестування встановлено, що вебзастосунок коректно обробляє як правильні, так і помилкові вхідні дані. Усі перевірені функції відповідають визначеним функціональним вимогам. У разі введення

коректних даних звернення успішно надсилається та зберігається в системі, а при введенні некоректних даних користувач отримує відповідні повідомлення про помилки із зазначенням причин відхилення введеної інформації.

Отримані результати свідчать про правильність реалізації механізмів валідації даних та працездатність функції надсилання звернень. Це дозволяє зробити висновок, що розроблений вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» відповідає функціональним вимогам і може бути рекомендований до подальшої експлуатації.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ПРЕДСТАВНИЦТВА ТА ВЗАЄМОДІЇ З КЛІЄНТАМИ ДИЗАЙН-СТУДІЇ «CREATIVE_DS»

У кваліфікаційній роботі розроблений вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

Функціональним призначенням вебзастосунку дизайн-студії «Creative_DS» є представлення діяльності дизайн-студії, демонстрація портфоліо виконаних робіт, забезпечення взаємодії з потенційними клієнтами, організація зворотного зв'язку.

Експлуатаційним призначенням вебзастосунку дизайн-студії «Creative_DS» є формуванням власного іміджу, забезпечення зручного доступу до інформації про діяльність студії, реалізованим проектам та переліку послуг, покращення комунікації між студією та користувачами, спрощення процесу подання заявок і забезпечення швидкого зворотного зв'язку.

Розроблене програмне забезпечення задовольняє вимогам, поставленим в технічному завданні. З даним вебзастосунком мають можливість працювати: адміністратор та клієнт.

Функції адміністратора:

- додавання, редагування та видалення інформації про студію;
- керування портфоліо робіт;
- оновлення інформації про послуги;
- перегляд та обробка повідомлень від клієнтів;
- адміністрування контенту вебзастосунку;
- підтримка актуальності інформації на сайті.

Функції клієнта:

- перегляд інформації про дизайн-студію «Creative_DS»;
- ознайомлення з переліком послуг студії;
- перегляд портфоліо виконаних дизайнерських робіт;

- отримання контактної інформації.

Для розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» було обрано сучасний стек технологій, який включає HTML5, CSS3, JavaScript, Bootstrap 5, PHP 8.3, Laravel та систему керування базами даних MySQL. Зазначені технології є взаємосумісними, широко використовуються у веброзробці та мають значну спільноту розробників, що спрощує їх використання та підтримку.

Для забезпечення роботи вебзастосунку необхідно використовувати технічні засоби з наступними мінімальними характеристиками:

- Сервер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 20 ГБ;
 - мережеве підключення зі швидкістю 100 Мбіт/с.
- Користувацький комп'ютер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 500 МБ;
 - мережеве підключення зі швидкістю 10 Мбіт/с.

Для забезпечення роботи вебзастосунку необхідна наявність встановленого програмного забезпечення з наступними мінімальними характеристиками:

- Сервер:
 - операційна система: Windows Server 2022 або Ubuntu Server 22.04 LTS;
 - вебсервер: Apache HTTP Server 2.4 або Nginx 1.20;
 - PHP версії 8.3;
 - система керування базами даних MySQL 8.0.
- Користувацький комп'ютер:
 - операційна система: Windows 10 або Ubuntu 22.04 LTS;
 - сучасний веббраузер із підтримкою HTML5, CSS3 та JavaScript, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

4 ОХОРОНА ПРАЦІ

4.1 Загальні принципи охорони праці

Охорона праці – це система правових, соціально-економічних, організаційно-технічних та лікувально-профілактичних заходів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Питання охорони праці регулюються Законом України «Про охорону праці» [13] та Кодексом законів про працю України [14].

Основними принципами охорони праці є:

- пріоритет життя і здоров'я працівників відносно результатів виробничої діяльності;
- повна відповідальність роботодавця за створення безпечних умов праці;
- залучення працівників до участі в управлінні охороною праці;
- навчання працівників правилам безпеки перед початком роботи.

Відповідно до законодавства, кожен працівник, який використовує персональний комп'ютер (ПК), повинен пройти інструктаж та ознайомитися з інструкціями з експлуатації обладнання.

4.2. Аналіз небезпечних і шкідливих факторів при роботі з ПК

Робота за комп'ютером пов'язана з впливом низки факторів, що можуть негативно впливати на стан здоров'я персоналу. Основними шкідливими факторами є:

– Фізичні фактори:

- Електромагнітне випромінювання від моніторів (навіть при використанні сучасних LCD-панелей, рівні випромінювання потребують контролю).

- Статична електрика, що накопичується на поверхнях.
- Недостатній рівень освітленості робочого місця, що спричиняє швидке зорове стомлення.

– Психофізіологічні фактори:

- Перенапруження зорового апарату: тривалий фокус очей на екрані при роботі з великими масивами даних.
- Монотонність праці: необхідність введення великої кількості однотипних даних.
- Статичне навантаження: тривале перебування у вимушеній позі («сидяча» робота), що призводить до захворювань опорно-рухового апарату, зокрема остеохондрозу та порушень постави.

– Ергономічні фактори:

- Невідповідність параметрів меблів антропометричним даним працівника.
- Нераціональне розташування клавіатури, миші та монітора, що провокує втомлення кисті.

Вплив цих факторів без належного контролю призводить до швидкої втомлюваності, зниження концентрації уваги та підвищення ризику професійних захворювань [15].

4.3. Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів

Для мінімізації негативного впливу на працівників розроблено комплекс технічних та організаційних заходів. Реалізація цих кроків спрямована на усунення джерел шкідливого впливу та створення сприятливого середовища для ефективної роботи з програмними системами.

Організаційні заходи

Організаційні заходи спрямовані на регламентацію робочого процесу та підвищення рівня відповідальності персоналу:

– Регламентация режиму праці та відпочинку: Впроваджується обов’язковий графік, згідно з яким працівник після кожної години роботи за персональним комп’ютером повинен робити перерву тривалістю 10–15 хвилин. У цей час рекомендується виконувати спеціальний комплекс вправ для зняття напруги з м’язів спини, шийного відділу хребта та гімнастику для очей, що дозволяє попередити виникнення хронічної втоми.

– Навчання та інструктажі: З метою дотримання безпеки життєдіяльності та ергономіки передбачено проведення регулярних інструктажів [13, 14]. Кожен працівник повинен володіти знаннями щодо методів безпечної експлуатації офісного обладнання та правильного розподілу фізичних навантажень під час робочого дня.

– Контроль виробничого середовища: Запроваджується система періодичного моніторингу параметрів робочих місць, зокрема перевірка рівня освітленості на відповідність чинним санітарним нормам, що дозволяє своєчасно виявляти та усувати небезпечні умови праці.

Ергономіка та вимоги до робочого місця

Організація робочого місця повинна відповідати ергономічним вимогам, що мінімізують статичне навантаження на м’язи та зоровий апарат. Робоче місце повинно забезпечувати можливість виконання всіх операцій з урахуванням характеру праці.

Основні вимоги до елементів робочого місця представлені в таблиці 4.1.

Таблиця 4.1 – Основні вимоги до елементів робочого місця

Елемент обладнання	Вимоги до параметрів
1	2
Робочий стіл	Висота столу має бути регульованою (в межах 650–750 мм). Поверхня повинна мати матове покриття для відсутності відблисків.

Кінець таблиці 4.1

1	2
Крісло (стілець)	Має бути підйомно-поворотним, з регульованою висотою сидіння та кутом нахилу спинки. Обов'язкова наявність підлокітників.
Монітор	Розташовується на відстані 600–700 мм від очей. Центр екрана повинен бути на рівні очей або трохи нижче.
Клавіатура	Розміщується на відстані 100–300 мм від краю столу на користь користувача. Кут нахилу – від 5° до 15°.

Додаткові вимоги:

- Простір для ніг: Робочий стіл повинен мати вільний простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм.
- Освітлення: Робоче місце не повинно бути розташоване так, щоб пряме світло від вікна потрапляло в очі або створювало відблиски на екрані. Бажано використовувати світильники з розсіювачем для рівномірного розподілу світла.
- Положення тіла: При роботі користувач повинен мати можливість тримати спину рівно, ступні ніг мають повністю спиратися на підлогу або підставку для ніг.

Неправильне розташування монітора або невідповідна висота крісла призводять до викривлення постави (сколіозу) та виникнення хронічного больового синдрому в ділянці ший та попереку, що суттєво знижує продуктивність роботи персоналу [15].

Технічні заходи забезпечення безпеки

Технічні заходи спрямовані на вдосконалення матеріально-технічної бази для забезпечення вищого рівня захисту працівників:

- Модернізація обладнання: Впровадження ергономічних клавіатур та використання килимків зі спеціальними гелевими опорами для зап'ястя. Це

рішення дозволяє суттєво знизити навантаження на суглоби кисті, що є надійним засобом профілактики тунельного синдрому [15].

– Оптимізація світлового середовища: Використання комбінованого освітлення (поєднання природного денного світла та штучних джерел) забезпечує оптимальні умови для зорової роботи, виключаючи виникнення відблисків, які є головною причиною швидкої втоми очей при тривалому перегляді інформації на дисплеї.

– Інтеграція обладнання: Забезпечення можливості індивідуального налаштування кута нахилу монітора та висоти столів, що дозволяє адаптувати робоче місце під антропометричні параметри кожного конкретного працівника.

Дотримання впроваджених заходів дозволяє суттєво знизити вплив шкідливих факторів при роботі з ПК. Це не лише зберігає здоров'я персоналу, а й підвищує ефективність використання програмних систем, оскільки знижується рівень помилок при введенні даних через втому.

ВИСНОВКИ

Метою даної кваліфікаційної роботи було розв’язання практичної задачі інженерії програмного забезпечення, а саме розробка сучасного, зручного та функціонального вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

Для досягнення поставленої мети було виконано такі завдання:

- здійснено аналіз практичної задачі інженерії програмного забезпечення;
- досліджено існуючі програмні аналоги та обґрунтовано доцільність розробки власного вебзастосунку;
- визначено вимоги до програмного продукту та сформовано технічне завдання;
- спроектовано архітектуру вебзастосунку;
- реалізовано програмне забезпечення відповідно до поставлених вимог;
- виконано перевірку та тестування роботи вебзастосунку;
- виконано документування розробки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дизайн інтер'єру від Наталі Гуменюк. URL: <https://gumenyuk.design/> (дата звернення: 02.05.2026).
2. Студія дизайну «Alex Deco». URL: <https://alexdeco.org/> (дата звернення: 02.05.2026).
3. Дизайн-студія Романа Москаленка. URL: <https://design-moskalenko.com/> (дата звернення: 02.05.2026).
4. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide: Covers UML 2.0. Boston : Addison-Wesley Professional, 2015. 431 p.
5. Доценко С. І. Організація та системи керування базами даних : навч. посіб. Харків : Український державний університет залізничного транспорту, 2023. 117 с.
6. HTML Living Standard. URL: <https://html.spec.whatwg.org/> (дата звернення: 27.04.2026).
7. Cascading Style Sheets. URL: <https://www.w3.org/Style/CSS/> (дата звернення: 27.04.2026).
8. JavaScript. URL: <https://developer.mozilla.org/docs/Web/JavaScript> (дата звернення: 27.04.2026).
9. Get started with Bootstrap. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 27.04.2026).
10. PHP. Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 27.04.2026).
11. The clean stack for Artisans and agents. URL: <https://laravel.com/> (дата звернення: 27.04.2026).
12. MySQL. URL: <https://dev.mysql.com/doc/> (дата звернення: 27.04.2026).

13. Про охорону праці : Закон України від 14.10.1992 № 2694-XII. Ред. від 16.05.2024. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 25.04.2026).

14. Кодекс законів про працю України : Закон України від 10.12.1971 № 322-VIII. Ред. від 01.01.2026. URL: <https://zakon.rada.gov.ua/laws/show/322-08> (дата звернення: 25.04.2026).

15. Жидецький В. Ц. Основи охорони праці : підручник. 5-те вид., перероб. і допов. Львів : Афіша, 2014. 376 с.

Додаток А Технічне завдання на розробку вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Вступ

Повна назва програмного продукту: Вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

1 Підстави для розробки

Підставою для розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» є наказ ректора Національного Університету Кораблебудування імені адмірала Макарова на затвердження тем кваліфікаційних робіт бакалаврів зі спеціальності 121 «Інженерія програмного забезпечення».

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням вебзастосунку дизайн-студії «Creative_DS» є представлення діяльності дизайн-студії, демонстрація портфоліо виконаних робіт, забезпечення взаємодії з потенційними клієнтами, організація зворотного зв'язку.

2.2 Експлуатаційне призначення

Експлуатаційним призначенням вебзастосунку дизайн-студії «Creative_DS» є формуванням власного іміджу, забезпечення зручного доступу до інформації про діяльність студії, реалізованим проєктам та переліку послуг, покращення комунікації між студією та користувачами, спрощення процесу подання заявок і забезпечення швидкого зворотного зв'язку.

3 Вимоги до програмного виробу

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

З даним вебзастосунком повинні мати можливість працювати: адміністратор та клієнт.

Функції адміністратора:

- керування інформацією про студію;
- керування портфоліо робіт;
- керування послугами;
- перегляд та обробка звернень від клієнтів;
- керування контентом вебзастосунку;
- підтримка актуальності інформації на сайті.

Функції клієнта:

- перегляд інформації про дизайн-студію «Creative_DS»;
- перегляд послуг студії;
- перегляд портфоліо виконаних дизайнерських робіт;
- перегляд контактів;
- надсилання звернень через форму зворотного зв'язку.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вимоги до організації вхідних даних

Введення інформації до вебзастосунку здійснюється користувачами системи (клієнтами та адміністратором) через графічний вебінтерфейс із використанням стандартних засобів взаємодії, таких як клавіатура, комп'ютерна миша або сенсорний екран мобільних пристроїв.

Інформація може надходити до системи такими способами:

- ручне введення даних у текстові поля;
- вибір значень із доступних випадаючих списків;
- автоматичне отримання даних із бази даних вебзастосунку.

Вимоги до організації вихідних даних

Відображення результатів роботи системи здійснюється у режимі реального часу на екранах користувачів через вебінтерфейс. Користувачам надається актуальна інформація про діяльність дизайн-студії, портфоліо, послуги, контакти та результати взаємодії із системою.

Основу інформаційного забезпечення вебзастосунку будуть складати відомості про користувачів системи, відомості про дизайн-студію, дані про виконані дизайнерські роботи, включаючи зображення та опис, інформація про звернення користувачів через форму зворотного зв'язку

3.2 Вимоги до надійності

Надійне (стійке) функціонування вебзастосунку повинне бути забезпечене виконанням сукупності організаційно-технічних заходів, перелік яких наведений нижче:

- організацією безперебійного живлення технічних засобів;
- використанням ліцензійного програмного забезпечення;
- резервним копіюванням даних.

3.3 Вимоги до умов експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, пропонованим до технічних засобів у частині умов їхньої експлуатації.

Кінцеві користувачі програми повинні мати практичні навички роботи з сучасним ПК.

3.4 Вимоги до складу та параметрів технічних засобів

Для забезпечення роботи вебзастосунку необхідно використовувати технічні засоби з наступними мінімальними характеристиками:

- Сервер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;

- обсяг вільного простору на HDD – 20 ГБ;
- мережеве підключення зі швидкістю 100 Мбіт/с.
- Користувачський комп'ютер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 500 МБ;
 - мережеве підключення зі швидкістю 10 Мбіт/с.

3.5 Вимоги до інформаційної та програмної сумісності

Вимоги до інформаційної та програмної сумісності

Для забезпечення роботи вебзастосунку необхідна наявність встановленого програмного забезпечення з наступними мінімальними характеристиками:

- Сервер:
 - операційна система: Windows Server 2022 або Ubuntu Server 22.04 LTS;
 - вебсервер: Apache HTTP Server 2.4 або Nginx 1.20;
 - PHP версії 8.3;
 - система керування базами даних MySQL 8.0.
- Користувачський комп'ютер:
 - операційна система: Windows 10 або Ubuntu 22.04 LTS;
 - сучасний веббраузер із підтримкою HTML5, CSS3 та JavaScript, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

4 Вимоги до програмної документації

Повний пакет документів повинен включати:

- технічне завдання;
- опис програми;
- інструкцію користувача;
- текст програми;

– програму й методику випробувань.

5 Техніко-економічні показники

Техніко-економічні показники в даній роботі не розраховувалися.

6 Етапи розробки вебзастосунку

Етапи розробки представлені у таблиці А.1

Таблиця А.1 – Етапи розробки

Номер етапу	Назва етапу розробки	Термін виконання
1.	Аналіз практичної задачі інженерії ПЗ	20.02.2026
2.	Аналіз вимог до ПЗ	15.03.2026
3.	Розробка архітектури ПЗ	01.04.2026
4.	Розробка проєкту ПЗ	15.04.2026
5.	Реалізація та тестування ПЗ	28.04.2026

7 Порядок контролю та приймання

Приймання програми здійснюється згідно з вимогами програми та методики випробувань, представником замовника та представником виконавця. Результати випробувань заносяться до «Акту приймання програми». У разі невідповідності яким-небудь вимогам технічного завдання, це відображається в «Акті приймання програми» і вимагає доопрацювання розробником програми протягом одного тижня. Після виконання доопрацювання, проводиться повторне приймання програми. Представник замовника повідомляється про повторне приймання не пізніше 4 днів до початку випробувань.

Додаток Б Код вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Додаток містить фрагменти коду розробленого програмного забезпечення.
Наведений інтерфейс та база даних.

```
-- =====
-- Database: Creative_DS
-- DBMS: MySQL 8.x
-- =====

CREATE DATABASE IF NOT EXISTS Creative_DS
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;

USE Creative_DS;

-- =====
-- USERS
-- =====

CREATE TABLE users (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(150) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    ON UPDATE CURRENT_TIMESTAMP
);

-- =====
-- ADMINS
-- =====

CREATE TABLE admins (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id BIGINT UNSIGNED NOT NULL UNIQUE,
    role_name VARCHAR(50) NOT NULL DEFAULT 'admin',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT fk_admin_user
        FOREIGN KEY (user_id)
        REFERENCES users(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

```

-- =====
-- STUDIO INFO
-- =====

CREATE TABLE studio_info (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    description TEXT,
    admin_id BIGINT UNSIGNED NOT NULL,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP,

    CONSTRAINT fk_studio_admin
        FOREIGN KEY (admin_id)
        REFERENCES admins(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- =====
-- SERVICES
-- =====

CREATE TABLE services (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    description TEXT,
    price DECIMAL(10,2) NOT NULL,
    admin_id BIGINT UNSIGNED NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP,

    CONSTRAINT fk_service_admin
        FOREIGN KEY (admin_id)
        REFERENCES admins(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- =====
-- PORTFOLIO CATEGORIES
-- =====

CREATE TABLE portfolio_categories (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL UNIQUE,
    description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP
);

```

```

-- =====
-- PORTFOLIO ITEMS
-- =====

CREATE TABLE portfolio_items (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    description TEXT,
    image_path VARCHAR(255) NOT NULL,
    category_id BIGINT UNSIGNED NOT NULL,
    admin_id BIGINT UNSIGNED NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP,

    CONSTRAINT fk_portfolio_category
        FOREIGN KEY (category_id)
        REFERENCES portfolio_categories(id)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,

    CONSTRAINT fk_portfolio_admin
        FOREIGN KEY (admin_id)
        REFERENCES admins(id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- =====
-- MESSAGES
-- =====

CREATE TABLE messages (
    id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_name VARCHAR(100) NOT NULL,
    email VARCHAR(150) NOT NULL,
    message TEXT NOT NULL,

    status ENUM('new','in_progress','done')
        DEFAULT 'new',

    user_id BIGINT UNSIGNED NULL,
    admin_id BIGINT UNSIGNED NULL,

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP,

    CONSTRAINT fk_message_user
        FOREIGN KEY (user_id)
        REFERENCES users(id)
        ON DELETE SET NULL

```

```

        ON UPDATE CASCADE,

        CONSTRAINT fk_message_admin
            FOREIGN KEY (admin_id)
            REFERENCES admins(id)
            ON DELETE SET NULL
            ON UPDATE CASCADE
    );

-- =====
-- INDEXES
-- =====

CREATE INDEX idx_services_name
ON services(name);

CREATE INDEX idx_portfolio_title
ON portfolio_items(title);

CREATE INDEX idx_portfolio_category
ON portfolio_items(category_id);

CREATE INDEX idx_messages_status
ON messages(status);

CREATE INDEX idx_messages_created
ON messages(created_at);

<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Creative_DS - Портфоліо</title>
<link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/bootst
rap.min.css" rel="stylesheet">
<style>
.hero{background:#343a40;color:#fff;padding:60px 0}
.card img{height:220px;object-fit:cover}
footer{background:#222;color:#fff;padding:20px;margin-
top:40px}
</style>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container">
<a class="navbar-brand" href="#">Creative_DS</a>
<button class="navbar-toggler" data-bs-toggle="collapse"
data-bs-target="#nav"><span class="navbar-toggler-
icon"></span></button>

```

```

    <div class="collapse navbar-collapse" id="nav">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item"><a class="nav-link"
href="#">Головна</a></li>
        <li class="nav-item"><a class="nav-link"
href="#">Послуги</a></li>
        <li class="nav-item"><a class="nav-link active"
href="#">Портфоліо</a></li>
        <li class="nav-item"><a class="nav-link"
href="#">Контакти</a></li>
      </ul></div></div></nav>

<section class="hero text-center">
<div class="container">
<h1>Портфоліо дизайн-студії</h1>
<p>Добірка реалізованих проєктів Creative_DS.</p>
</div>
</section>

<div class="container my-5">
<div class="row mb-4">
<div class="col-md-4">
<select class="form-select">
<option>Усі категорії</option>
<option>Логотипи</option>
<option>Вебдизайн</option>
<option>Поліграфія</option>
</select>
</div>
</div>

<div class="row g-4">

<div class="col-md-4">
<div class="card h-100 shadow-sm">

<div class="card-body">
<h5 class="card-title">Проект 1</h5>
<p class="card-text">Короткий опис дизайнерської роботи.</p>
<button class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#m1">Детальніше</button>
</div>
</div>
</div>

<div class="modal fade" id="m1">
<div class="modal-dialog modal-lg"><div class="modal-
content">
<div class="modal-header"><h5 class="modal-title">Проект
1</h5>
<button class="btn-close" data-bs-
dismiss="modal"></button></div>

```

```

    <div class="modal-body">
    
    <p>Розширений опис виконаного проекту, використаних
технологій та дизайнерських рішень.</p>
    </div></div></div></div>

    <div class="col-md-4">
    <div class="card h-100 shadow-sm">
    
    <div class="card-body">
    <h5 class="card-title">Проект 2</h5>
    <p class="card-text">Короткий опис дизайнерської роботи.</p>
    <button class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#m2">Детальніше</button>
    </div>
    </div>
    </div>

    <div class="modal fade" id="m2">
    <div class="modal-dialog modal-lg"><div class="modal-
content">
    <div class="modal-header"><h5 class="modal-title">Проект
2</h5>
    <button class="btn-close" data-bs-
dismiss="modal"></button></div>
    <div class="modal-body">
    
    <p>Розширений опис виконаного проекту, використаних
технологій та дизайнерських рішень.</p>
    </div></div></div></div>

    <div class="col-md-4">
    <div class="card h-100 shadow-sm">
    
    <div class="card-body">
    <h5 class="card-title">Проект 3</h5>
    <p class="card-text">Короткий опис дизайнерської роботи.</p>
    <button class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#m3">Детальніше</button>
    </div>
    </div>
    </div>

    <div class="modal fade" id="m3">
    <div class="modal-dialog modal-lg"><div class="modal-
content">
    <div class="modal-header"><h5 class="modal-title">Проект
3</h5>

```

```

        <button class="btn-close" data-bs-
dismiss="modal"></button></div>
        <div class="modal-body">
            
            <p>Розширений опис виконаного проєкту, використаних
технологій та дизайнерських рішень.</p>
        </div></div></div></div>

        <div class="col-md-4">
            <div class="card h-100 shadow-sm">
                
                <div class="card-body">
                    <h5 class="card-title">Проект 4</h5>
                    <p class="card-text">Короткий опис дизайнерської роботи.</p>
                    <button class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#m4">Детальніше</button>
                </div>
            </div>
        </div>

        <div class="modal fade" id="m4">
            <div class="modal-dialog modal-lg"><div class="modal-
content">
                <div class="modal-header"><h5 class="modal-title">Проект
4</h5>
                <button class="btn-close" data-bs-
dismiss="modal"></button></div>
                <div class="modal-body">
                    
                    <p>Розширений опис виконаного проєкту, використаних
технологій та дизайнерських рішень.</p>
                </div></div></div></div>

        <div class="col-md-4">
            <div class="card h-100 shadow-sm">
                
                <div class="card-body">
                    <h5 class="card-title">Проект 5</h5>
                    <p class="card-text">Короткий опис дизайнерської роботи.</p>
                    <button class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#m5">Детальніше</button>
                </div>
            </div>
        </div>

        <div class="modal fade" id="m5">
            <div class="modal-dialog modal-lg"><div class="modal-
content">

```

```

    <div class="modal-header"><h5 class="modal-title">Проект
5</h5>
    <button class="btn-close" data-bs-
dismiss="modal"></button></div>
    <div class="modal-body">
    
    <p>Розширений опис виконаного проекту, використаних
технологій та дизайнерських рішень.</p>
    </div></div></div></div>

    <div class="col-md-4">
    <div class="card h-100 shadow-sm">
    
    <div class="card-body">
    <h5 class="card-title">Проект 6</h5>
    <p class="card-text">Короткий опис дизайнерської роботи.</p>
    <button class="btn btn-primary" data-bs-toggle="modal" data-
bs-target="#m6">Детальніше</button>
    </div>
    </div>
    </div>

    <div class="modal fade" id="m6">
    <div class="modal-dialog modal-lg"><div class="modal-
content">
    <div class="modal-header"><h5 class="modal-title">Проект
6</h5>
    <button class="btn-close" data-bs-
dismiss="modal"></button></div>
    <div class="modal-body">
    
    <p>Розширений опис виконаного проекту, використаних
технологій та дизайнерських рішень.</p>
    </div></div></div></div>

</div></div>
<footer><div class="container text-center">
<p>© 2026 Creative_DS. Усі права захищені.</p>
</div></footer>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstra
p.bundle.min.js"></script>
</body></html>

<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">

```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Creative_DS - Панель адміністратора</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootst
rap.min.css" rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet">
    <style>
    body{background:#f4f6f9}
    .sidebar{min-height:100vh;background:#212529}
    .sidebar a{color:#ddd;text-
decoration:none;display:block;padding:.8rem 1rem}
    .sidebar a:hover,.sidebar
a.active{background:#0d6efd;color:#fff}
    </style>
    </head><body>
    <div class="container-fluid">
    <div class="row">
    <nav class="col-md-2 sidebar">
    <h4 class="text-white p-3">Creative_DS</h4>
    <a class="active" href="#"><i class="bi bi-grid"></i>
Панель</a>
    <a href="#"><i class="bi bi-images"></i> Портфоліо</a>
    <a href="#"><i class="bi bi-briefcase"></i> Послуги</a>
    <a href="#"><i class="bi bi-envelope"></i> Повідомлення</a>
    <a href="#"><i class="bi bi-building"></i> Про студію</a>
    </nav>
    <main class="col-md-10 p-4">
    <div class="d-flex justify-content-between align-items-center
mb-4">
    <h2>Керування портфоліо</h2>
    <div><span class="badge bg-
primary">Адміністратор</span></div>
    </div>
    <div class="card">
    <div class="card-body">
    <div class="row mb-3">
    <div class="col-md-4"><input class="form-control"
placeholder="Пошук..."></div>
    <div class="col-md-3">
    <select class="form-select">
    <option>Усі
категорії</option><option>Логотипи</option><option>Вебдизайн</opti
on>
    </select>
    </div>
    <div class="col text-end">
    <button class="btn btn-success" data-bs-toggle="modal" data-
bs-target="#addModal"><i class="bi bi-plus-circle"></i> Додати
роботу</button>
    </div>
    </div>

```

```

<table class="table table-striped align-middle">
  <thead><tr><th>ID</th><th>Назва</th><th>Категорія</th><th>Дата</th><th>Статус</th><th>Дії</th></tr></thead>
  <tbody>
    <tr><td>1</td><td>Проект 1</td><td>Вебдизайн</td><td>2026-06-11</td><td><span class='badge bg-success'>Опубліковано</span></td><td><button class='btn btn-sm btn-warning'><i class='bi bi-pencil'></i></button> <button class='btn btn-sm btn-danger'><i class='bi bi-trash'></i></button></td></tr>
    <tr><td>2</td><td>Проект 2</td><td>Вебдизайн</td><td>2026-06-12</td><td><span class='badge bg-success'>Опубліковано</span></td><td><button class='btn btn-sm btn-warning'><i class='bi bi-pencil'></i></button> <button class='btn btn-sm btn-danger'><i class='bi bi-trash'></i></button></td></tr>
    <tr><td>3</td><td>Проект 3</td><td>Вебдизайн</td><td>2026-06-13</td><td><span class='badge bg-success'>Опубліковано</span></td><td><button class='btn btn-sm btn-warning'><i class='bi bi-pencil'></i></button> <button class='btn btn-sm btn-danger'><i class='bi bi-trash'></i></button></td></tr>
    <tr><td>4</td><td>Проект 4</td><td>Вебдизайн</td><td>2026-06-14</td><td><span class='badge bg-success'>Опубліковано</span></td><td><button class='btn btn-sm btn-warning'><i class='bi bi-pencil'></i></button> <button class='btn btn-sm btn-danger'><i class='bi bi-trash'></i></button></td></tr>
    <tr><td>5</td><td>Проект 5</td><td>Вебдизайн</td><td>2026-06-15</td><td><span class='badge bg-success'>Опубліковано</span></td><td><button class='btn btn-sm btn-warning'><i class='bi bi-pencil'></i></button> <button class='btn btn-sm btn-danger'><i class='bi bi-trash'></i></button></td></tr>
  </tbody></table>
  <nav><ul class="pagination justify-content-end">
    <li class="page-item disabled"><a class="page-link"><<</a></li>
    <li class="page-item active"><a class="page-link">1</a></li>
    <li class="page-item"><a class="page-link">2</a></li>
    <li class="page-item"><a class="page-link">>></a></li>
  </ul></nav>
</div></div>
</main></div></div>

<div class="modal fade" id="addModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header"><h5>Додати роботу</h5><button class="btn-close" data-bs-dismiss="modal"></button></div>
      <div class="modal-body">
        <div class="mb-3"><label class="form-label">Назва</label><input class="form-control"></div>
        <div class="mb-3"><label class="form-label">Категорія</label><select class="form-select"><option>Вебдизайн</option><option>Логотипи</option></select></div>

```

```

    <div class="mb-3"><label class="form-
label">Опис</label><textarea class="form-
control"></textarea></div>
    <div class="mb-3"><label class="form-
label">Зображення</label><input type="file" class="form-
control"></div>
  </div>
  <div class="modal-footer">
    <button class="btn btn-secondary" data-bs-
dismiss="modal">Скасувати</button>
    <button class="btn btn-primary">Зберегти</button>
  </div>
</div></div></div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstra
p.bundle.min.js"></script>
</body></html>

```

```
/* Creative_DS common styles */
```

```

:root{
  --primary:#0d6efd;
  --secondary:#6c757d;
  --success:#198754;
  --dark:#212529;
  --light:#f8f9fa;
  --body:#f4f6f9;
  --radius:12px;
  --shadow:0 6px 18px rgba(0,0,0,.08);
}

*{box-sizing:border-box}

body{
  font-family: "Segoe UI", Tahoma, Arial, sans-serif;
  background:var(--body);
  color:#333;
  line-height:1.6;
}

.navbar{
  box-shadow:var(--shadow);
}

.hero{
  background:linear-gradient(135deg,#212529,#495057);
  color:#fff;
  padding:70px 0;
}

.hero h1{

```

```

    font-weight:700;
}

.section-title{
    font-weight:700;
    margin-bottom:1.5rem;
}

.card{
    border:none;
    border-radius:var(--radius);
    overflow:hidden;
    box-shadow:var(--shadow);
    transition:.25s ease;
}

.card:hover{
    transform:translateY(-5px);
}

.card img{
    height:220px;
    object-fit:cover;
}

.btn{
    border-radius:10px;
}

.form-control,
.form-select{
    border-radius:10px;
}

.table{
    background:#fff;
}

.table thead{
    background:var(--dark);
    color:#fff;
}

.badge{
    font-size:.8rem;
}

footer{
    background:var(--dark);
    color:#fff;
    margin-top:50px;
    padding:25px 0;
}

```

```

.sidebar{
  min-height:100vh;
  background:var(--dark);
}
.sidebar h4{
  color:#fff;
  padding:1.25rem 1rem;
  margin:0;
}
.sidebar a{
  color:#dee2e6;
  display:block;
  text-decoration:none;
  padding:.85rem 1rem;
  transition:.2s;
}
.sidebar a:hover,
.sidebar a.active{
  background:var(--primary);
  color:#fff;
}
.admin-header{
  background:#fff;
  border-radius:var(--radius);
  padding:15px 20px;
  box-shadow:var(--shadow);
  margin-bottom:20px;
}
.content-card{
  background:#fff;
  border-radius:var(--radius);
  box-shadow:var(--shadow);
  padding:20px;
}
.pagination .page-link{
  border-radius:8px;
  margin:0 2px;
}

.modal-content{
  border-radius:var(--radius);
}
@media (max-width:768px){

.hero{
  padding:50px 0;
}
.card img{
  height:180px;
}

.sidebar{
  min-height:auto;
}

```

}

}

Додаток В Опис вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

1 Загальні відомості

Вебзастосунок призначений для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

Область застосування – як предметна галузь розглядається діяльність дизайн-студії «Creative_DS» з метою власного представництва в мережі інтернет, демонстрації портфоліо виконаних робіт, забезпечення взаємодії з потенційними клієнтами, організації зворотного зв'язку та спрощення процесу комунікації між студією та користувачами вебресурсу.

Для розробки вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» було обрано сучасний стек технологій, який включає HTML5, CSS3, JavaScript, Bootstrap 5, PHP 8.3, Laravel та систему керування базами даних MySQL. Зазначені технології є взаємосумісними, широко використовуються у веброзробці та мають значну спільноту розробників, що спрощує їх використання та підтримку.

Для забезпечення роботи вебзастосунку необхідна наявність встановленого програмного забезпечення з наступними мінімальними характеристиками:

- Сервер:
 - операційна система: Windows Server 2022 або Ubuntu Server 22.04 LTS;
 - вебсервер: Apache HTTP Server 2.4 або Nginx 1.20;
 - PHP версії 8.3;
 - система керування базами даних MySQL 8.0.
- Користувачський комп'ютер:
 - операційна система: Windows 10 або Ubuntu 22.04 LTS;

– сучасний веббраузер із підтримкою HTML5, CSS3 та JavaScript, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

2 Функціональне призначення розробки

Вебзастосунок виконує наступні основні функції:

Функції адміністратора:

- керування інформацією про студію;
- керування портфоліо робіт;
- керування послугами;
- перегляд та обробка звернень від клієнтів;
- керування контентом вебзастосунку;
- підтримка актуальності інформації на сайті.

Функції користувача:

- перегляд інформації про дизайн-студію «Creative_DS»;
- перегляд послуг студії;
- перегляд портфоліо виконаних дизайнерських робіт;
- перегляд контактів;
- надсилання звернень через форму зворотного зв'язку.

3 Опис логічної структури

Система побудована на основі багаторівневої архітектури, яка включає рівень представлення, рівень бізнес-логіки та рівень доступу до даних.

На рівні представлення знаходиться вебінтерфейс користувача, через який клієнти та адміністратор взаємодіють із системою. Рівень бізнес-логіки реалізовано за допомогою контролерів та сервісів, які забезпечують обробку запитів, реалізацію бізнес-правил та взаємодію з моделями даних. Рівень доступу до даних представлений моделями ORM, які забезпечують взаємодію з реляційною базою даних MySQL.

4 Технічні засоби

Для забезпечення роботи вебзастосунку необхідно використовувати технічні засоби з наступними мінімальними характеристиками:

– Сервер:

- процесор – двоядерний із тактовою частотою 2,0 ГГц;
- оперативна пам'ять – 4 ГБ;
- обсяг вільного простору на HDD – 20 ГБ;
- мережеве підключення зі швидкістю 100 Мбіт/с.

– Користувацький комп'ютер:

- процесор – двоядерний із тактовою частотою 2,0 ГГц;
- оперативна пам'ять – 4 ГБ;
- обсяг вільного простору на HDD – 500 МБ;
- мережеве підключення зі швидкістю 10 Мбіт/с.

5 Виклик і завантаження

Для запуску вебзастосунку необхідно відкрити встановлений на користувацькому комп'ютері веббраузер та набрати адресу сайту у рядку адреси.

6 Вхідні дані

Введення інформації до вебзастосунку здійснюється користувачами системи (клієнтами та адміністратором) через графічний вебінтерфейс із використанням стандартних засобів взаємодії, таких як клавіатура, комп'ютерна миша або сенсорний екран мобільних пристроїв.

Інформація може надходити до системи такими способами:

- ручне введення даних у текстові поля;
- вибір значень із доступних випадаючих списків;
- автоматичне отримання даних із бази даних вебзастосунку.

Основу інформаційного забезпечення вебзастосунку складають відомості про користувачів системи, відомості про дизайн-студію, дані про виконані

дизайнерські роботи, включаючи зображення та опис, інформація про звернення користувачів через форму зворотного зв'язку

7. Вихідні дані

Відображення результатів роботи системи здійснюється у режимі реального часу на екранах користувачів через вебінтерфейс. Користувачам надається актуальна інформація про діяльність дизайн-студії, портфоліо, послуги, контакти та результати взаємодії із системою.

Додаток Г Інструкція користувача вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

1 Призначення програми

1.1 Функціональне призначення програми

Функціональним призначенням вебзастосунку дизайн-студії «Creative_DS» є представлення діяльності дизайн-студії, демонстрація портфоліо виконаних робіт, забезпечення взаємодії з потенційними клієнтами, організація зворотного зв'язку.

1.2 Експлуатаційне призначення програми

Експлуатаційним призначенням вебзастосунку дизайн-студії «Creative_DS» є формуванням власного іміджу, забезпечення зручного доступу до інформації про діяльність студії, реалізовані проєкти та перелік послуг, покращення комунікації між студією та користувачами, спрощення процесу подання заявок і забезпечення швидкого зворотного зв'язку.

1.3 Склад функцій програми

Програмне забезпечення підтримує роботу двох категорій дійових осіб: адміністратора та клієнта.

Функції адміністратора:

- керування інформацією про студію;
- керування портфоліо робіт;
- керування послугами;
- перегляд та обробка звернень від клієнтів;
- керування контентом вебзастосунку;
- підтримка актуальності інформації на сайті.

Функції користувача:

- перегляд інформації про дизайн-студію «Creative_DS»;

- перегляд послуг студії;
- перегляд портфоліо виконаних дизайнерських робіт;
- перегляд контактів;
- надсилання звернень через форму зворотного зв'язку.

2 Умови виконання вебзастосунку

Вимоги до технічних засобів

Для забезпечення роботи вебзастосунку необхідно використовувати технічні засоби з наступними мінімальними характеристиками:

- Сервер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 20 ГБ;
 - мережеве підключення зі швидкістю 100 Мбіт/с.
- Користувацький комп'ютер:
 - процесор – двоядерний із тактовою частотою 2,0 ГГц;
 - оперативна пам'ять – 4 ГБ;
 - обсяг вільного простору на HDD – 500 МБ;
 - мережеве підключення зі швидкістю 10 Мбіт/с.

Вимоги до програмних засобів

Для забезпечення роботи вебзастосунку необхідна наявність встановленого програмного забезпечення з наступними мінімальними характеристиками:

- Сервер:
 - операційна система: Windows Server 2022 або Ubuntu Server 22.04 LTS;
 - вебсервер: Apache HTTP Server 2.4 або Nginx 1.20;
 - PHP версії 8.3;
 - система керування базами даних MySQL 8.0.
- Користувацький комп'ютер:
 - операційна система: Windows 10 або Ubuntu 22.04 LTS;

– сучасний веббраузер із підтримкою HTML5, CSS3 та JavaScript, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

3 Виконання програми

Під час розробки було створено зручний та інтуїтивно зрозумілий користувацький інтерфейс, що забезпечує комфортну взаємодію користувачів із вебзастосунком. Для адміністратора реалізовано функціонал управління інформаційним наповненням сайту, який дозволяє оперативно оновлювати дані без внесення змін до програмного коду.

Для запуску вебзастосунку необхідно відкрити встановлений на користувацькому комп'ютері веббраузер та набрати адресу сайту у рядку адреси.

Головна сторінка вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» наведена на рисунку Г.1.

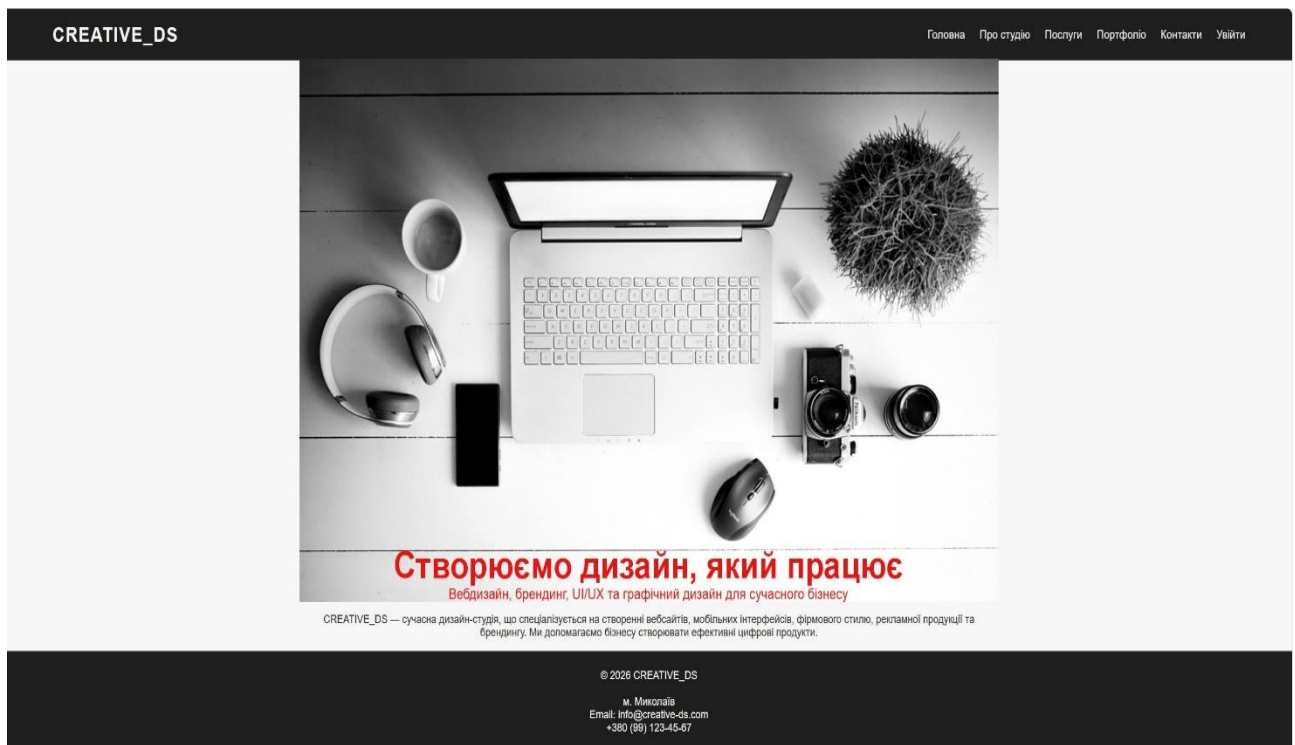


Рисунок Г.1 – Головна сторінка вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Сторінка «Контакти» вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» наведена на рисунку Г.2.

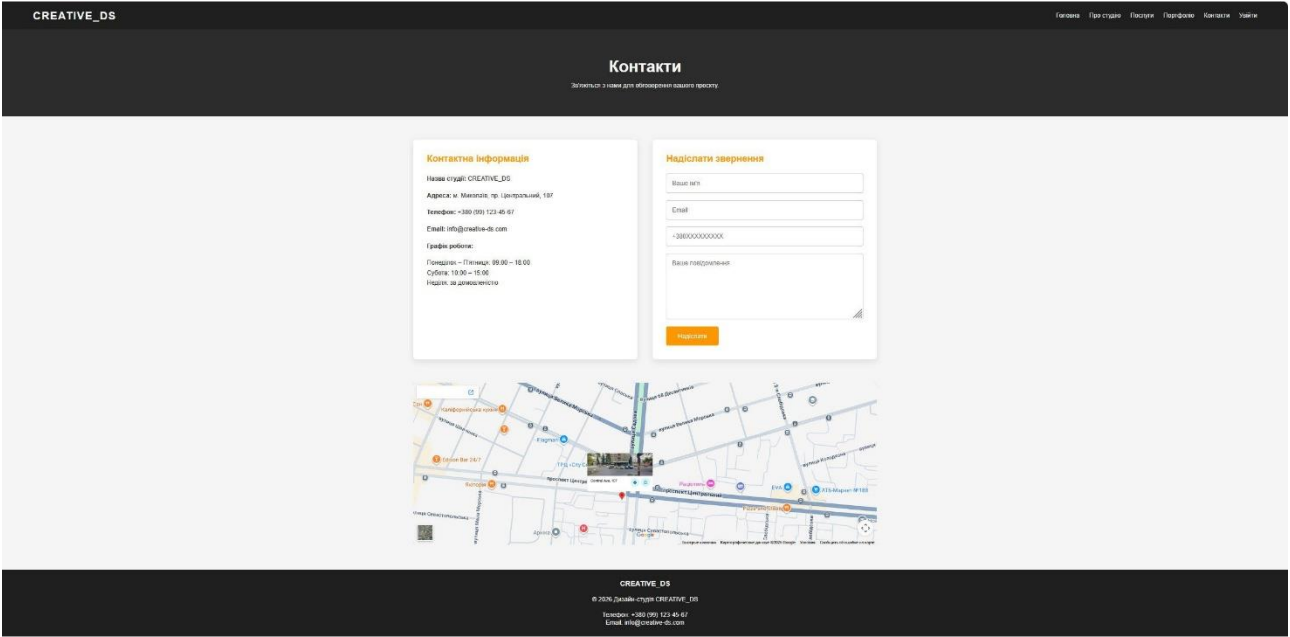


Рисунок Г.2 – Сторінка «Контакти» вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Сторінка «Панель адміністратора – керування портфоліо» вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS» наведена на рисунку Г.3.

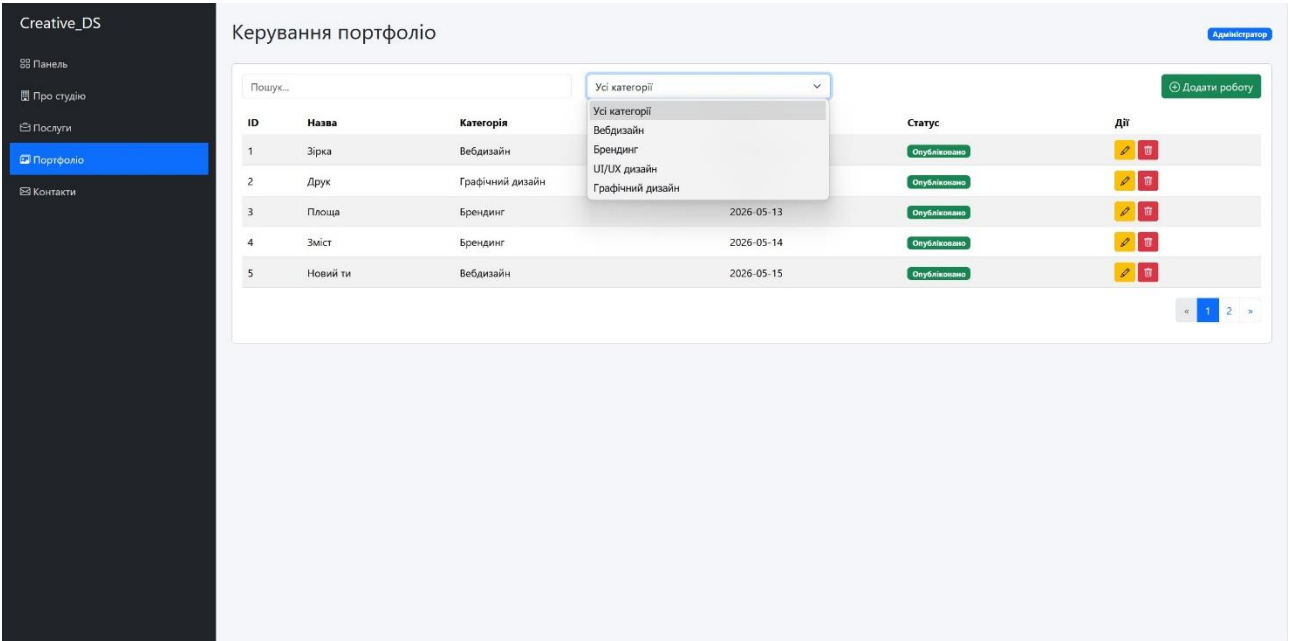


Рисунок Г.3 – Сторінка «Панель адміністратора – керування портфоліо» вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

Додаток Д Програма та методика випробування вебзастосунку для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS»

1 Об'єкт випробувань

Об'єктом випробувань є вебзастосунок для представництва та взаємодії з клієнтами дизайн-студії «Creative_DS».

Область застосування – як предметна галузь розглядається діяльність дизайн-студії «Creative_DS» з метою власного представництва в мережі інтернет, демонстрації портфоліо виконаних робіт, забезпечення взаємодії з потенційними клієнтами, організації зворотного зв'язку та спрощення процесу комунікації між студією та користувачами вебресурсу.

2 Мета випробувань

Метою випробувань є перевірка відповідності розробленого програмного забезпечення вимогам, які викладені в технічному завданні.

3 Вимоги до програми

Під час випробування програмного забезпечення необхідно перевірити, чи відповідають її функціональні характеристики тим вимогам, що зазначені у розділі «Вимоги до функціональних характеристик» технічного завдання.

4. Вимоги до програмної документації

Програмна документація, що надається на випробування, має містити:

- технічне завдання;
- опис програми;
- програму та методику випробувань;
- інструкцію користувача;
- тексти програмних модулів.

5 Засоби та порядок випробувань

Програмні засоби випробувань

Програмними засобами, необхідними для випробувань, є:

- операційна система: Windows 10 або Ubuntu 22.04 LTS;
- сучасний веббраузер із підтримкою HTML5, CSS3 та JavaScript, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

Апаратні засоби випробувань

Апаратними засобами, необхідними для випробувань, є:

- процесор – двоядерний із тактовою частотою 2,0 ГГц;
- оперативна пам'ять – 4 ГБ;
- обсяг вільного простору на HDD – 500 МБ;
- мережеве підключення зі швидкістю 10 Мбіт/с.

Порядок проведення випробувань

Проведення випробувань вебзастосунку має відбуватися в наступному порядку:

- виконання інструкцій до кожної функції;
- аналіз результатів і встановлення відповідності вебзастосунку вимогам і системних документам.

У випадку виявлення помилок у роботі, необхідно скласти перелік знайдених проблем і визначити терміни їх виправлення.

Після виправлення зазначених помилок, необхідно протестувати програмне забезпечення повторно у повному обсязі.

6 Методи випробувань

Випробування будемо проводити за стратегією «чорного ящика». Через велику кількість функцій буде представлено декілька результатів випробувань функцій програми. У таблиці Д.1 наведено програму випробувань вебзастосунку.

Таблиця Д.1 – Програма випробувань вебзастосунку

№	Об'єкт випробування	Методика проведення випробування	Очікуваний результат
1	Перегляд інформації про студію	Відкрити головну сторінку вебзастосунку	Відображається актуальна інформація про студію без помилок
2	Перегляд портфоліо	Перейти до сторінки «Портфоліо»	Відображається список проєктів із зображеннями та описом
3	Перегляд послуг	Відкрити сторінку «Послуги»	Відображається перелік послуг із відповідною інформацією
4	Перегляд контактів	Перейти до сторінки «Контакти»	Відображаються контактні дані студії
5	Надсилання звернення	Заповнити форму коректними даними та натиснути кнопку «Надіслати»	Звернення успішно збережено (або надіслано), користувач отримує повідомлення про успішне виконання
6	Авторизація адміністратора	Ввести коректний логін і пароль	Виконується успішна авторизація та відкривається панель адміністратора
7	Керування інформацією про студію	Увійти в адміністративну панель, змінити інформацію та зберегти зміни	Інформація успішно оновлюється та відображається на сайті
8	Керування послугами	Додати, змінити або видалити послугу	Зміни успішно зберігаються та відображаються користувачам
9	Керування портфоліо	Додати новий проєкт до портфоліо	Новий проєкт успішно додається та відображається на сторінці портфоліо
10	Перегляд звернень	Відкрити список звернень в адміністративній панелі	Відображається перелік отриманих звернень
11	Обробка звернень	Змінити статус звернення або позначити його як оброблене	Статус звернення успішно оновлюється
12	Керування контентом	Змінити текстовий або графічний контент сторінок	Зміни успішно зберігаються та відображаються після оновлення сторінки

За результатами проведених випробувань встановлено, що всі основні функціональні можливості вебзастосунку працюють відповідно до встановлених вимог. Під час виконання випробувань критичних помилок не виявлено, що свідчить про готовність програмного забезпечення до експлуатації.