

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Чорноморський національний університет
імені Петра Могили**

Факультет комп'ютерних наук

Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____ Ю. П. Кондратенко

« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

ІНТЕРНЕТ-ПОРТАЛ ПОРТФОЛІО ВЕБДИЗАЙНЕРА

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.22010227

Виконав студент 4-го курсу, групи 402

_____ *А. В. Федоренко*

«18» червня 2024 р.

*Керівник: професор, д-р техн. наук, професор
кафедри ІПЗ*

_____ *А. В. Швед*

«18» червня 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.

_____ Ю. П. Кондратенко
« ____ » _____ 2024 р.

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Видано студенту групи 402 факультету комп'ютерних наук Федоренку Андрію Володимировичу.

1. Тема кваліфікаційної роботи «Інтернет-портал портфоліо вебдизайнера».
- Керівник роботи Швед Альона Володимирівна, проф, д-р. техн. наук, професор.
Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2023 р. № 271
2. Строк представлення кваліфікаційної роботи студентом «18» червня 2024 р.
 3. Вхідні (початкові) дані до роботи: аналіз існуючих аналогів, технології та алгоритми розробки.
- Очікуваний результат: розробка інтернет-порталу.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз предметної області та аналогічних порталів;
- вибір технологій та фреймворків розробки;
- проектування та програмна реалізація інтернет-порталу;
- розробка бази даних, тестування та апробація вебзастосунку.

5. Перелік графічного матеріалу: слайди презентації.

6. Завдання до спеціальної частини: «Інтернет-портал портфоліо вебдизайнера»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва А. О., доцент кафедри екології	

Керівник роботи проф, д-р техн. наук, професор Швед А. В.

(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання Федоренко А. В.

(прізвище та ініціали) _____

(підпис)

Дата видачі завдання «14» січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Інтернет-портал портфоліо вебдизайнера

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників КРБ	10.11.2023	15.11.2023	Виконано
2	Отримання завдання на виконання КРБ	10.01.2024	15.01.2024	Виконано
3	Складання календарного плану роботи на весь період виконання КРБ	16.01.2024	30.01.2024	Виконано
4	Отримання завдання на переддипломну практику	15.04.2024	29.04.2024	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до КРБ	29.04.2024	11.05.2024	Виконано
6	Розробка звіту з переддипломної практики	12.05.2024	15.05.2024	Виконано
7	Виконання КРБ: огляд існуючих інтернет порталів, аналіз та вибір технологій розробки, створення бази даних та програмна реалізація	13.05.2024	22.06.2024	Виконано
8	Перший попередній захист КРБ на засіданні комісії кафедри	27.05.2024	27.05.2024	Виконано
9	Доробка та остаточне оформлення КРБ	28.05.2024	09.06.2024	Виконано
10	Другий попередній захист КРБ на засіданні комісії кафедри	10.06.2024	10.06.2024	Виконано
11	Подання КРБ рецензенту	13.06.2024	13.06.2024	Виконано
11	Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту	18.06.2024	18.06.2024	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	25.06.2024	25.06.2024	Виконано

Розробив студент Федоренко А. В.
(прізвище, ім'я, по батькові студента)

_____ (підпис)

Керівник роботи проф, д-р техн. наук, професор Швед А. В.
(посада, прізвище, ім'я, по батькові)

_____ (підпис)

« 29 » _____ 01 _____ 2024 р.

АНОТАЦІЯ

**кваліфікаційної роботи бакалавра групи 402 ЧНУ ім. Петра Могили
Федоренка Андрія Володимировича**

Тема: «Інтернет-портал портфоліо вебдизайнера»

Актуальність роботи полягає в активному розвитку сфери вебдизайну та необхідності таких ресурсів, як інтернет-портал на якому дизайнери зможуть розміщувати свої роботи для пошуку клієнтів та привертання уваги до своїх проєктів, обміну досвідом, отримання корисної інформації зі статей.

Об'єктом роботи є процес розробки інтернет-порталу.

Предметом роботи є програмні засоби та інформаційні технології розробки вебзатосунків.

Метою є допомога дизайнерам у розповсюдженні та популяризації їх проєктів та залучення клієнтів, за рахунок розробки інтернет-порталу.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі описано аналіз предметної області, огляд аналогів та постановку задачі.

У другому розділі було проаналізовано та зроблено вибір щодо технологій розробки, фреймворків, бази даних та сервісу для створення вебінтерфейсів.

У третьому розділі було виконано моделювання та проєктування порталу.

У четвертому розділі описана програмна реалізація інтернет порталу. Розробка основних сторінок та реалізація формування рейтингів, додавання робіт та створення адмін-панелі.

В результаті розроблено зручний та функціональний інтернет-портал.

Кваліфікаційна робота містить 60 сторінок, 24 рисунки, та 25 використаних джерел.

Ключові слова: інтернет-портал, вебдизайнер, вебсайт.

ABSTRACT

**to the bachelor's qualification work by the student of the group 402 of Petro Mohyla
Black Sea National University**

Fedorenko Andrii

Internet portal of a web designer's portfolio

The relevance of the work lies in the active development of the field of web design and the need for such resources as an Internet portal where designers can post their work to find clients and attract attention to their projects, exchange experience, and receive useful information from articles.

The object of the work is the process of developing an Internet portal.

The subject of the work is software and information technologies for the development of web applications.

The purpose of the work is to help designers in the distribution and popularization of their projects and attracting customers through the development of an Internet portal.

The qualification work consists of an introduction, four sections, conclusions and appendices.

The first chapter describes the analysis of the subject area, a review of analogues and the statement of the problem.

The second section analyzed and made a choice regarding development technologies, frameworks, databases and services for creating web interfaces.

In the third section, the portal was modeled and designed.

The fourth chapter describes the software implementation of the Internet portal. Development of the main pages and implementation of the formation of ratings, adding works and creating an admin panel.

As a result, a convenient and functional Internet portal was developed.

The qualification paper contains 60 pages, 24 figures, and 25 used sources.

Keywords: internet portal, web designer, website.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНТЕРНЕТ-ПОРТАЛІВ ТА ТЕХНОЛОГІЙ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Аналіз предметної області розробки онлайн-порталів.....	6
1.2 Аналіз існуючих онлайн-порталів для вебдизайнерів	7
1.3 Постановка задачі.....	11
Висновки до розділу 1	14
2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОНЛАЙН- ПОРТАЛУ	15
2.1 Вибір мови програмування	15
2.2 Вибір фреймворків	16
2.3 Вибір бази даних	20
2.4 Вибір сервісу для створення вебінтерфейсів	23
3 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ	28
3.1 Розробка діаграми варіантів використання	28
3.2 Розробка діаграми класів.....	29
3.3 Розробка діаграм послідовностей дій порталу.....	30
3.4 Розробка мокапів порталу	31
Висновки до розділу 3	34
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ	36
4.1 Створення дизайну основних сторінок сайту	36
4.2 Розробка функціональних розділів	37
4.3 Розробка адмін-панелі	46

4.4 Фізична модель бази даних	49
Висновки до розділу 4	52
ВИСНОВКИ.....	53
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	54
ДОДАТОК А Лістинг коду програмної реалізації інтернет-порталу	56

ПЕРЕЛІК СКОРОЧЕНЬ

СУБД	–	система управління базами даних
CSS	–	Cascading Style Sheets
MVC	–	Model-View-Controller
ORM	–	Object-Relational Mapping
PHP	–	Hypertext Preprocessor
UI	–	User Interface
UML	–	Unified Modeling Language
UX	–	User Experience

ВСТУП

Темою кваліфікаційної роботи є створення інтернет-порталу з портфоліо вебдизайнерів. Створення такого порталу є досить актуальним та корисним. Ось чому:

1) онлайн-портал може об'єднати вебдизайнерів з усього світу, кожен може зареєструватися та додати свої роботи;

2) вебдизайнери можуть викладати свої проекти на порталі, демонструючи їх потенційним клієнтам. Крім того, клієнти можуть залишати відгуки та рейтинги, що допоможе іншим користувачам оцінити професійність та якість роботи дизайнера;

3) важливою функцією порталу може бути публікація оглядів новітніх тенденцій у вебдизайні, а також новин та подій у галузі.

Створення такого порталу може значно полегшити роботу вебдизайнерів та покращити їхні можливості для розвитку та співпраці з майбутніми клієнтами.

Об'єктом роботи є процес розробки інтернет-порталу.

Предметом роботи є програмні засоби та інформаційні технології розробки вебзатосунків.

Метою роботи є допомога дизайнерам у розповсюдженні та популяризації їх проектів та залучення клієнтів, за рахунок розробки інтернет-порталу.

Завданням бакалаврської кваліфікаційної роботи є створення інтернет-порталу з портфоліо вебдизайнерів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНТЕРНЕТ-ПОРТАЛІВ ТА ТЕХНОЛОГІЙ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області розробки онлайн-порталів

Поговоримо про аналіз предметної області розробки онлайн-порталів для вебдизайнерів.

Онлайн-портал - це вебсайт або платформа, яка забезпечує доступ до різних ресурсів, інструментів та інформації через Інтернет. Вони можуть бути різних типів: загальні портали з різноманітною інформацією, спеціалізовані портали для певної аудиторії або індустрії, корпоративні портали для внутрішнього використання компаніями тощо [3].

Для чого вебдизайнерам онлайн-портали:

1) платформа для показу своїх робіт: онлайн-портал може служити вебдизайнерам як візитна картка, де вони можуть виставляти свої роботи, портфоліо, щоб потенційні клієнти могли переглядати їхній стиль та компетенції;

2) інструменти для співпраці: портали можуть включати інструменти для співпраці, спрощуючи комунікацію між вебдизайнерами та їх клієнтами, обмін файлами, коментарі та зворотний зв'язок [3];

3) навчання та розвиток: деякі портали можуть пропонувати навчальні ресурси, форуми або спільноти для вебдизайнерів, де вони можуть обмінюватися досвідом, вивчати нові техніки та ідеї.

Тенденції та особливості ринку:

1) мобільна оптимізація: зростаюча кількість користувачів переходять на мобільні пристрої, тому важливо, щоб портали були добре оптимізовані для мобільних платформ;

2) естетика та UX/UI дизайн: користувачі стають все вимогливішими до дизайну та взаємодії з сайтами, тому важливо, щоб портали були естетично привабливими та мали інтуїтивно зрозумілий інтерфейс;

3) персоналізація та аналітика: ключовими функціями можуть бути можливості персоналізації для користувачів, які шукають вебдизайнерів, а також інструменти аналітики, щоб вебдизайнери могли відстежувати взаємодію зі своїми ресурсами [20].

Конкурентне середовище:

1) популярні платформи для вебдизайнерів: існує кілька відомих платформ, таких як Behance, Dribbble, або Adobe Portfolio, які надають вебдизайнерам можливість показати свої роботи та взаємодіяти зі спільнотою;

2) нові ініціативи та стартапи: на ринку також можуть з'являтися нові стартапи або ініціативи, які пропонують інноваційні підходи до розвитку онлайн-порталів для вебдизайнерів, наприклад, з використанням штучного інтелекту або блокчейн технологій.

Аналіз цієї предметної області може допомогти вебдизайнерам зрозуміти потреби своєї аудиторії та знайти оптимальні рішення для представлення своїх навичок та робіт в Інтернеті.

1.2 Аналіз існуючих онлайн-порталів для вебдизайнерів

Детальний аналіз онлайн-порталів вебдизайнерів може включати наступні аспекти.

Дизайн та використання користувацького інтерфейсу (UI/UX) [23]:

1) оцінка естетики та функціональності вебсайту. Чи є сучасним та привабливим вигляд? Чи зручно навігувати;

2) UX. Аналіз користувацького досвіду. Чи забезпечує сайт інтуїтивний та ефективний досвід для користувачів? Чи є шляхи для покращення зручності використання?

Вміст та інформаційна якість:

1) оцінка різноманітності, актуальності та якості статей та інших матеріалів про вебдизайн;

2) аналіз проєктів, які відображаються на порталі. Чи є різноманітність стилів, платформ та типів проєктів? Чи якісно представлені роботи в портфоліо.

Технічні характеристики:

1) продуктивність та швидкість, чи завантажується вебсайт швидко? Чи відповідає він сучасним вимогам продуктивності;

2) сумісність та адаптивність, чи коректно відображається вебсайт на різних пристроях та браузерах? Чи дотримується він стандартів доступності та реагує на різні розміри екранів?

Функціональність та інтерактивність:

1) чи надає вебсайт корисний та різноманітний функціонал для користувачів, такий як фільтри для пошуку, можливість зберігання улюблених проєктів, можливість зв'язку з дизайнерами тощо?

2) чи використовуються на сайті ефективні інтерактивні елементи, такі як анімація, прокрутка, плавне завантаження сторінок тощо?

Спільнота та взаємодія:

1) чи є можливість обміну думками та досвідом між вебдизайнерами за допомогою форумів та обговорень? Чи активні та залучені користувачі в спільноті?

2) чи присутня вебдизайнерська спільнота на соціальних мережах? Чи публікуються актуальні новини та інші цікаві матеріали на блозі порталу?

Ці аспекти допомагають здійснити повний аналіз та оцінку якості та корисності онлайн-порталів вебдизайнерів для користувачів.

Розглянемо кілька існуючих порталів для вебдизайнерів та проведемо їх аналіз:

Behance:

1) Behance відомий своїм різноманіттям та високою якістю проєктів у портфоліо. Тут можна знайти різні типи дизайну, від вебсайтів до ілюстрацій;

2) портал пропонує користувачам високу інтерактивність - можливість відслідковувати роботи улюблених дизайнерів, коментувати проєкти та навіть співпрацювати над проєктами;

3) UI/UX [23]. Behance має сучасний та привабливий дизайн, який дозволяє користувачам легко знаходити та переглядати проекти;

4) доступність та вартість. Портал є безкоштовним для користувачів.

Нижче зображено фрагмент головної сторінки порталу Behance (рис. 1.1).

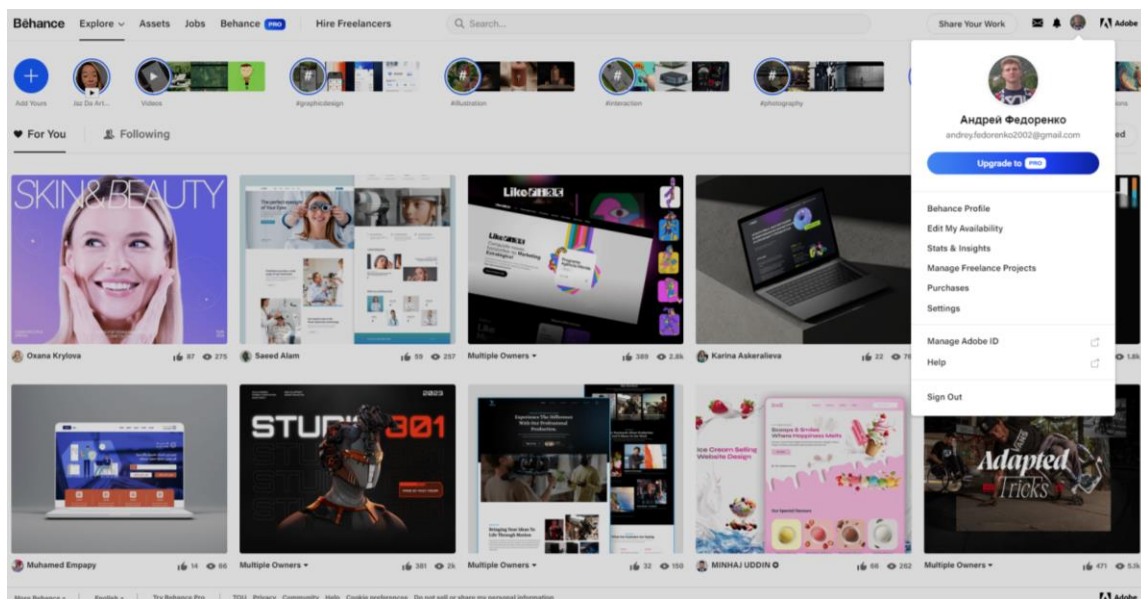


Рисунок 1.1 – Головна сторінка порталу Behance

Dribbble:

1) вміст та якість матеріалів Dribbble славиться своїми короткими переглядовими зразками дизайну. Тут можна знайти велику кількість візуально привабливих проєктів;

2) портал дозволяє користувачам створювати та взаємодіяти зі спільнотою, ділитися своїми роботами та отримувати відгуки;

3) дизайн та використання користувацького інтерфейсу (UI/UX). Dribbble має чистий та стильний дизайн, що спрощує перегляд та пошук робіт;

4) деякі функції можуть бути доступні тільки за плату, але базовий доступ є безкоштовним.

Нижче зображено фрагмент головної сторінки порталу Dribbble (рис. 1.2).

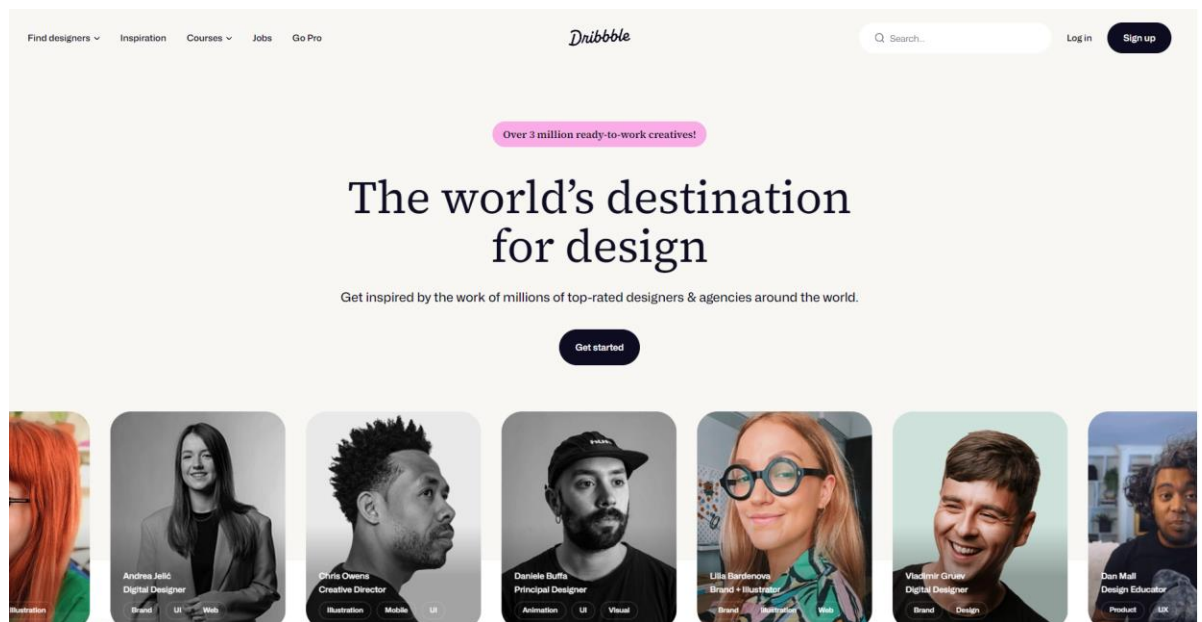


Рисунок 1.2 – Головна сторінка порталу Dribbble

Awwwards:

- 1) Awwwards визначає та нагороджує найкращі вебдизайни. Тут можна знайти надзвичайно якісні та творчі роботи;
- 2) портал дозволяє користувачам голосувати за проекти, додавати їх до обраного, а також отримувати натхнення від найкращих дизайнів;
- 3) Awwwards має стильний та досить простий дизайн, який забезпечує зручний перегляд та голосування за роботи;
- 4) деякі функції, такі як доступ до додаткового контенту, можуть бути доступні за плату.

Нижче зображено фрагмент головної сторінки порталу Awwwards (рис. 1.3).

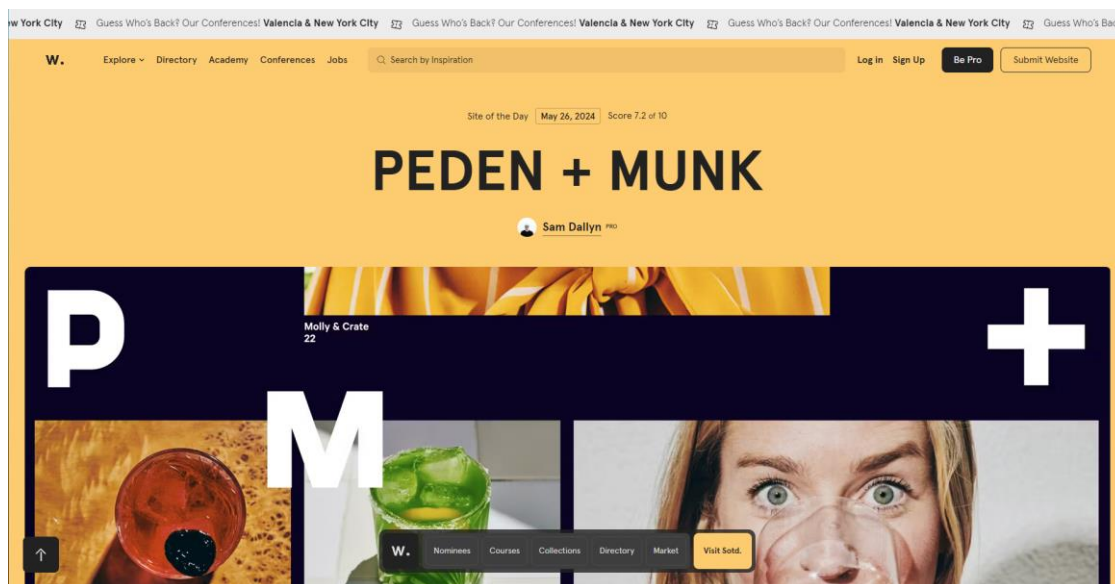


Рисунок 1.3 – Головна сторінка порталу Awwards

1.3 Постановка задачі

Перед початком розробки інтернет-порталу, постали такі основні задачі:

- 1) розробити дизайн сторінок вебсайту для верстання;
- 2) визначитись з мовою програмування та фреймворком для створення порталу;
- 3) визначитись з системою бази даних;
- 4) реалізувати основні алгоритми та функціонал порталу.

Постановка задачі: створення онлайн-порталу

Розробка онлайн-порталу, де користувачі можуть реєструватися, додавати статті, створювати профілі дизайнерів та додавати їхні роботи. Користувачі також повинні мати можливість оцінювати дизайнерів, залишаючи відгуки та рейтинги.

Основні вимоги.

1. Реєстрація користувачів.
2. Авторизація користувачів.
3. Управління профілями користувачів.
4. Додавання та управління статтями.
5. Створення профілів дизайнерів.

6. Додавання робіт дизайнерів.
7. Оцінка та рейтинг дизайнерів.
8. Адміністративна панель.

Функціональні вимоги.

Реєстрація користувачів:

- 1) форма реєстрації в якій вказуються електронна пошта та пароль.

Авторизація користувачів:

- 1) форма авторизації з електронною поштою/іменем користувача та паролем;
- 2) механізм для відновлення паролю через електронну пошту.

Управління профілями користувачів-дизайнерів:

- 1) редагування імені, прізвища, фото профілю, біографії може здійснювати як зареєстрований користувач так і головний адміністратор порталу;
- 2) головний адміністратор має можливість видаляти профілі зареєстрованих користувачів.

Додавання та управління статтями:

- 1) форма додавання статей із полями заголовку, тексту, категорії, дати публікації, зображення;
- 2) можливість редагування та видалення власних статей;
- 3) можливість додавання коментарів до статей.

Додавання робіт дизайнерів:

- 1) форма додавання роботи з назвою, описом, категорією та зображенням;
- 2) можливість редагування та видалення робіт.

Оцінка та рейтинг дизайнерів:

- 1) можливість залишати позитивні оцінки;
- 2) відображення середнього рейтингу для кожного дизайнера.

Адміністративна панель:

- 1) керування користувачами: перегляд, редагування та видалення користувачів;
- 2) керування статтями: модерація, редагування та видалення статей;

3) керування дизайнерами та їх роботами: перегляд, редагування та видалення профілів дизайнерів та їх робіт.

Безпека даних:

- 1) шифрування паролів: зберігання паролів у зашифрованому вигляді.
- 2) захист від SQL-ін'єкцій: використання підготовлених запитів.

Нефункціональні вимоги.

Продуктивність:

- 1) швидке завантаження сторінок: оптимізація запитів до бази даних та використання кешування.
- 2) масштабованість: можливість масштабування серверної інфраструктури для підтримки великої кількості користувачів.

Юзабіліті:

- 1) зручний інтерфейс: інтуїтивно зрозумілий користувацький інтерфейс та мінімалістичний дизайн;
- 2) мобільна адаптивність: адаптивний дизайн для зручного використання на мобільних пристроях.

Технологічні вимоги:

- 1) серверна частина реалізована на мові програмування PHP з використанням фреймворку Laravel;
- 2) у клієнтській, фронтенд частині використані HTML, CSS, Tailwind CSS;
- 3) база даних - MySQL.

Вимоги до технічного забезпечення.

Відсутні жорсткі вимоги до технічного забезпечення. Персональний комп'ютер чи ноутбук з технічними характеристиками, що підтримують будь-який сучасний браузер.

Апаратний інтерфейс.

Апаратним інтерфейсом є пристрій користувача (ПК чи мобільний пристрій), який буде використано для роботи з порталом.

Загальні обмеження.

Обмеження для роботи з ПЗ – наявність доступу до мережі Інтернет.

Ця постановка задачі охоплює всі необхідні функціональні та нефункціональні вимоги для створення повноцінного онлайн-порталу з можливістю реєстрації користувачів, додавання статей, створення профілів дизайнерів, додавання їх робіт та оцінювання дизайнерів.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи було виконано аналіз предметної області, огляд та аналіз аналогів та постановка задачі для розробки.

Аналіз предметної області дозволяє зрозуміти, в якій сфері буде здійснюватися розробка, які існують тенденції, вимоги та особливості. Це створює фундамент для розробки, забезпечуючи, що вона буде відповідати актуальним потребам та специфіці галузі.

Огляд та аналіз аналогів дозволяє виявити існуючі рішення, ідентифікувати їх сильні та слабкі сторони, а також визначити ринкові ніші для інновацій.

Постановка задачі формулює чіткі цілі та завдання розробки, забезпечуючи фокус на конкретних аспектах, визначення функціональних вимог та оцінку успішності проекту. Ці етапи також мінімізують ризики, дозволяють планувати ресурси і час при роботі.

2 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ОНЛАЙН-ПОРТАЛУ

2.1 Вибір мови програмування

PHP (Hypertext Preprocessor) - це мова програмування загального призначення, яка особливо популярна для розробки вебсайтів [1]. Ось кілька причин, чому PHP є гарним вибором для створення вебсайтів.

1. Простота вивчення та використання: PHP вважається досить легкою мовою програмування для вивчення, особливо для тих, хто вже має деякий досвід у веброзробці. Синтаксис PHP подібний до синтаксису мови C, тому багато програмістів знайомі з цим стилем програмування.

2. Велика спільнота та підтримка. PHP має велику та активну спільноту розробників. Це означає, що завжди є багато ресурсів, документації, форумів та інших джерел підтримки, які можна використовувати для отримання допомоги або порад.

3. Широкі можливості. PHP має багатий функціональний набір, що дозволяє розробникам створювати різноманітні вебсайти, включаючи блоги, електронні комерційні платформи, соціальні мережі та інші типи вебдодатків [5].

4. Швидкість. PHP зазвичай працює досить швидко, що робить його ідеальним для вебсайтів з великою кількістю відвідувачів. Багато вебсерверів, таких як Apache та Nginx, підтримують PHP та дозволяють встановлювати оптимізації для підвищення швидкості роботи вебсайту. Він також підтримує багато різних систем управління базами даних, таких як MySQL, PostgreSQL, SQLite та інші, що робить його універсальним інструментом для веброзробки.

5. Безкоштовність та відкритість. PHP є вільним програмним забезпеченням, що означає, що ви можете використовувати його безкоштовно. Крім того, відкритий код PHP дозволяє розробникам вносити зміни та покращення до мови, що сприяє її постійному розвитку та вдосконаленню [6].

Загалом, PHP є потужним інструментом для розробки вебсайтів, який може

задовольнити потреби різних користувачів, від початківців до досвідчених професіоналів. Його простота, потужність та широкий функціонал роблять його одним із найпопулярніших виборів для веброзробників у всьому світі [7].

2.2 Вибір фреймворків

Laravel

Переваги:

1) Laravel відомий своїм чистим і зрозумілим синтаксисом, що полегшує розробку;

2) містить багато інструментів для типових задач, таких як маршрутизація, сесії, кешування, аутентифікація;

3) велика екосистема з пакетами, такими як Laravel Nova, Laravel Horizon, Laravel Dusk;

4) відмінна документація та велика спільнота користувачів.

Недоліки:

1) може вимагати більше серверних ресурсів порівняно з деякими іншими фреймворками;

2) може бути складним для новачків, які не знайомі з концепціями MVC або іншими сучасними патернами.

Symfony

Переваги:

1) дуже гнучкий фреймворк, що дозволяє використовувати окремі компоненти або весь фреймворк;

2) Symfony має професійну підтримку та комерційні послуги від SensioLabs;

3) відмінна документація та велика кількість навчальних ресурсів;

4) дотримання найкращих стандартів та практик PHP.

Недоліки:

1) відмінна документація та велика кількість навчальних ресурсів;

2) дотримання найкращих стандартів та практик РНР висока складність та довге навчання, особливо для новачків;

3) може бути менш продуктивним на невеликих проєктах через складність налаштування.

CodeIgniter

Переваги:

1) простий та легкий фреймворк, який легко освоїти;

2) висока продуктивність та мала вимогливість до ресурсів;

3) легко налаштовується та розширюється.

Недоліки:

1) менше вбудованих функцій порівняно з Laravel чи Symfony;

2) менша спільнота та менше навчальних ресурсів у порівнянні з популярнішими фреймворками.

Yii

Переваги:

1) пропонує високу швидкість розробки завдяки генераторам коду та зручним інструментам;

2) вбудовані засоби для забезпечення безпеки, такі як захист від SQL-ін'єкцій та XSS;

3) хороша документація та навчальні матеріали.

Недоліки:

1) може мати круту криву навчання для новачків;

2) менша популярність порівняно з Laravel та Symfony, що може означати менше ресурсів та підтримки від спільноти.

З перерахованих фреймворків РНР вибір був зроблений на користь Laravel як найбільш зрозумілого та розповсюдженого.

Laravel - це один з найпопулярніших та потужних вебфреймворків для розробки вебдодатків на мові програмування РНР. Він був випущений у 2011 році та швидко завоював велику популярність завдяки своїй простоті використання,

багатим функціоналом та ефективним механізмом розробки.

Елегантний синтаксис та зрозуміла структура коду. Laravel надає зручний та читабельний синтаксис, який дозволяє розробникам швидко створювати та розуміти код. Він пропонує чітку структуру каталогів та організацію файлів, що спрощує розробку та підтримку проєктів.

Маршрутизація та контролери Laravel надає потужний механізм маршрутизації, що дозволяє легко визначати URL-адреси та зв'язувати їх з контролерами.

Контролери дозволяють групувати логіку пов'язану з відповідними ділянками додатку та обробляти запити від користувачів.

Eloquent ORM. Laravel включає Eloquent ORM, що надає простий та елегантний спосіб взаємодії з базою даних. Він дозволяє використовувати звичні методи PHP для роботи з даними, такі як створення, читання, оновлення та видалення записів в базі даних.

Шаблонізація та Blade. Laravel має власний механізм шаблонізації, відомий як Blade, який дозволяє легко створювати та управляти шаблонами вебсторінок.

Blade надає можливість використовувати зручні директиви та включення, що спрощує створення та підтримку інтерфейсів користувача.

Автентифікація та авторизація. Laravel має вбудовану систему автентифікації та авторизації, яка дозволяє легко налаштовувати механізми аутентифікації користувачів. Це дозволяє створювати безпечні вебдодатки з функціями реєстрації, входу, виходу та керування правами доступу.

Міграції та сіди. Laravel пропонує механізми міграцій та сидів, які дозволяють легко керувати структурою бази даних та наповнювати її тестовими даними. Це робить процес розгортання та оновлення бази даних більш простим та безпечним.

Кешування та оптимізація продуктивності. Laravel має вбудовану підтримку різних механізмів кешування, таких як Memcached та Redis, для підвищення продуктивності додатків [11].

Також надаються інші інструменти та оптимізації, такі як відслідковування

запитів та логування, для покращення продуктивності та відладки додатків.

Це лише декілька основних особливостей Laravel, і фреймворк має ще багато інших функцій та можливостей, що роблять його одним з найпопулярніших виборів для розробки вебдодатків на PHP.

Нижче зображено логотип фреймворку Laravel (рис. 2.1).



Рисунок 2.1 – Логотип фреймворку Laravel [9]

2.2.1. Css-фреймворк Tailwind CSS

Tailwind CSS - це потужний та популярний CSS-фреймворк, який дозволяє розробникам швидко створювати стильні та сучасні вебінтерфейси. Ось детальний опис особливостей та переваг Tailwind CSS.

Унікальний підхід до дизайну:

1) Tailwind CSS відрізняється від інших CSS-фреймворків своїм унікальним підходом до створення інтерфейсів. Замість написання CSS-класів для кожного елемента окремо, Tailwind CSS надає набір універсальних класів, які можна застосувати безпосередньо до HTML-елементів.

Масштабованість та гнучкість:

1) завдяки своїй концепції класів, Tailwind CSS дозволяє швидко створювати та масштабувати вебінтерфейси без необхідності написання власного CSS-коду;

2) він також надає гнучкість для кастомізації за допомогою конфігураційних файлів та можливості перевизначення значень змінних.

Набір компонентів:

1) Tailwind CSS поставляється з набором вбудованих компонентів та утиліт, таких як кнопки, картки, форми, навігація та багато інших, що полегшують створення складних вебінтерфейсів [25].

Документація та ресурси:

1) фреймворк має докладну та зрозумілу документацію, яка допомагає новачкам швидко засвоїти основні концепції та можливості;

2) також існує багато онлайн-ресурсів, відеоуроків та спільнот, які надають додаткові матеріали та поради щодо використання Tailwind CSS.

Інтеграція та розширення:

1) Tailwind CSS можна легко інтегрувати з іншими інструментами та фреймворками, такими як React, Vue.js, або Angular;

2) крім того, існують різні розширення та плагіни для Tailwind CSS, які розширюють його функціональність та додають нові можливості.

Швидкість та продуктивність:

1) Tailwind CSS створений з урахуванням продуктивності та швидкості розробки. Використання готових класів дозволяє зберігати час на написанні та відлагодженні CSS-коду.

Загалом, Tailwind CSS є потужним інструментом для швидкої та ефективної розробки стильних вебінтерфейсів, що зробив його дуже популярним серед веброзробників.

2.3 Вибір бази даних

Вибір бази даних є важливим компонентом для зберігання, управління та аналізу даних у багатьох додатках. Існують різні типи баз даних, кожен з яких має свої переваги та недоліки. Далі проведемо аналіз основних типів баз даних.

Реляційні бази даних (RDBMS)

MySQL.

Переваги:

1) широко використовується, велика спільнота користувачів;

2) безкоштовне використання під ліцензією GPL;

3) оптимізована особливо для вебзастосунків.

Недоліки:

1) може мати обмеження при роботі з дуже великими обсягами даних;

2) деякі функції, як-от повний текстовий пошук, можуть бути обмеженими у безкоштовній версії.

PostgreSQL

Переваги:

1) підтримка складних запитів, тригерів, процедур і т.д;

2) безкоштовне використання під ліцензією PostgreSQL;

3) підтримує великі бази даних і високу навантаженість.

Недоліки:

1) може бути повільніше ніж MySQL у певних сценаріях;

2) може бути складнішою в налаштуванні та управлінні для новачків.

NoSQL бази даних

MongoDB

Переваги:

1) дозволяє зберігати дані в неструктурованій формі;

2) легко масштабується горизонтально через шардінг;

3) висока продуктивність для записів і читань.

Недоліки:

1) менш потужні запити в порівнянні з SQL;

2) підтримка ACID транзакцій була додана відносно недавно і не така сильна, як у реляційних базах даних.

Cassandra

Переваги:

1) спроектована для масштабування у великих розподілених системах;

2) висока доступність і відсутність єдиної точки відмови;

3) висока продуктивність для великих обсягів даних.

Недоліки:

- 1) вимагає значних зусиль для налаштування та підтримки;
- 2) не підтримує складні запити, як реляційні бази даних.

З вищезазначеного переліку баз даних вибір при розробці порталу пав на MySQL.

MySQL є однією з найпопулярніших відкритих систем управління базами даних (СУБД), що використовується для зберігання та управління великими обсягами даних у різноманітних типах додатків, від вебсайтів до корпоративних систем. Переваги MySQL починаються з її відкритості та безкоштовності. Це означає, що будь-який може завантажити, використовувати і розповсюджувати MySQL безкоштовно, що дозволяє розробникам та організаціям економити витрати на ліцензії програмного забезпечення [17].

Однією з ключових переваг MySQL є його висока швидкодія та ефективність. Вона дозволяє виконувати швидкі та ефективні операції зберігання та отримання даних, що робить її ідеальним вибором для високонавантажених додатків. Крім того, MySQL має великий набір оптимізацій та індексів, що дозволяють підтримувати високу продуктивність навіть при обробці великих обсягів даних.

Іншою перевагою MySQL є його простота в установці та налаштуванні. Він має простий та зрозумілий процес встановлення, а також інтуїтивно зрозумілий інтерфейс адміністратора, що дозволяє швидко розпочати роботу з базою даних без необхідності великої підготовки або додаткових зусиль.

MySQL також відомий своєю масштабованістю та гнучкістю. Він дозволяє легко масштабувати вашу базу даних з ростом обсягу даних або трафіку, що робить його ідеальним рішенням для ростучих проєктів. Крім того, MySQL підтримує різні типи реплікації та кластеризації, що дозволяє розробникам створювати розподілені та високодоступні системи баз даних.

Багатофункціональність. MySQL має багато розширених функцій та можливостей, таких як транзакції, підтримка зовнішніх ключів, збережені процедури, функції та тригери [18].

Безпека. MySQL надає широкі можливості забезпечення безпеки даних, включаючи різні рівні автентифікації, шифрування даних та контроль доступу.

Широке співтовариство та підтримка. Оскільки MySQL є однією з найпопулярніших баз даних у світі, існує велика спільнота користувачів та розробників, яка надає підтримку, допомогу та розробляє різні розширення та інструменти.

MySQL залишається одним з найбільш популярних та використовуваних рішень для зберігання та управління даними у світі, завдяки своїй надійності, продуктивності та широкому спектру функціональності.

Нижче зображено логотип бази даних MySQL (рис. 2.2).



Рисунок 2.2 – Логотип системи баз даних MySQL [5]

2.4 Вибір сервісу для створення вебінтерфейсів

Сервіси для створення інтерфейсів (UI/UX) допомагають дизайнерам створювати ефективні, естетичні та зручні для користувача інтерфейси. Ось аналіз деяких популярних сервісів для створення інтерфейсів, їх переваги та недоліки. [21]

Figma.

Переваги:

1) дозволяє кільком користувачам одночасно працювати над одним проектом;

2) не потребує встановлення програмного забезпечення, доступ через браузер;

3) легка інтеграція з інструментами, такими як Slack, Zeplin, і JIRA.

Недоліки:

1) вимагає стабільного інтернет-з'єднання для роботи;

2) деякі функції доступні тільки у платних версіях.

Sketch.

Переваги:

1) багато плагінів та ресурсів від спільноти;

2) оптимізований для macOS, швидкий та стабільний;

3) інтуїтивний інтерфейс, легкість у вивченні.

Недоліки:

1) недоступний для користувачів Windows та Linux;

3) не підтримує одночасну роботу декількох користувачів над проєктом.

Adobe XD.

Переваги:

1) легка інтеграція з іншими продуктами Adobe, такими як Photoshop та Illustrator;

2) потужні інструменти для створення інтерактивних прототипів та анімацій;

3) підтримка спільної роботи та відгуків у реальному часі.

Недоліки:

1) деякі функції доступні тільки у платних версіях;

2) може вимагати більше ресурсів комп'ютера, особливо при роботі з великими проєктами.

InVision

Переваги:

1) легке створення інтерактивних прототипів з анімаціями;

2) можливість коментування та обговорення проєктів у команді;

3) інтеграція з різними інструментами, такими як Slack, JIRA, Trello.

Недоліки:

- 1) не такий потужний для повного циклу дизайну, як інші інструменти;
- 1) платні плани можуть бути дорогими для невеликих команд.

Також найпопулярніший графічний редактор Adobe Photoshop теж може слугувати інструментом для створення інтерфейсу онлайн порталу, але враховуючи такі його недоліки як складність інтерфейсу, необхідність оплати для користування а також більша затратність часу на розробку дизайну у порівнянні з аналогами він не став оптимальним вибором для роботи [22].

Вибір пав на онлайн-інструмент Figma і ось чому.

Figma - це онлайн-інструмент для дизайну та прототипування вебінтерфейсів та мобільних додатків. [19] Ось детальний огляд особливостей та можливостей Figma.

1. Інтуїтивний інтерфейс користувача.

Figma має зручний та легкий у використанні інтерфейс, що дозволяє розробникам швидко оволодіти програмою та почати роботу над проектами.

2. Колаборація та спільна робота.

Однією з основних переваг Figma є можливість спільно працювати над проектами з іншими користувачами в реальному часі. Кілька користувачів можуть працювати над одним проектом, редагуючи його одночасно та спілкуючись у чаті.

3. Шаблони та бібліотеки компонентів.

Figma надає багато вбудованих шаблонів та бібліотек компонентів, що полегшують створення дизайну та прототипування інтерфейсів.

4. Інтеграція з іншими інструментами.

Figma інтегрується з різними іншими інструментами та сервісами, такими як Slack, Jira, Zeplin, InVision тощо, що полегшує роботу в команді та забезпечує плавну робочу потік.

5. Прототипування та анімація.

За допомогою Figma можна створювати інтерактивні прототипи, а також додавати анімацію та переходи між екранами, що дозволяє краще відтворювати функціональність майбутнього додатку або вебсайту.

6. Безпека та зберігання даних.

Figma забезпечує високий рівень безпеки для даних користувачів, використовуючи шифрування та інші заходи захисту інформації.

7. Масштабованість та універсальність.

Figma підходить для роботи над різноманітними проєктами - від маленьких дизайнерських завдань до великих та складних командних проєктів.

Загалом, Figma є потужним інструментом для дизайну та прототипування, який надає багато можливостей для розробників та дизайнерів, сприяє колаборації та полегшує роботу в команді.

Вибір і порівняння мов програмування, програмних фреймворків, баз даних і сервісів для створення вебінтерфейсів є важливим через кілька ключових причин.

Кожен проєкт має свої унікальні вимоги, такі як масштабованість, продуктивність, безпека та інтеграція з іншими системами. Вибір відповідних технологій допомагає задовольнити ці вимоги.

Різні фреймворки та бази даних мають різні характеристики продуктивності. Наприклад, деякі бази даних оптимізовані для швидкого зчитування даних, тоді як інші - для швидкого запису. Вибір правильного інструменту може значно покращити загальну продуктивність системи.

Фреймворки та сервіси з великою спільнотою розробників та хорошою документацією можуть значно спростити розробку. Вони надають доступ до великої кількості прикладів, навчальних матеріалів та підтримки.

Порівняння та вибір фреймворків, баз даних і сервісів дозволяє прийняти обґрунтоване рішення, яке допоможе створити ефективний, безпечний та масштабований продукт, що відповідає всім вимогам проєкту.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи було виконано аналіз та вибір технологій розробки та системи бази даних

Аналіз та вибір необхідних технологій, фреймворків та сервісів є дуже важливими етапами перед стартом реалізації проєкту кваліфікаційної роботи бакалавра.

Обравши та проаналізувавши мови програмування, фреймворки та сервіси для створення макету було приступлено до безпосередньої розробки порталу з впровадженням усіх необхідних технологій.

3 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ

3.1 Розробка діаграми варіантів використання

Діаграма варіантів використання (Use Case Diagram) є візуальним інструментом, який зображає функціональні можливості системи та те, як різні користувачі або зовнішні системи (актори) взаємодіють з цією системою.

Основна задача такої діаграми полягає в ілюстрації взаємодії між системою та її користувачами, показуючи, які конкретні дії (варіанти використання) можуть виконуватися. Це допомагає чітко визначити, що система повинна робити, щоб задовольнити потреби користувачів і забезпечити правильну реалізацію функціональних вимог. Нижче зображена сама діаграма (рис. 3.1).

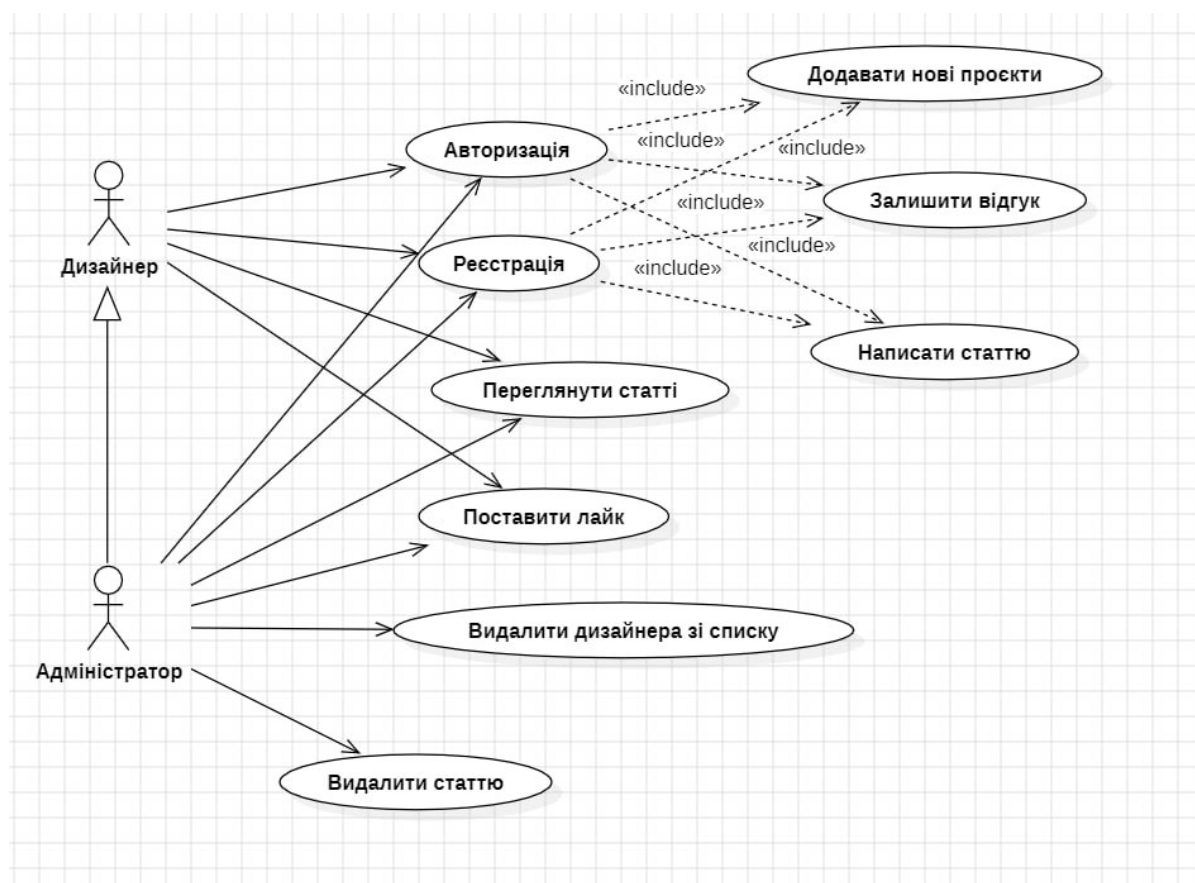


Рисунок 3.1 – Діаграма варіантів використання інтернет-порталу

3.2 Розробка діаграми класів

Діаграма класів є одним з ключових типів діаграм у мові моделювання UML (Unified Modeling Language). Вона призначена для візуалізації структури програмного забезпечення, зокрема класів, їх властивостей (атрибутів), методів (операцій) та взаємозв'язків між класами. Основна мета діаграми класів полягає в тому, щоб продемонструвати, як класи системи взаємодіють між собою і як вони співпрацюють для реалізації функцій системи. Це допомагає розробникам краще розуміти архітектуру системи і забезпечувати правильну реалізацію програмного коду. Нижче зображена сама діаграма (рис. 3.2).

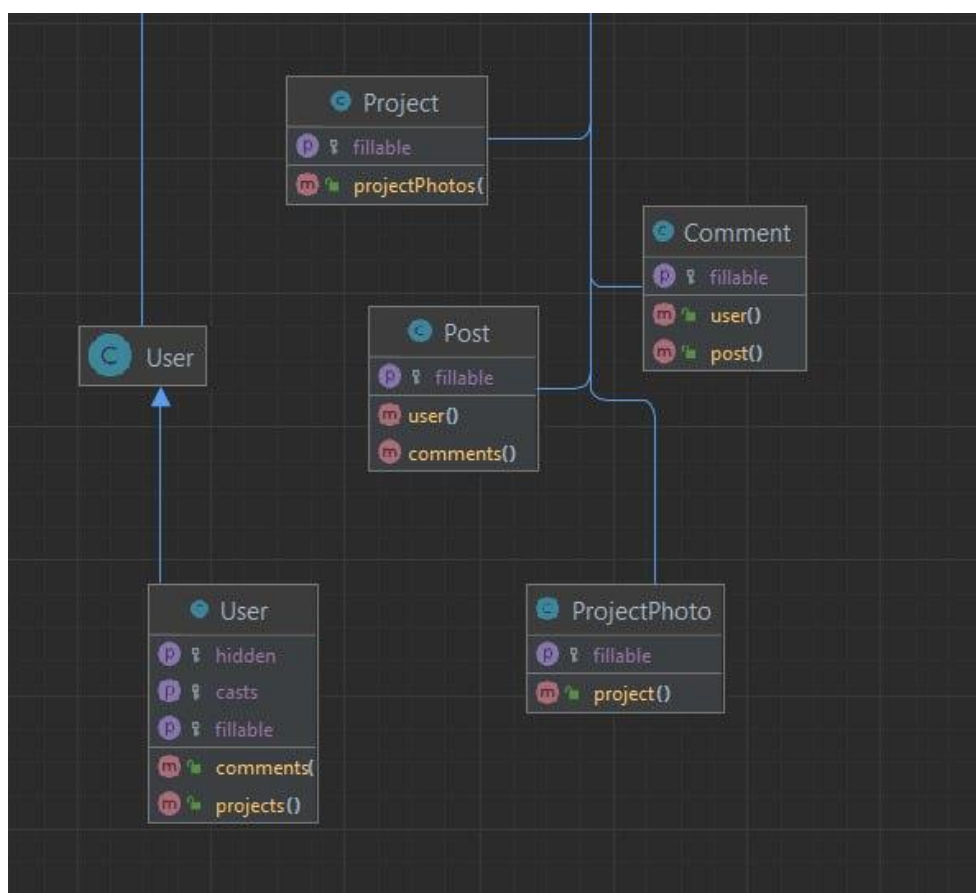


Рисунок 3.2 – Діаграма класів інтернет-порталу

3.3 Розробка діаграм послідовностей дій порталу

Діаграма послідовності є типом діаграми в мові моделювання UML (Unified Modeling Language), яка використовується для відображення взаємодії між об'єктами в рамках певного процесу або сценарію. Вона ілюструє порядок обміну повідомленнями між об'єктами в часовій послідовності.

Основна мета діаграми послідовності полягає в тому, щоб показати, як об'єкти взаємодіють між собою у визначеному порядку для виконання конкретного сценарію або функції. Це дозволяє деталізувати послідовність подій, які відбуваються під час виконання певної функції чи операції, і визначити, як різні об'єкти співпрацюють у цьому процесі. Нижче зображені ці діаграми (рис. 3.3).

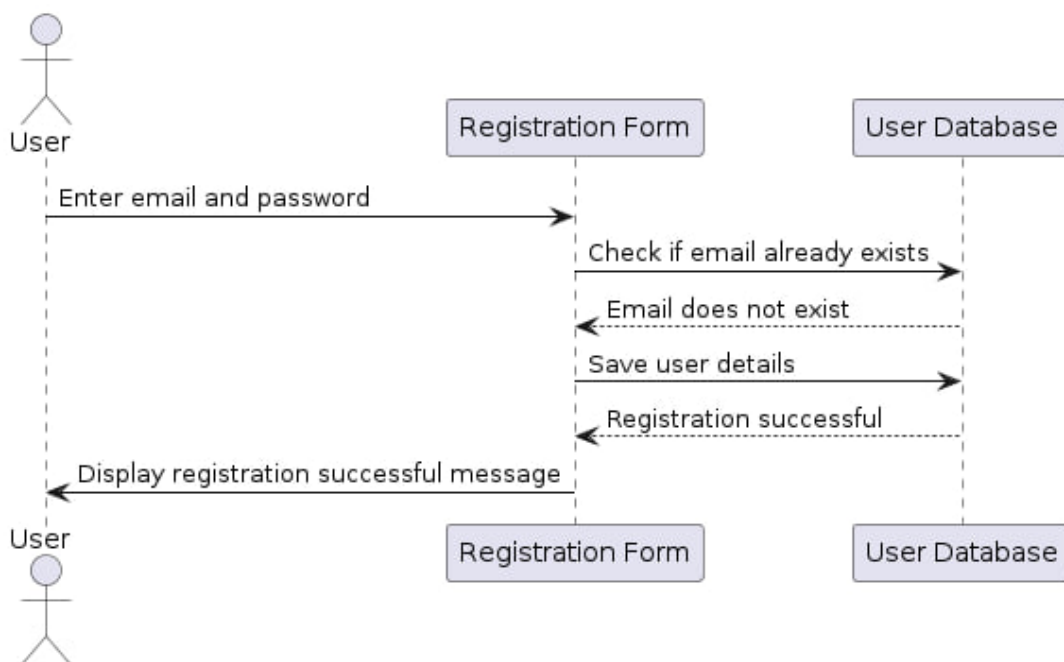


Рисунок 3.3 – Діаграма послідовностей дій під час реєстрації користувача

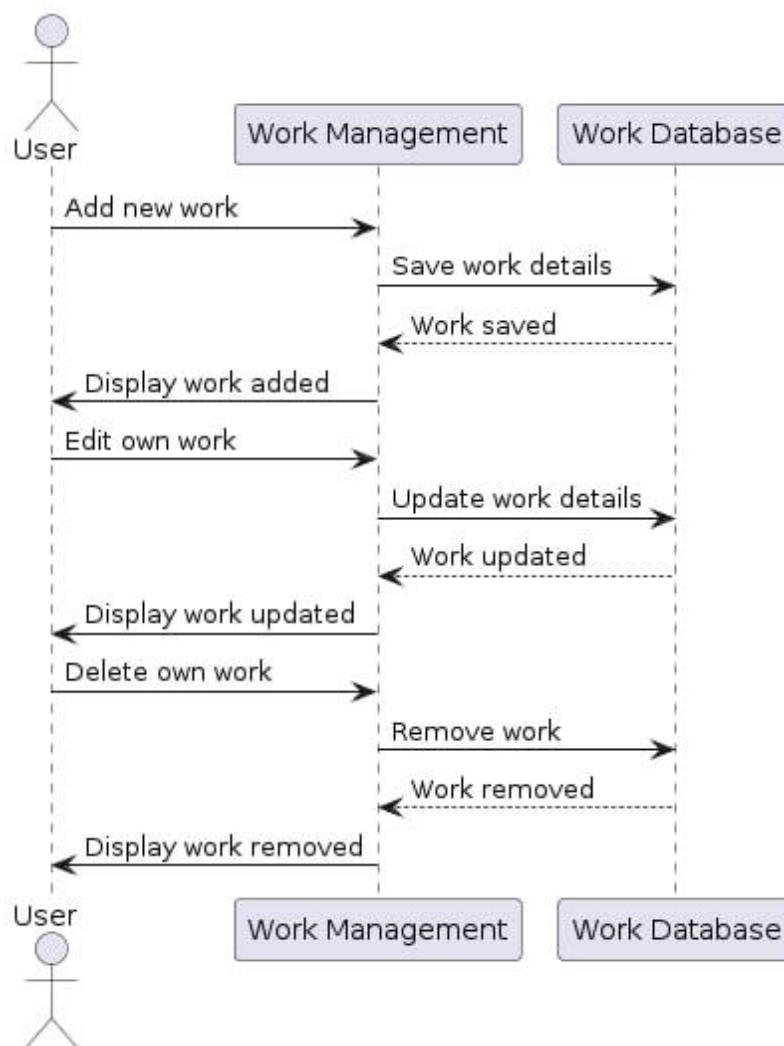


Рисунок 3.4 – Діаграма послідовностей дій керування роботами дизайнерами

3.4 Розробка мокапів порталу

Мокапи (mockups) є важливим етапом у процесі розробки вебдизайну, представляючи собою високоякісні статичні зображення, які показують, як виглядатиме кінцевий продукт. Вони дозволяють візуалізувати ідеї, ефективно комунікувати з клієнтами, узгоджувати бачення з командою, перевіряти зручність та логічність інтерфейсу, а також слугують основою для створення прототипів та написання коду.

Мокапи допомагають усім учасникам проєкту уявити кінцевий результат, отримати зворотний зв'язок і внести необхідні коригування перед переходом до

більш затратних етапів розробки. Вони також використовуються для створення документації та навчальних матеріалів.

Створення мокапів включає вивчення вимог, розробку вайрфреймів, створення детальних візуальних зображень, отримання зворотного зв'язку та затвердження дизайну. Інструменти, такі як Adobe XD, Sketch, Figma, InVision та Balsamiq, полегшують процес створення мокапів.

Мокапи дозволяють візуалізувати дизайн та функціональність майбутнього продукту ще до початку кодування. Це дає можливість клієнтам, замовникам та команді розробників побачити, як буде виглядати кінцевий продукт, і зробити необхідні коригування на ранніх етапах.

Мокапи розроблялись у сервісі Figma, як обраного найкращим серед інших аналогів. Figma дозволяє швидко та без складнощів зробити мокапи майбутнього проєкту, з яких потім і був створений повноцінний дизайн основних сторінок інтернет-порталу.

Таким чином, мокапи є ключовим інструментом, що підвищує ефективність комунікації та допомагає уникнути багатьох потенційних проблем на пізніших етапах проєкту.

Були розроблені мокапи головної сторінки, сторінки статей, та сторінки «Детальніше» для перегляду робіт дизайнерів.

Мокапи є потужним інструментом, який покращує процес розробки програмного забезпечення та вебдодатків, забезпечуючи чітку візуалізацію, полегшуючи комунікацію, зменшуючи кількість змін на пізніх етапах та забезпечуючи кращу інтеграцію дизайну та функціональності. Нижче зображено мокапи головної сторінки (рис. 3.5) та сторінки статей (рис. 3.6).

Кафедра інтелектуальних інформаційних систем
Інтернет-портал портфоліо вебдизайнера



Рисунок 3.5 – Мокап головної сторінки вебпорталу



Рисунок 3.6 – Мокап сторінки зі статтю

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи були спроектовані основні важливі діаграми та розроблені мокапи, які описують принципи роботи порталу:

- 1) діаграма варіантів використання;
- 3) діаграма класів;
- 4) діаграма послідовностей дій;
- 5) мокапи основних сторінок сайту.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

4.1 Створення дизайну основних сторінок сайту

Перше з чого почалась розробка проєкту інтернет-порталу, це створення макету – дизайну основних сторінок сайту. Були спроектовані – головна сторінка (рис. 4.1), сторінка зі статтями, сторінка для перегляду більшої кількості проєктів дизайнерів.

Спочатку було створено 3 вступні блоки без особливого функціоналу – вони відповідають за презентацію проєкту – для чого потрібен портал, та які задачі він вирішує.



Рисунок 4.1 – Меню та перші розділи головної сторінки порталу

4.2 Розробка функціональних розділів

Наступний розділ сайту (рис. 4.2) є одним з найголовніших – в ньому закладений принцип додавання робіт та профіля дизайнерів. Кожна людина може зареєструватись у порталі, додати інформацію про себе та фото своїх робіт. Їх профіль з'явиться у цьому розділі, а у слайдері буде перелік фото робіт. Також реалізована можливість рейтингу – сердечка означають кількість позитивних відгуків, які люди можуть залишати під роботами дизайнерів, по ним формується рейтинг дизайнера – на першому місці дизайнер з найвищим рейтингом.

РОБОТИ

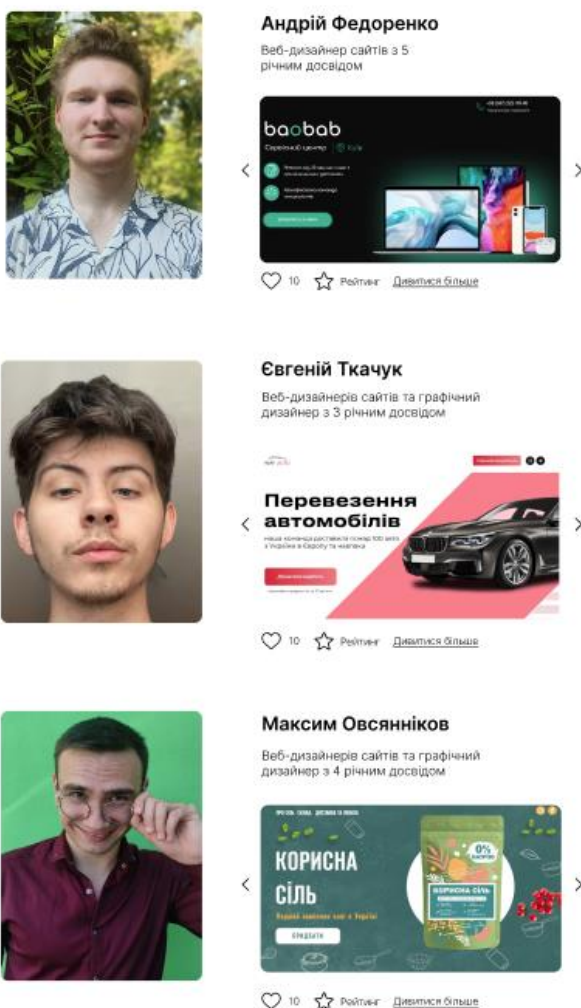


Рисунок 4.2 – Розділ з профілями дизайнерів та їх роботами

Лістинг коду з реалізацією цього розділу.

```
<?php
namespace App\Http\Controllers;
class MainPageController extends Controller
{
    public function mainpage(){
        $designers = User::query()
            ->where('role','designer')
            ->withSum('projects', 'likes')
            ->orderByDesc('projects_sum_likes')
            ->limit(5)
            ->get();
        $posts = Post::all();
        return view('main', compact(['designers','posts']));
    }
    public function postById(Post $post){
        return view('post', compact(['post']));
    }
    public function designerById(User $designer){
        return view('designer', compact(['designer']));
    }
    public function add_comment(Request $request){
        $post_id = $request->input('post_id');
        $comment_text = $request->input('text');
        $user = Auth::user();
```

Цей код представляє контролер для обробки запитів, які стосуються головної сторінки, постів та дизайнерів у додатку. Ось опис його функцій.

1) ``mainpage()``. Цей метод отримує топ-5 дизайнерів за кількістю лайків на їх проекти та всі пости. Він використовує модель ``User`` для отримання дизайнерів і модель ``Post`` для отримання постів. Потім передає отримані дані у представлення ``main``.

2) ``postById(Post $post)``. Цей метод приймає пост за ідентифікатором і передає його у представлення ``post``.

3) ``designerById(User $designer)``. Цей метод приймає дизайнера за ідентифікатором і передає його у представлення ``designer``.

4) ``add_comment(Request $request)``. Цей метод обробляє запит на додавання коментаря до поста. Він отримує ідентифікатор поста та текст коментаря з запиту, а потім створює новий коментар у базі даних за допомогою моделі ``Comment``. Після створення коментаря він перенаправляє користувача на сторінку поста з повідомленням про успішне додавання коментаря.

5) ``addLikes(Request $request)``. Цей метод обробляє запит на додавання лайка до проекту. Він отримує ідентифікатор дизайнера та ідентифікатор проекту з запиту, знаходить відповідний проєкт у базі даних та збільшує кількість лайків. Після збереження змін в базі даних він перенаправляє користувача на сторінку дизайнера з повідомленням про успішне додавання лайка.

6) ``deleteDesigner(User $designer)``. Цей метод видаляє дизайнера з бази даних. Він отримує об'єкт дизайнера, викликає метод ``delete()`` для видалення його з бази даних та перенаправляє користувача на головну сторінку з повідомленням про успішне видалення дизайнера.

Усі ці методи використовуються для обробки різноманітних запитів, пов'язаних з головною сторінкою, постами та дизайнерами у вебдодатку.

Далі було реалізовано розділ з додаванням статей на портал (рис. 4.3). Функціонал даного розділу дозволяє відкривати опис статті по натисканню кнопки «Детальніше» (рис. 4.4). В функціоналі сайту є можливість додавати нові статті – кожна нова стаття буде з'являтися в цьому розділі з відповідним описом, датою публікації та картинкою.

СТАТТІ ПРО ВЕБ ДИЗАЙН

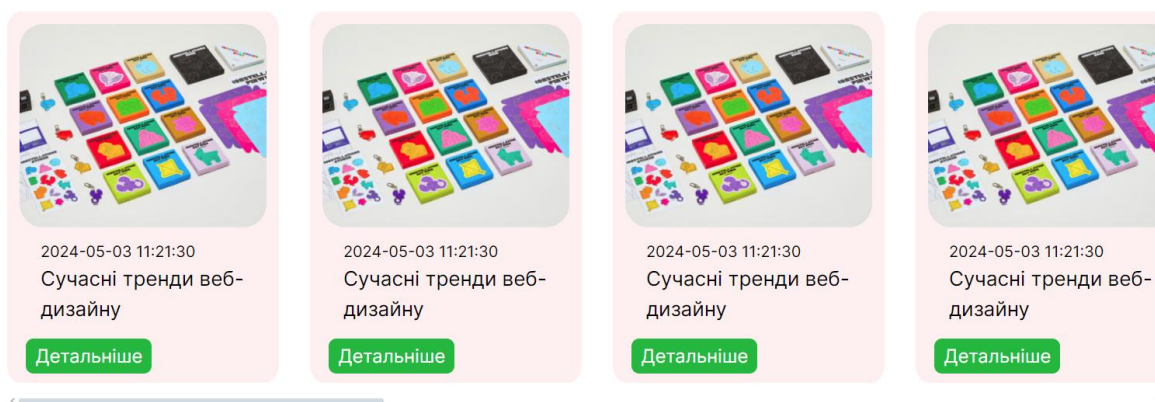


Рисунок 4.3 – Розділ зі статтями

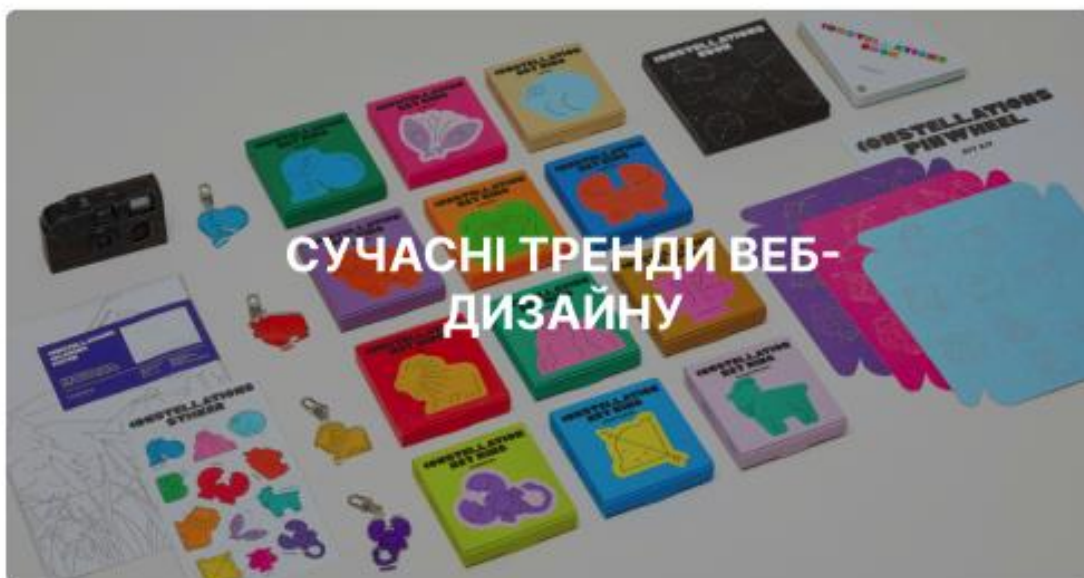
Розділ зі статтями оснащений слайдером для зручного перегляду. Також до кожної статі можна додавати коментарі (рис. 4.5).

DESIGN.IO

ПРО ПОРТАЛ

РОБОТИ ДИЗАЙНЕРІВ

СТАТТІ



СУЧАСНІ ТРЕНДИ ВЕБ-ДИЗАЙНУ

21.04.24  Андрій Федоренко

На сьогоднішній день, веб-дизайн постійно еволюціонує, але ось деякі з основних трендів, які можна спостерігати:

Соковиті та яскраві кольори — це одна із тенденцій, яка збережеться в 2023 році. Вони допоможуть веб-сайту виділитися серед конкурентів та створити візуальну ідентифікацію, яка запам'ятається надовго. Крім того, варто не забувати про використання психології кольорів, адже вона допоможе зрозуміти, яким чином барви впливають на емоції.

До речі, кольором 2023 року від Pantone стає Viva Magenta — дійсно сміливий та яскравий відтінок, який миттєво впадає в око. Це малиново-червоний тон, який зберігає баланс між теплом та прохолодою. Його можна описати як напористий, але не агресивний, потужний, але не зарозумілий, безстрашний, але не безтурботний. Незважаючи на те, що в багатьох інтерфейсах переважають спокійні та лаконічні відтінки, ніщо не заважає спробувати живі кольори, які створюють неординарну та привабливу індивідуальність вашого бренду.

Рисунок 4.4 – Сторінка з описом статті

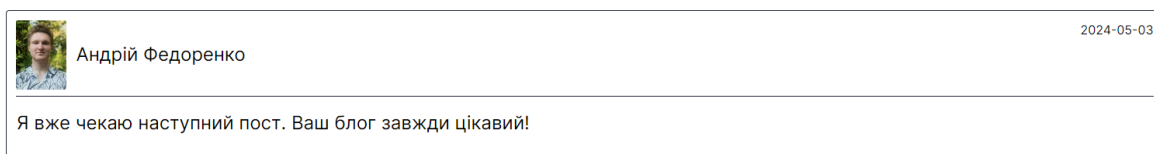


Рисунок 4.5 – Додавання коментарів до статей

Лістинг коду з реалізацією статей.

```
<?php
namespace App\Http\Controllers;
class PostController extends Controller
{
    public function store(Request $request)
    {
        $user = Auth::user();
        Post::create([
            'title' => $request->post_title,
            'text' => $request->post_description,
            'photo_url'=>$request->file('post_photo_url')->store('photos','uploads'),
            'user_id'=>$user->id
        ]);
        return redirect()->route('profile');
    }
}
```

Для чого в інтернет-порталі потрібні статті. Статті на інтернет-порталі вебдизайнерів можуть відігравати декілька важливих ролей:

1) статті можуть надавати корисні поради, практичні вправи, уроки та реальні приклади для навчання або вдосконалення навичок у вебдизайні. Вони можуть охоплювати різні аспекти дизайну, включаючи вебінтерфейси, вебграфіку, вебанімацію, взаємодію з користувачем тощо;

2) статті можуть відслідковувати останні тенденції у вебдизайні, нові технології та інструменти. Вони допомагають дизайнерам бути в курсі новинок та адаптувати свої практики до змін у галузі;

3) вебдизайнерські статті можуть містити корисні поради, які допомагають вирішити конкретні проблеми або вдосконалити процес роботи. Це може включати оптимізацію швидкості завантаження вебсайтів, вибір кольорової палітри, використання шрифтів тощо;

4) статті створюють можливість для спілкування та обміну досвідом між вебдизайнерами. Коментарі та обговорення під статтями можуть стати місцем для обміну ідеями, порадами та навіть співпраці;

5) продуктивність та ефективність статей можуть також присвячуватися питанням продуктивності та ефективності в роботі вебдизайнера. Вони можуть надавати поради щодо кращого організації робочого процесу, використання інструментів для управління завданнями та проєктами, а також методів для збереження та використання дизайнерських ресурсів.

Отже, статті на інтернет-порталі вебдизайнерів відіграють важливу роль у навчанні, оновленні знань та навичок, спільноті вебдизайнерів і підвищенні продуктивності в цій галузі.



Веб-сайт для продажу онлайн-курсів зі стилю

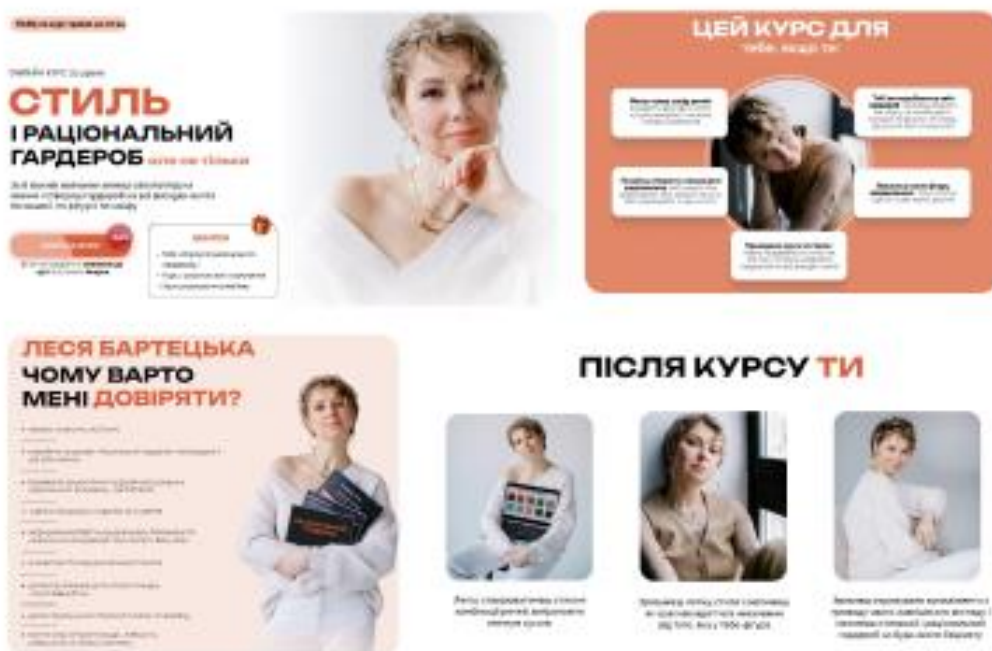


Рисунок 4.5 – Сторінка, де розміщена більшість проектів дизайнерів

На цій сторінці виводяться усі фото проєктів, які можуть додати вебдизайнери після реєстрації на порталі (рис. 4.5). Вони можуть додати фото та опис, а користувачі, які будуть їх передивлятися можуть ставити позитивні відгуки, з яких формується рейтинг дизайнера та буде змінюватись на головній сторінці сайту в його профілі.

Лістинг коду з цього розділу наведено нижче.

```
<?php
class ProjectController extends Controller
{
    public function store(Request $request)
    {
        $user = Auth::user();
        $project = Project::create([
            'title' => $request->title,
            'description' => $request->description,
            'user_id'=>$user->id
        ]);
        if ($request->hasFile('photo_urls')) {
            foreach ($request->file('photo_urls') as $file) {
                $path = $file->store('photos','uploads');
                $project->projectPhotos()->create([
                    'photo_url' => $path,
                ]);
            }
        }
        return redirect()->route('profile');
    }
}
```

Цей код відповідає за обробку запитів, пов'язаних із створенням нового проєкту у системі. Ось опис його функцій.

`store(Request $request)`. Цей метод отримує дані з форми, яку відправляє користувач для створення нового проєкту. Спочатку він отримує об'єкт користувача, який відправив запит, за допомогою фасаду `Auth`. Потім він створює новий проєкт у базі даних за допомогою моделі `Project`, використовуючи дані, що надійшли в запиті (назва проєкту, опис тощо).

Після цього метод перевіряє, чи були надіслані файли (фотографії) у запиті. Якщо так, то для кожного надісланого файлу виконується наступне:

1) файл зберігається у зазначеній директорії за допомогою методу `store()` з використанням диска "uploads";

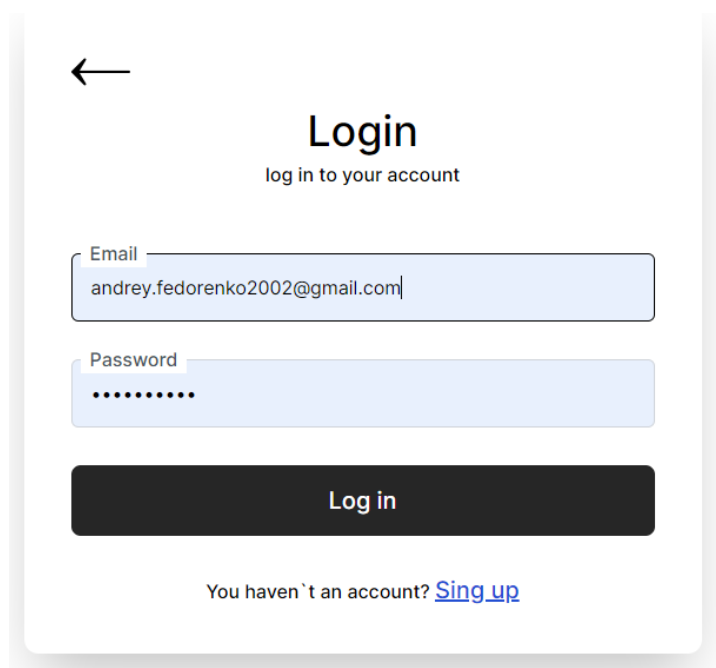
2) шлях до збереженого файлу записується до бази даних, як пов'язане з проєктом фото, за допомогою методу `projectPhotos()->create()`, де `projectPhotos()` - це відношення в моделі `Project`, що вказує на фотографії проєкту.

Нарешті, після успішного створення проєкту та збереження фотографій метод перенаправляє користувача на сторінку профілю.

Отже, цей контролер відповідає за обробку та збереження нових проєктів, які створюються користувачами у системі, включаючи збереження фотографій проєкту.

4.3 Розробка адмін-панелі

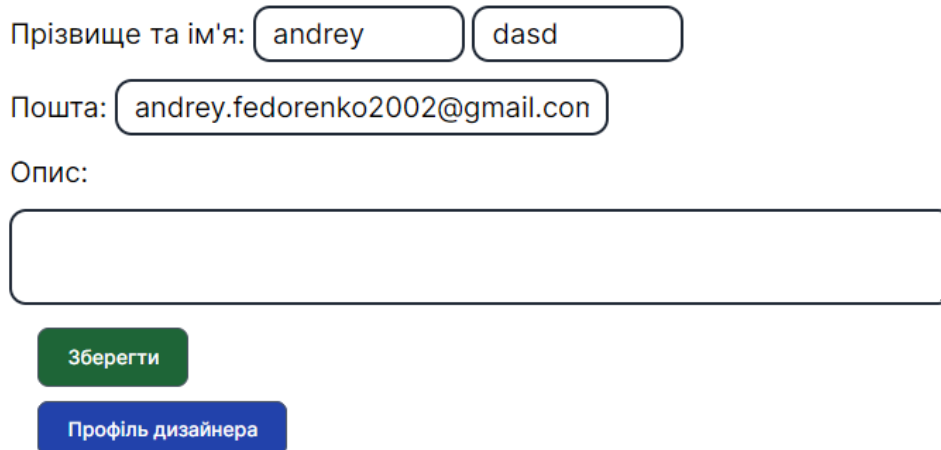
Також була створена адмін-панель для керування, видалення, додавання інформації, про вебдизайнерів (рис. 4.7), статті (рис. 4.9) та проєкти (рис. 4.8) та окремо форма реєстрації користувачів (рис. 4.6).



The image shows a login form on a white background. At the top left is a back arrow. The title 'Login' is centered, with the subtitle 'log in to your account' below it. There are two input fields: 'Email' containing 'andrey.fedorenko2002@gmail.com' and 'Password' with masked characters. A black 'Log in' button is below the fields. At the bottom, it says 'You haven't an account? [Sing up](#)'.

Рисунок 4.6 – Форма для реєстрації користувача на порталі

Інформація про користувача



The form contains several input fields: 'Прізвище та ім'я:' with two sub-fields containing 'andrey' and 'dasd'; 'Пошта:' with a field containing 'andrey.fedorenko2002@gmail.com'; and 'Опис:' with a large empty text area. Below the form are two buttons: a green 'Зберегти' button and a blue 'Профіль дизайнера' button.

Рисунок 4.7 – Форма для редагування та додавання профілів вебдизайнерів

Додати проект

Назва проекту

Опис проекту

Напишіть опис тут

Загрузити фото

Рисунок 4.8 – Форма для додавання проєктів

Додати пост

Назва статті

Текст статті

Напишіть текст тут

Загрузити фото

Рисунок 4.9 – Форма для додавання статей

4.4 Фізична модель бази даних

Як було описано у першому розділі, для функціонування порталу – збереження даних про користувачів, фото проєктів та статей – було обрано системи керування базами даних MySQL. Розробка бази даних передбачає в себе проєктування фізичної моделі (рис. 4.10).

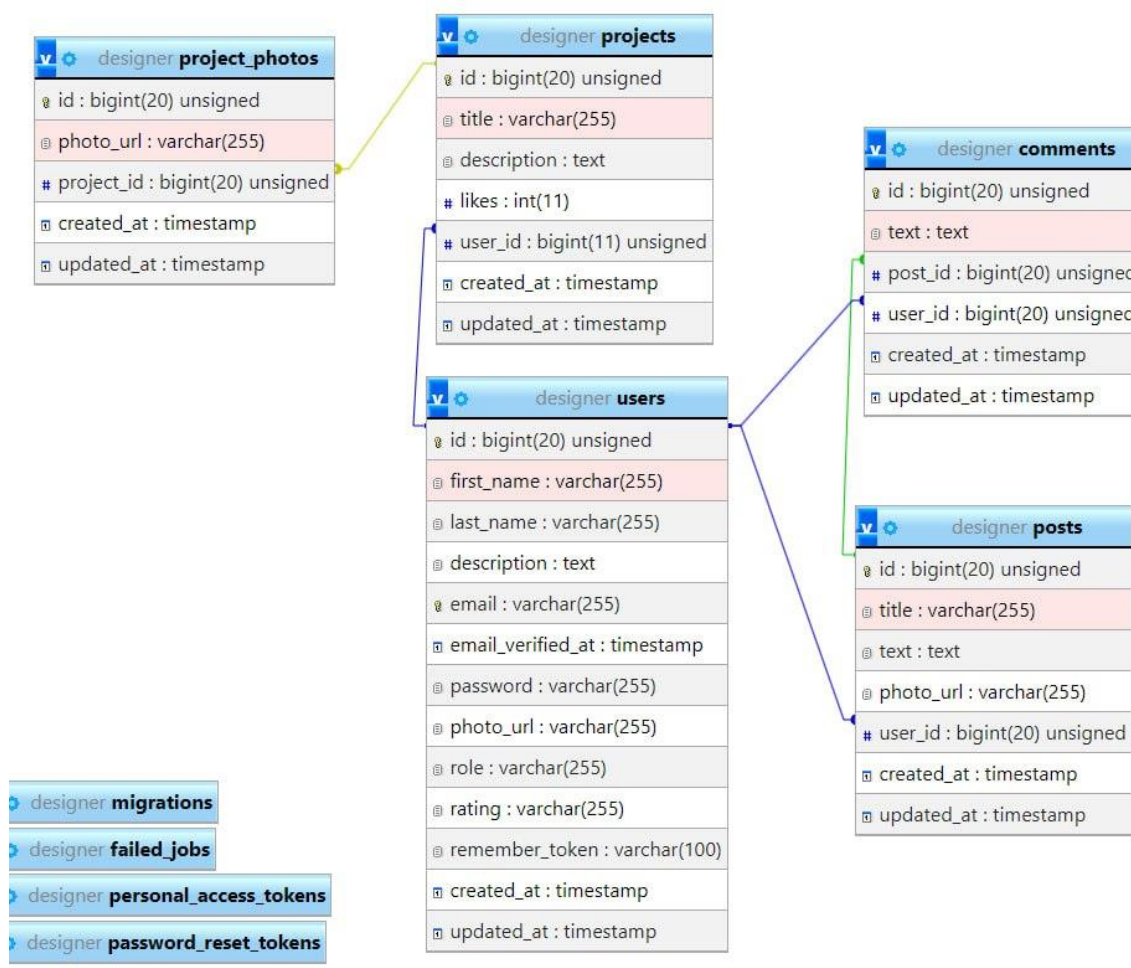


Рисунок 4.10 – Фізична модель бази даних

Далі розглянемо структуру кожної таблиці. Нижче зображена структура таблиці posts (рис. 4.11).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 text	text	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	4 photo_url	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше ▾
<input type="checkbox"/>	5 user_id 🔑	bigint		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	6 created_at	timestamp			Так	CURRENT_TIMESTAMP		DEFAULT_GENERATED	Більше ▾
<input type="checkbox"/>	7 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 4.11 – Структура таблиці posts

Нижче зображена структура таблиці projects (рис. 4.12).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Так	NULL			Більше ▾
<input type="checkbox"/>	4 likes	int			Ні	0			Більше ▾
<input type="checkbox"/>	5 user_id 🔑	bigint		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	6 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	7 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 4.12 – Структура таблиці projects

Нижче зображена структура таблиці project_photos (рис. 4.12).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	photo_url	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	project_id	bigint		UNSIGNED	Ні	Немає			Більше
4	created_at	timestamp			Так	CURRENT_TIMESTAMP		DEFAULT_GENERATED	Більше
5	updated_at	timestamp			Так	NULL			Більше

Рисунок 4.13 – Структура таблиці project_photos

Нижче зображена структура таблиці users (рис. 4.14).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	first_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	last_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
4	description	text	utf8mb4_unicode_ci		Так	NULL			Більше
5	email	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
6	email_verified_at	timestamp			Так	NULL			Більше
7	password	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
8	photo_url	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше
9	role	varchar(255)	utf8mb4_unicode_ci		Так	designer			Більше
10	rating	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше
11	remember_token	varchar(100)	utf8mb4_unicode_ci		Так	NULL			Більше
12	created_at	timestamp			Так	NULL			Більше
13	updated_at	timestamp			Так	NULL			Більше

Рисунок 4.14 – Структура таблиці users

Нижче зображена структура таблиці comments (рис. 4.14).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 text	text	<i>utf8mb4_unicode_ci</i>		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 post_id 🔑	bigint		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	4 user_id 🔑	bigint		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	5 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	6 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 4.15 – Структура таблиці comments

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи було створено дизайн та розроблено програмну реалізацію основних сторінок порталу та їх функціонал і підключення бази даних:

- 1) головна сторінка та додавання профілів та робіт дизайнерів і статей;
- 2) сторінка зі статтями та можливість додавання коментарів;
- 3) сторінка «Детальніше» з іншими роботами дизайнерів, та додавання до них позитивних відгуків;
- 4) адмін-панель для внесення змін до наповнення порталу.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи бакалавра здійснено розробку інтернет-порталу вебдизайнера. Досліджено та застосовано на практиці програмні методи та засоби розробки вебсайтів. Розроблено інтернет-портал, який дає змогу реєстрації, зареєстровані дизайнери можуть завантажувати свої роботи, які потім користувачі зможуть оцінювати, також реалізована можливість статей. Було розроблено функціональний онлайн-портал зі зручним дизайном. Для досягнення поставленої мети виконано наступні завдання:

- 1) проаналізована предметна область та аналогічні портали;
- 2) обрані технології та фреймворки розробки;
- 3) виконано проєктування та програмна реалізація інтернет-порталу;
- 4) розроблена база даних, проведено тестування та апробацію вебзастосунок.

Практичне значення отриманих результатів полягає у тому що якісний та функціональний інтернет-портал зможе допомогти вебдизайнерам у просуванні та популяризації своїх робіт, а позитивні відгуки від користувачів сприятимуть зацікавленості майбутніх клієнтів у послугах вебдизайнерів. Розроблений застосунок може бути вдосконалений за допомогою ще більшого розширення функціоналу та додання нових можливостей.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. PHP. A popular general-purpose scripting language that is especially suited to web development. URL: <https://www.php.net> (дата звернення: 01.06.2024).
2. Чому варто обрати Laravel при розробці вебпорталу. *ItProfit*. URL: <https://ru.itprofit.dev/blog/laravel-ili-cms-vordpress-ili-bitriks>. (дата звернення: 01.06.2024).
3. Для чого потрібен інтернет-портал. URL: <https://goldwebsolutions.com/uk/blog/shho-take-internet-portal-ta-v-chomu-jogo-zruchnist/> (дата звернення: 02.06.2024).
4. TailwindCSS Framework. URL: <https://tailwindcss.com> (дата звернення: 02.06.2024).
5. MySQL База даних. URL: <https://www.mysql.com> (дата звернення: 03.06.2024).
6. Ulman Larry. PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide. Peachpit Press, 2014. 156 с.
7. Yuliano T. Pengenaln Php //IlmuKomputer. com. 2007. 143 с.
8. Mitchell W. H. PHP. 2008. 181 с.
9. Stauffer M. Laravel: Up & Running. " O'Reilly Media, Inc.", 2023. 147 с.
10. Bean M. Laravel 5 essentials. Packt Publishing Ltd, 2015. 123 с.
11. Фреймворк Laravel: підручник. Armel J. Web application development with Laravel PHP Framework version 4. 2014. 322 с.
12. Фреймворк Laravel: підручник. McCool S. Laravel starter. Packt Publishing, 2012. 356 с.
13. Фреймворк Laravel: підручник. Sinha S., Dave H. J. Beginning Laravel. Packt Publishing, 2017. 341 с.
14. Фреймворк Laravel: підручник. He R. Y. Design and implementation of web based on Laravel framework //2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014). Atlantis Press, 2015. С. 301-304.

15. Php docs. URL: <https://www.php.net/docs.php> (дата звернення: 05.06.2024)
16. Laravel docs. URL: <https://laravel.com/docs/11.x/readme> (дата звернення: 05.06.2024)
17. MySQL docs: вебсайт. URL: <https://dev.mysql.com/doc/> (дата звернення: 06.06.2024)
18. Kristopher Wilson. The Clean Architecture in PHP. 2019.
19. Що таке Figma: функції, інструменти та переваги. *WeZom Academy*. URL: <https://wezom.academy/ua/chto-takoe-figma-funktsii-instrumenty-ipreimuschestva/> (дата звернення: 07.06.2024).
20. Jenifer Tidwell, Charles Brewer, Aynne Valencia. Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly Media, 2018. 171 с.
21. Randy J. Hunt. Product Design for the Web: Principles of Designing & Releasing Web Products. New Riders Pub, 2013. 161 с.
22. Alla Kholmatova. Design Systems: A practical guide for creating design languages for digital products. Smashing Magazine, 2017. 263 с.
23. Ben Forta. MySQL Crash Course. Sams Publishing, 2017. 261 с.
24. Steve Krug. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. New Riders, 2013. 122 с.
25. Eric Meyer. CSS: The Definitive Guide: Web Layout and Presentation. O'Reilly Media, 2023.

ДОДАТОК А**Лістинг коду програмної реалізації інтернет-порталу**

Реалізація реєстрації користувачів.

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'first_name',
        'last_name',
        'email',
        'password',
        'description',
        'photo_url',
        'role',
        'rating'
    ];

    /**
```

```

* The attributes that should be hidden for serialization.
*
* @var array<int, string>
*/
protected $hidden = [
    'password',
    'remember_token',
];

/**
* The attributes that should be cast.
*
* @var array<string, string>
*/
protected $casts = [
    'email_verified_at' => 'datetime',
    'password' => 'hashed',
];

public function comments()
{
    return $this->hasMany(Comment::class);
}

public function projects()
{
    return $this->hasMany(Project::class);
}
}

```

Лістинг коду головної сторінки порталу.

```

<?php

namespace App\Http\Controllers;

use App\Models\Comment;
use App\Models\Post;

```

```
use App\Models\Project;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Redirect;
use function Laravel\Prompts\text;

class MainPageController extends Controller
{
    public function mainpage(){
        $designers = User::query()
            ->where('role','designer')
            ->withSum('projects', 'likes')
            ->orderByDesc('projects_sum_likes')
            ->limit(5)
            ->get();
        $posts = Post::all();

        return view('main', compact(['designers','posts']));
    }

    public function postById(Post $post){
        return view('post', compact(['post']));
    }

    public function designerById(User $designer){

        return view('designer', compact(['designer']));
    }

    public function add_comment(Request $request){
        $post_id = $request->input('post_id');
        $comment_text = $request->input('text');
        $user = Auth::user();

        Comment::create([
            'text' => $comment_text,
```

```

        'post_id' => $post_id,
        'user_id'=>$user->id
    ]);
    return Redirect::route('post',['post'=>$post_id])->with('status', 'post-comment-added');
}

public function addLikes(Request $request){
    $designer_id = $request->input('designer_id');
    $project_id = $request->input('project_id');
    $model = Project::find($project_id);
    $model->likes += 1;
    $model->save();
    return Redirect::route('designer',['designer'=>$designer_id])->with('status', 'likes-added');
}

public function deleteDesigner(User $designer){
    $designer->delete();
    return Redirect::route('main')->with('status', 'designer-deleted');
}
}
}

```

Лістинг коду роботи бази даних з користувачами.

```
<?php
```

```

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
 */
class UserFactory extends Factory
{
    /**

```



```
* The current password being used by the factory.
*/
protected static ?string $password;

/**
 * Define the model's default state.
 *
 * @return array<string, mixed>
 */
public function definition(): array
{
    return [
        'name' => fake()->name(),
        'email' => fake()->unique()->safeEmail(),
        'email_verified_at' => now(),
        'password' => static::$password ??= Hash::make('password'),
        'remember_token' => Str::random(10),
    ];
}

/**
 * Indicate that the model's email address should be unverified.
 */
public function unverified(): static
{
    return $this->state(fn (array $attributes) => [
        'email_verified_at' => null,
    ]);
}
}
```