

Полтавський університет економіки і торгівлі  
Навчально-науковий інститут денної освіти  
Форма навчання денна  
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту  
Завідувач кафедри  
\_\_\_\_\_ Олена ОЛЬХОВСЬКА  
(підпис)

« \_\_\_ » \_\_\_\_\_ 2026 р.

## **КВАЛІФІКАЦІЙНА РОБОТА**

**на тему**  
**«РОЗРОБКА ВЕБ-САЙТУ ПЕРСОНАЛЬНОГО ПОРТФОЛІО»**

**зі спеціальності 122 Комп'ютерні науки**  
**освітня програма «Комп'ютерні науки»**  
**ступеня бакалавра**

**Виконавець роботи** Сірооченко Аліна Богданівна  
\_\_\_\_\_ « \_\_\_ » \_\_\_\_\_ 2026 р.  
(підпис)

**Науковий керівник** к.пед.н., доцент, Кошова Оксана Петрівна  
\_\_\_\_\_ « \_\_\_ » \_\_\_\_\_ 2026 р.  
(підпис)

**Рецензент**

**ПОЛТАВА 2026 р.**

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри \_\_\_\_\_ Олена ОЛЬХОВСЬКА  
(підпис)  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

## **ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему «Розробка веб-сайту персонального портфоліо»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Сірооченко Аліна Богданівна

Затверджена наказом ректора № 213-Н від «01» жовтня 2025 р.

Термін подання студентом роботи « \_\_\_\_ » \_\_\_\_\_ 2026 р.

Вихідні дані до кваліфікаційної роботи: статті та документації з теми розробки веб застосунків та сайтів.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

### ВСТУП

#### РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

#### РОЗДІЛ 2. ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ

- 2.1. Аналіз існуючих вебресурсів аналогічного призначення
- 2.2. Порівняльна характеристика сучасних підходів і технічних реалізацій
- 2.3. Переваги та недоліки існуючих програмних рішень
- 2.4. Сучасні підходи до розробки вебресурсів портфоліо

#### РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

- 3.1. Характеристика використаних технологій і засобів розробки
- 3.2. Проектування структури та основних модулів веб-сайту портфоліо
- 3.3. Опис архітектури системи та взаємодії її компонентів

#### РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

- 4.1. Розробка структури даних та основних сутностей веб-сайту портфоліо
- 4.2. Реалізація клієнтської частини та динамічної поведінки веб-сайту
- 4.3. Розробка користувацького інтерфейсу та основних функціональних можливостей

### ВИСНОВКИ

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

### ДОДАТОК А

Перелік графічного матеріалу: 3 аркуші блок-схем, 24 ілюстрації.

## Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

## Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
Вступ		
Вивчення методичних рекомендацій та стандартів та звіт керівнику		
Постановка задачі		
Інформаційний огляд джерел бібліотек та інтернету		
Теоретична частина		
Практична частина		
Закінчення оформлення		
Доповідь студента на кафедрі		
Доробка (за необхідністю), рецензування		

Дата видачі завдання «\_\_» \_\_\_\_\_ 2025 р.

Здобувач вищої освіти \_\_\_\_\_ Сірооченко Аліна Богданівна  
(підпис)Науковий керівник \_\_\_\_\_ к.пед.н., доц. Кошова О.П.  
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)**Результати захисту кваліфікаційної роботи**Кваліфікаційна робота оцінена на \_\_\_\_\_  
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № \_\_\_\_\_ від «\_\_» \_\_\_\_\_ 2026 р.

Секретар ЕК \_\_\_\_\_  
(підпис) (ініціали та прізвище)

**Затверджую**

Зав. кафедрою \_\_\_\_\_  
к.ф.-м.н. Олена ОЛЬХОВСЬКА  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

**Погоджено**

Науковий керівник \_\_\_\_\_  
к.пед.н., Оксана КОШОВА  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

**План**

дипломного проекту з фаху  
спеціальності 122 Комп'ютерні науки  
освітня програма 122 Комп'ютерні науки  
на тему «Розробка веб-сайту персонального портфоліо»  
Прізвище, ім'я, по батькові Сірооченко Аліна Богданівна

**ВСТУП****РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ**

- 2.1. Аналіз існуючих вебресурсів аналогічного призначення
- 2.2. Порівняльна характеристика сучасних підходів і технічних реалізацій
- 2.3. Переваги та недоліки існуючих програмних рішень
- 2.4. Сучасні підходи до розробки вебресурсів портфоліо

**РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА**

- 3.1. Характеристика використаних технологій і засобів розробки
- 3.2. Проектування структури та основних модулів веб-сайту портфоліо
- 3.3. Опис архітектури системи та взаємодії її компонентів

**РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА**

- 4.1. Розробка структури даних та основних сутностей веб-сайту портфоліо
- 4.2. Реалізація клієнтської частини та динамічної поведінки веб-сайту
- 4.3. Розробка користувацького інтерфейсу та основних функціональних можливостей

**ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А**

Здобувач вищої освіти \_\_\_\_\_ А. Б. Сірооченко

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

## РЕФЕРАТ

Записка: 105 сторінок, 24 рисунків, 1 додаток, 14 літературних джерел.

**Мета дипломної роботи** – розробка веб-сайту персонального портфоліо, який забезпечує зручну презентацію професійного досвіду та відповідає сучасним технічним вимогам.

**Об'єкт дослідження** – процес створення та підтримки персонального вебсайту як інструменту професійної самопрезентації.

**Предмет дослідження** – методи, технології та програмні засоби, що забезпечують створення сучасного веб-сайту портфоліо

**Методи дослідження** – у процесі виконання роботи були застосовані методи аналізу предметної області, порівняльного аналізу існуючих вебресурсів аналогічного призначення, методи проектування інформаційної архітектури, адаптивного веб-дизайну, а також сучасні підходи до реалізації клієнтської частини веб-сайту. Для створення веб-сайту використано HTML5, CSS3, JavaScript, бібліотеки Google Material Icons та IonIcons. Організацію стилів реалізовано на основі CSS Grid та Flexbox. Для динамічної поведінки застосовано чистий JavaScript без використання додаткових фреймворків.

У межах дослідження було визначено основні вимоги до функціоналу веб-сайту персонального портфоліо, зокрема вимоги до інформаційного наповнення, зручності користувацького інтерфейсу, адаптивності, продуктивності та доступності. Проведено аналіз існуючих вебресурсів аналогічного призначення, визначено їх переваги та недоліки, що дозволило обґрунтувати вибір підходів до реалізації власного програмного рішення. Описано й обґрунтовано вибір технологічного стеку, структури веб-сайту та засобів реалізації інтерфейсу користувача.

У процесі виконання дипломної роботи спроектовано структуру веб-сайту, розроблено основні програмні модулі та реалізовано клієнтську частину системи. Створено вісім інформаційних секцій для збереження та відображення

даних про автора, портфоліо, резюме, відгуки, послуги, блог та контакти. Розроблено механізми динамічної поведінки, які забезпечують роботу бургер-меню, зміну стилю хедера при прокручуванні та додавання нових відгуків без перезавантаження сторінки.

Окрему увагу приділено реалізації адаптивного дизайну, який забезпечує коректне відображення сайту на пристроях з різною шириною екрану (від настільних моніторів до смартфонів). У системі передбачено валідацію введених даних у контактній формі, що дозволяє контролювати коректність заповнення полів перед додаванням відгуку. Для представлення архітектури та логіки роботи програмного рішення підготовлено блок-схеми та діаграми взаємодії компонентів.

У результаті виконання дипломної роботи створено функціональний веб-сайт персонального портфоліо, який поєднує інформаційні, презентаційні та комунікаційні можливості в межах єдиної системи. Розроблене програмне рішення забезпечує зручну демонстрацію проєктів (з інтеграцією відеоконтенту з Instagram), структуроване представлення професійного досвіду та освіти, можливість отримання відгуків від клієнтів, інформування про перелік послуг, а також канали зв'язку через контактну форму та соціальні мережі. Сайт є легким, швидким, не потребує серверної інфраструктури та може бути розміщений на будь-якому хостингу статичних файлів.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>8</b>
<b>РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>12</b>
<b>РОЗДІЛ 2. ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ.....</b>	<b>16</b>
<b>2.1. Аналіз існуючих вебресурсів аналогічного призначення.....</b>	<b>16</b>
<b>2.2. Порівняльна характеристика сучасних підходів і технічних реалізацій.....</b>	<b>24</b>
<b>2.3. Переваги та недоліки існуючих програмних рішень .....</b>	<b>28</b>
<b>2.4. Сучасні підходи до розробки вебресурсів портфоліо.....</b>	<b>32</b>
<b>РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА .....</b>	<b>38</b>
<b>3.1. Характеристика використаних технологій і засобів розробки .....</b>	<b>38</b>
<b>3.2. Проектування структури та основних модулів веб-сайту портфоліо</b>	<b>43</b>
<b>3.3. Опис архітектури системи та взаємодії її компонентів.....</b>	<b>49</b>
<b>РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА .....</b>	<b>56</b>
<b>4.1. Розробка структури даних та основних сутностей веб-сайту портфоліо .....</b>	<b>56</b>
<b>4.2. Реалізація клієнтської частини та динамічної поведінки веб-сайту..</b>	<b>65</b>
<b>4.3. Розробка користувацького інтерфейсу та основних функціональних можливостей .....</b>	<b>73</b>
<b>ВИСНОВКИ .....</b>	<b>83</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>86</b>
<b>ДОДАТОК А.....</b>	<b>89</b>

## ВСТУП

Швидкий розвиток цифрових технологій і постійний доступ до мережі Інтернет суттєво змінили способи професійної самопрезентації. Якщо раніше основними інструментами спеціаліста були класичне резюме та портфоліо у вигляді окремого документа, то сьогодні центральне місце займає персональний вебсайт. Саме він дозволяє об'єднати в одному середовищі інформацію про професійні навички, досвід, виконані проекти, публікації, контакти та рекомендації. Персональний сайт став не лише «візитівкою», а повноцінним інструментом формування професійного бренду та взаємодії з роботодавцями, клієнтами й професійною спільнотою.

Актуальність теми дипломної роботи «Розробка вебсайту персонального портфоліо» визначається кількома важливими чинниками. Насамперед зростає роль цифрової репутації. У сучасних умовах роботодавці та замовники часто починають знайомство зі спеціалістом саме з пошуку інформації в інтернеті. Якість вебсайту, його швидкодія, адаптивність та зручність використання формують перше враження про професійність людини. Крім того, у багатьох сферах діяльності важливо демонструвати не лише кінцевий результат роботи, а й процес вирішення задач, застосовані інструменти та досягнуті результати. Саме тому формат кейс-стаді, який детально описує виконані проекти, став стандартом професійної комунікації.

Сучасні технології веброзробки значно спростили створення персональних сайтів. Використання статичної генерації сторінок, headless CMS, безсерверних сервісів і автоматизованих систем деплою дозволяє створювати швидкі, безпечні та зручні в підтримці вебресурси навіть невеликими командами або однією людиною. Це робить тему створення персонального портфоліо особливо актуальною для сучасних спеціалістів.

Розробка вебсайту портфоліо є не лише питанням дизайну чи верстки. Важливим аспектом є правильне проектування інформаційної архітектури. Для потенційного роботодавця важливо швидко отримати інформацію про навички, технології та досвід кандидата. Для клієнта необхідними є чіткий опис послуг, приклади реалізованих проєктів і зручні способи зв'язку. Для професійної спільноти важливими можуть бути публікації, участь у конференціях або відкритих проєктах. Таким чином, структура сайту повинна враховувати потреби різних категорій користувачів.

Окрему увагу необхідно приділити підтримваності сайту. Персональний бренд постійно розвивається: з'являються нові проєкти, технології та професійні напрями. Тому архітектура сайту має забезпечувати можливість швидкого оновлення контенту без необхідності повної перебудови структури. Для цього використовуються модульні компоненти інтерфейсу, повторювані шаблони сторінок та автоматизація типових процесів, зокрема оптимізації зображень, перевірки продуктивності та оновлення контенту.

Не менш важливою складовою є доступність вебресурсу. Персональний сайт має бути зручним для користувачів із різними можливостями. Це передбачає правильну семантичну структуру сторінок, підтримку навігації з клавіатури, достатній контраст кольорів та наявність альтернативного тексту для зображень. Дотримання принципів доступності не лише розширює аудиторію користувачів, а й позитивно впливає на загальну якість вебресурсу.

Важливим критерієм якості вебсайту є продуктивність. Швидкість завантаження сторінок безпосередньо впливає на користувацький досвід та ефективність взаємодії із сайтом. Для сучасних вебресурсів особливого значення набувають оптимізація зображень, мінімізація JavaScript і CSS, використання сучасних форматів файлів та систем доставки контенту (CDN). Окрім цього, важливо забезпечити стабільність інтерфейсу під час завантаження сторінок та адаптивність для мобільних пристроїв.

Для персонального портфоліо також важливими є пошукова оптимізація та підтримка соціального шарингу. Використання коректних заголовків, метаописів, структурованих даних `schema.org` та `Open Graph` дозволяє покращити видимість сайту в пошукових системах і зробити посилання більш привабливими в соціальних мережах. Наявність `sitemap.xml` та `robots.txt` сприяє правильній індексації сторінок пошуковими системами.

Окрему роль відіграє безпека та захист даних користувачів. Навіть простий сайт-портфоліо може містити форми зворотного зв'язку, тому необхідно забезпечити захист від спаму, використання `HTTPS` та коректну обробку введених даних. Також важливо регулярно оновлювати залежності проєкту та перевіряти сторонні бібліотеки для запобігання можливим уразливостям.

Серед сучасних підходів до створення вебсайтів портфоліо найбільш популярними є статична генерація сторінок (SSG), серверний рендеринг (SSR) та використання `headless CMS`. Для персональних сайтів із відносно стабільним контентом оптимальним рішенням часто є SSG, оскільки такий підхід забезпечує високу швидкість роботи, надійність і низькі витрати на інфраструктуру. У випадках, коли необхідне часте оновлення даних або інтеграція зі сторонніми сервісами, можуть застосовуватися гібридні рішення.

У дипломній роботі основна увага приділяється не лише вибору технологій, а й відповідності вебсайту сучасним вимогам щодо зручності використання, продуктивності, доступності, SEO та підтримуваності. Практична частина роботи передбачає створення архітектури сайту, реалізацію адаптивного інтерфейсу, налаштування системи керування контентом та автоматизацію процесів розробки й деплою.

Метою дипломної роботи є розробка вебсайту персонального портфоліо, який забезпечує зручну презентацію професійного досвіду та відповідає сучасним технічним вимогам. Для досягнення цієї мети необхідно виконати аналіз сучасних рішень, визначити функціональні та нефункціональні вимоги,

обґрунтувати вибір технологічного стеку, спроектувати структуру сайту та реалізувати його основні функції.

Основними завданнями роботи є аналіз сучасних підходів до створення персональних вебсайтів, визначення вимог до структури та функціональності портфоліо, проектування інформаційної архітектури та моделі даних, розробка адаптивного інтерфейсу користувача, реалізація сторінок сайту та форми зворотного зв'язку, забезпечення доступності, продуктивності й безпеки вебресурсу, а також проведення тестування та підготовка сайту до публікації.

Об'єктом дослідження є процес створення та підтримки персонального вебсайту як інструменту професійної самопрезентації. Предметом дослідження виступають методи, технології та програмні засоби, що забезпечують створення сучасного вебсайту портфоліо.

Практична значущість роботи полягає у створенні готового вебресурсу, який може використовуватися для демонстрації професійних навичок і проєктів. Також результати роботи можуть бути застосовані як методичний приклад для створення подібних сайтів у майбутньому.

Структурно дипломна робота складається з чотирьох розділів. У першому розділі розглядаються постановка задачі, мета та вимоги до вебсайту. Другий розділ присвячений аналізу сучасних рішень і технологій створення персональних портфоліо. У третьому розділі описуються архітектура системи, структура даних та принципи побудови інтерфейсу. Четвертий розділ містить практичну реалізацію проєкту, тестування, результати оцінювання якості та опис процесу деплою. Завершують роботу висновки, список використаних джерел і додатки з додатковими матеріалами.

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Розробка сучасних вебсайтів, орієнтованих на представлення професійних компетенцій окремого фахівця, є актуальним напрямом у галузі інформаційних технологій, оскільки такі рішення забезпечують впорядковане подання відомостей про досвід, портфоліо робіт, публікації та контакти, спрощують доступ аудиторії до ключової інформації та підвищують ефективність комунікації між автором і зацікавленими сторонами. У сфері персональної самопрезентації потреба у впровадженні подібних засобів набуває особливої ваги, оскільки професійна діяльність пов'язана з необхідністю демонстрації реалізованих проєктів, чіткої фіксації компетенцій, формування зрозумілого переліку послуг, публікації матеріалів та забезпечення зручного каналу зворотного зв'язку. Використання вебсайту персонального портфоліо дає можливість поєднати зазначені функції в межах єдиного програмного середовища, що позитивно впливає на довіру до фахівця, зменшує комунікаційні бар'єри та сприяє впорядкуванню інформаційних потоків.

Основне завдання дипломної роботи полягає у створенні вебсайту персонального портфоліо, який забезпечує інформаційну підтримку професійної діяльності автора та реалізує основні функції, необхідні для повсякденного використання ресурсу цільовими користувачами. Розроблюване програмне рішення повинно поєднувати можливості централізованого керування контентом (опис компетенцій, проєктів, публікацій), уніфіковане подання кейсів із фіксацією ролі та результатів, а також підтримку механізмів комунікації через контактну форму та посилання на зовнішні професійні профілі. Вебсайт має бути зручним у використанні як для відвідувачів (швидкий доступ до ключових розділів, передбачувана навігація), так і для адміністратора, відповідального за оновлення змісту та підтримку актуальності матеріалів. Важливою умовою є забезпечення цілісності структури застосунку, логічної організації його компонентів та узгодженої взаємодії між рівнем

подання (представлення сторінок), джерелом даних (контентною моделлю) і допоміжними службами (обробка форм, аналітика).

Метою дипломної роботи є розробка вебсайту персонального портфоліо. Досягнення поставленої мети передбачає розв'язання комплексу взаємопов'язаних завдань, що охоплюють аналітичні та проєктно-реалізаційні етапи. Насамперед необхідно дослідити особливості предметної області персональних портфоліо, проаналізувати існуючі вебресурси аналогічного призначення для виявлення їх сильних і слабких сторін, типових структур і підходів до подання матеріалів. Такий аналіз дає змогу визначити перелік функціональних можливостей, які доцільно реалізувати у власному рішенні, а також обґрунтувати вибір архітектурних і технологічних підходів.

Подальшим кроком є формування функціональних вимог до системи. До них належать забезпечення можливості структурованого подання інформації про автора (компетенції, підходи, інструменти), підтримка каталогу проєктів із тегуванням і фільтрами, уніфіковані сторінки кейсів із розділами “контекст — задача — роль — процес — артефакти — результати — висновки”, публікація матеріалів (статті, виступи) за наявності, а також організація зручних сценаріїв зв'язку (форма контакту, альтернативні канали). Окремого значення набувають вимоги до зручності інтерфейсу (читабельність типографіки, послідовна дизайн-система), логічності навігації (очікувані маршрути та зрозуміла ієрархія сторінок), коректності обробки введених даних та адаптивності сторінок для різних типів пристроїв. Функціональні можливості мають бути достатніми для підтримки типових інформаційних і комунікаційних процесів персонального бренду, без створення надмірної складності використання.

Важливим етапом є обґрунтування загальної структури програмного рішення. У межах роботи доцільно передбачити поділ системи на логічно пов'язані компоненти, що відповідають за подання (шаблони сторінок і повторно використовувані інтерфейсні елементи), організацію доступу до контенту (контентна модель і шари перетворення даних), допоміжні сервіси

(опрацювання форм, генерація метаданих, звітність) та операційні процеси (збірка, оптимізація зображень, формування карти сайту, публікація). Такий підхід забезпечує чіткий розподіл функцій між частинами системи, підвищує прозорість реалізації й створює передумови для подальшого супроводу та розвитку ресурсу. Крім того, структура має враховувати особливості предметної області, зокрема наявність різних типів контенту (проекти, публікації, відгуки), необхідність уніфікованих метаданих (заголовки, описи, структуровані дані) та сталі, людиночитабельні посилання.

Оскільки основою функціонування вебсайту портфоліо є робота з контентом, окрему увагу слід приділити проектуванню інформаційного наповнення та принципам організації даних. Потрібно визначити основні сутності (проект, публікація/виступ, відгук, контактний канал), їхні атрибути (назва, короткий опис, slug, часові рамки, роль автора, стек інструментів, розділи кейсу, набір ілюстрацій із підписами та альтернативними текстами, теги), а також зв'язки між ними. Важливим є забезпечення можливості збереження, оновлення, пошуку й відображення інформації у зручному для користувача вигляді. При цьому система має підтримувати коректне формування метаданих для пошукових систем і соціальних мереж (title, description, Open Graph, Twitter Cards) та генерацію карти сайту для індексації.

З огляду на значення якості взаємодії, слід передбачити вимоги до доступності: семантична розмітка, достатній контраст тексту, коректні підписи до форм і повідомлення про помилки, можливість повної навігації з клавіатури, наявність альтернативних текстів для змістових зображень. Не менш суттєвими є вимоги до продуктивності: використання адаптивних зображень у сучасних форматах, відкладене завантаження нетермінових ресурсів, мінімізація блокувальних ресурсів, коректна конфігурація кешування та орієнтація на ключові вебметрики. У частині SEO доцільно забезпечити унікальні заголовки та описи сторінок, дружні URL, валідні структуровані дані schema.org та відсутність технічних помилок, що перешкоджають індексації.

Щодо вимог до забезпечення коректності введення та опрацювання інформації, усі дані, що надходять через форми, повинні перевірятися на відповідність визначеним умовам із належною обробкою помилок і захистом від автоматизованих відправлень. Доречно також передбачити мінімальний, прозорий підхід до збирання персональних даних і чітку політику приватності, яка пояснює цілі та способи обробки інформації. Важливою є підтримка безпечного каналу передавання даних (HTTPS) та регулярне оновлення залежностей для зниження ризиків.

У межах дипломної роботи необхідно не лише реалізувати основні функціональні можливості системи, а й забезпечити її зручність, логічність побудови та відповідність сучасним вимогам до вебресурсів. Це означає, що створюване програмне рішення повинно мати послідовну інформаційну архітектуру, уніфіковані шаблони для сторінок кейсів, зрозумілу навігацію, коректну маршрутизацію та узгоджену взаємодію всіх компонентів. Практична цінність вебсайту полягає в тому, що він може бути використаний як інструмент підтримки професійної комунікації, а також як основа для подальшого розширення функціоналу відповідно до потреб конкретного фахівця (локалізація, нові розділи, розширені типи матеріалів).

Отже, у дипломній роботі планується розробити цілісний вебсайт персонального портфоліо, орієнтований на систематизацію основних процесів професійної самопрезентації. Реалізація такого програмного рішення потребує аналізу предметної області, формування функціональних і нефункціональних вимог, проектування інформаційної структури, організації контентної моделі, забезпечення коректної обробки даних і створення зручного користувацького інтерфейсу з урахуванням вимог доступності, продуктивності, SEO та безпеки. Усе це формує теоретичну та практичну основу для подальшого розгляду сучасних аналогів, обґрунтування вибору технологічного стеку та безпосередньої реалізації програмного продукту в наступних розділах роботи.

## РОЗДІЛ 2. ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ

### 2.1. Аналіз існуючих вебресурсів аналогічного призначення

На етапі проектування вебсайту персонального портфоліо ключовим є аналіз уже існуючих рішень аналогічного призначення, оскільки він дозволяє виявити сталі підходи до структури контенту, способів взаємодії з користувачем, організації навігації, подачі кейсів і технічних практик, які визначають загальну якість продукту. Такий огляд доцільно проводити у кількох площинах: за професійними доменами (портфоліо розробників, дизайнерів, фахівців креативних індустрій), за стилістикою та глибиною викладу (від лаконічних "візиток" до розгорнутих колекцій кейс-стаді), а також за технічною організацією (статичні сайти, рішення на основі CMS, гібридні підходи з використанням API та зовнішніх сервісів).

Для більш детального розуміння сучасних тенденцій доцільно розглянути кілька конкретних прикладів.

#### Приклад 1. Портфоліо відео- та фоторедактора

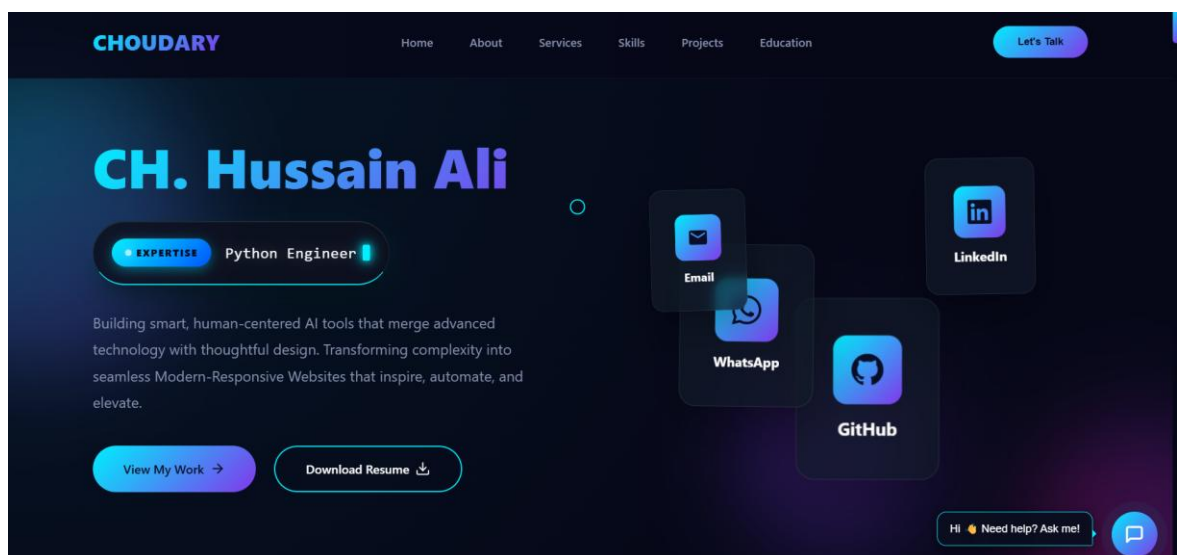


Рисунок 2.1 – Головна сторінка портфоліо відео- та фоторедактора  
(джерело: choudaryhussainali.online)

Цей приклад демонструє сучасний підхід до портфоліо відео- та фоторедактора. Головна сторінка містить чітке позиціонування, демонстрацію робіт у вигляді сітки карток та кнопки заклику до дії. Використано ефекти скляної морфології (glassmorphism) та плавні анімації. Сайт оптимізований для мобільних пристроїв та має високі показники продуктивності, що підтверджує правильний вибір статичної архітектури.

## Приклад 2. Портфоліо графічного дизайнера



Рисунок 2.2 – Головна сторінка портфоліо графічного дизайнера  
(джерело: behance.net)

Характерною особливістю цього типу портфоліо є акцент на візуальній складовій. Мінімалістичний дизайн, великі зображення проєктів та мінімум тексту дозволяють зосередити увагу на роботах автора. Такий підхід є ефективним для демонстрації візуальних навичок. Крім того, великий заголовок з ім'ям автора та роком одразу ідентифікує власника портфоліо, а темний фон створює контраст для візуального контенту.

### Приклад 3. Інноваційне портфоліо з ефектом "живого" асистента

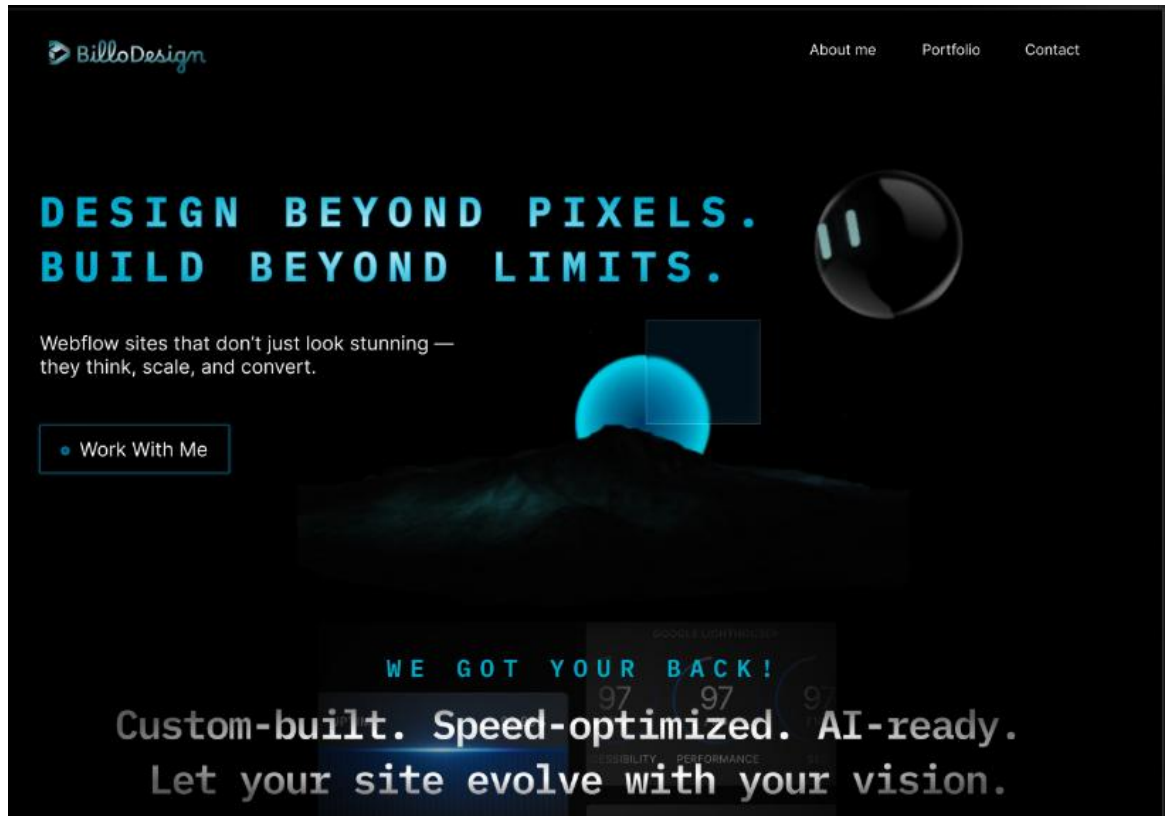


Рисунок 2.3 – Головна сторінка портфоліо з інтерактивним елементом (джерело: [awwwards.com/sites/billodesign-living-portfolio](https://awwwards.com/sites/billodesign-living-portfolio))

Цей приклад демонструє інноваційний підхід до побудови портфоліо. На сайті реалізовано інтерактивний елемент у вигляді "живої" сфери, що реагує на рухи курсора. Мікро-анімації, звуковий супровід та плавний скрол створюють унікальний користувацький досвід. Цей підхід отримав номінацію на премію Awwwards. Такий підхід демонструє, як за допомогою сучасних веб-технологій (Three.js, WebGL) можна створити запам'ятовуваний інтерактивний досвід.

## Приклад 4. Портфоліо фотографа

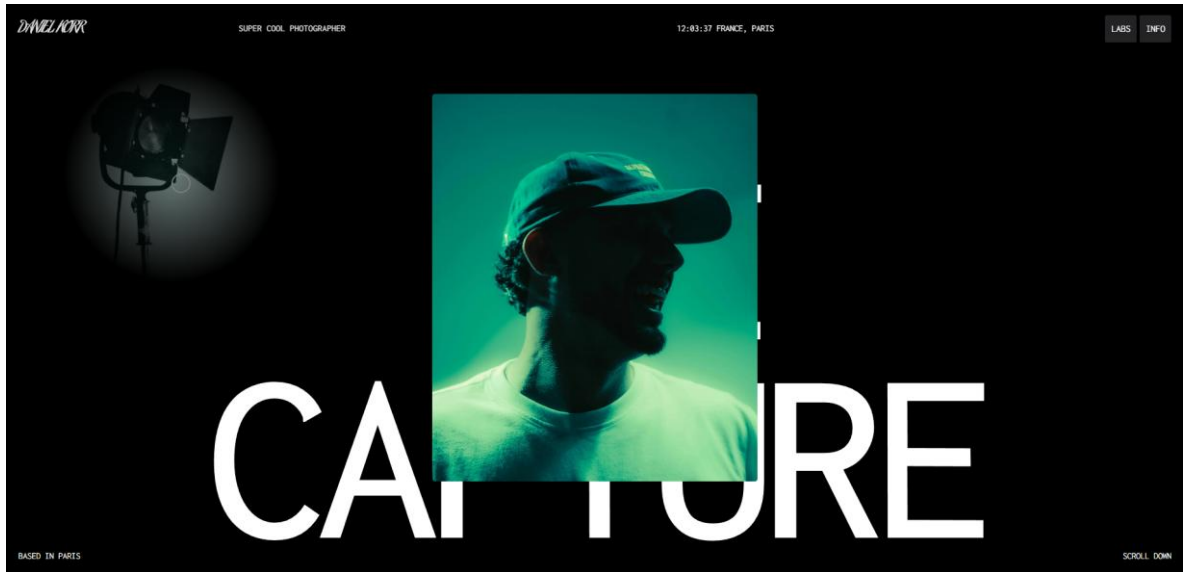


Рисунок 2.4 – Головна сторінка портфоліо фотографа (джерело: Daniel Korrr Photographer Portfolio)

Фотографічне портфоліо побудоване навколо візуального контенту. Темне, кінематографічне оформлення, плавні переходи між зображеннями та мінімальний текст дозволяють зосередити увагу на роботах автора. Сайт оптимізований для швидкого завантаження великих зображень. Бургер-меню, яке з'являється при зменшенні екрану, забезпечує зручність використання на мобільних пристроях.

Розглянуті приклади демонструють, що для персонального портфоліо фото та відео редактора оптимальним є статичний сайт з акцентом на візуальний контент. Спільними рисами успішних рішень є: адаптивний дизайн, висока продуктивність, мінімалістичний інтерфейс та зручна навігація. Використання статичних технологій (HTML, CSS, JS) дозволяє досягти швидкого завантаження сторінок, що є критичним для демонстрації відео- та фотоконтенту [1,2].

У сегменті портфоліо розробників помітною є тенденція акцентувати на реальних проектах, відкритих репозиторіях та невеликих технічних есеях, що ілюструють підходи до розв'язання інженерних задач. Типова головна сторінка

у таких ресурсів містить чіткий заголовок із позиціонуванням (спеціалізація, стек, напрямні інтересів), короткий абзац про підходи до роботи, добірку найактуальніших або найрепрезентативніших проєктів із мініатюрами та лінками на деталі, а також видимі переходи до розділів “Про мене”, “Проєкти”, “Публікації” та “Контакти”. Каталог проєктів здебільшого організовано у вигляді сітки карток однакового формату, кожна з яких має стабільні атрибути: назву, ролі або технології, коротку анотацію та прев’ю зображення. В окремих прикладах додаються фільтри або теги (наприклад, “Frontend”, “Backend”, “Performance”, “Accessibility”), які дозволяють швидко зосередити увагу на релевантних прикладах. На сторінках кейсів послідовно викладається контекст (яка була задача і чому вона важлива), роль і внесок автора, процес (які були альтернативи та чому обрано той чи інший підхід), ключові артефакти (скріншоти, діаграми, порівняння “було/стало”), а завершенням стають результати — бажано у вимірюваних метриках (скорочення часу завантаження, поліпшення Core Web Vitals, скорочення TTFB, зменшення розміру бандла, зростання конверсії тощо). Такий кістяк спостерігається у більшості вдалих прикладів і забезпечує очікувану передбачуваність структури для відвідувача.

Портфоліо дизайнерів і спеціалістів у візуальних напрямках (UI/UX, брендинг, ілюстрація, фотографія) зазвичай більше уваги приділяють композиції та якості зображень. Успішні приклади вирізняються виваженою типографікою, “дихаючою” сіткою, що не перенавантажує сторінку, та зручним ритмом подачі візуальних матеріалів: великі мініатюри для каталогу робіт, можливість повноекранного перегляду, ретельно підібрані підписи, які пояснюють зміст, а не лише ілюструють. На рівні кейсу у цій категорії також формується тенденція до розповідного викладу: від дослідження задачі та емпатії до користувача — через варіанти концепцій і проміжні рішення — до фінального дизайну з аргументами, чому саме обране рішення краще. Позитивним елементом є наявність розділів з процесом (скетчі, сітки,

прототипи), що дозволяють оцінити мислення автора, а не тільки естетичний результат.

Окремий підтип становлять портфоліо фахівців креативних індустрій (копірайтинг, відеопродакшн, саунд-дизайн), де суттєвою є інтеграція медіа-матеріалів: відео, аудіо, візуальних колекцій. Вдалими вважаються приклади, у яких багатомедійні блоки завантажуються без надмірного навантаження на сторінку, мають прев'ю з можливістю “лінивої” підміни важкого контенту, а також супроводжуються релевантними метаданими (тривалість ролика, рік, роль автора, інструменти виробництва, відзнаки або публікації). Цей сегмент особливо чутливий до продуктивності, оскільки велика вага медіафайлів може зруйнувати досвід користувача, якщо не передбачено адаптивні практики та CDN.

Серед спільних рис вдалих рішень виділяється консистентна інформаційна архітектура: передбачувана навігація, уніфіковані шаблони сторінок (особливо для кейсів), чисті дружні URL-адреси, чітка ієрархія заголовків і зрозумілі СТА. Окрім цього, повторюваною практикою є грамотне використання SEO-метаданих і структурованих даних [schema.org](https://schema.org) (Person, CreativeWork/Project, Article), що підвищує якість індексації та покращує зовнішній вигляд сторінок у пошуку й соціальному шерингу. Водночас у менш вдалих прикладах спостерігаються типові огріхи: надто дрібна типографіка, недостатній контраст, відсутність альтернативних текстів для зображень, “скачки” верстки через несинхронізоване завантаження ресурсів, важкі анімації без користі для змісту.

З точки зору технічної організації портфоліо традиційно будуються одним із трьох шляхів. Перший — статична генерація сторінок (SSG) з публікацією на CDN; у такому підході контент зберігається у структурованих файлах або забирається з headless-CMS під час збірки, а відвідувач отримує готову HTML-розмітку миттєво. Другий — класичні CMS із серверним рендерингом, які пропонують зручні панелі адміністрування, але вимагають

дисципліни продуктивності та безпеки. Третій — гібридні схеми, коли контент редагується в headless-CMS, а фронтенд генерує сторінки інкрементально або рендерить їх на сервері за потреби. Кожен із цих підходів зустрічається у практиці персональних портфоліо, і їхня доцільність залежить від частоти оновлень, вимог до інтерактивності та досвіду автора із засобами підтримки.

Підсумовуючи, аналіз наявних рішень підтверджує кілька важливих висновків для розробки власного портфоліо. По-перше, структура та мова подачі мають бути продумані не менше, ніж візуальна складова: саме вони керують увагою відвідувача та формують “карту” шляху до потрібного змісту. По-друге, уніфіковані шаблони сторінок (особливо кейсів) зменшують когнітивне навантаження та спрощують підтримку. По-третє, продуктивність і доступність — це параметри першого класу, які слід закладати з початку, а не доробляти як “опції” наприкінці. Нарешті, для індивідуального проєкту цінною є технологічна стриманість: вибір простого, надійного та відтворюваного підходу, що не вимагатиме надмірних зусиль у підтримці [5].

Проведений аналіз дозволяє зробити кілька додаткових узагальнень, важливих для практичної реалізації власного проєкту. Перш за все, варто відзначити, що всі розглянуті приклади об'єднує спільна риса — орієнтація на конкретну цільову аудиторію. Для портфоліо відео- та фоторедактора ключовим є швидке завантаження візуального контенту, тому автори обирають статичну архітектуру без зайвих динамічних елементів, які могли б сповільнити роботу сайту. Для портфоліо графічного дизайнера на перший план виходить візуальна складова — мінімалістичний інтерфейс, великі зображення, мінімум тексту, що дозволяє зосередити увагу на роботах. Інноваційне портфоліо з інтерактивними елементами розраховане на іншу аудиторію — воно демонструє технічну майстерність автора, його здатність працювати зі складними веб-технологіями, тому виправдано використовує Three.js та WebGL, навіть ціною дещо нижчої продуктивності. Фотографічне портфоліо, своєю чергою, робить акцент на якості зображень та плавності переходів між ними, що досягається

завдяки ретельній оптимізації графічних файлів та використанню сучасних форматів.

Ще одним важливим спостереженням є те, що більшість успішних портфоліо використовують підхід "спочатку контент", тобто структура та дизайн сайту підпорядковуються змісту, а не навпаки. Це означає, що автори спочатку визначають, які саме роботи вони хочуть показати, яку інформацію про себе надати, і лише потім обирають відповідний шаблон або розробляють власний дизайн. Такий підхід дозволяє уникнути ситуації, коли красивий дизайн не вміщує реальний контент або спотворює його сприйняття.

З технічного боку, аналіз підтверджує, що для персонального портфоліо оптимальним є використання статичних технологій – HTML, CSS та JavaScript. На відміну від складних CMS, статичний сайт не потребує обслуговування бази даних, регулярних оновлень системи безпеки та адміністрування серверної частини. Всі файли можна розмістити на безкоштовному хостингу, наприклад GitHub Pages, що дозволяє взагалі уникнути витрат на інфраструктуру. Крім того, статичний сайт набагато простіше переносити між хостингами – достатньо скопіювати папку з файлами, тоді як для перенесення CMS потрібно експортувати базу даних та налаштувати серверне оточення.

Окремо слід зупинитися на питанні продуктивності. У всіх розглянутих прикладах, за винятком інтерактивного, показники Google Lighthouse для швидкодії перевищують 90 балів. Це досягається завдяки кільком факторам: мінімізація CSS та JavaScript (видалення зайвих пробілів, коментарів, невикористовуваного коду), використання атрибуту `loading="lazy"` для зображень, що знаходяться за межами першого екрану, а також правильне налаштування кешування на стороні сервера. У випадку з GitHub Pages кешування налаштоване автоматично, що є додатковою перевагою цього хостингу.

Щодо доступності, то варто зазначити, що не всі проаналізовані портфоліо приділяють їй достатньо уваги. Наприклад, у деяких прикладах

відсутні альтернативні тексти для зображень, що ускладнює сприйняття контенту користувачами з вадами зору, а також погіршує SEO. Іншою типовою проблемою є недостатній контраст тексту – сірий текст на сірому тлі важко читається навіть людям з нормальним зором, не кажучи вже про людей з порушеннями. Ці помилки слід враховувати при розробці власного сайту, щоб уникнути їх повторення.

Таким чином, на основі проведеного аналізу можна зробити остаточний висновок, що для персонального портфоліо фото та відео редактора найкращим вибором є статичний сайт, створений з використанням чистих HTML, CSS та JavaScript, з акцентом на візуальний контент, адаптивний дизайн та високу продуктивність. Всі додаткові функції, такі як динамічне додавання відгуків, мають бути реалізовані за допомогою клієнтського JavaScript без ускладнення серверної частини. Такий підхід забезпечує оптимальне співвідношення функціональності, швидкості роботи та простоти підтримки, що є критичним для індивідуального проєкту, який розробляється та підтримується однією людиною.

## **2.2. Порівняльна характеристика сучасних підходів і технічних реалізацій**

Порівняльний аналіз технічних підходів, що застосовуються у персональних портфоліо, доцільно проводити крізь призму трьох взаємопов'язаних аспектів: операційні властивості (простота публікації, безпека, потреба в інфраструктурі), експлуатаційні характеристики (продуктивність, доступність, SEO), а також авторський процес (зручність оновлення контенту, повторюваність, інструменти якості).

Таблиця 2.1 – Порівняльна характеристика підходів до створення веб-сайтів портфоліо

<b>Критерій</b>	<b>Статична генерація (SSG)</b>	<b>Класичні CMS (WordPress, Joomla)</b>	<b>Headless CMS (Strapi, Sanity)</b>
<b>Продуктивність</b>	Висока (готовий HTML)	Середня (динамічне формування сторінок)	Висока (SSG + API)
<b>Час завантаження</b>	Миттєве	Залежить від кешування	Дуже високе
<b>Безпека</b>	Висока (немає БД, серверної логіки)	Середня (потрібні оновлення, плагіни)	Висока (контент відділено)
<b>Зручність редагування</b>	Низька/Середня (потрібен Git)	Висока (візуальний редактор)	Висока (графічний інтерфейс + API)
<b>Вартість хостингу</b>	Низька (безкоштовний)	Середня/Висока	Низька/Середня
<b>Складність підтримки</b>	Низька	Висока (оновлення ядра, плагінів)	Середня
<b>Оптимальність для персонального портфоліо</b>	Так (швидкість і безпека)	Для невеликих сайтів – надмірна	Так (для часто оновлюваних)

З таблиці 2.1 видно, що для невеликого персонального портфоліо з відносно стабільним контентом статична генерація або повністю статичний сайт є оптимальним вибором, оскільки вони забезпечують найкращу продуктивність, безпеку та найнижчу вартість підтримки [13].

Статична генерація (SSG) пропонує просту архітектуру: сторінки випікаються на етапі збірки, файли потрапляють на CDN і роздаються з мінімальною затримкою. Це забезпечує чудову початкову швидкість, зменшує поверхню атаки (відсутній постійний застосунковий сервер) і спрощує масштабування за рахунок георозподіленої роздачі. До обмежень належить потреба у збірці для оновлення контенту; утім, у сучасних інструментах доступні інкрементальні оновлення та вебхуки, що дають змогу тригерити генерацію частинами. Для персонального портфоліо, де зміни зазвичай відбуваються періодично і не потребують миттєвої публікації щохвилини, цей підхід часто виявляється оптимальним.

Класичні CMS (із серверним рендерингом) історично привабливі дружнім бек-офісом і численними плагінами. Вони дають змогу без спеціальних знань додавати сторінки, зображення, пости, а також швидко змінювати структуру навігації. Водночас постійна наявність серверної частини накладає вимоги до безпеки (оновлення ядра та плагінів), продуктивності (кешування, оптимізація шаблонів, мініфікація ресурсів), а також до інфраструктури (база даних, резервні копії, моніторинг). Для портфоліо цей шлях залишається доцільним, якщо автору критично важливий візуальний редактор і можливість частих правок “на живому сайті” без пайплайна збірки, а також якщо є досвід з оперативною підтримкою CMS.

Headless-підходи поєднують зручність керування контентом у CMS із сучасним фронтендом. Контент редагується у графічному інтерфейсі, подається через API, а фронтенд будує сторінки на етапі збірки (або інкрементально) й отримує переваги SSG. Це рішення зручне у довгостроковій перспективі, оскільки відділяє дані від подання, але вимагає дисципліни кешування й

налаштування доступу до API, а також може мати вартість, якщо використовується комерційний сервіс. У контексті персонального портфоліо headless-модель вигідна, якщо очікується регулярне поповнення кейсів і публікацій, а також якщо важлива колаборація без прямого доступу до репозиторію.

З огляду на експлуатаційні характеристики, всі три підходи здатні забезпечити високу доступність і продуктивність за умови належної практики. Водночас саме SSG найпростіше гарантує високі показники Lighthouse “із коробки” завдяки готовому HTML на першому завантаженні та мінімуму клієнтського коду. Класичні CMS вимагають ретельного кешування та оптимізації, щоби уникнути “важких” сторінок і повільного TTFB. Headless-підходи дозволяють поєднати комфорт редагування з якістю фронтенду, але технічно складніші в налаштуванні.

Що стосується авторського процесу, важливим чинником є повторюваність. Портфоліо виграє від уніфікованих шаблонів: якщо кожен кейс доповнюється однаково структурованими розділами, процес підготовки матеріалів стає передбачуваним, а якість подачі — стабільною. В інструментальному сенсі це означає наявність контентної моделі зі строго визначеними полями (title, slug, role, timeline, stack, sections, media з alt-описами, теги), системи перевірок перед публікацією (лінтинг розмітки, валідація метаданих, перевірки доступності, контроль ваги зображень) та автоматизованих кроків у пайплайні (генерація прев'ю зображень, побудова карти сайту, оновлення OG-карток).

Змістовний огляд сучасних прикладів також показує, що відходити від “візиткової” моделі до розповідної подачі кейсів стало де-факто стандартом. Відвідувач очікує не просто галерею, а доказову історію: який був контекст, що саме зроблено, чому це рішення краще, де підтвердження ефекту. З технічного погляду це означає, що у фронтенді варто приділити увагу змістовним підписам до зображень, належним заголовкам і спискам, код-блокам або псевдокоду

(якщо релевантно), а також акуратному “скрол-сценарію” сторінки, щоби читач не губився в тексті.

Узагальнюючи порівняння, можна стверджувати, що для персонального портфолію оптимальним є або SSG (з інкрементальними оновленнями) при помірній частоті змін, або headless-зв'язка при регулярних оновленнях і бажанні мати графічний бекофіс. Обидва підходи добре узгоджуються з вимогами до продуктивності, доступності та SEO і дозволяють вибудувати зрозумілий авторський процес з уніфікованими шаблонами та перевітками якості [3,5].

### **2.3. Переваги та недоліки існуючих програмних рішень**

Оцінюючи поширені технічні рішення, доцільно розглядати їх крізь вплив на життєвий цикл портфолію: первинну розробку, публікацію, оновлення, безпеку та довготривалу підтримку. Статична генерація приваблює простотою інфраструктури і базово високою продуктивністю. Її переваги особливо виражені для портфолію: готова розмітка, мінімум клієнтського коду, легке кешування та масштабування через CDN. Разом з тим при дуже частих оновленнях контенту потрібна дисципліна побудови пайплайна, щоби оновлення не перетворювалось на ручну рутину; інкрементальна генерація та автоматичні тригери частково нівелюють цей недолік.

Класичні CMS надають авторові комфорт редагування, проте створюють залежність від стану серверної частини та плагінів. Для збереження продуктивності потрібні додаткові механізми кешування сторінок і запитів, оптимізація шаблонів, мінімізація шрифтів і скриптів, а також системний підхід до безпеки: регулярні оновлення, резервне копіювання, обмеження прав і моніторинг активності. Для невеликих портфолію, що рідко оновлюються, така

гнучкість може виявитися надмірною, якщо немає постійного догляду за бекендом.

Headless-підходи комбінують сильні сторони двох світів: зручність керування контентом і високоякісний фронтенд. Серед мінусів — необхідність налаштування кешування API, обліку квот (у хмарних сервісів) і потенційна вартість. Водночас для портфоліо, що динамічно зростає, це часто найраціональніший шлях: автор отримує інструменти для регулярних публікацій без контакту з кодом, а користувач — швидкі сторінки, які добре індексуються.

Поза вибором стеку є перехресні ризики та переваги, універсальні для будь-якої реалізації. До позитивних практик належать уніфікація шаблонів кейсів, централізація генерації метаданих ([title/description/OG/schema.org](https://title/description/OG/schema.org)), автоматична оптимізація зображень (уключно з багаторозмірними прев'ю та сучасними форматами), а також перевірки доступності (axe) і продуктивності (Lighthouse) в автоматизованому конвеєрі. Типові помилки — невизначені правила заповнення контенту (різні поля для схожих речей), відсутність альтернативних текстів і описів до зображень, важкі фонові відео без користі для змісту, надлишок клієнтських бібліотек, що не впливають на результативність сприйняття портфоліо.

З огляду на виконання даної дипломної роботи, окремо варто розглянути доцільність використання статичного підходу (HTML/CSS/JS) порівняно з іншими технологіями. Для персонального портфоліо фото та відео редактора, яке не передбачає щоденного оновлення контенту та не потребує складної серверної логіки, статичний сайт є оптимальним рішенням з кількох причин.

По-перше, статичні файли можна розмістити на безкоштовному хостингу (GitHub Pages, Netlify, Vercel), що повністю виключає витрати на інфраструктуру. На відміну від CMS, які потребують оренди сервера з підтримкою PHP та MySQL, статичний сайт може обслуговуватися звичайним CDN, що значно дешевше та простіше в адмініструванні [13].

По-друге, швидкодія статичного сайту є максимально можливою. Оскільки сервер віддає вже готові HTML-сторінки, а не генерує їх "на льоту" з бази даних, час першого завантаження (TTFB) є мінімальним. Це позитивно впливає на користувацький досвід та SEO-ранжування, що є критичним для портфоліо, яке має привертати увагу потенційних клієнтів [12].

По-третє, безпека статичного сайту є значно вищою порівняно з динамічними рішеннями. Відсутність серверної частини, бази даних та сценаріїв авторизації унеможлиблює більшість типів атак, таких як SQL-ін'єкції, підбір паролів або злам адміністративної панелі. Для персонального портфоліо, яке не зберігає конфіденційних даних користувачів, це є додатковою перевагою.

По-четверте, простота підтримки є важливим фактором для індивідуального розробника. Для внесення змін до контенту (додавання нового проєкту, оновлення резюме, зміна контактної інформації) достатньо відредагувати HTML-файл та виконати коміт у Git-репозиторій. Весь процес публікації може бути автоматизовано за допомогою GitHub Actions або аналогічних CI/CD-інструментів.

Отже, для даного проєкту свідомо обрано статичний підхід, який найкраще відповідає поставленим вимогам: забезпечити швидко, безпечно, дешево в підтримці та зручну для демонстрації візуального контенту платформу.

Таким чином, переваги сучасних рішень полягають у можливості обрати рівень інструментальної підтримки, який відповідає частоті оновлень і технічній підготовці автора, та побудувати стабільний цикл "редактура → перевірки → публікація". Недоліки виникають у випадках, коли відсутня методична основа для контенту, а технічні рішення обрано без урахування реальних потреб (наприклад, надмірно складна інфраструктура за рідких оновлень або, навпаки, важкий CMS-бекенд для простого статичного сайту) [4].



## 2.4. Сучасні підходи до розробки вебресурсів портфоліо

Сучасні підходи до створення персональних портфоліо концентруються на чотирьох взаємодоповнювальних принципах: змістовність і структурованість контенту, доступність для різних користувацьких сценаріїв, висока продуктивність і дисциплінована інженерна практика.

Перший принцип — контент-спершу. Інформаційна архітектура, шаблони сторінок, дизайн-система та технічні рішення вибудовуються на основі реального вмісту, а не підганяють його під декоративні форми. Це означає визначення незмінного ядра моделі даних для кейсу, повторюваних розділів і обов'язкових метаданих та підпорядкування візуальних рішень задачі виразної, зручної і верифікованої подачі.

Другий принцип — доступність як невід'ємний критерій якості. Йдеться про коректну семантику HTML, продуману фокус-навігацію, адекватний контраст, інформативні альтернативні описи, передбачувану взаємодію з формами та помилками. На практиці це вимагає перевірок (автоматичних і ручних), заборони “антипатернів” (наприклад, текст на зображеннях без альтернативи) і дотримання ієрархії заголовків, що має не лише естетичний, а й навігаційний сенс.

Третій принцип — продуктивність як базове функціональне обмеження. Візуальні матеріали мають бути адаптивними за розміром і форматом, скрипти — мінімізованими, CSS — організованими так, щоб перший екран з'являвся без відчутних затримок. Паралельно налаштовується кешування, оглядаються ключові вебметрики (LCP, CLS, INP), а також передбачаються сценарії деградації на слабких пристроях або повільних мережах.

Четвертий принцип — інженерна гігієна: версіонування, автоматизовані збірки, статичний аналіз коду й контенту, захист ланцюжка поставок (оновлення залежностей, аудит уразливостей), а також прозорий процес релізів. Для персонального портфоліо це не “забаганка”, а спосіб гарантувати

відтворюваність якості при кожному оновленні й уникати випадкових регресів у продуктивності чи доступності [5].

З точки зору інструментів інтерфейсу, особливу роль відіграє послідовна дизайн-система: типографічна шкала, палітра з перевіреними контрастами, сітка та відступи, узгоджені стани компонентів. Вона дає змогу швидко формувати нові сторінки без “винаходу” стилів щоразу і зберігати єдність візуальної мови. Для SEO практичним є централізований модуль метаданих, що підхоплює поля моделі кейсу (назва, слоган, прев’ю) і автоматично формує title/description/OG-картки та schema.org-блоки, знижуючи ризик людських помилок.

Окремо слід підкреслити роль змістовних підписів до зображень і схем. У портфоліо вони виконують не декоративну, а пояснювальну функцію: коротко фіксують, на що саме дивиться читач і в чому цінність наведеного прикладу. Такі підписи працюють і на доступність, і на глибину сприйняття, і на індексацію.

Розглянемо кожен із цих принципів більш детально, оскільки вони є фундаментальними для створення якісного веб-ресурсу.

Принцип "контент спершу" передбачає, що всі дизайн-рішення повинні прийматися виходячи з реального вмісту, а не навпаки. Це означає, що перед початком розробки необхідно проаналізувати, які саме типи контенту будуть представлені на сайті: чи це будуть переважно зображення, відео, текстові кейси, або комбінація різних форматів. На основі цього аналізу визначається структура сторінок, розміри та співвідношення блоків, типографіка та кольорова гама. Наприклад, для портфоліо фотографа доцільно використовувати великі зображення на всю ширину екрану та мінімальну кількість тексту, тоді як портфоліо копірайтера або журналіста потребує зручної типографіки для читання довгих текстів. У розробленому проекті цей принцип реалізовано через чітке розділення секцій: спочатку представляється автор, потім демонструються роботи, далі надається професійний досвід, і лише

потім – контактна інформація. Така послідовність відповідає очікуванням відвідувача, який спочатку хоче побачити, що вміє автор, а потім – як з ним зв'язатися.

Принцип доступності набуває особливого значення в контексті персонального портфоліо, оскільки воно є публічним ресурсом, доступним для будь-якої аудиторії. Дотримання стандартів WCAG (Web Content Accessibility Guidelines) не лише розширює потенційну аудиторію, включаючи людей з порушеннями зору, слуху, моторики або когнітивними розладами, але й покращує SEO-показники та загальну якість коду. Основні вимоги доступності включають: забезпечення достатнього контрасту між текстом та фоном (рекомендований коефіцієнт не менше 4.5:1 для звичайного тексту), можливість повної навігації за допомогою клавіатури (без використання миші), наявність альтернативних текстів для всіх змістових зображень, коректну семантичну структуру заголовків (h1 використовується один раз на сторінку, подальші заголовки будуються за ієрархією), а також надання користувачеві достатнього часу для виконання дій. У сучасній веб-розробці існують автоматизовані інструменти для перевірки доступності, такі як axe, Lighthouse, WAVE, які дозволяють виявити більшість типових проблем ще на етапі розробки.

Принцип продуктивності є критичним для утримання уваги відвідувача. Дослідження показують, що збільшення часу завантаження сторінки з 1 секунди до 3 секунд підвищує ймовірність відмови (відмова користувача від подальшого перегляду) на 32%. Для портфоліо, яке часто містить великі зображення та відео, це особливо актуально. Основні метрики продуктивності, які слід відстежувати, включають: LCP (Largest Contentful Paint) – час появи найбільшого елемента контенту, який не повинен перевищувати 2.5 секунди; CLS (Cumulative Layout Shift) – стабільність верстки при завантаженні, яка не повинна перевищувати 0.1; INP (Interaction to Next Paint) – час відгуку на взаємодію, який не повинен перевищувати 200 мілісекунд. Для досягнення високих показників продуктивності рекомендується використовувати сучасні

формати зображень (WebP, AVIF), які забезпечують краще стиснення при збереженні якості, застосовувати атрибут `loading="lazy"` для зображень, що знаходяться за межами першого екрану, мінімізувати CSS та JavaScript (видаляти зайві пробіли, коментарі, невикористовуваний код), а також використовувати CDN для прискорення доставки статичних файлів. У розробленому проєкті всі зображення попередньо оптимізовані, CSS-код організовано з використанням змінних та медіа-запитів, а сам сайт є повністю статичним, що забезпечує мінімальний час завантаження.

Принцип інженерної гігієни часто недооцінюється при створенні невеликих проєктів, однак він є запорукою довготривалої підтримки та розвитку. Версіонування коду за допомогою Git дозволяє відстежувати історію змін, повертатися до попередніх версій у разі помилок, а також експериментувати з новими функціями в окремих гілках без ризику зламати основний код. Автоматизовані збірки дозволяють виконувати типові задачі (наприклад, оптимізацію зображень, мінімізацію коду, перевірку синтаксису) автоматично при кожному збереженні змін. Статичний аналіз коду (за допомогою інструментів ESLint для JavaScript, Stylelint для CSS) допомагає виявити потенційні помилки та підтримувати єдиний стиль написання коду в команді. Захист ланцюжка поставок означає регулярне оновлення залежностей (бібліотек, фреймворків) для запобігання вразливостям, а також використання перевірених джерел при додаванні сторонніх бібліотек. Навіть для персонального портфоліо, яке розробляється однією людиною, дотримання цих принципів значно спрощує внесення змін через півроку або рік, коли деталі реалізації вже забулися.

Таким чином, застосування всіх чотирьох принципів у комплексі дозволяє створити веб-сайт, який є не лише візуально привабливим, але й зручним для різних категорій користувачів, швидко завантажується на будь-яких пристроях, легко підтримується та розвивається. Саме такий підхід було

обрано при створенні веб-сайту персонального портфоліо в межах даної дипломної роботи.

Застосування розглянутих підходів у даному проєкті.

У розробленому веб-сайті портфоліо реалізовано всі чотири описані принципи.

Структура сайту формувалася виходячи з реального вмісту: портфоліо відеоробіт, резюме, відгуки клієнтів, перелік послуг. Кожен розділ отримав логічне місце на сторінці, а навігація побудована таким чином, щоб ключова інформація (портфоліо, контакти) була доступна за 1-2 кліки.

Семантична розмітка HTML забезпечує коректну роботу скрінрідерів. Використано достатній контраст кольорів (перевірено через Lighthouse). Для всіх зображень додано атрибути `alt` з описом вмісту, що підвищує доступність для користувачів із вадами зору.

Сайт є повністю статичним, що забезпечує мінімальний час завантаження. Зображення оптимізовано (формати JPEG/PNG зі стисненням), використано "ліниве" завантаження для нетермінових ресурсів. CSS-код організовано з використанням змінних та медіа-запитів для адаптивності.

Код сайту зберігається в Git-репозиторії (GitHub), що забезпечує версіонування та відстеження змін. Проєкт має чітку файлову структуру (окремі файли для HTML, CSS, JS, директорія для зображень). Для автоматизації публікації використовується GitHub Pages.

Узагальнюючи, сучасні підходи до розробки портфоліо формують методологію, у якій технології служать змісту, а не навпаки. Вибраний стек має підтримувати цю методологію: забезпечувати стабільну продуктивність та індексацію, пропонувати керуваність процесу публікацій і не створювати надмірної інфраструктурної складності там, де вона не додає цінності для користувача чи автора [5].



## РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

### 3.1. Характеристика використаних технологій і засобів розробки

Під час розробки веб-сайту персонального портфоліо для фото та відео редактора важливо було забезпечити не лише привабливий візуальний вигляд, але й високу продуктивність, адаптивність до різних типів пристроїв, зручність навігації та можливість інтерактивної взаємодії з відвідувачами. Для досягнення цих цілей було обрано набір сучасних веб-технологій, які дозволяють створити легкий, швидкий і візуально насичений ресурс. Нижче детально описано основні засоби, застосовані у проекті, та їхню роль у реалізації функціональних можливостей сайту.

HTML5 як основа структури веб-сайту.

HTML5 став базовою технологією для побудови семантичної структури всіх сторінок веб-ресурсу. Використання сучасних семантичних елементів, таких як `<header>`, `<section>`, `<nav>`, `<article>` та `<footer>`, дозволило створити логічно організовану розмітку, яка є зрозумілою не лише для браузерів, але й для пошукових систем. Кожна важлива частина сайту – головне меню навігації, секція портфоліо, секція резюме, секція про автора, блок відгуків клієнтів, перелік послуг, блог, контактна форма та футер – отримала своє логічне місце у структурі документа.

Завдяки HTML5 вдалося реалізувати інтеграцію з зовнішніми ресурсами, зокрема посилання на відеоролики у Instagram Reels, які демонструють приклади робіт автора. Крім того, семантична розмітка забезпечує коректне відображення сайту на різних пристроях і створює надійну основу для подальшого стилізування за допомогою CSS та додавання динамічної поведінки через JavaScript. Важливим аспектом є також використання мета-

тегів для налаштування viewport, що забезпечує коректне масштабування сторінки на мобільних пристроях [1].

CSS3 та підходи до стилізації інтерфейсу.

Для створення візуального оформлення сайту було використано CSS3, що дозволило реалізувати сучасний, естетично привабливий інтерфейс із темною кольоровою гамою. Основними кольорами стали темно-сірі та чорні відтінки (темне тло сторінок), білий текст для контрасту та жовтий (#bac964) як акцентний колір для кнопок, заголовків та інтерактивних елементів. Така кольорова схема не лише виглядає стильно, але й створює належний контраст для читання тексту та розгляду портфолійних зображень.

У проєкті було застосовано сучасні механізми CSS, зокрема Flexbox та Grid для побудови гнучких сіткових структур. Секція портфоліо використовує display: grid із властивістю grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr)), що автоматично адаптує кількість колонок під ширину екрану. Секції резюме, відгуків, послуг, блогу та контактів також побудовані на основі Grid або Flexbox, що забезпечує єдиний підхід до компоновання [2]. Завдяки медіа-запитам (media queries) реалізовано повну адаптивність сайту: для екранів шириною до 1200px зменшуються відступи, для 768px ховається звичайне меню і з'являється бургер-меню, а для 450px зменшується базовий розмір шрифту.

CSS-змінні (custom properties) та анімації.

Для спрощення підтримки кольорової схеми та забезпечення узгодженості дизайну було використано CSS-змінні, визначені у псевдокласі :root. Це дозволило централізовано керувати кольорами, тінями та іншими параметрами, що значно спрощує внесення змін до дизайну в майбутньому. Наприклад, змінні --white, --dark-gray, --light-gray, --dark-bg, --light-bg, --yellow, --box-shadow використовуються по всьому проєкту, забезпечуючи єдиний стиль [2].

Окрім статичного оформлення, у проєкті реалізовано плавні анімаційні ефекти, які покращують користувацький досвід. Наведення курсора на картки проєктів у портфоліо викликає ефект масштабування зображення (`transform: scale(1.2)`) та появу підпису, що виїжджає знизу. Кнопки мають ефект зміни позиції іконки при наведенні. Хедер сайту змінює свій зовнішній вигляд при прокручуванні сторінки – зникає напівпрозорість, з'являється білий фон і темні посилання. Усі переходи налаштовано через властивість `transition: 0.2s linear`, що робить інтерфейс більш плавним та сучасним.

JavaScript для динамічної поведінки.

Для реалізації інтерактивних функцій на стороні клієнта було використано чистий JavaScript (vanilla JS) без використання додаткових бібліотек або фреймворків [3,4]. Такий підхід дозволив мінімізувати обсяг коду, що завантажується, та забезпечити високу продуктивність навіть на мобільних пристроях. У проєкті реалізовано кілька ключових JavaScript-функцій.

Першою важливою функцією є робота з бургер-меню на мобільних пристроях. При кліку на іконку меню до навігаційної панелі додається клас "show", який змінює трансформацію елемента, і меню з'їжджає з правого краю екрану. Це забезпечує зручну навігацію на смартфонах та планшетах.

Другою функцією є зміна стилю хедера при прокручуванні сторінки. JavaScript відстежує позицію прокрутки, і коли вона перевищує нуль, хедер отримує клас "active", який змінює його фон, внутрішні відступи та колір посилань. Це покращує візуальну ієрархію та робить навігацію більш помітною.

Третьою, і найбільш складною функцією, є динамічне додавання відгуків клієнтів через форму. При натисканні на кнопку "send message" JavaScript отримує значення з полів імені та повідомлення, перевіряє їх на заповненість, а потім створює новий HTML-елемент (блок відгуку) і додає його до контейнера відгуків. Після цього форма очищується. Важливо зазначити, що ця функція працює без перезавантаження сторінки (завдяки виклику `e.preventDefault()`), що забезпечує безшовний досвід взаємодії з сайтом.



Інтеграція з зовнішніми сервісами та іконковими бібліотеками.

Для збагачення візуального наповнення та розширення функціональних можливостей сайту було інтегровано кілька зовнішніх ресурсів. По-перше, це Google Material Icons – бібліотека іконок, яка використовується для відображення календаря в секції резюме, іконок сервісів (камера, відео, редагування), а також для іконки меню-бургера. Підключення здійснюється через CDN, що забезпечує швидке завантаження та актуальність версій [9].

По-друге, для іконок соціальних мереж у футері було використано бібліотеку IonIcons, яка надає стилізовані логотипи Instagram, TikTok та Facebook [10]. Ці іконки є клікабельними посиланнями на реальні профілі автора, що забезпечує додатковий канал комунікації з потенційними клієнтами.

По-третє, портфолійні проекти представлені у вигляді карток із зображеннями-прев'ю, які ведуть на реальні відеоролики в Instagram Reels. Це дозволяє відвідувачам сайту не лише побачити статичні приклади робіт, але й переглянути динамічний відеоконтент безпосередньо на платформі соціальної мережі.

Оптимізація зображень та візуального контенту.

Оскільки сайт є портфоліо візуального контенту, велика увага приділялася оптимізації зображень. Усі графічні матеріали (фонове зображення на головній сторінці, прев'ю проектів, фото автора, аватарки клієнтів) були попередньо оптимізовані для вебу: зменшено розмір файлів без помітної втрати якості, використано формати JPEG та PNG із належним рівнем стиснення. Крім того, для фонового зображення на головній сторінці було застосовано фільтр градації сірого (`filter: grayscale(100%)`), що створює певний художній ефект і підкреслює стиль сайту [2]. Зображення у портфоліо мають однакову висоту, що забезпечує рівномірність сітки, а при наведенні вони масштабуються, додаючи динаміки.

Система контролю версій та зберігання файлів.

Для зберігання коду та файлів проєкту використовується система контролю версій Git, а для хостингу репозиторію – платформа GitHub. Це дозволяє відстежувати історію змін, працювати над проєктом з різних пристроїв, а також мати доступ до коду з будь-якої точки світу [12]. Крім того, важливі файли, такі як резюме автора (CV Sirgochenko.pdf), зберігаються в репозиторії та доступні для завантаження через відповідну кнопку на сайті.

Підсумок вибору технологій.

Вибір наведених технологій був зумовлений необхідністю створення легкого, швидкого, візуально привабливого та функціонального веб-сайту персонального портфоліо. HTML5 забезпечив семантичну структуру, CSS3 – сучасний адаптивний дизайн з анімаціями та гнучким компонованням, JavaScript – динамічну інтерактивну поведінку без потреби у важких фреймворках. Інтеграція зовнішніх бібліотек іконок та соціальних мереж дозволила розширити функціонал без надмірного збільшення обсягу коду. Усі ці технології разом утворюють цілісну технічну основу, яка відповідає сучасним вимогам до розробки веб-ресурсів такого типу.

### **3.2. Проєктування структури та основних модулів веб-сайту портфоліо**

Проєктування структури веб-сайту персонального портфоліо здійснювалося з урахуванням необхідності поєднання інформаційної, презентаційної та комунікаційної складових у межах єдиної системи. Головним завданням було створити ресурс, який би ефективно демонстрував професійні компетенції автора у сфері фото та відео продакшну, надавав структуровану інформацію про досвід роботи та освіти, а також забезпечував зручний канал зворотного зв'язку з потенційними клієнтами. Нижче детально описано архітектурну побудову сайту, основні модулі та їхню взаємодію.

### Загальна архітектура веб-сайту.

Сайт являє собою односторінковий веб-ресурс із плавною навігацією, де основні розділи розташовані послідовно один за одним, а перехід між ними здійснюється через якірні посилання у верхньому меню. Така архітектура є оптимальною для персонального портфоліо, оскільки дозволяє користувачеві отримати повну інформацію про автора без необхідності переходу між окремими сторінками. Користувач, прокручуючи сторінку, поступово знайомиться з портфоліо, резюме, інформацією про автора, відгуками, послугами, блогом і контактами. При цьому фіксований хедер (верхня панель) завжди залишається видимим після прокрутки, що забезпечує швидкий доступ до навігації з будь-якої точки сайту.

Загальна структура сайту формує багаторівневу інформаційну систему. На верхньому рівні знаходиться навігаційне меню, яке забезпечує швидкий перехід між розділами. Далі розташована головна секція (hero section), яка представляє автора та його професію. Потім ідуть змістовні блоки: портфоліо (демонстрація робіт), резюме (освіта та досвід), про автора, відгуки клієнтів, послуги, блог та контактна форма. Завершує сторінку футер із посиланнями на соціальні мережі.



Рисунок 3.2 – Структура модулів веб-сайту персонального портфоліо

Модуль навігації та хедер.

Одним із ключових модулів сайту є модуль навігації, реалізований у вигляді фіксованого хедера. У стандартному стані хедер є напівпрозорим, а

його фон зливається з головним зображенням, створюючи ефект занурення. При прокручуванні сторінки JavaScript виявляє зміну позиції і додає хедеру клас "active", який змінює фон на білий, збільшує внутрішні відступи та змінює колір посилань на темний. Це робить навігацію більш помітною на тлі різноманітного контенту та покращує загальний користувацький досвід.

На мобільних пристроях модуль навігації трансформується: горизонтальне меню зникає, а замість нього з'являється іконка бургер-меню. При кліку на іконку меню з'їжджає з правого краю екрану, займаючи всю висоту сторінки, а пункти навігації стають вертикальними. Це забезпечує зручну навігацію на пристроях з обмеженою шириною екрану. Такий підхід до адаптивності є стандартом сучасної веб-розробки.

Модуль портфоліо.

Центральним елементом сайту є модуль портфоліо, оскільки саме він демонструє практичні результати роботи автора. Модуль реалізований у вигляді сітки карток проєктів. Кожна картка містить зображення-прев'ю (кадр з відеоролика), а сама картка є клікабельним посиланням на відповідний відеоролик у Instagram Reels. Така архітектура дозволяє, з одного боку, дати користувачеві швидке візуальне уявлення про роботу, а з іншого – перенаправити його на платформу з оригінальним контентом для детального перегляду.

При наведенні курсора на картку відбувається анімація: зображення масштабується, а з нижнього краю виїжджає напівпрозорий блок із назвою проєкту. Цей ефект не лише привертає увагу, але й надає додаткову інформацію (назву або номер проєкту). Сітка портфоліо побудована з використанням CSS Grid, що забезпечує однаковий розмір карток і автоматичне підлаштування кількості колонок під ширину екрану [2,7].

Модуль резюме (освіта та досвід).

Модуль резюме виконує функцію представлення професійного бекграунду автора. Він поділений на три колонки, кожна з яких присвячена

окремому періоду або місцю роботи/навчання. Перша колонка містить інформацію про освіту (бакалавр комп'ютерних наук у Полтавському університеті економіки і торгівлі), дві наступні – про досвід роботи (контент-креатор/відеоредактор в AutoBy Servisas та відео- та фоторедактор у Dvizh Vilnius).

Кожен блок у колонці має уніфіковану структуру: дата (з іконкою календаря), назва позиції, опис обов'язків та назва установи. Така уніфікація забезпечує легкість сприйняття інформації та дозволяє швидко порівнювати різні періоди професійного життя. Колонки побудовані з використанням CSS Grid, що гарантує рівномірне розташування блоків незалежно від їхньої висоти.

Модуль "Про автора".

Модуль "Про автора" складається з двох колонок: лівої, яка містить фотографію автора, та правої – з текстовим описом. Текст опису підкреслює творчий характер діяльності, захоплення малюванням, зйомкою та відеомонтажем, а також постійне прагнення до розвитку. У цьому ж модулі розташована кнопка завантаження резюме (CV), яка веде на PDF-файл, збережений у репозиторії проєкту. Такий підхід дозволяє потенційним роботодавцям або клієнтам отримати повну версію резюме в стандартному форматі.

Модуль відгуків клієнтів (testimonials).

Цей модуль є важливим елементом соціального доказу компетентності автора. Він реалізує дві взаємопов'язані функції: по-перше, відображення існуючих відгуків у вигляді сітки карток; по-друге, можливість додавання нових відгуків через форму у контактній секції. Кожна картка відгуку містить аватар клієнта (умовне зображення), текст відгуку та ім'я автора (з можливістю вказати посаду або компанію).

JavaScript-функціонал забезпечує динамічне додавання нових відгуків: користувач заповнює поля імені та повідомлення у контактній формі, натискає кнопку, і новий блок відгуку додається до контейнера без перезавантаження

сторінки. Дані при цьому не зберігаються на сервері (лише в DOM-дереві поточної сесії), що є спрощеним, але достатнім для демонстрації функціоналу в межах дипломного проекту.

Модуль послуг (services).

Модуль послуг представляє перелік того, що автор пропонує потенційним клієнтам. Він містить три картки: відеозйомка, відеомонтаж та фотографія. Кожна картка має велику іконку (з бібліотеки Material Icons), заголовок, детальний опис послуги та кнопку "learn more". Хоча на поточному етапі кнопки "learn more" ведуть на порожні посилання (символ "#"), у майбутньому вони можуть бути замінені на посилання на сторінки з детальним описом кожної послуги, прикладами робіт або прайс-листом.

Опис кожної послуги є досить детальним. Для відеозйомки акцент робиться на техніках мобільної зйомки, які забезпечують стабільне та якісне зображення. Для відеомонтажу – на динамічному редагуванні, корекції кольору, музичному синхроні та адаптації під вертикальний формат. Для фотографії – на вмінні передавати настрій та стиль, а також на натуральній колірній корекції.

Модуль блогу (blogs).

Модуль блогу на поточному етапі розробки є заглушкою, оскільки автор ще не підготував публікації. У ньому відображаються три картки з позначкою "coming soon", датою (Jan 9, 2022), кількістю коментарів (5 comments). Незважаючи на те, що контент відсутній, структура модулю вже закладена і дозволяє в майбутньому легко додавати реальні публікації, замінивши зображення, заголовки та тексти.

Модуль контактів та зворотного зв'язку.

Модуль контактів складається з двох колонок. Ліва колонка містить форму для надсилання повідомлення (поля для імені, email, телефону, тексту повідомлення та кнопка "send message"). Права колонка містить контактні деталі: email, український та литовський номери телефонів, а також адресу у

Вільнюсі.

Важливо зазначити, що поточна реалізація форми надсилання повідомлення є демонстраційною – вона додає відгук до модуля відгуків, але не надсилає дані на сервер або електронну пошту. Однак архітектура модулю закладена таким чином, що в майбутньому можна легко інтегрувати реальний бекенд (наприклад, через Netlify Forms, Formspree або власний серверний обробник), не змінюючи фронтенд-частину.

Модуль футера та соціальних мереж.

Футер містить посилання на соціальні мережі автора: Instagram, TikTok та Facebook. Кожна іконка є клікабельним посиланням, яке відкриває відповідний профіль у новій вкладці. Цей модуль виконує важливу функцію розширення каналів комунікації та дозволяє потенційним клієнтам познайомитися з автором ближче через його активність у соціальних мережах.

Отже, проектування структури веб-сайту персонального портфоліо виконано на основі модульного підходу, за якого окремі частини системи відповідають за власний функціональний напрям, але працюють у межах єдиної візуальної та навігаційної логіки. Модулі портфоліо, резюме, відгуків, послуг та контактів утворюють цілісну картину професійної діяльності автора, а адаптивний дизайн та динамічна поведінка (бургер-меню, зміна хедера, додавання відгуків) забезпечують зручність використання на будь-яких пристроях. Запропонована структура повністю відповідає завданням персональної професійної самопрезентації та відображає логіку побудови розробленого програмного продукту.

### **3.3. Опис архітектури системи та взаємодії її компонентів**

Архітектура розробленого веб-сайту персонального портфоліо побудована за принципом статичного клієнт-серверного застосунку з

елементами динамічної поведінки на стороні клієнта. Усі файли сайту (HTML, CSS, JavaScript, зображення) є статичними та зберігаються на веб-хостингу. Коли користувач переходить за адресою сайту, браузер завантажує ці файли, після чого подальша робота відбувається локально, без постійного звернення до сервера (за винятком завантаження зображень та зовнішніх ресурсів, таких як іконки) [12]. Така архітектура є простою, надійною та високопродуктивною, оскільки сервер займається лише роздачею статичних файлів.

Клієнтська архітектура та структура файлів.

Проект має чітко організовану файлову структуру, яка сприяє легкості підтримки та подальшого розвитку. Основними файлами є `index.html` (головний HTML-документ), `styles.css` (файл каскадних таблиць стилів), `script.js` (файл JavaScript-коду) та директорія `images/`, у якій зберігаються всі графічні матеріали. Okремо варто відзначити файл `CV Siroochenko.pdf`, який є резюме автора та доступний для завантаження.

Уся логіка розділена за типами ресурсів: структура та вміст сторінки – в HTML, візуальне оформлення – в CSS, динамічна поведінка – в JavaScript. Це відповідає принципам розділення відповідальності (*separation of concerns*) та дозволяє розробникові легко вносити зміни: наприклад, змінювати кольори або відступи в одному файлі, не торкаючись HTML або JS.

Взаємодія компонентів на стороні клієнта.

Взаємодія компонентів системи може бути описана наступною послідовністю кроків при завантаженні сторінки та подальшій роботі користувача.

Етап 1: Завантаження та парсинг HTML. Коли користувач вводить URL-адресу сайту, браузер надсилає HTTP-запит до сервера. Сервер відповідає файлом `index.html`. Браузер починає парсити HTML-код, будуючи DOM-дерево. Під час парсингу він зустрічає теги `<link rel="stylesheet" href="styles.css">` та `<script src="script.js">`, які вказують на необхідність завантаження додаткових ресурсів.

Етап 2: Завантаження CSS та JavaScript. Браузер надсилає додаткові запити на сервер для отримання `styles.css` та `script.js`. Отримавши CSS, браузер будує CSSOM-дерево (об'єктну модель CSS). Отримавши JavaScript, браузер виконує його. У нашому випадку, JavaScript-код (файл `script.js`) одразу виконується після завантаження: додаються обробники подій для кнопки меню, для вікна прокрутки, для кнопки надсилання повідомлення.

Етап 3: Об'єднання DOM та CSSOM. Браузер об'єднує DOM-дерево та CSSOM-дерево, формуючи дерево рендерингу (render tree) [6,7,8]. На основі цього дерева браузер обчислює позиції та розміри кожного елемента (layout або reflow) та виконує відмальовку (paint) пікселів на екрані. Користувач бачить готову сторінку.

Етап 4: Взаємодія з навігацією. Коли користувач натискає на пункт меню (наприклад, "portfolio"), спрацьовує стандартна поведінка якірних посилань: браузер плавно прокручує сторінку до елемента з відповідним id (в даному випадку – до секції з `id="portfolio"`). Це реалізовано засобами HTML/CSS без додаткового JavaScript-коду.

Етап 5: Прокручування сторінки. Коли користувач прокручує сторінку, спрацьовує обробник події `scroll`, доданий у JavaScript. Функція отримує поточну позицію прокрутки (`window.scrollY`) і, якщо вона більша за 0, додає хедеру клас "active". У відповідь на це CSS змінює вигляд хедера: фон стає білим, посилання – темними, збільшуються внутрішні відступи. Якщо позиція прокрутки дорівнює 0 (сам верх сторінки), клас "active" видаляється, і хедер повертається до початкового напівпрозорого стану.

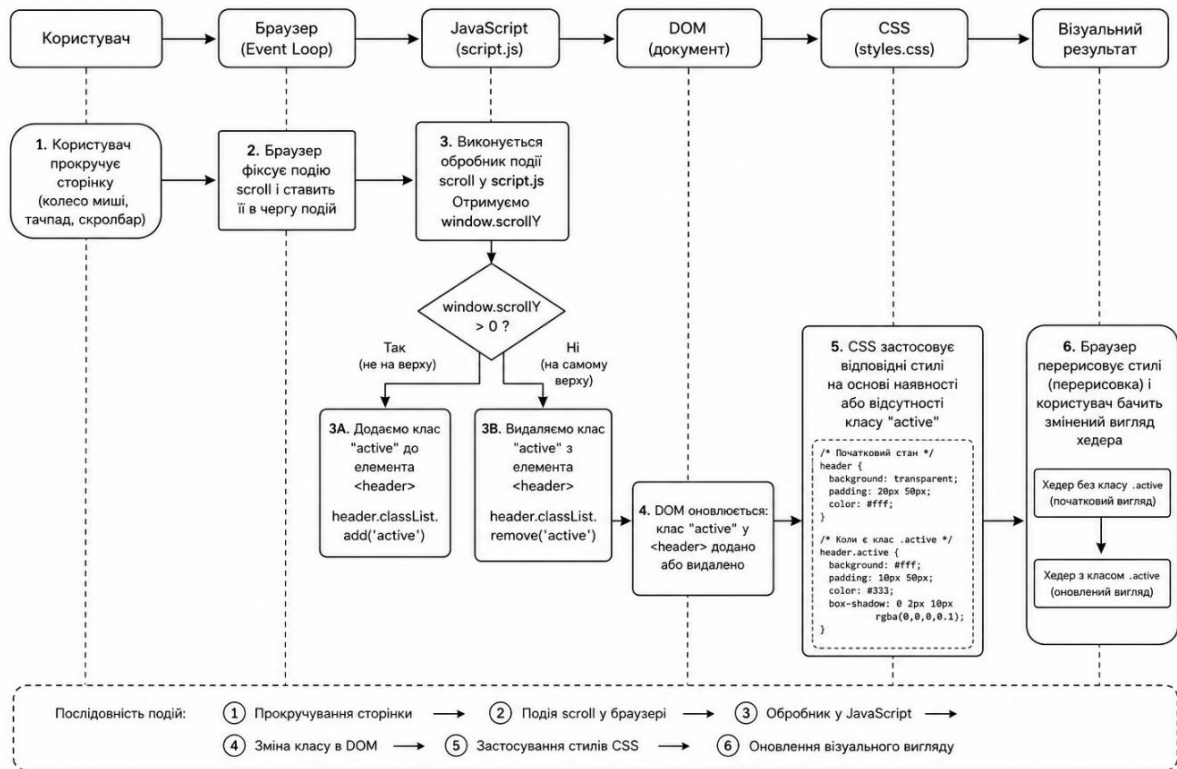


Рисунок 3.1 – Взаємодія компонентів при прокручуванні сторінки

Етап 6: Додавання нового відгуку. Це найбільш складний та показовий сценарій динамічної взаємодії. Він відбувається наступним чином:

1. Користувач заповнює поля "your name" та "write a message" у контактній формі.
2. Користувач натискає кнопку "send message".
3. Спрацьовує обробник події click, доданий до кнопки. Обробник викликає `e.preventDefault()`, запобігаючи стандартному відправленню форми та перезавантаженню сторінки.
4. JavaScript зчитує значення з полів введення (`nameInput.value` та `messageInput.value`) і зберігає їх у змінні.
5. Виконується перевірка: якщо обидва поля не порожні, процес триває; якщо будь-яке з полів порожнє, з'являється повідомлення alert із проханням заповнити поля.
6. JavaScript створює новий HTML-елемент: `document.createElement('div')`, додає йому клас "box", а потім за допомогою

властивості `innerHTML` наповнює його вмістом (зображення-аватар, текст відгуку, ім'я користувача).

7. Отриманий новий елемент додається до контейнера відгуків за допомогою методу `appendChild`.
8. Поля форми очищуються (`nameInput.value = ""; messageInput.value = ""`);  
Роль CSS-змінних у забезпеченні узгодженості стилів.

Важливим елементом архітектури є використання CSS-змінних. Вони визначені у псевдокласі `:root` і доступні глобально для всіх елементів. Коли розробнику потрібно змінити акцентний колір (наприклад, з жовтого на інший), достатньо змінити значення змінної `--yellow` в одному місці, і зміна застосується до всіх елементів, які використовують цю змінну (кнопки, заголовки, посилання, іконки). Це значно спрощує підтримку та дозволяє швидко експериментувати з дизайном.

Архітектура зберігання даних.

У поточній версії проєкту всі дані (тексти відгуків, зображення, інформація про проєкти) зберігаються статично, безпосередньо в HTML-файлі. Винятком є нові відгуки, які додаються динамічно, але зберігаються лише у пам'яті браузера (в DOM-дереві) і втрачаються при перезавантаженні сторінки. Це є спрощеною моделлю, достатньою для демонстрації архітектурних рішень у межах дипломного проєкту.

Схема архітектури системи.

Для наочного представлення взаємодії всіх компонентів системи можна узагальнити архітектуру за допомогою наступної схеми:

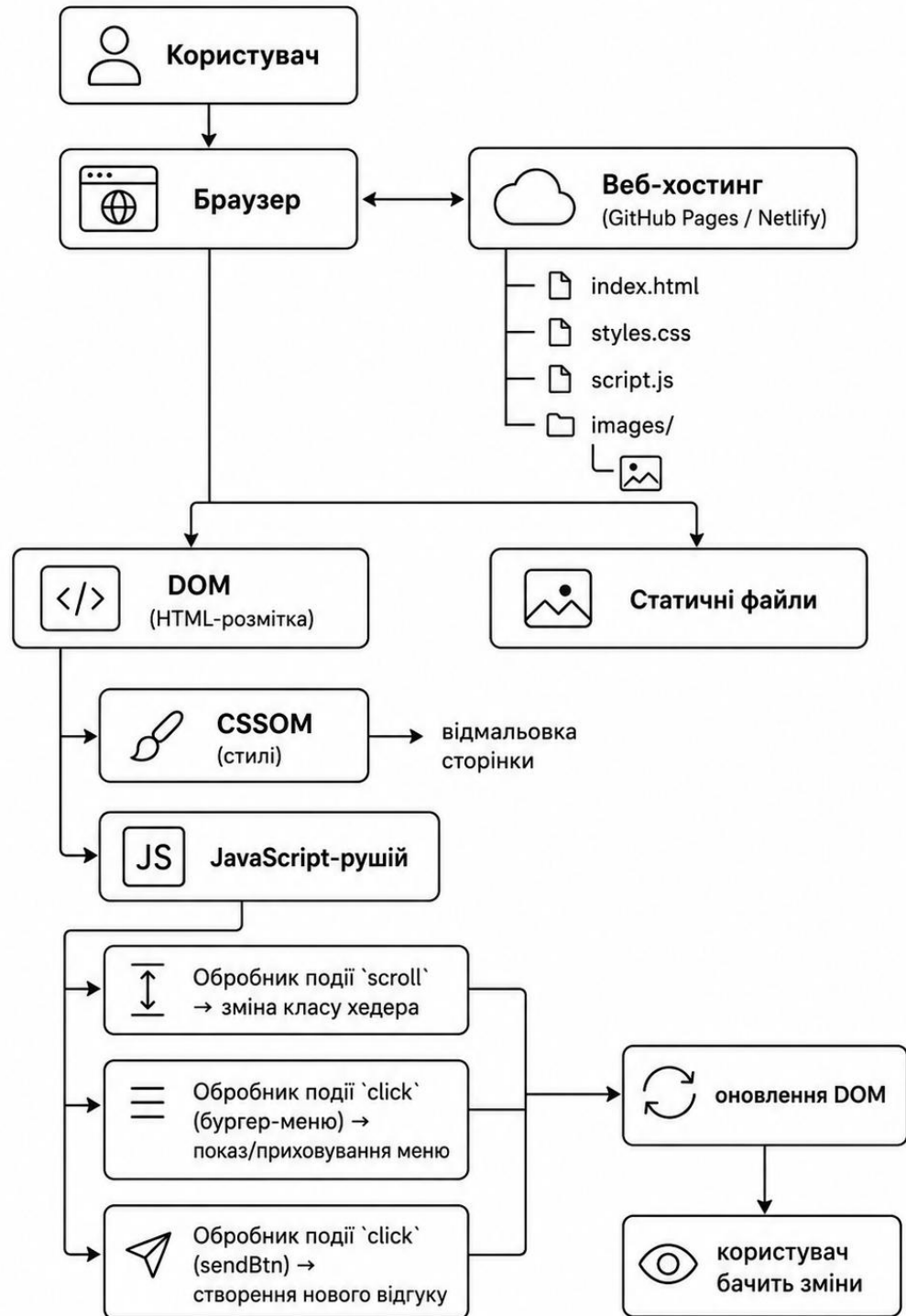


Рисунок 3.3 – Архітектурна схема взаємодії компонентів веб-сайту

Отже, архітектура розробленого веб-сайту персонального портфоліо являє собою статичну клієнт-серверну систему з елементами динамічної поведінки на стороні клієнта. Статичні файли (HTML, CSS, JavaScript, зображення, PDF-резюме) зберігаються на веб-хостингу та роздаються браузеру

за запитом. Весь візуальний контент та базова структура закладені в HTML, стилі винесені в окремий CSS-файл із використанням змінних та адаптивних механізмів, а інтерактивність забезпечується JavaScript-кодом, який реагує на дії користувача (прокручування, кліки) та маніпулює DOM-деревом (додавання класів, створення нових елементів). Така архітектура забезпечує високу продуктивність, простоту підтримки та зручність використання, що є ключовими вимогами до веб-ресурсу такого типу.

## РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

### 4.1. Розробка структури даних та основних сутностей веб-сайту портфоліо

Розробка структури даних для веб-сайту персонального портфоліо має свою специфіку, оскільки на відміну від складних вебзастосунків із серверною частиною та реляційними базами даних, даний проєкт є статичним веб-сайтом. Це означає, що вся інформація, яка відображається на сторінках, зберігається безпосередньо у файлах HTML, CSS та JavaScript, а також у вигляді графічних файлів у директорії images/. Такий підхід є цілком виправданим для персонального портфоліо, оскільки він забезпечує високу швидкодію, простоту розгортання та відсутність потреби в адмініструванні бази даних [1,2]. У межах розробленого проєкту дані організовані у вигляді логічних сутностей, які відповідають структурі контенту сайту. До цих сутностей належать: проєкти портфоліо, інформація про освіту та досвід роботи, відгуки клієнтів, перелік послуг, публікації блогу, контактна інформація та дані про соціальні мережі. Така організація свідчить про те, що сайт реалізовано як цілісний інформаційний ресурс, орієнтований на професійну самопрезентацію.

Організація файлової структури проєкту.

Файлова структура проєкту є основою для зберігання всіх даних. Вона має наступний вигляд:

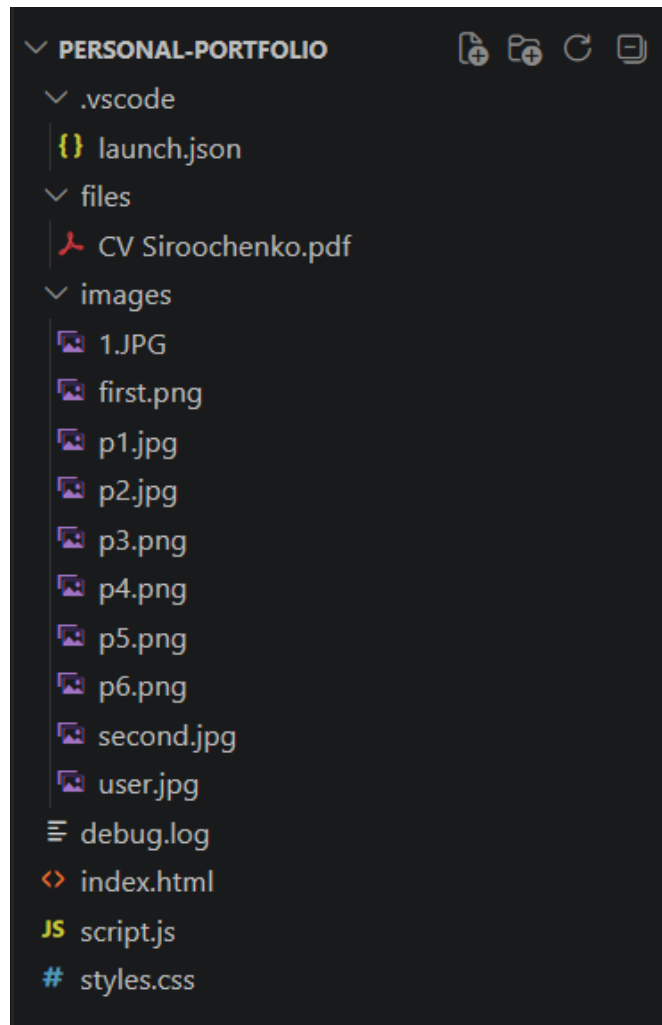


Рисунок 4.1 – Файлова структура веб-сайту персонального портфоліо

Така організація є типовою для статичних веб-сайтів і забезпечує легкість підтримки та подальшого розширення. Всі дані розподілені за типами: структура та текстовий вміст – у HTML, стилі – у CSS, динамічна поведінка – у JS, медіафайли – у директорії images/, документи – у директорії files/.

Сутність «Проект портфоліо».

Центральною сутністю сайту є проект портфоліо. Оскільки автор спеціалізується на фото та відео контенті, кожен проект представлений у вигляді картки із зображенням-прев'ю (кадр з відеоролика) та посиланням на оригінальний відеоролик у Instagram Reels.

У структурі HTML-коду кожен проєкт реалізований наступним чином:

```
html
```

```
<div class="box">
```

```
  <a href="https://www.instagram.com/reel/..." target="_blank">
```

```
    <div class="image">
```

```
      
```

```
      <div class="content">
```

```
        <h2>project #1</h2>
```

```
      </div>
```

```
    </div>
```

```
  </a>
```

```
</div>
```

Кожен проєкт містить наступні атрибути: посилання на відеоролик (атрибут href тегу <a>), зображення-прев'ю (тег <img> з атрибутом src), назву проєкту (вміст тегу <h2>). Така структура є достатньою для демонстрації портфоліо, оскільки основний контент (відеоролики) знаходиться на зовнішній платформі Instagram, а сайт виконує роль каталогу-навігатора. На рисунку нижче представлено зовнішній вигляд секції портфоліо.

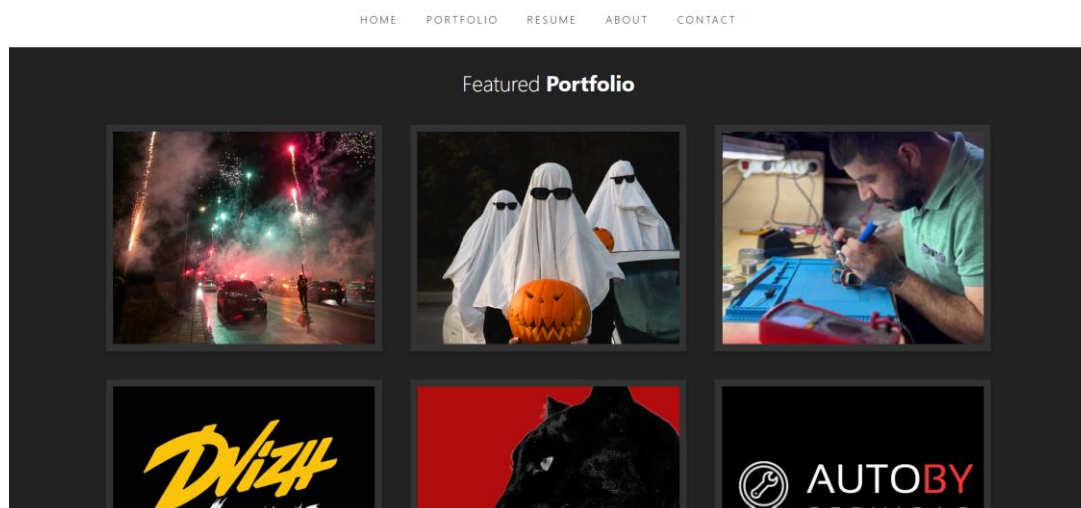


Рисунок 4.2 – Секція портфоліо веб-сайту

Сутність «Освіта та досвід роботи».

Інформація про освіту та професійний досвід є ключовою для персонального портфоліо, оскільки саме вона формує довіру потенційних клієнтів та роботодавців. У розробленому сайті ця інформація зберігається безпосередньо в HTML-коді секції resume. Кожен блок освіти або досвіду має уніфіковану структуру:

```
html
<div class="box">
  <span><i class="material-icons"> calendar_today </i> Sep 2022 -
Present</span>
  <h3>Bachelor in Computer Science</h3>
  <p>
    I am studying Computer Science, focusing on web development,
    interface design, multimedia content creation...
  </p>
  <p class="university">Poltava University of Economics and Trade
(PUET)</p>
</div>
```

Кожен такий блок містить наступні поля: дата або період з іконкою календаря, назва навчальної програми або посади, детальний опис обов'язків або напрямів навчання, назва установи або компанії. Така уніфікація забезпечує легкість сприйняття та дозволяє швидко порівнювати різні періоди професійного життя.

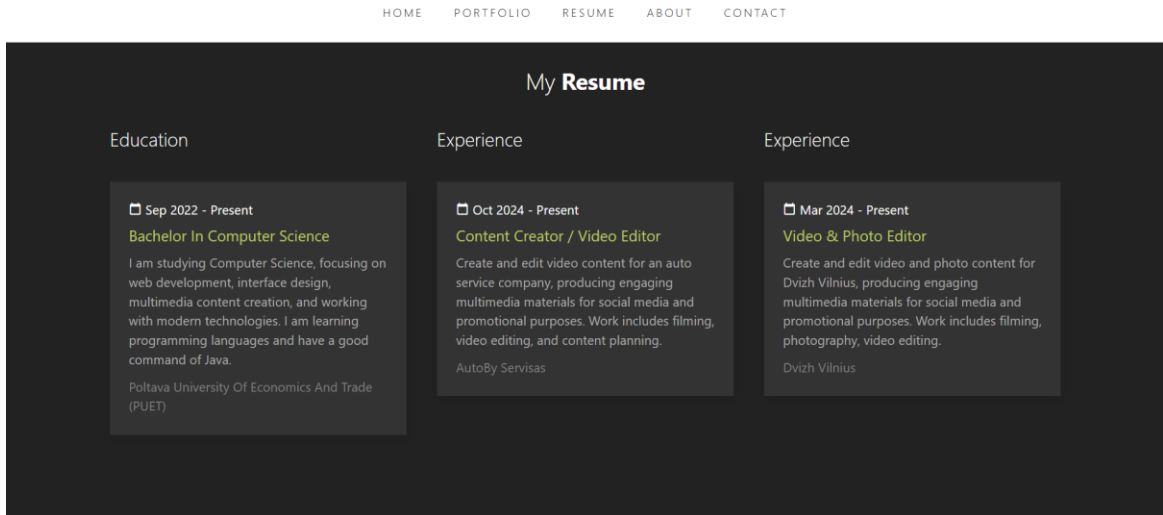


Рисунок 4.3 – Секція резюме (освіта та досвід роботи)

Сутність «Відгук клієнта» (Testimonial).

Відгуки клієнтів є важливим елементом соціального доказу компетентності автора. У проєкті ця сутність реалізована у вигляді карток, які відображаються у секції clients. Кожен відгук має наступну структуру:

html

```
<div class="box">
  <div class="image">
    
  </div>
  <div class="content">
    <p>"Great work! Highly recommended."</p>
    <p>- Ethan, CEO <span>XYZ Inc</span></p>
  </div>
</div>
```

Кожен відгук містить: зображення-аватар клієнта (умовне, однакове для всіх, але може бути кастомізоване), текст відгуку, ім'я клієнта та, за наявності, його посаду або назву компанії. Важливою особливістю є можливість динамічного додавання нових відгуків через форму у контактній секції. Новий відгук створюється JavaScript-кодом і додається до контейнера відгуків без

перезавантаження сторінки. Нижче наведено фрагмент коду, який відповідає за це:

```

javascript
const newTestimonial = document.createElement('div');
newTestimonial.classList.add('box');
newTestimonial.innerHTML = `
  <div class="image">
    
  </div>
  <div class="content">
    <p>"${message}"</p>
    <p>- ${name}</p>
  </div>
`;
testimonialsContainer.appendChild(newTestimonial);

```

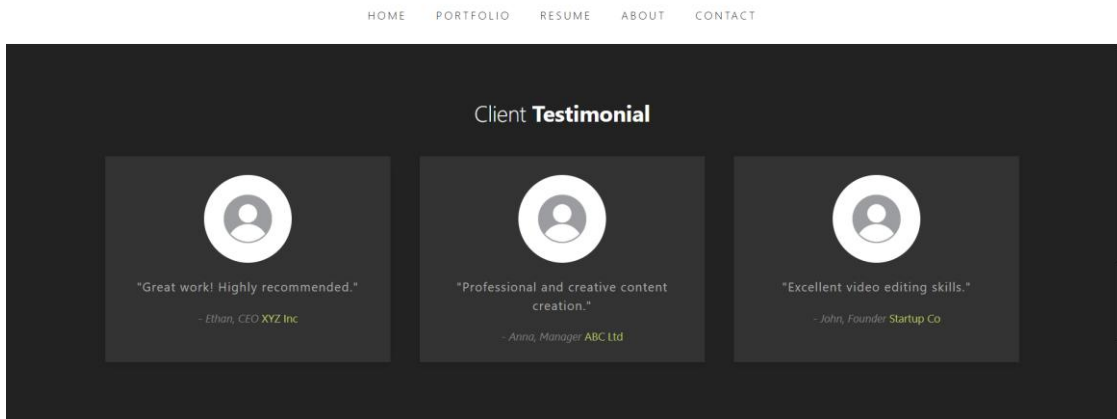


Рисунок 4.4 – Секція відгуків клієнтів

Сутність «Послуга» (Service).

Секція послуг містить інформацію про те, що автор пропонує потенційним клієнтам. Кожна послуга представлена окремою карткою з наступною структурою:

```
html
```

```
<div class="box">
  <i class="icon material-icons">videocam</i>
  <h2>Video Shooting</h2>
  <p>
    I use mobile videography techniques that deliver sharp, stable,
    and visually appealing footage...
  </p>
  <a href="#" class="btn-2">learn more <i class="material-
icons">navigate_next</i></a>
</div>
```

Кожна картка послуги містить: іконку (з бібліотеки Google Material Icons), назву послуги, детальний опис (перелік того, що входить у послугу), кнопку "learn more" (на поточному етапі веде на порожнє посилання #, але в майбутньому може бути замінена на сторінку з детальним описом). На рисунку нижче представлено зовнішній вигляд секції послуг.

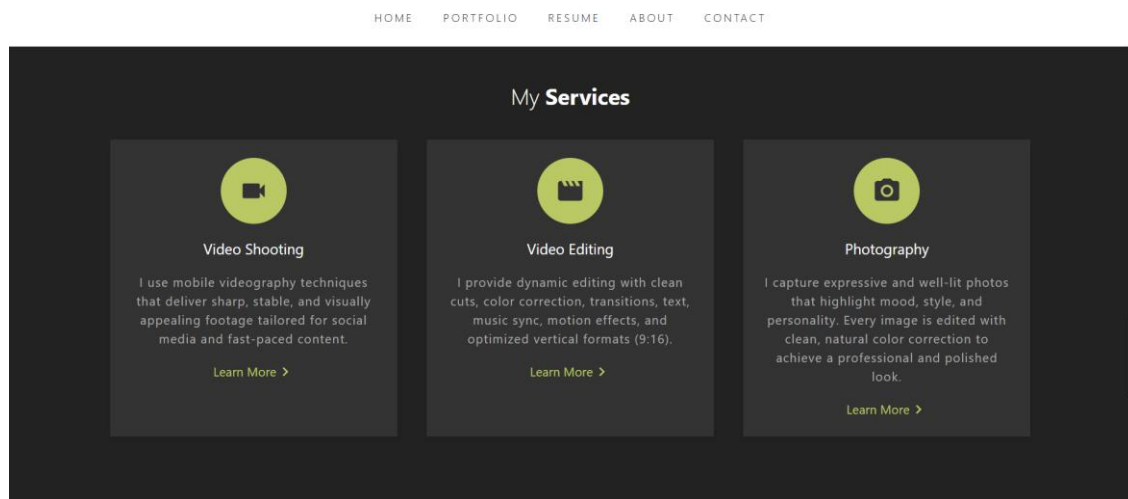


Рисунок 4.5 – Секція послуг

Сутність «Публікація блогу» (Blog).

Секція блогу на поточному етапі розробки є заглушкою, оскільки автор ще не підготував публікації. Однак структура вже закладена:

html

```
<div class="box">
  <div class="image">
    
  </div>
  <div class="content">
    <h2>coming soon</h2>
    <p>coming soon</p>
    <span><i class="material-icons"> calendar_today </i> Jan 9,
2022</span>
    <span><i class="material-icons"> chat_bubble_outline </i> 5
comments</span>
  </div>
</div>
```

Кожна публікація має містити: зображення-прев'ю, заголовок, короткий опис, дату публікації та кількість коментарів. Наявність цієї структури дозволяє в майбутньому легко додавати реальні публікації шляхом заміни вмісту карток.

Сутність «Контактна інформація».

Контактна інформація зберігається у правій колонці секції contact:

html

```
<div class="col">
  <h2>my contact details</h2>
  <p>email</p>
  <span>example@gmail.com</span>
  <br />
  <br />
  <p>phone ua</p>
  <span>+380 (XXX) XX XXXX</span>
  <br />
```

```

<br />
<p>phone lt</p>
<span>+370 (XXX) XX XXX</span>
<br />
<br />
<p>address</p>
<span>Vilnius <br />
  Lithuania</span>
</div>

```

Ця сутність містить: електронну пошту, український та литовський номери телефонів, фізичну адресу. Всі ці дані є статичними та зберігаються безпосередньо в HTML.

Сутність «Соціальні мережі».

У футері сайту розташовані посилання на соціальні мережі автора:

html

```

<div class="socials">
  <a href="https://www.instagram.com/alina_si.prod/" target="_blank">
    <ion-icon name="logo-instagram"></ion-icon>
  </a>
  <a href="https://www.tiktok.com/@alina_si.prod" target="_blank">
    <ion-icon name="logo-tiktok"></ion-icon>
  </a>
  <a href="https://www.facebook.com/profile.php?id=100017327880603"
target="_blank">
    <ion-icon name="logo-facebook"></ion-icon>
  </a>
</div>

```

Кожна соціальна мережа представлена іконкою (з бібліотеки IonIcons) та клікабельним посиланням на реальний профіль автора.

Отже, структура даних розробленого веб-сайту персонального портфоліо є статичною та організованою без використання реляційної бази даних. Всі сутності (проекти портфоліо, освіта та досвід, відгуки, послуги, публікації, контактна інформація, соціальні мережі) зберігаються безпосередньо у файлах HTML, CSS та JavaScript, а також у вигляді графічних файлів. Такий підхід є оптимальним для персонального портфоліо, оскільки забезпечує високу продуктивність, простоту розгортання та відсутність потреби в адмініструванні бази даних. Всі дані логічно структуровані за секціями, що полегшує їх підтримку та подальше оновлення.

#### **4.2. Реалізація клієнтської частини та динамічної поведінки веб-сайту**

Оскільки розроблений веб-сайт є статичним і не має серверної частини, вся логіка обробки подій, маніпуляції з DOM-деревом та динамічна зміна вмісту реалізована на стороні клієнта за допомогою мови JavaScript. Це є типовим підходом для статичних веб-сайтів, де не потрібне зберігання даних на сервері або складна бізнес-логіка. JavaScript-код виконується безпосередньо в браузері користувача після завантаження HTML-сторінки та CSS-стилів. У розробленому проєкті реалізовано три основні групи клієнтської логіки: керування навігацією (бургер-меню та зміна стилю хедера), динамічне додавання відгуків, а також обробка подій прокручування сторінки. Нижче детально описано кожен з цих груп.

Керування навігаційним меню (бургер-меню).

На мобільних пристроях з шириною екрану менше 768px стандартне горизонтальне меню ховається, а замість нього з'являється іконка меню (бургер-меню). Це реалізовано за допомогою медіа-запиту в CSS та JavaScript-обробника події click. Коли користувач натискає на іконку меню, до навігаційної панелі додається клас "show", який змінює значення CSS-

властивості `transform` з `translateX(100%)` на `translateX(0)`, в результаті чого меню "виїжджає" з правого краю екрану.

```
javascript
const navbar = document.querySelector('.header .navbar');
const menuButton = document.querySelector('.header .menu');

menuButton.addEventListener('click', () => {
  navbar.classList.toggle('show');
});
```

При повторному натисканні клас "show" видаляється, і меню ховається за межі екрану. Крім того, при прокручуванні сторінки меню автоматично закривається (клас "show" видаляється), що покращує користувацький досвід [7]. На рисунку нижче представлено зовнішній вигляд бургер-меню на мобільному пристрої.

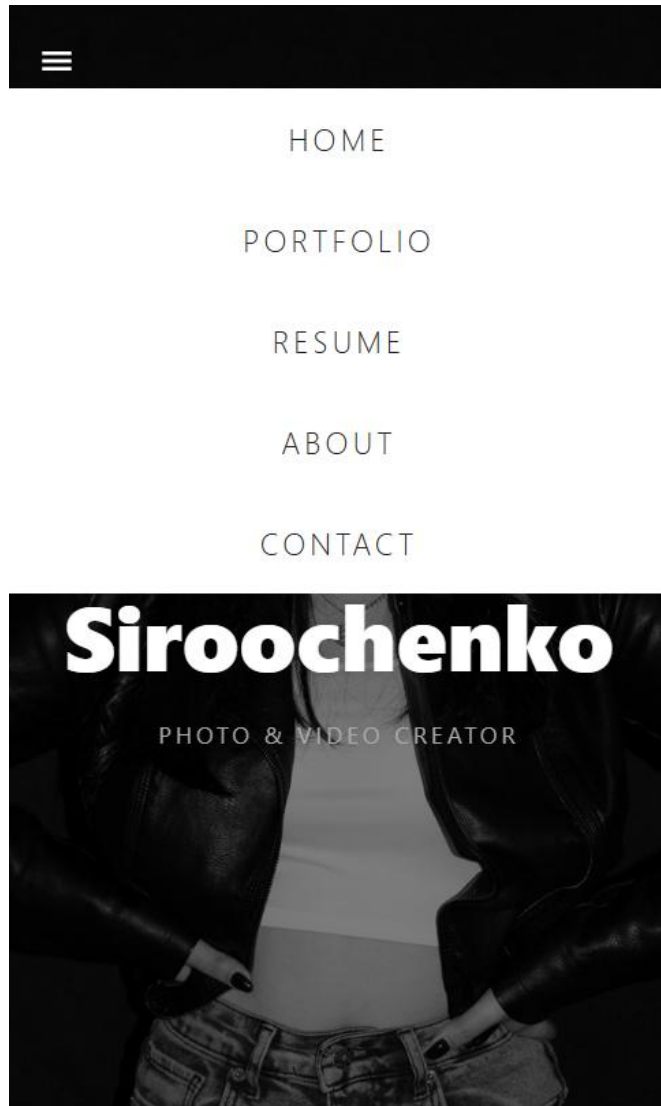


Рисунок 4.6 – Бургер-меню на мобільному пристрої

Зміна стилю хедера при прокручуванні сторінки.

Для покращення візуальної ієрархії та забезпечення кращої помітності навігації при прокручуванні сторінки, хедер змінює свій зовнішній вигляд. Коли позиція прокрутки (`window.scrollY`) стає більшою за 0, до хедера додається клас "active". Цей клас змінює фон хедера з напівпрозорого на білий, колір посилань – з білого на темний, а внутрішні відступи зменшуються.

```
javascript
```

```
window.addEventListener('scroll', () => {
  navbar.classList.remove('show');
  const header = document.querySelector('.header');
```

```

if (window.scrollY > 0) {
    header.classList.add('active');
} else {
    header.classList.remove('active');
}
});

```

Такий підхід є стандартним для сучасних веб-сайтів і забезпечує зручність навігації, оскільки хедер залишається видимим і контрастним на тлі будь-якого контенту. Нижче наведено порівняння вигляду хедера у верхній частині сторінки та після прокручування.

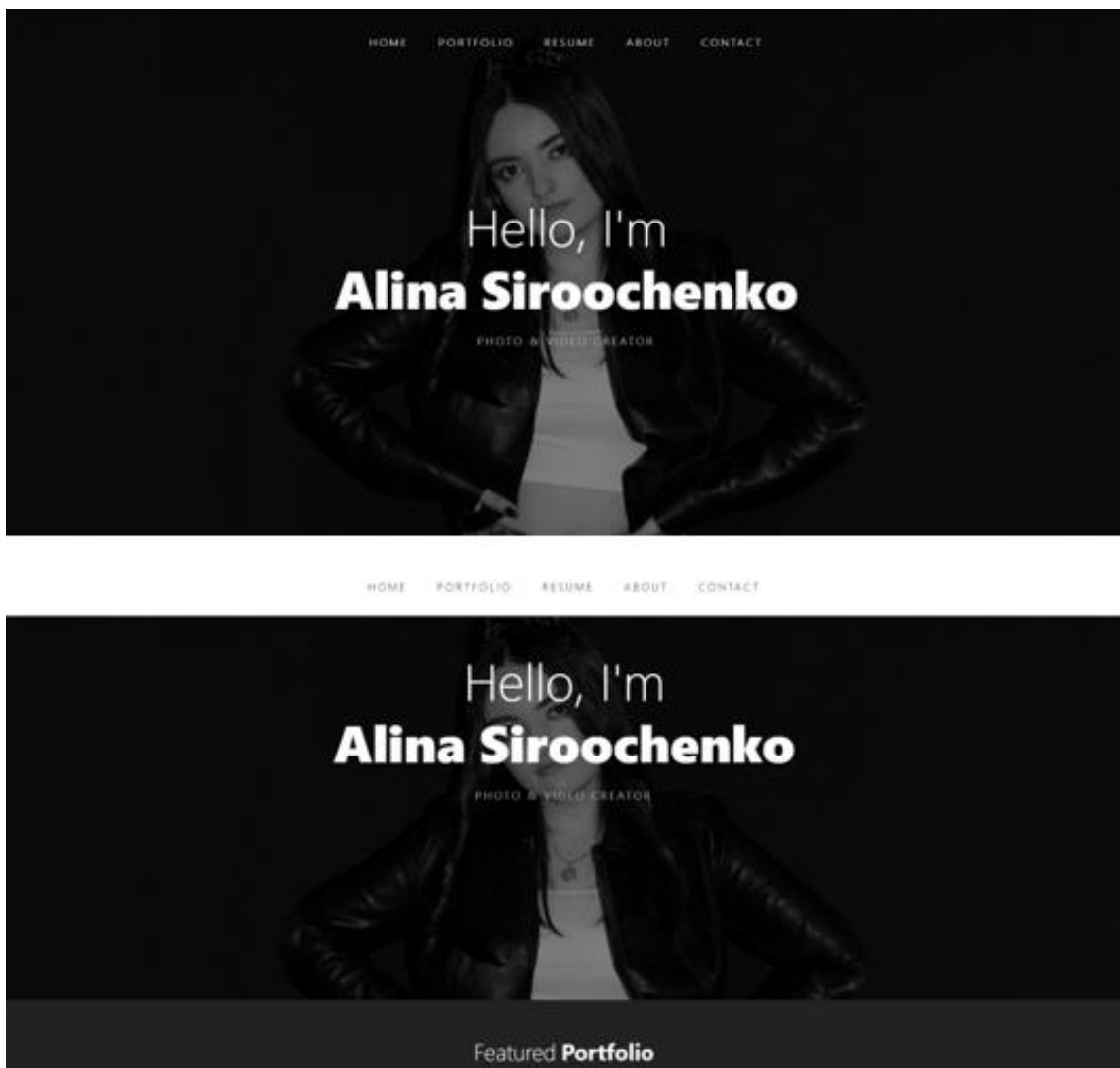


Рисунок 4.7 – Зміна стилю хедера при прокручуванні сторінки

Динамічне додавання відгуків клієнтів.

Найбільш складною та показовою функцією клієнтської логіки є динамічне додавання нових відгуків через форму у контактній секції. Цей механізм демонструє можливість інтерактивної взаємодії з користувачем без перезавантаження сторінки. Схема роботи наступна:

1. Користувач заповнює поля "your name" та "write a message" у контактній формі.
2. Користувач натискає кнопку "send message".
3. Спрацьовує обробник події click, доданий до кнопки. Обробник викликає `e.preventDefault()`, що запобігає стандартному відправленню форми та перезавантаженню сторінки.
4. JavaScript зчитує значення з полів введення за допомогою властивості `value`.
5. Виконується перевірка: якщо обидва поля не порожні, процес триває; якщо будь-яке з полів порожнє, з'являється повідомлення `alert` із проханням заповнити поля.
6. JavaScript створює новий HTML-елемент (`div`) за допомогою методу `document.createElement()`, додає йому клас "box", а потім за допомогою властивості `innerHTML` наповнює його вмістом.
7. Отриманий новий елемент додається до контейнера відгуків за допомогою методу `appendChild()`.
8. Поля форми очищуються (`nameInput.value = ""; messageInput.value = ""`).

Код, який реалізує цю логіку, наведено нижче:

```
javascript
const sendBtn = document.getElementById('sendBtn');
const nameInput = document.getElementById('name');
const messageInput = document.getElementById('message');
const testimonialsContainer =
document.getElementById('testimonialsContainer');
```

```

sendBtn.addEventListener('click', (e) => {
  e.preventDefault();

  const name = nameInput.value.trim();
  const message = messageInput.value.trim();

  if (name === "" || message === "") {
    alert('Please enter your name and message!');
    return;
  }

  const newTestimonial = document.createElement('div');
  newTestimonial.classList.add('box');
  newTestimonial.innerHTML = `
    <div class="image">
      
    </div>
    <div class="content">
      <p> "${message}" </p>
      <p>- ${name}</p>
    </div>
  `;
  testimonialsContainer.appendChild(newTestimonial);

  nameInput.value = "";
  messageInput.value = "";
});

```

Важливо зазначити, що на поточному етапі розробки нові відгуки не

зберігаються на сервері і втрачаються при перезавантаженні сторінки. Однак для демонстрації функціональних можливостей у межах дипломного проєкту цей підхід є достатнім. У майбутньому може бути додана інтеграція з серверним бекендом або хмарним сховищем для постійного зберігання відгуків. На рисунку нижче представлено процес додавання нового відгуку.

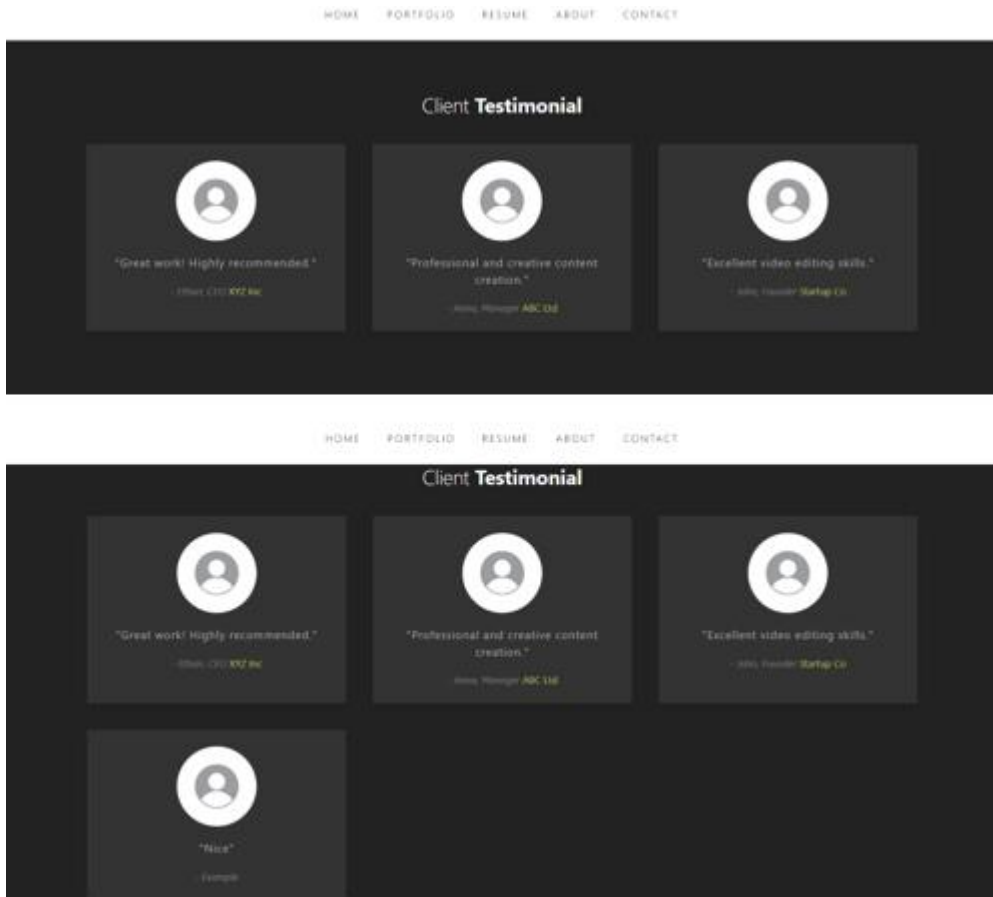


Рисунок 4.8 – Динамічне додавання відгуку через контактну форму

Обробка події завантаження сторінки.

Для того, щоб хедер одразу мав правильний вигляд при завантаженні сторінки (якщо користувач відкрив сторінку не з самого верху), додано обробник події load:

```
javascript
window.addEventListener('load', () => {
  const header = document.querySelector('.header');
  if (window.scrollY > 0) {
```

```

        header.classList.add('active');
    } else {
        header.classList.remove('active');
    }
});

```

Це гарантує, що незалежно від того, з якої позиції користувач починає перегляд, стиль хедера буде коректним.

Взаємодія з зовнішніми ресурсами та бібліотеками.

Для роботи з іконками використовуються дві зовнішні бібліотеки: Google Material Icons та IonIcons. Вони підключені через CDN (мережу доставки контенту) у секції <head> HTML-документа:

```

html
<link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet" />
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>

```

Такий підхід дозволяє використовувати велику бібліотеку іконок без необхідності зберігати їх локально, що економить місце та спрощує підтримку.

Таким чином, клієнтська частина веб-сайту персонального портфоліо реалізована засобами чистого JavaScript без використання додаткових фреймворків. У ній реалізовано три ключові функціональні блоки: адаптивне бургер-меню для мобільних пристроїв, динамічна зміна стилю хедера при прокручуванні та інтерактивне додавання нових відгуків через форму. Вся логіка побудована на обробниках подій (click, scroll, load) та маніпуляціях з DOM-деревом (додавання/видалення класів, створення нових елементів, додавання до контейнерів). Такий підхід забезпечує високу продуктивність, оскільки весь код виконується локально в браузері користувача, та не потребує серверної інфраструктури [3,4].

### **4.3. Розробка користувацького інтерфейсу та основних функціональних можливостей**

Розробка користувацького інтерфейсу веб-сайту персонального портфоліо виконувалася з урахуванням необхідності поєднання естетичної привабливості, зручної навігації, адаптивності до різних типів пристроїв та підтримки інтерактивної взаємодії з користувачем. У межах проєкту інтерфейс реалізовано засобами HTML5, CSS3 та JavaScript з використанням зовнішніх бібліотек іконок (Google Material Icons та IonIcons). Такий підхід дозволив сформувати цілісне візуальне середовище, у якому кожна секція сайту виконує свою функцію: головна сторінка представляє автора, портфоліо демонструє приклади робіт, резюме інформує про освіту та досвід, секція "про себе" розкриває творчу особистість, відгуки підтверджують компетентність, послуги описують пропозиції, блог (в розробці) буде публікувати статті, а контакти забезпечують зворотний зв'язок.

Загальна структура інтерфейсу та візуальне оформлення.

Сайт являє собою односторінковий веб-ресурс із плавною навігацією, де основні розділи розташовані послідовно один за одним. Така архітектура є оптимальною для персонального портфоліо, оскільки дозволяє користувачеві отримати повну інформацію про автора без необхідності переходу між окремими сторінками. Візуальне оформлення виконане в темній кольоровій гамі: темно-сіре тло (#222222 для основного фону, #333333 для фону карток), білий текст для контрасту та жовтий колір (#bac964) як акцентний для кнопок, заголовків та інтерактивних елементів. Така кольорова схема є сучасною, створює професійне враження та забезпечує належний контраст для читання тексту [5].

Головна сторінка (hero section) містить велике фонове зображення із застосованим фільтром градації сірого (filter: grayscale(100%)), що створює драматичний художній ефект. Поверх зображення розташований текст: "hello,

i'm Alina Siroochenko" та "Photo & Video Creator". Це одразу дає відвідувачеві зрозуміти, хто є автором сайту та якою діяльністю він займається.

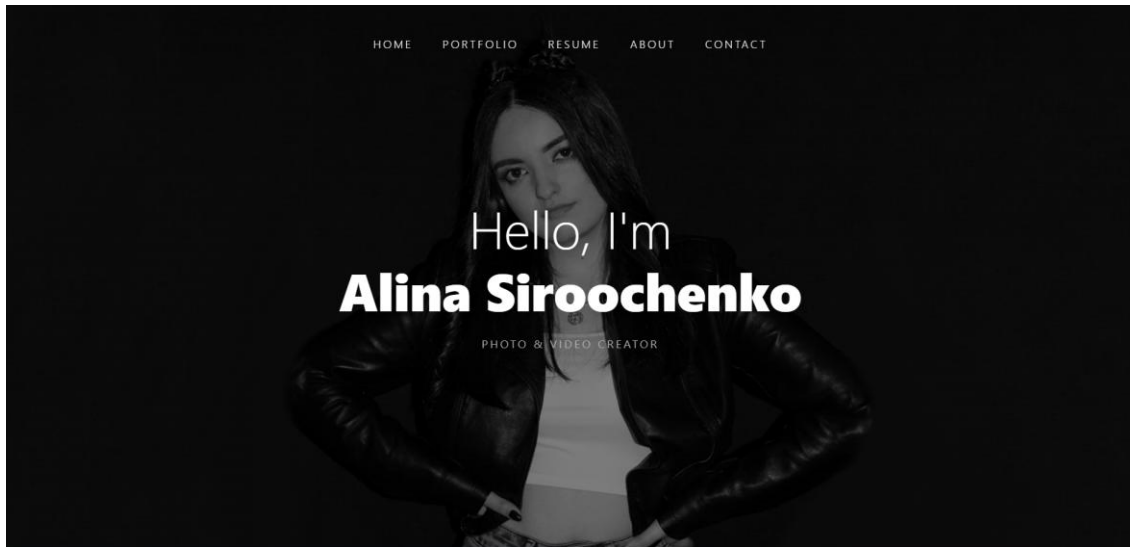


Рисунок 4.9 – Головна сторінка (hero section) веб-сайту

Адаптивний дизайн та медіа-запити.

Однією з ключових вимог до сучасного веб-сайту є коректне відображення на різних пристроях: від широких моніторів до смартфонів. Для забезпечення адаптивності використано медіа-запити (media queries) у файлі styles.css. Визначено три контрольні точки (breakpoints):

- 1200px – для ноутбуків та невеликих моніторів: зменшуються відступи хедера та секцій.
- 991px – для планшетів: зменшується базовий розмір шрифту (font-size: 60%).
- 768px – для мобільних пристроїв: горизонтальне меню ховається, з'являється бургер-меню; меню стає вертикальним, займає всю ширину екрану [7].
- 450px – для маленьких смартфонів: додаткове зменшення базового розміру шрифту (font-size: 55%) та розміру тексту в hero-секції.

Приклад медіа-запиту для мобільних пристроїв:

css

```
@media (max-width: 768px) {
```

```
.header .menu {
  display: block;
}
.header .navbar {
  position: fixed;
  top: 6rem;
  left: 0;
  right: 0;
  background: var(--white);
  display: flex;
  flex-flow: column;
  text-align: center;
  transform: translateX(100%);
}
.header .navbar a {
  color: var(--dark-bg);
  margin: 2rem 0;
  font-size: 2rem;
}
header .navbar.show {
  transform: translateX(0);
}
}
```

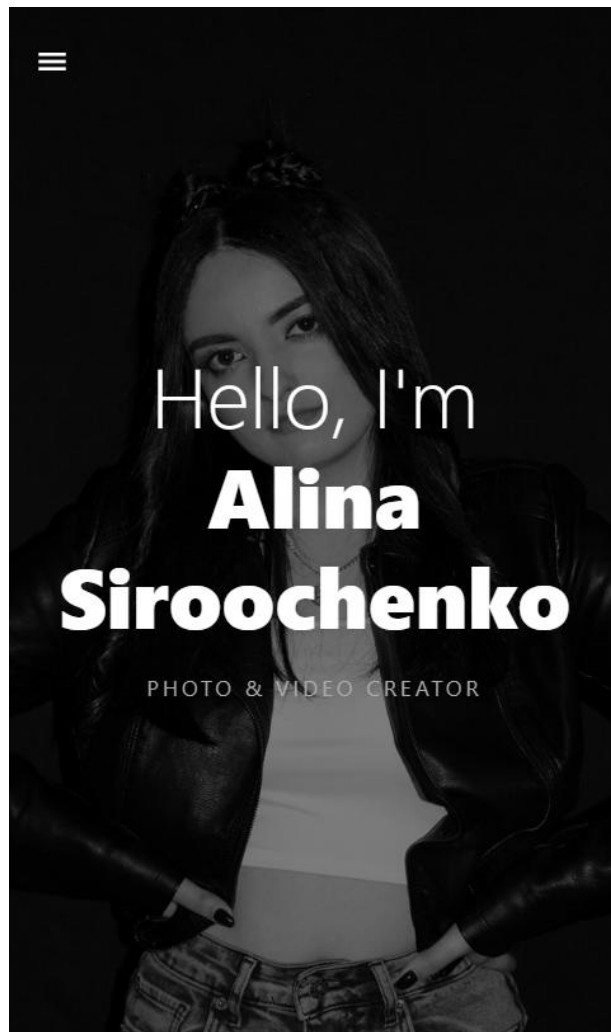


Рисунок 4.10 – Адаптивне відображення сайту на мобільному пристрої

Секція портфолію.

Секція портфолію є візуальним центром сайту. Вона реалізована у вигляді сітки карток, кожна з яких містить зображення-прев'ю проєкту. При наведенні курсора відбувається ефект масштабування зображення (`transform: scale(1.2)`) та з'являється підпис з назвою проєкту, що "виїжджає" знизу. Клік по картці відкриває відповідний відеоролик в Instagram Reels у новій вкладці (атрибут `target="_blank"`). Сітка побудована з використанням CSS Grid, що забезпечує рівномірне розташування карток:

```
css
.portfolio .box-container {
  display: grid;
```

```

grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
gap: 4rem;
}

```

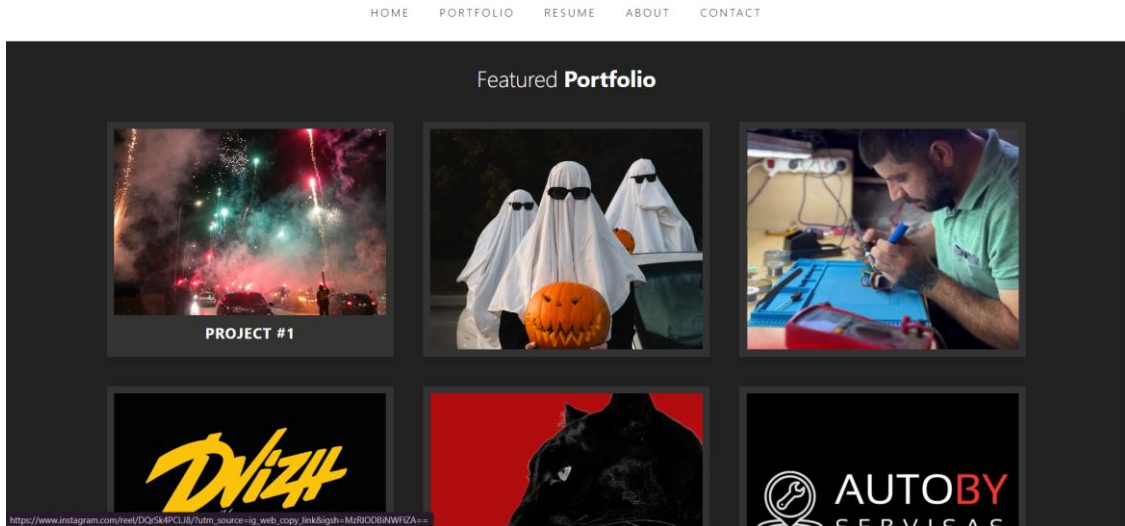


Рисунок 4.11 – Секція портфоліо з ефектом при наведенні

Секція резюме.

Секція резюме побудована з використанням CSS Grid і поділена на три колонки. Кожна колонка містить блоки з інформацією про освіту або досвід роботи. Важливою особливістю є використання іконок календаря з бібліотеки Google Material Icons для візуального виділення дат. Кольорове виділення назв установ (клас .university) дозволяє швидко ідентифікувати місце навчання або роботи. Блоки мають трохи темніший фон (`background-color: var(--light-bg)`) та тінь (`box-shadow: var(--box-shadow)`), що створює ефект глибини.



Рисунок 4.12 – Деталізація блоків освіти та досвіду з іконками

#### Секція "Про себе" (About).

Ця секція складається з двох колонок: лівої – з фотографією автора, правої – з текстовим описом. Фотографія займає 100% висоти контейнера завдяки властивості `object-fit: cover`. Текстовий опис підкреслює творчий характер діяльності, захоплення малюванням, зйомкою та відеомонтажем. Кнопка "download CV" (оформлена класом `.btn`) дозволяє завантажити резюме у форматі PDF. Кнопка має жовтий фон, білий текст та заокруглені кути (`border-radius: 2rem`), що робить її помітною та клікабельною.

html

```
<a href="files/CV Siroochenko.pdf" download class="btn">download CV</a>
```

Використання атрибута `download` у тегу `<a>` підказує браузеру, що файл потрібно завантажити, а не відкривати у вікні.

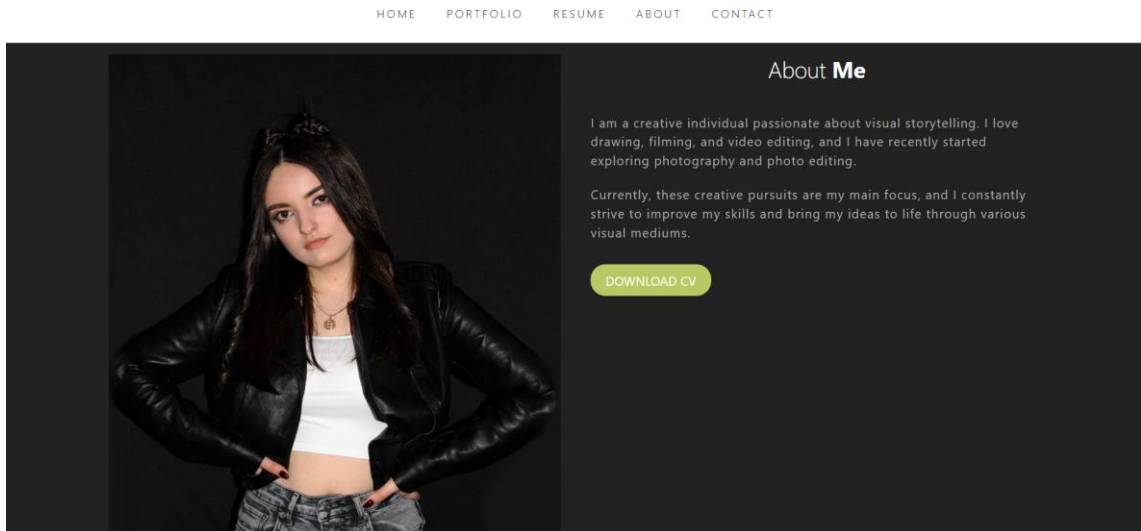


Рисунок 4.13 – Секція "Про себе" з фотографією автора та кнопкою завантаження резюме

Секція відгуків клієнтів.

Секція відгуків демонструє картки з аватарами клієнтів, текстами відгуків та іменами авторів. Ключовою особливістю є можливість додавання нових відгуків через форму у контактній секції (цей механізм детально описано у підрозділі 4.2). Візуально картки відгуків мають центроване вирівнювання тексту, аватар у вигляді кола (`border-radius: 50%`) та жовтий колір для виділення імен клієнтів.

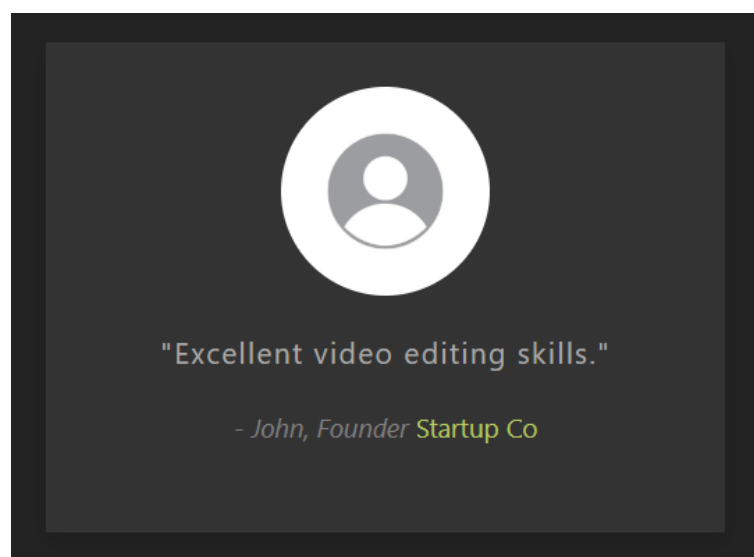


Рисунок 4.14 – Дизайн карток відгуків клієнтів

Секція послуг.

Секція послуг містить три картки з іконками (з бібліотеки Google Material Icons), заголовками та детальними описами. Іконки мають великий розмір (font-size: 4rem), жовтий фон та білий колір, що робить їх візуальним центром кожної картки. Опис кожної послуги є детальним та інформативним. Кнопка "learn more" має ефект зміщення іконки при наведенні (hover і { padding-left: 1.5rem; }), що додає інтерактивності.

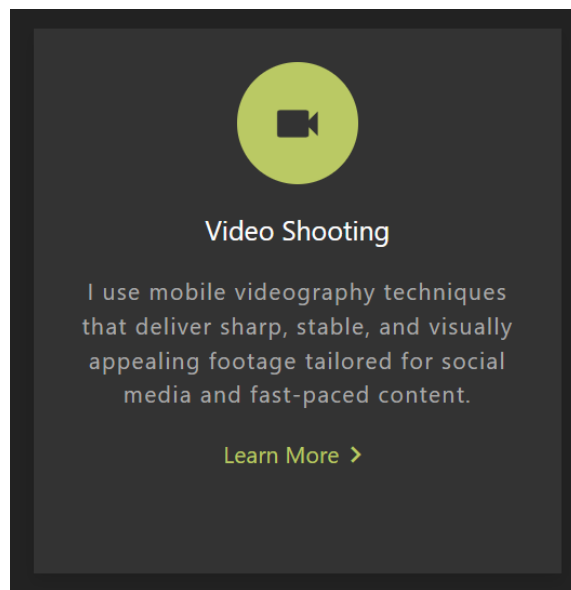


Рисунок 4.15 – Картки послуг з великими іконками

Секція блогу.

Секція блогу наразі є заглушкою, але її візуальне оформлення вже закладено. Кожна картка публікації має зображення, заголовок, короткий опис, дату та кількість коментарів. У майбутньому, коли з'являться реальні публікації, достатньо буде замінити вміст карток, не змінюючи структуру HTML та CSS.

Контактна секція.

Контактна секція складається з двох колонок. Ліва колонка містить форму для залишення повідомлення (поля для імені, email, телефону та тексту повідомлення). Всі поля мають однакове оформлення: темний фон (background-color: var(--light-bg)), білий текст, заокруглені кути (border-radius: 0.5rem) та

внутрішні відступи. Кнопка "send message" має такий самий стиль, як і кнопка завантаження резюме (жовтий фон, заокруглені кути).

Права колонка містить контактні деталі: електронну пошту, номери телефонів (український та литовський), фізичну адресу у Вільнюсі. Назви полів (email, phone ua, phone lt, address) виділені жовтим кольором, що полегшує сканування інформації.

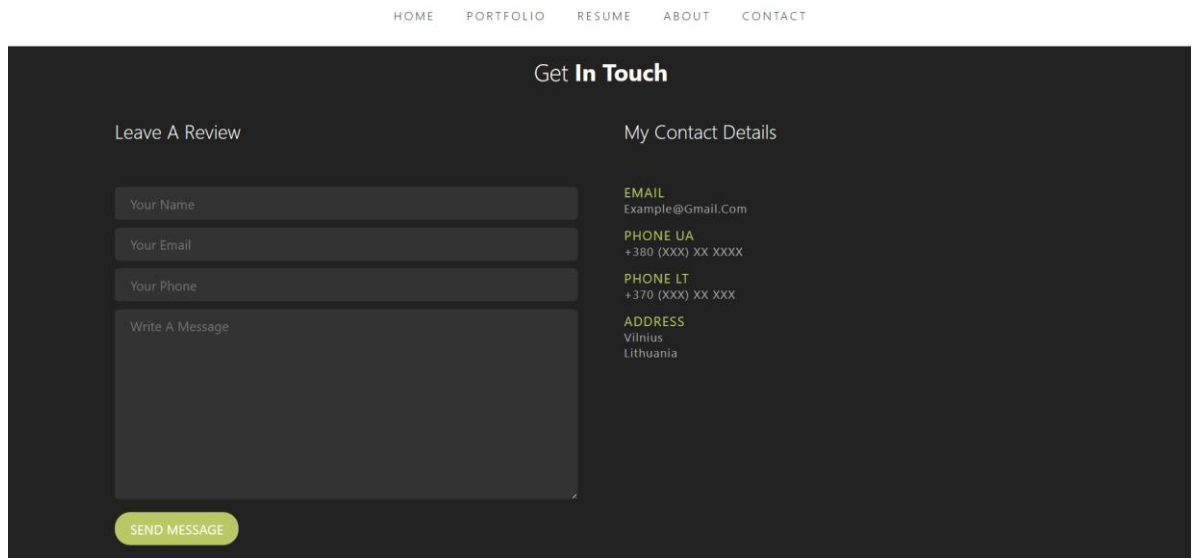


Рисунок 4.16 – Контактна секція з формою та контактними деталями

Футер та соціальні мережі.

Футер містить іконки соціальних мереж (Instagram, TikTok, Facebook) з бібліотеки IonIcons. Іконки мають жовтий колір, темне коло (background-color: var(--light-bg), border-radius: 50%) та ефект при наведенні (курсор змінюється на pointer). Посилання відкриваються у новій вкладці завдяки атрибуту target="\_blank".

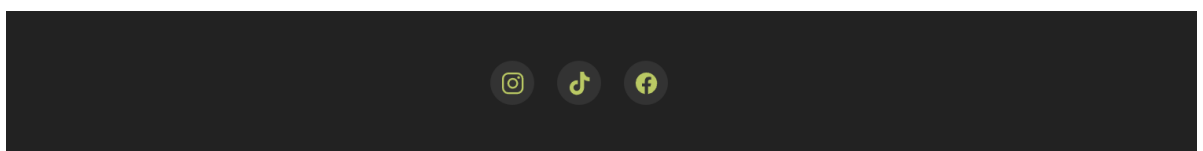


Рисунок 4.17 – Футер з посиланнями на соціальні мережі

Отже, користувацький інтерфейс розробленого веб-сайту персонального портфоліо побудовано як цілісну односторінкову систему з плавною

навігацією. Реалізація інтерфейсу засобами HTML5, CSS3 та JavaScript забезпечила семантичну структуру, адаптивність дизайну (завдяки медіа-запитам та гнучким сіткам Grid/Flexbox), візуальну привабливість (темна тема, жовті акценти, ефекти при наведенні) та інтерактивність (бургер-меню, зміна стилю хедера, додавання відгуків). Основні функціональні можливості веб-сайту охоплюють демонстрацію портфоліо, представлення професійного досвіду та освіти, інформацію про автора, відгуки клієнтів, перелік послуг, контактну форму (з можливістю додавання відгуків) та посилання на соціальні мережі. Саме така організація інтерфейсу забезпечує практичну придатність системи для професійної самопрезентації [1,2,5].

## ВИСНОВКИ

У результаті виконання дипломної роботи досягнуто поставленої мети, яка полягала у розробці веб-сайту персонального портфоліо для фото та відео редактора.

У ході роботи було виконано основні завдання дослідження. Проведено аналіз існуючих веб-ресурсів аналогічного призначення, зокрема портфоліо розробників, дизайнерів та представників креативних індустрій, що дозволило виявити типові структури, сильні та слабкі сторони таких рішень. На основі цього аналізу було сформовано основні функціональні вимоги до веб-сайту портфоліо, які включають представлення інформації про автора, демонстрацію проєктів, відображення професійного досвіду та освіти, публікацію відгуків клієнтів, перелік послуг, а також забезпечення зворотного зв'язку через контактну форму.

Обґрунтовано вибір технологій і засобів розробки. Для реалізації проєкту використано HTML5, CSS3 та JavaScript як базові технології веб-розробки. Додатково застосовано бібліотеки Google Material Icons для іконографії та IonIcons для іконок соціальних мереж. Такий набір технологій забезпечив можливість створення легкого, швидкого, адаптивного та візуально привабливого веб-сайту, який не потребує серверної інфраструктури та може бути розміщений на будь-якому хостингу статичних файлів.

У роботі спроектовано структуру веб-сайту, розроблено архітектуру клієнтської частини та реалізовано основні модулі системи. Створено вісім основних інформаційних секцій: головна сторінка (hero section) з представленням автора, секція портфоліо для демонстрації проєктів із посиланнями на відеоролики в Instagram Reels, секція резюме з інформацією про освіту та досвід роботи, секція "про себе" з фотографією автора, творчим описом та кнопкою завантаження резюме, секція відгуків клієнтів, секція

послуг з детальним описом пропозицій, секція блогу (заглушка для майбутніх публікацій) та контактна секція з формою зворотного зв'язку та контактними деталями. Завершує сторінку футер із посиланнями на соціальні мережі автора (Instagram, TikTok, Facebook).

Окрему увагу приділено реалізації адаптивного дизайну з використанням медіа-запитів, що забезпечує коректне відображення сайту на пристроях з різною шириною екрану (монітори, планшети, смартфони). Для мобільних пристроїв реалізовано бургер-меню, яке з'являється замість горизонтальної навігації при ширині екрану менше 768px. CSS Grid та Flexbox використано для побудови гнучких сіткових структур у секціях портфоліо, резюме, відгуків, послуг, блогу та контактів, що гарантує рівномірне розташування елементів незалежно від їхнього вмісту.

Розроблено користувацький інтерфейс веб-сайту в темній кольоровій гамі з акцентним жовтим кольором, що створює професійне враження, забезпечує належний контраст для читання тексту та підкреслює інтерактивні елементи. Використано CSS-змінні для централізованого керування кольорами та іншими параметрами, що значно спрощує підтримку та внесення змін до дизайну. Реалізовано плавні анімаційні ефекти при наведенні на елементи портфоліо (масштабування зображень, поява підписів) та на кнопках (зміщення іконок), що покращує користувацький досвід.

У процесі реалізації застосунку забезпечено динамічну поведінку за допомогою чистого JavaScript без використання додаткових фреймворків. Реалізовано три ключові функціональні блоки: адаптивне бургер-меню (відкриття/закриття при кліку, автоматичне закриття при прокручуванні), динамічну зміну стилю хедера при прокручуванні сторінки (зміна фону, кольору посилань та внутрішніх відступів), а також інтерактивне додавання нових відгуків клієнтів через контактну форму без перезавантаження сторінки. Для форми додавання відгуків реалізовано валідацію введених даних (перевірка на порожнечу полів) та інформування користувача про помилки.

Проведено перевірку працездатності системи, що підтвердила коректність реалізації основних функціональних можливостей: коректне відображення всіх секцій на різних пристроях, робота навігаційного меню, функціонування бургер-меню на мобільних пристроях, зміна стилю хедера при прокручуванні, відкриття відеороликів у Instagram Reels при кліку на картки портфоліо, завантаження PDF-резюме при натисканні на відповідну кнопку, додавання нових відгуків через контактну форму з валідацією введених даних.

За результатами виконання дипломної роботи опубліковано тези на XLIX Міжнародній науковій конференції студентів та аспірантів «Актуальні питання розвитку науки та забезпечення якості освіти у XXI столітті» (м. Полтава, 23 квітня 2026 р.) [4].

Отже, розроблений веб-сайт персонального портфоліо може бути використаний як практичне програмне рішення для професійної самопрезентації фото та відео редактора. Сайт забезпечує зручну демонстрацію портфоліо (з інтеграцією відеоконтенту з Instagram), структуроване представлення професійного досвіду та освіти, можливість отримання відгуків від клієнтів, інформування про перелік послуг, а також канали зв'язку через контактну форму та соціальні мережі. Завдяки використанню статичних технологій (HTML, CSS, JavaScript) сайт є легким, швидким, не потребує серверної інфраструктури та може бути розміщений на будь-якому хостингу статичних файлів (GitHub Pages, Netlify, Vercel тощо). Адаптивний дизайн забезпечує коректне відображення на всіх типах пристроїв – від настільних моніторів до смартфонів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дакетт Дж. HTML та CSS: розробка та дизайн веб-сайтів / Джон Дакетт. – Indianapolis : Wiley, 2011. – 490 с.
2. Крокфорд Д. JavaScript: сильні сторони / Дуглас Крокфорд. – Sebastopol : O'Reilly Media, 2008. – 172 с.
3. Кошова О. П., Ольховська О. В., Тацій Д. С., Олексійчук Ю. Ф., Черненко О. О. Розробка веб-додатків та сервісів на платформі Node.js // Таврійський науковий вісник. Серія: Технічні науки. – 2023. – Вип. 2. – С. 78–89. – DOI: <https://doi.org/10.32782/tnv-tech.2023.2.9>
4. Сірооченко А. Б. Проектування веб-сайту власного портфоліо // Актуальні питання розвитку науки та забезпечення якості освіти у XXI столітті : тези доповідей XLIX Міжнародної наукової конференції студентів та аспірантів (м. Полтава, 23 квітня 2026 р.). – Полтава : ПУЕТ, 2026. – С. 455.
5. Документація MDN Web Docs: HTML [Електронний ресурс] / Mozilla Developer Network. – Режим доступу: <https://developer.mozilla.org/uk/docs/Web/HTML> (дата звернення: 02.06.2026).
6. Документація MDN Web Docs: CSS [Електронний ресурс] / Mozilla Developer Network. – Режим доступу: <https://developer.mozilla.org/uk/docs/Web/CSS> (дата звернення: 02.06.2026).
7. Документація MDN Web Docs: JavaScript [Електронний ресурс] / Mozilla Developer Network. – Режим доступу: <https://developer.mozilla.org/uk/docs/Web/JavaScript> (дата звернення: 02.06.2026).

8. Документація MDN Web Docs: CSS Grid Layout [Електронний ресурс] / Mozilla Developer Network. – Режим доступу: [https://developer.mozilla.org/uk/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/uk/docs/Web/CSS/CSS_Grid_Layout) (дата звернення: 02.06.2026).
9. Документація MDN Web Docs: Flexbox [Електронний ресурс] / Mozilla Developer Network. – Режим доступу: [https://developer.mozilla.org/uk/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout](https://developer.mozilla.org/uk/docs/Web/CSS/CSS_Flexible_Box_Layout) (дата звернення: 02.06.2026).
10. Документація MDN Web Docs: Медіа-запити [Електронний ресурс] / Mozilla Developer Network. – Режим доступу: [https://developer.mozilla.org/uk/docs/Web/CSS/Media\\_Queries](https://developer.mozilla.org/uk/docs/Web/CSS/Media_Queries) (дата звернення: 02.06.2026).
11. Google Material Icons – офіційна документація [Електронний ресурс] / Google Fonts. – Режим доступу: <https://fonts.google.com/icons> (дата звернення: 02.06.2026).
12. IonIcons – офіційна документація [Електронний ресурс] / Ionic Framework. – Режим доступу: <https://ionic.io/ionicons> (дата звернення: 02.06.2026).
13. Що таке GitHub Pages? [Електронний ресурс] / GitHub Docs. – Режим доступу: <https://docs.github.com/uk/pages> (дата звернення: 02.06.2026).
14. Ольховська О. В., Черненко О. О. Методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 122 Комп'ютерні науки. – Полтава : ПУЕТ, 2024. – 67 с.
15. Бражніченко, А. О., Ольховська, О. В., Черненко, О. О., Лисенко, Д. В. (2026). ВЕБ-ДОДАТОК ДЛЯ МОНІТОРИНГУ ПРОЄКТІВ З АВТО-СПОВІЩЕННЯМИ ПРО РИЗИКИ. Таврійський науковий вісник. Серія: Технічні науки, 1(1), 29-34.  
<https://doi.org/10.32782/tnv-tech.2026.1.1.3>

16. Жуля А.П., Черненко О.О., Шустваль А.С. Програмний засіб для відстеження робочого часу з автоформуванням аналітичних звітів Вісник Кременчуцького національного університету імені Михайла Остроградського. Кременчук: КрНУ, Випуск 6/2025(155), С.165-172. DOI <https://doi.org/10.32782/1995-0519.2025.6.21>

## ДОДАТОК А

### Код програми

```
=====
Файл: index.html
=====
```

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Personal Portfolio Website</title>
  <!-- google font icons -->
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
  <!-- local css file -->
  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <!-- header section starts -->
  <header class="header" id="header">
    <a<i class="menu material-icons"> menu </i></a>
    <nav class="navbar">
      <a href="#">home</a>
      <a href="#portfolio">portfolio</a>
      <a href="#resume">resume</a>
      <a href="#about">about</a>
      <a href="#contact">contact</a>
    </nav>
  </header>
  <!-- header section ends -->

  <!-- home section starts -->
  <section class="home" id="home">
    <h1 class="hero-text">
      hello, i'm <br />
      <span>Alina Siroochenko</span>
    </h1>
    <p>Photo & Video Creator</p>
  </section>
  <!-- home section ends -->

  <!-- portfolio section starts -->
  <section class="portfolio" id="portfolio">
    <h1 class="heading">featured <span>portfolio</span></h1>

    <div class="box-container">
      <div class="box">
```

```


  <div class="image">
    
    <div class="content">
      <h2>project #1</h2>
    </div>
  </div>
</a>
</div>
<div class="box">

```

```


  <div class="image">
    
    <div class="content">
      <h2>project #2</h2>
    </div>
  </div>
</a>
</div>
<div class="box">

```

```


  <div class="image">
    
    <div class="content">
      <h2>project #3</h2>
    </div>
  </div>
</a>
</div>
<div class="box">

```

```


  <div class="image">
    
    <div class="content">
      <h2>project #4</h2>
    </div>
  </div>
</a>
</div>
<div class="box">

```

```


  <div class="image">

```

```

    
    <div class="content">
      <h2>project #5</h2>
    </div>
  </div>
</a>
</div>
<div class="box">
  <a
href="https://www.instagram.com/reel/DMahazGt_13/?utm_source=ig_web_copy_link&igsh=MzR
IODBiNWFIZA==" target="_blank">
    <div class="image">
      
      <div class="content">
        <h2>project #6</h2>
      </div>
    </div>
  </a>
</div>
</div>
</section>
<!-- portfolio section ends -->

<!-- resume section starts -->
<section class="resume" id="resume">
  <h1 class="heading">my <span>resume</span></h1>
  <div class="col-container">
    <div class="col">
      <h2>education</h2>
      <div class="box">
        <span><i class="material-icons"> calendar_today </i> Sep 2022 - Present</span>
        <h3>Bachelor in Computer Science</h3>
        <p>
          I am studying Computer Science, focusing on web development, interface design, multimedia
          content creation,
          and working with modern technologies. I am learning programming languages and have a
          good command of Java.
        </p>
        <p class="university">Poltava University of Economics and Trade (PUET)</p>
      </div>
    </div>
    <div class="col">
      <h2>experience</h2>
      <div class="box">
        <span><i class="material-icons"> calendar_today </i> Oct 2024 - Present</span>
        <h3>Content Creator / Video Editor</h3>
        <p>
          Create and edit video content for an auto service company, producing engaging multimedia
          materials
          for social media and promotional purposes. Work includes filming, video editing, and content
          planning.
        </p>
      </div>
    </div>
  </div>

```

```

    </p>
    <p class="university">AutoBy Servisas</p>
  </div>
</div>
<div class="col">
  <h2>experience</h2>
  <div class="box">
    <span><i class="material-icons"> calendar_today </i> Mar 2024 - Present</span>
    <h3>Video & Photo Editor</h3>
    <p>
      Create and edit video and photo content for Dvizh Vilnius, producing engaging multimedia
      materials
      for social media and promotional purposes. Work includes filming, photography, video
      editing.
    </p>
    <p class="university">Dvizh Vilnius</p>
  </div>
</div>
</div>
</section>
<!-- resume section ends -->

<!-- about section starts -->
<section class="about" id="about">
  <div class="box-container">
    <div class="box">
      <div class="image">
        
      </div>
    </div>
    <div class="box">
      <h1 class="heading">about <span>me</span></h1>
      <p>
        I am a creative individual passionate about visual storytelling.
        I love drawing, filming, and video editing, and I have recently started
        exploring photography and photo editing.
      </p>
      <p>
        Currently, these creative pursuits are my main focus,
        and I constantly strive to improve my skills and bring my ideas to life
        through various visual mediums.
      </p>
      <a href="files/CV Siroochenko.pdf" download class="btn">download CV</a>
    </div>
  </div>
</section>
<!-- about section ends -->

<!-- clients section starts -->
<section class="clients" id="clients">
  <h1 class="heading">client <span>testimonial</span></h1>

```

```
<div class="box-container" id="testimonialsContainer">
```

```
<!-- Приклад існуючого відгуку -->
```

```
<div class="box">
```

```
<div class="image">
```

```

```

```
</div>
```

```
<div class="content">
```

```
<p>"Great work! Highly recommended."</p>
```

```
<p>- Ethan, CEO <span>XYZ Inc</span></p>
```

```
</div>
```

```
</div>
```

```
<div class="box">
```

```
<div class="image">
```

```

```

```
</div>
```

```
<div class="content">
```

```
<p>"Professional and creative content creation."</p>
```

```
<p>- Anna, Manager <span>ABC Ltd</span></p>
```

```
</div>
```

```
</div>
```

```
<div class="box">
```

```
<div class="image">
```

```

```

```
</div>
```

```
<div class="content">
```

```
<p>"Excellent video editing skills."</p>
```

```
<p>- John, Founder <span>Startup Co</span></p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- clients section ends -->
```

```
<!-- services section starts -->
```

```
<section class="services" id="services">
```

```
<h1 class="heading">my <span>services</span></h1>
```

```
<div class="box-container">
```

```
<!-- Video Shooting -->
```

```
<div class="box">
```

```
<i class="icon material-icons">videocam</i>
```

```
<h2>Video Shooting</h2>
```

```
<p>
```

```
I use mobile videography techniques that deliver sharp, stable,  
and visually appealing footage tailored for social media and fast-paced content.
```

```
</p>
```

```
<a href="#" class="btn-2">learn more <i class="material-icons">navigate_next</i></a>
```

```
</div>
```

```
<!-- Video Editing -->
```

```
<div class="box">
```

```
<i class="icon material-icons">movie</i>
```

```
<h2>Video Editing</h2>
```

```
<p>
```

```
I provide dynamic editing with clean cuts, color correction, transitions,  
text, music sync, motion effects, and optimized vertical formats (9:16).
```

```
</p>
```

```
<a href="#" class="btn-2">learn more <i class="material-icons">navigate_next</i></a>
```

```
</div>
```

```
<!-- Photography -->
```

```
<div class="box">
```

```
<i class="icon material-icons">photo_camera</i>
```

```
<h2>Photography</h2>
```

```
<p>
```

```
I capture expressive and well-lit photos that highlight mood, style, and personality.
```

```
Every image is edited with clean, natural color correction to achieve a professional and  
polished look.
```

```
</p>
```

```
<a href="#" class="btn-2">learn more <i class="material-icons">navigate_next</i></a>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- services section ends -->
```

```
<!-- blogs section starts -->
```

```
<section class="blogs" id="blogs">
```

```
<h1 class="heading">my <span>blogs</span></h1>
```

```
<div class="box-container">
```

```
<div class="box">
```

```
<div class="image">
```

```

```

```
</div>
```

```
<div class="content">
```

```
<h2>coming soon</h2>
```

```
<p>
```

```
coming soon
```

```
</p>
```

```
<span><i class="material-icons">calendar_today </i> Jan 9, 2022</span>
```

```
<span><i class="material-icons">chat_bubble_outline </i> 5 comments</span>
```

```
</div>
```

```
</div>
```

```
<div class="box">
```

```
<div class="image">
```

```

```

```
</div>
```

```
<div class="content">
```

```

    <h2>coming soon</h2>
    <p>
      coming soon
    </p>
    <span><i class="material-icons"> calendar_today </i> Jan 9, 2022</span>
    <span><i class="material-icons"> chat_bubble_outline </i> 5 comments</span>
  </div>
</div>
<div class="box">
  <div class="image">
    
  </div>
  <div class="content">
    <h2>coming soon</h2>
    <p>
      coming soon
    </p>
    <span><i class="material-icons"> calendar_today </i> Jan 9, 2022</span>
    <span><i class="material-icons"> chat_bubble_outline </i> 5 comments</span>
  </div>
</div>
</div>
</section>
<!-- blogs section ends -->

<!-- contact section starts -->
<section class="contact" id="contact">
  <h1 class="heading">get <span>in touch</span></h1>
  <div class="col-container">
    <div class="col">
      <h2>leave a review</h2>
      <input type="text" id="name" placeholder="your name" />
      <input type="email" id="email" placeholder="your email" />
      <input type="text" id="phone" placeholder="your phone" />
      <textarea id="message" placeholder="write a message" cols="30" rows="10"></textarea>
      <button id="sendBtn" class="btn">send message</button>
    </div>
    <div class="col">
      <h2>my contact details</h2>
      <p>email</p>
      <span>alinasiroochenko@gmail.com</span>
      <br />
      <br />
      <p>phone ua</p>
      <span>+380 (970) 51 0197</span>
      <br />
      <br />
      <p>phone lt</p>
      <span>+370 (634) 83 437</span>
      <br />
      <br />

```

```

    <p>address</p>
    <span>Vilnius <br />
    Skroblų g. 25</span>
  </div>
</div>
</section>
<!-- contact section ends -->

<!-- footer section starts -->
<section class="footer" id="footer">
  <div class="socials">

    <!-- Instagram -->
    <a href="https://www.instagram.com/alina_si.prod/" target="_blank">
      <ion-icon name="logo-instagram"></ion-icon>
    </a>

    <!-- TikTok -->
    <a href="https://www.tiktok.com/@alina_si.prod?_r=1&_t=ZM-926yphPZaS8"
target="_blank">
      <ion-icon name="logo-tiktok"></ion-icon>
    </a>

    <!-- Facebook -->
    <a href="https://www.facebook.com/profile.php?id=100017327880603&locale=ru_RU"
target="_blank">
      <ion-icon name="logo-facebook"></ion-icon>
    </a>

  </div>
</section>
<!-- footer section ends -->

<!-- ionicons cdn -->
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
<!-- local js file -->
<script src="script.js"></script>
</body>

</html>

```

```

=====
Файл: styles.css
=====

```

```

* {
  font-family: 'Segoe UI';
  margin: 0;
  padding: 0;
  box-sizing: border-box;

```

```
outline: none;
border: none;
text-decoration: none;
text-transform: capitalize;
transition: 0.2s linear;
}

/* css variables */
:root {
  --white: #fff;
  --dark-gray: #ababab;
  --light-gray: #808080;
  --dark-bg: #222222;
  --light-bg: #333333;
  --yellow: #bac964;
  --box-shadow: 0 0.5rem 1rem rgba(0, 0, 0, 0.1);
}

html {
  font-size: 62.5%;
  overflow-x: hidden;
  scroll-behavior: smooth;
  scroll-padding-top: 6rem;
}

html::-webkit-scrollbar {
  width: 1rem;
}

html::-webkit-scrollbar-track {
  background: var(--white);
}

html::-webkit-scrollbar-thumb {
  background: var(--light-bg);
}

body {
  background-color: var(--dark-bg);
}

section {
  padding: 5rem 9%;
}

/* home */
.home {
  height: 100vh;
  background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('images/first.png');
  filter: grayscale(100%);
  background-size: cover;
```

```
background-position: center;
background-attachment: fixed;
```

```
display: flex;
align-items: center;
justify-content: center;
flex-direction: column;
text-align: center;
}
```

```
.home .hero-text {
font-size: 7rem;
color: var(--white);
font-weight: 200;
line-height: 1.2;
}
```

```
.home .hero-text span {
font-weight: 800;
}
```

```
.home p {
color: var(--dark-gray);
font-size: 1.5rem;
margin-top: 2rem;
text-transform: uppercase;
letter-spacing: 0.3rem;
}
```

```
/* home */
```

```
/* header */
```

```
.header {
position: fixed;
top: 0;
left: 0;
right: 0;
padding: 5rem 9%;
z-index: 100;
}
```

```
.header .navbar {
display: flex;
align-items: center;
justify-content: center;
}
```

```
.header .navbar a {
font-size: 1.5rem;
color: var(--white);
padding: 0 2rem;
```

```
letter-spacing: 0.3rem;
font-weight: 300;
text-transform: uppercase;
}

.header .menu {
font-size: 2.5rem;
color: var(--white);
cursor: pointer;
text-align: left;
display: none;
}

.header.active {
background-color: var(--white);
padding: 3rem 9%;
}

.header.active .navbar a {
color: var(--dark-bg);
}

.header.active .navbar a:hover {
color: var(--yellow);
}

.header.active .menu {
color: var(--dark-bg);
}

/* header */

/* portfolio */
/* utility class */
.heading {
color: var(--white);
font-size: 3rem;
font-weight: 200;
text-align: center;
padding-bottom: 4rem;
}

.heading span {
font-weight: 700;
}

/* utility class */

.portfolio .box-container .box .image {
height: 30rem;
overflow: hidden;
```

```
    position: relative;
  }

.portfolio .box-container .box .image img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}

.portfolio .box-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
  gap: 4rem;
}

.portfolio .box-container .box {
  background-color: var(--light-bg);
  padding: 1rem;
  box-shadow: var(--box-shadow);
}

.portfolio .box-container .box:hover img {
  transform: scale(1.2);
}

.portfolio .box-container .box .image .content {
  position: absolute;
  left: 0;
  bottom: 0;
  right: 0;
  background-color: var(--light-bg);
  padding: 1rem;
  display: flex;
  align-items: center;
  justify-content: center;
  transform: translateY(100%);
}

.portfolio .box-container .box .content h2 {
  font-size: 2rem;
  color: var(--white);
  letter-spacing: 0.1rem;
  text-transform: uppercase;
}

.portfolio .box-container .box:hover .image .content {
  transform: translateY(0);
}

/* portfolio */
/* add scrollbar */
```

```
/* fix header */

/* resume */
.resume .col .box {
  background-color: var(--light-bg);
  padding: 2.5rem;
  margin: 2rem 0;
  box-shadow: var(--box-shadow);
}

.resume .col-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
  gap: 4rem;
}

.resume .col h2 {
  font-size: 2.5rem;
  color: var(--white);
  font-weight: 200;
  padding-bottom: 2rem;
}

.resume .col .box span {
  color: var(--white);
  font-size: 1.7rem;
  display: flex;
  align-items: center;
}

.resume .col .box span i {
  padding-right: 0.5rem;
  font-size: 1.7rem;
}

.resume .col .box h3 {
  color: var(--yellow);
  font-size: 2rem;
  margin-top: 1rem;
  font-weight: 400;
}

.resume .col .box p {
  font-size: 1.7rem;
  color: var(--dark-gray);
  margin-top: 1rem;
  line-height: 1.5;
  text-transform: none;
}

.resume .col .box .university {
```

```
color: var(--light-gray);
text-transform: capitalize;
}

/* resume */

/* about */
.about .box-container .box .image {
  height: 100%;
  overflow: hidden;
}

.about .box-container .box .image img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}

.about .box-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
  gap: 4rem;
}

.about .box-container .box p {
  color: var(--dark-gray);
  margin-bottom: 2rem;
  font-size: 1.7rem;
  letter-spacing: 0.1rem;
  line-height: 1.5;
  text-transform: none;
}

/* utility class */
.btn {
  background-color: var(--yellow);
  font-size: 1.7rem;
  padding: 1rem 2rem;
  border-radius: 2rem;
  color: var(--white);
  text-transform: uppercase;
  margin-right: 1rem;
  display: inline-block;
  margin-top: 1rem;
}

/* utility class */
/* about */

/* clients */
.clients .box-container .box .image {
```

```
width: 12rem;
height: 12rem;
overflow: hidden;
margin: 0 auto;
}

.clients .box-container .box .image img {
width: 100%;
height: 100%;
object-fit: cover;
border-radius: 50%;
}

.clients .box-container .box {
text-align: center;
background-color: var(--light-bg);
padding: 2.5rem;
box-shadow: var(--box-shadow);
}

.clients .box-container {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
gap: 4rem;
}

.clients .box-container .box .content {
margin-top: 2rem;
}

.clients .box-container .box .content p:nth-child(1) {
font-size: 1.7rem;
color: var(--dark-gray);
margin-bottom: 2rem;
line-height: 1.5;
letter-spacing: 0.1rem;
text-transform: none;
}

.clients .box-container .box .content p:nth-child(2) {
font-style: italic;
font-size: 1.5rem;
color: var(--light-gray);
}

.clients .box-container .box .content p:nth-child(2) span {
color: var(--yellow);
font-style: normal;
}

/* clients */
```

```
/* services */
.services .box-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
  gap: 4rem;
}

.services .box-container .box {
  background-color: var(--light-bg);
  padding: 2.5rem;
  text-align: center;
  box-shadow: var(--box-shadow);
}

.services .box-container .box .icon {
  font-size: 4rem;
  color: var(--light-bg);
  background-color: var(--yellow);
  padding: 2.5rem;
  border-radius: 50%;
  margin-bottom: 2rem;
}

.services .box-container .box h2 {
  font-size: 2rem;
  color: var(--white);
  font-weight: 400;
  margin-bottom: 2rem;
}

.services .box-container .box p {
  font-size: 1.7rem;
  color: var(--dark-gray);
  margin-bottom: 2rem;
  line-height: 1.5;
  text-transform: none;
  letter-spacing: 0.1rem;
}

.services .box-container .box .btn-2 {
  font-size: 1.7rem;
  color: var(--yellow);
  display: flex;
  justify-content: center;
}

.services .box-container .box .btn-2:hover i {
  padding-left: 1.5rem;
}
```

```
/* services */

/* blogs */
.blogs .box-container .box .image {
  height: 30rem;
  overflow: hidden;
}

.blogs .box-container .box .image img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}

.blogs .box-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(30rem, 1fr));
  gap: 4rem;
}

.blogs .box-container .box {
  background-color: var(--light-bg);
}

.blogs .box-container .box .content {
  padding: 2.5rem;
}

.blogs .box-container .box .content h2 {
  font-size: 2rem;
  color: var(--white);
  font-weight: 400;
  margin-bottom: 2rem;
}

.blogs .box-container .box .content p {
  color: var(--dark-gray);
  margin-bottom: 2rem;
  font-size: 1.7rem;
}

.blogs .box-container .box .content span {
  font-size: 1.5rem;
  color: var(--light-gray);
  margin-right: 2rem;
}

.blogs .box-container .box .content span i {
  font-size: 1.7rem;
  vertical-align: -0.3rem;
}
```

```
.blogs .box-container .box .content span:hover {
  color: var(--yellow);
  cursor: pointer;
}

/* blogs */

/* contact */
.contact .col-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(25rem, 1fr));
  gap: 6rem;
}

.contact .col-container .col h2 {
  font-size: 2.5rem;
  color: var(--white);
  font-weight: 200;
  padding-bottom: 5rem;
}

.contact .col-container .col input,
.contact .col-container .col textarea {
  background-color: var(--light-bg);
  width: 100%;
  padding: 1rem 2rem;
  border-radius: 0.5rem;
  font-size: 1.7rem;
  color: var(--dark-gray);
  margin: 0.5rem 0;
  resize: vertical;
}

.contact .col-container .col p {
  font-size: 1.7rem;
  color: var(--yellow);
  text-transform: uppercase;
  letter-spacing: 0.1rem;
}

.contact .col-container .col span {
  font-size: 1.5rem;
  color: var(--dark-gray);
  letter-spacing: 0.1rem;
}

/* contact */

/* footer */
.footer {
```

```
border-top: 0.2rem solid var(--light-bg);
text-align: center;
}
```

```
.footer .socials ion-icon {
font-size: 2.5rem;
color: var(--yellow);
background-color: var(--light-bg);
border-radius: 50%;
padding: 1rem;
margin: 0 1rem;
cursor: pointer;
}
```

```
.footer p {
font-size: 1.7rem;
color: var(--dark-gray);
margin-top: 3rem;
}
```

```
/* footer */
```

```
/* media queries */
```

```
@media (max-width: 1200px) {
.header {
padding: 3rem 2rem;
}
}
```

```
.header.active {
padding: 2rem;
}
```

```
section {
padding: 3rem 2rem;
}
}
```

```
@media (max-width: 991px) {
html {
font-size: 60%;
}
}
```

```
@media (max-width: 768px) {
.header .menu {
display: block;
}
}
```

```
.header .navbar {
position: fixed;
top: 6rem;
}
```

```

    left: 0;
    right: 0;
    background: var(--white);
    display: flex;
    flex-flow: column;
    text-align: center;
    transform: translateX(100%);
  }

.header .navbar a {
  color: var(--dark-bg);
  margin: 0;
  font-size: 2rem;
  margin: 2rem 0;
}

header .navbar.show {
  transform: translateX(0);
}
}

@media (max-width: 450px) {
  html {
    font-size: 55%;
  }

  .home .hero-text {
    font-size: 6rem;
  }
}

/* media queries */

```

=====

Файл: script.js

=====

```

// Navbar toggle
const navbar = document.querySelector('.header .navbar');
const menuButton = document.querySelector('.header .menu');

menuButton.addEventListener('click', () => {
  navbar.classList.toggle('show');
});

// Header scroll effect
window.addEventListener('scroll', () => {
  navbar.classList.remove('show');
  const header = document.querySelector('.header');
  if (window.scrollY > 0) {
    header.classList.add('active');
  }
});

```

```

    } else {
      header.classList.remove('active');
    }
  });

// On page load
window.addEventListener('load', () => {
  const header = document.querySelector('.header');
  if (window.scrollY > 0) {
    header.classList.add('active');
  } else {
    header.classList.remove('active');
  }
});

// Testimonials form
const sendBtn = document.getElementById('sendBtn');
const nameInput = document.getElementById('name');
const messageInput = document.getElementById('message');
const testimonialsContainer = document.getElementById('testimonialsContainer');

sendBtn.addEventListener('click', (e) => {
  e.preventDefault(); // щоб кнопка не перезавантажувала сторінку

  const name = nameInput.value.trim();
  const message = messageInput.value.trim();

  if (name === "" || message === "") {
    alert('Please enter your name and message!');
    return;
  }

  // Creating a new feedback block
  const newTestimonial = document.createElement('div');
  newTestimonial.classList.add('box');
  newTestimonial.innerHTML = `
    <div class="image">
      
    </div>
    <div class="content">
      <p>"${message}"</p>
      <p>- ${name}</p>
    </div>
  `;

  // Adding feedback to the container
  testimonialsContainer.appendChild(newTestimonial);

  // Cleaning the form
  nameInput.value = "";
  messageInput.value = "";

```

};