

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «ПРОЕКТУВАННЯ ТА РОЗРОБКА
ВЕБСАЙТУ ФОТОГРАФА З ВИКОРИСТАННЯМ
БІБЛІОТЕКИ REACT»

Виконав: студент 4 курсу, групи 6.1219-2пі
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми програмна інженерія
(назва освітньої програми)

С.В. Босенко

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент, к.т.н. Лимаренко Ю.О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри електроніки, інформаційних систем та
програмного забезпечення ІННІ ім. Ю.М. Потебні ЗНУ,
доцент, к.т.н. Заяц В.І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

“ 26 ” січня 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Босенку Степану Вадимовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Проектування та розробка вебсайту фотографа з використанням бібліотеки React

керівник роботи Лимаренко Юлія Олексіївна, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Проектування та розробка вебсайту фотографа з використанням бібліотеки React.

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

презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 07.02.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.02.2023	
2.	Збір вихідних даних.	21.02.2023	
3.	Обробка методичних та теоретичних джерел.	16.03.2023	
4.	Розробка першого та другого розділу.	14.04.2023	
5.	Розробка третього розділу.	15.05.2023	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	01.06.2023	
7.	Захист кваліфікаційної роботи.	22.06.2023	

Студент _____
(підпис)

С.В. Босенко _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

Ю.О. Лимаренко _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Проектування та розробка вебсайту фотографа з використанням бібліотеки React»: 46 с., 30 рис., 11 джерел.

CSS, HTML, JAVASCRIPT, PHOTOGRAPHY, PORTFOLIO, REACT.

Об'єкт дослідження – вебсайт портфоліо фотографа.

Мета роботи: розробка вебсайту портфоліо фотографа.

Методи дослідження – методи ручного кодування вебсайтів.

У дипломному проекті викладено теоретичні відомості про види і призначення вебсайтів та приклади сучасних поширених систем. На основі даних теоретичних відомостей розроблено проект вебсайту фотографа. Було спроектовано та розроблено макет у онлайн-сервісі Figma. Використовуючи мови гіпертекстової розмітки HTML, CSS та мову програмування JavaScript був реалізований вебсайт з дотриманням усіх вимог.

SUMMARY

Bachelor's qualifying paper «Design and Development of the Photographer's Website using the React Library»: 46 pages, 30 figures, 11 references.

CSS, HTML, JAVASCRIPT, PHOTOGRAPHY, PORTFOLIO, REACT.

The object of the study is the website of the photographer's portfolio.

development of a photographer's portfolio website.

Research methods – methods of manual coding of websites.

The diploma project presents theoretical information about the types and purposes of websites and examples of modern common systems. Based on these theoretical information, the project of the photographer's website was developed. The layout was designed and developed in the Figma online service. Using the hypertext markup languages HTML, CSS, and the JavaScript programming language, a fully compliant website was implemented.

ЗМІСТ

Завдання на кваліфікаційну роботу	2
Реферат	4
Summary	5
Вступ.....	8
1 Сучасні вебсайти	10
1.1 Поняття вебсайту	10
1.2 Призначення вебсайтів	10
1.3 Класифікація вебсайтів.....	12
1.4 Структура вебсайтів.....	13
1.5 Засоби для створення вебсайту.....	15
1.5.1 Огляд способів створення вебсайтів.....	16
2 Розробка проекту вебсайту.....	24
2.1 Технічне завдання	24
2.1.1 Найменування і область застосування.....	24
2.1.2 Призначення розробки	24
2.1.3 Технічні вимоги до програмного продукту	24
2.2 Етапи створення вебсайтів.....	24
2.3 Проектування макету.....	25
2.4 Мови розробки	27
2.4.1 HTML	27
2.4.2 CSS.....	27
2.4.3 JavaScript	28
2.4.4 Бібліотека React.js	29
2.5 Програмні засоби	31
2.5.1 Проектування.....	31
2.5.2 Дизайн	32
3 Реалізація вебсайту	36

3.1 Приклади реалізації вебсайту	36
Висновки	45
Перелік посилань.....	46

ВСТУП

У наш час інформаційні технології присутні у багатьох галузях людської діяльності. Підприємства та організації використовують вебсайти, адже це сприяє збільшенню цільової аудиторії, прибутків підприємства.

Правильно підібраний та розроблений вебсайт надає багато переваг організації та їх клієнтам:

- зменшують час обслуговування;
- надають швидкий і зручний пошук за інформацією;
- надають можливість легко та швидко ознайомитися з діяльністю організації або підприємства;
- чудовий засіб реклами (надають більший обсяг охоплення аудиторії);
- надають можливість слідкувати за діяльністю та успіхами організації.

Мета дослідження – розробка вебсайту портфоліо фотографа, для цього були сформульовані наступні завдання:

- визначити поняття вебсайту та їх призначення;
- проаналізувати наявні види вебсайтів;
- описати інструменти, за допомогою яких вебсайт може бути реалізована;
- розробити проект вебсайту портфоліо фотографа;

Об'єкт дослідження – основні методи та засоби розробки вебсайтів.

Предмет дослідження – вебсайт портфоліо фотографа.

Для виконання поставлених задач використовувалися як електронні, так і друковані ресурси..

Дипломний проект складається зі вступу, трьох розділів, висновків та переліку посилань.

Перший розділ містить основні відомості про вебсайти, а саме: визначення, призначення, класифікацію та структуру. Також в розділі описуються технології, які можуть бути використані для реалізації сучасного

вебсайту.

Другий розділ містить технічне завдання, технічні вимоги до програмного продукту та макети застосунку.

Третій розділ містить приклади реалізації кожної частини застосунку – зображення розроблених сторінок. Також виконаний опис застосунку.

1 СУЧАСНІ ВЕБСАЙТИ

1.1 Поняття вебсайту

Вебсайт – це колекція вебсторінок, пов'язаних між собою гіперпосиланнями і доступних через Інтернет [1]. Він представляє собою електронну інформаційну систему, яка може містити різні типи контенту, такі як текст, зображення, відео, аудіо та інші елементи.

Вебсайти можуть мати різні цілі та функції. Деякі з них служать інформаційним цілям, надаючи вміст, новини або посібники з використання продуктів або послуг. Інші вебсайти можуть бути інтернет-магазинами, де користувачі можуть купувати товари або послуги онлайн. Сайти також можуть бути блогами, форумами, соціальними мережами або платформами для обміну інформацією та комунікації.

Для доступу до вебсайту користувач зазвичай використовує веббраузер, такий як Google Chrome, Mozilla Firefox або Safari. Користувач вводить адресу вебсайту, відому як URL (Uniform Resource Locator), в адресному рядку браузера, і браузер завантажує відповідні вебсторінки, відображаючи їх на екрані користувача.

Сьогодні вебсайти є невід'ємною частиною нашого життя і широко використовуються в різних галузях, включаючи бізнес, освіту, розваги, комунікацію та багато іншого.

1.2 Призначення вебсайтів

Вебсайти мають різноманітні призначення і використовуються для різних цілей. Ось кілька загальних призначень вебсайтів [2]:

– інформаційне призначення: багато вебсайтів створюються для надання

- інформації користувачам (це можуть бути сайти компаній з представленням послуг, новинні портали, блоги, онлайн-енциклопедії та інші джерела інформації);
- електронна комерція: вебсайти електронної комерції призначені для продажу товарів або послуг через інтернет (вони можуть бути інтернет-магазинами, платформами для продажу та покупки товарів, аукціонами тощо);
 - спілкування та соціальні зв'язки: соціальні мережі та спільноти створюються для забезпечення зв'язку, спілкування та обміну інформацією між користувачами (вони дозволяють створювати профілі, ділитися контентом, обговорювати теми та встановлювати контакти);
 - освітнє призначення: вебсайти, присвячені освіті, надають навчальний матеріал, курси, лекції та інші ресурси для навчання та саморозвитку (вони можуть бути сайтами університетів, платформами для онлайн-навчання, мовними курсами тощо);
 - розваги та медіа: багато вебсайтів створюються для розваг та споживання медійного контенту (це можуть бути сайти з музикою, фільмами, відеоіграми, новинні портали, мультимедійні платформи тощо);
 - портфоліо та особисті сайти: деякі люди створюють вебсайти для представлення своїх робіт, навичок, творчості або особистої інформації (це можуть бути вебсайти художників, дизайнерів, фотографів, письменників та інших творчих професій);
 - урядові та неурядові організації: вебсайти урядових та неурядових організацій служать для надання інформації про діяльність, послуги, програми, законодавство та інші відомості організаційного характеру.

Варто зауважити, що це лише загальні призначення, і вебсайти можуть поєднувати кілька функцій або мати унікальні цілі залежно від конкретного контексту та потреб користувачів.

1.3 Класифікація вебсайтів

Існує безліч способів класифікації вебсайтів залежно від їхнього напрямку і функціональності. Ось деякі загальні категорії вебсайтів за їхнім напрямком:

- інформаційні сайти: це найпоширеніший тип вебсайтів, які надають інформацію про певну тему або предмет (це можуть бути новинні сайти, блоги, онлайн-енциклопедії, сайти компаній з представленням послуг і багато іншого);
- електронна комерція: це вебсайти, на яких користувачі можуть купувати товари або послуги онлайн (такі сайти можуть бути інтернет-магазинами, платформами для продажу та покупки товарів, аукціонами та іншими);
- соціальні мережі: це вебсайти, які дозволяють користувачам спілкуватися, обмінюватися інформацією, створювати профілі та побудовувати соціальні мережі (найвідомішими прикладами є facebook, twitter, instagram і linkedin);
- блоги та форуми: це вебсайти, на яких користувачі можуть писати свої думки, ділитися досвідом, обговорювати теми та спілкуватися між собою (блоги можуть бути особистими, професійними, тематичними або новинними);
- освітні та навчальні сайти: це вебсайти, які надають освітній матеріал, навчальні ресурси, курси або інформацію про навчання (це можуть бути сайти університетів, онлайн-платформ навчання, мовні курси тощо);
- портфоліо та особисті сайти: це вебсайти, на яких люди представляють свої роботи, творчість, навички або особисту інформацію (вони часто використовуються художниками, дизайнерами, фотографами, письменниками та іншими професіоналами);
- медіа та розваги: це вебсайти, що надають медійний контент, такий як

новини, музика, фільми, відеоігри, телепередачі та інші форми розваг.

Це лише кілька загальних категорій вебсайтів. Часто сайти можуть поєднувати кілька напрямків і функцій, а класифікація може бути більш детальною в залежності від конкретного контексту.

1.4 Структура вебсайтів

Існує кілька типів структур вебсайтів, які використовуються для організації та взаємозв'язку сторінок і контенту. Розглянемо деякі загальні види структур вебсайтів.

Стандартна. Основна вебсторінка містить посилання на інші документи вебсайту, а документи містять посилання, відповідно, на основну вебсторінку. Це найпростіший і найпоширеніший спосіб організації вебсайту (див. рис 1.1).

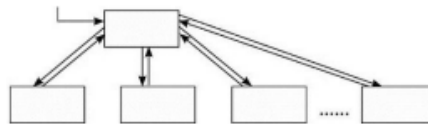


Рисунок 1.1 – Стандартна структура вебсайту

Каскад. У цьому випадку посилання в документах задані таким чином, що існує тільки один шлях обходу сторінок вебсайту. За каскадного способу організації сторінок відвідувачі сайту можуть переміщуватися тільки в одному з напрямків – вперед або назад (див. рис. 1.2).



Рисунок 1.2 – Каскадна структура сайту

Хмарочос. У цій моделі відвідувачі можуть опинитися на деяких сторінках, тільки якщо вони йдуть правильним шляхом. Це нагадує підйом до потрібної кімнати у великому хмарочосі (див. рис. 1.3).

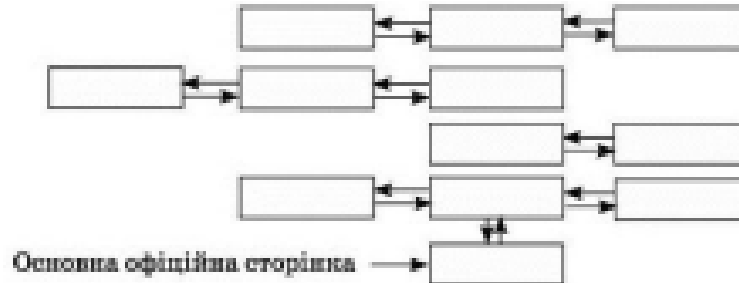


Рисунок 1.3 – Структура сайту типу «Хмарочос»

Павутина. У цьому випадку всі сторінки вебсайту містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу. Ця схема може перетворитися на лабіринт, якщо вийде з-під контролю, але вона популярна в тих випадках, коли посиланнями на документи користуються не надто часто (див. рис. 1.4).

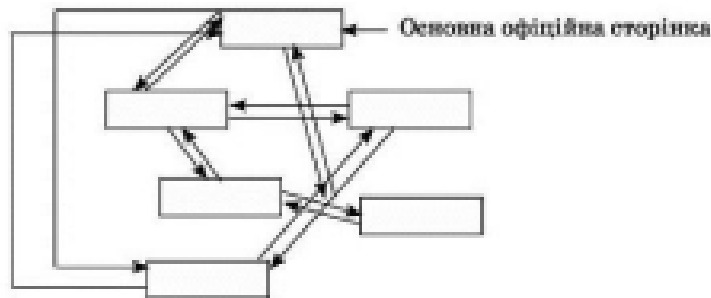


Рисунок 1.4 – Структура сайту типу «Павутина»

Ці типи структур надають різноманітні способи організації вебсайту, залежно від його мети, обсягу контенту та потреб користувачів. Конкретний вибір структури залежить від вимог проекту та зручності навігації для користувачів.

1.5 Засоби для створення вебсайту

Існує кілька способів створення вебсайтів, залежно від потреб, рівня технічних знань та доступних ресурсів. Ось декілька загальних видів створення вебсайтів [6].

Ручне кодування. Цей метод вимагає знання HTML, CSS, JavaScript та інших вебтехнологій. Розробник створює вебсайт, написавши вручну код для кожної сторінки та її функціональності. Це найбільш гнучкий метод, але вимагає високого рівня технічних навичок.

Вебплатформи для створення сайтів. Існують різні вебплатформи, такі як WordPress, Joomla, Drupal та інші, які надають зручне середовище для створення вебсайтів без необхідності програмування. Вони мають готові теми та плагіни, що спрощують процес створення сайту.

Використання веббудівельних інструментів. Існують інструменти, які дозволяють створювати вебсайти шляхом перетягування та розміщення елементів на сторінці. Ці інструменти, такі як Wix, Weebly, Squarespace, надають інтуїтивно зрозумілий інтерфейс для швидкого створення сайту.

Використання шаблонів. Багато вебсайтів пропонують готові шаблони, які можна налаштувати та використовувати для створення власного вебсайту. Це спрощує процес проектування та розробки, особливо для тих, хто не має великого досвіду в програмуванні.

Замовлення розробки. Якщо у вас немає достатньо часу, навичок або бажання створювати вебсайт самостійно, ви можете замовити розробку вебсайту у професійних веброзробників або вебстудіях. Вони створять сайт згідно з вашими вимогами та специфікаціями.

Вибір методу створення вебсайту залежить від вашої експертизи, доступних ресурсів та потреб проекту.

1.5.1 Огляд способів створення вебсайтів

Написання вебсайту власноруч вимагає базових знань веброзробки і використання різних технологій [7]. Ось загальні кроки, які можуть допомогти вам розпочати [4].

Вибір платформи: розгляньте, яку платформу ви бажаєте використовувати для створення вебсайту. Найпоширеніші платформи включають WordPress, Joomla, Drupal і самотійну розробку з використанням HTML, CSS і JavaScript.

Планування і структура: визначте мету вашого вебсайту і його структуру. Розгляньте, які сторінки ви хочете включити і як вони будуть пов'язані між собою [3].

Дизайн: створіть дизайн вашого вебсайту. Ви можете використовувати готові шаблони, створити власний дизайн або найняти професіонала для дизайну. Використовуйте HTML і CSS для створення структури і вигляду сторінок [5].

Розробка: розробка включає створення коду вебсайту. Використовуйте HTML для створення структури сторінок, CSS для оформлення і JavaScript для додавання інтерактивності і функцій.

Контент: додайте вміст до вашого вебсайту, такий як текст, зображення, відео, аудіо і інше. Створіть вміст, який відповідає меті вашого вебсайту і цікавий для ваших відвідувачів.

Оптимізація: виконайте оптимізацію вебсайту для покращення швидкості завантаження, відповідності стандартам веброзробки і покращення рейтингу в пошукових системах. Використовуйте оптимізацію SEO (пошукова оптимізація) для поліпшення видимості вашого вебсайту в пошукових системах.

Тестування: перевірте і протестируйте ваш вебсайт, щоб переконатися, що він працює належним чином на різних пристроях і веббраузерах. Виправте

будь-які помилки або проблеми, з якими зіткнулися під час тестування.

Розгортання: розмістіть ваш вебсайт на вебсервері, щоб він став доступним для користувачів через Інтернет. Зареєструйте доменне ім'я (адресу вашого вебсайту) і виберіть хостинговий план, який відповідає вашим потребам.

Вебсайт можна написати за допомогою текстових редакторів та зберегти у файлі з розширенням html. Для написання коду можна використовувати будь-який текстовий редактор, наприклад Notepad++, Visual Studio Code, SublimeText та інші.

Notepad++ – це текстовий редактор для тих користувачів, котрих бракує можливостей звичайного блокноту Windows. В Notepad++ ви можете використати підсвічування синтаксису, залежно від мови програмування обраної користувачем, згортати блоки коду згідно синтаксису мови програмування, автодоповнення слова, що набирається, підтримка великої кількості мов, динамічна зміна вікон, перетягування фрагментів коду або тексту, плагіни, макроси (див. рис. 1.5).

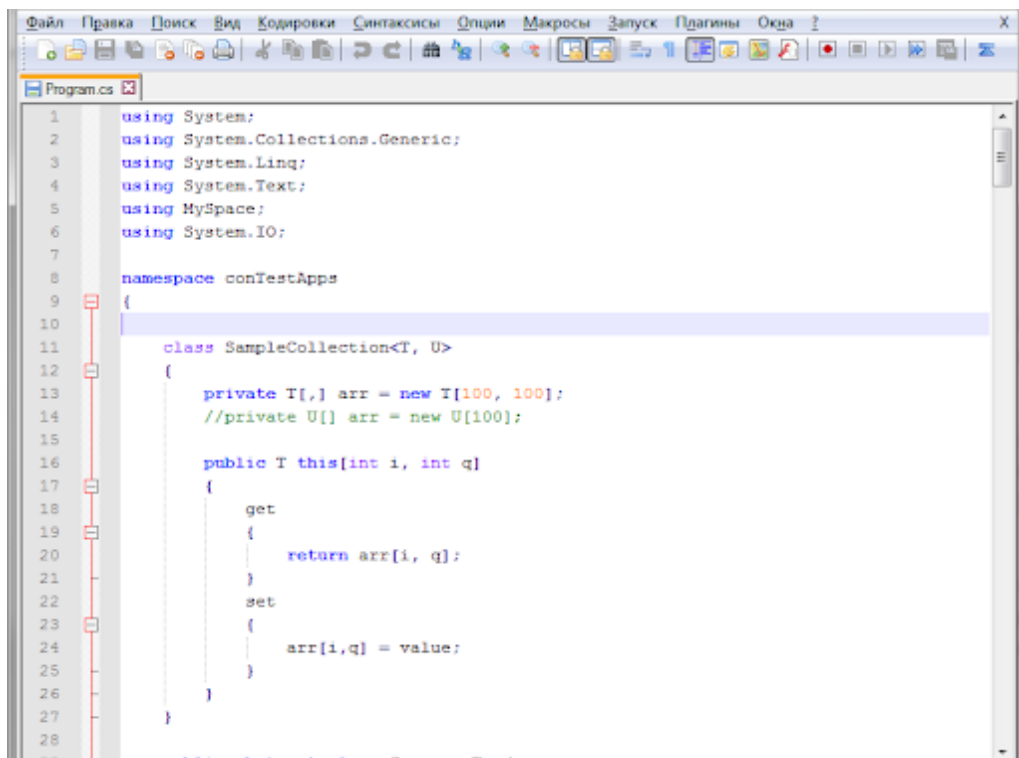


Рисунок 1.5 – Інтерфейс Notepad++

SublimeText – це кросплатформенний текстовий редактор з підтримкою плагінів, розроблених на мові програмування Python. Редактор містить візуальні теми. Одною з особливостей інтерфейсу є міні-карта усього коду у правій частині екрану, натискаючи на яку можна здійснювати навігацію. Як і у більшості текстових редакторів є автодоповнення, підсвічування синтаксису. А також є цікава функція «заготовки» (збережені фрагменти коду, ключові слова для їх запуску) (див. рис. 1.6).

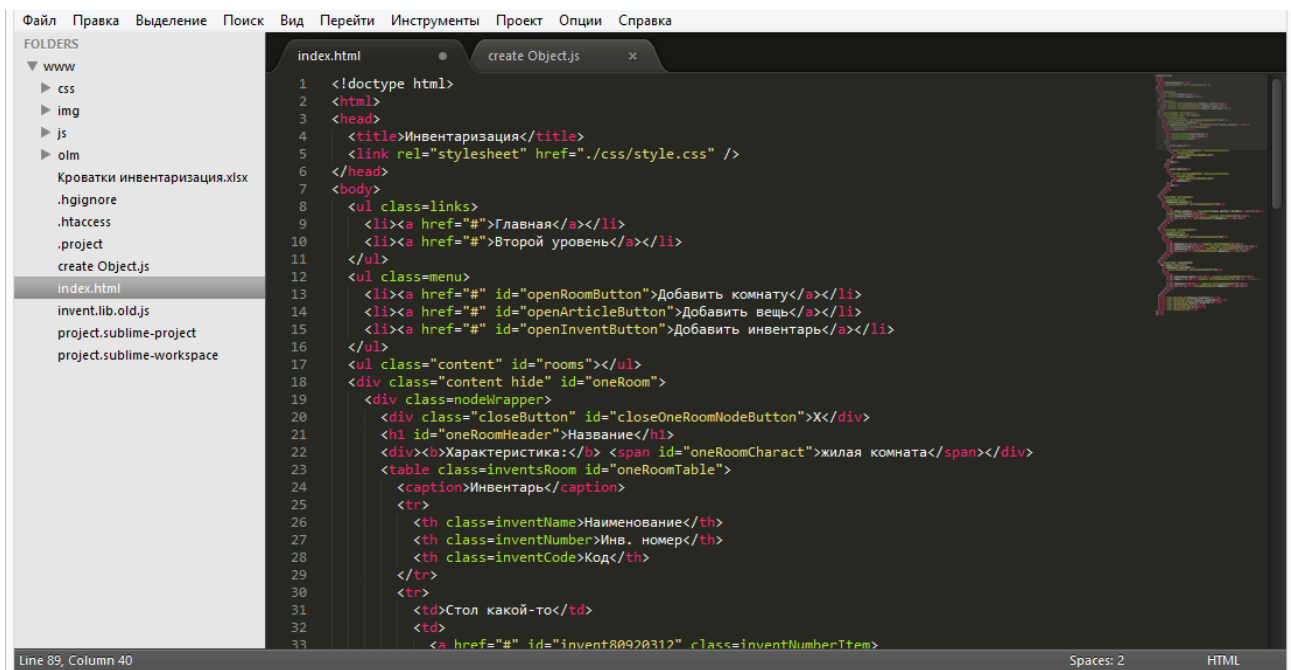


Рисунок 1.6 – Інтерфейс SublimeText

Visual Studio Code, який також зазвичай називають **VS Code** – це редактор вихідного коду, створений Microsoft із Electron Framework для Windows, Linux і macOS. Функції включають підтримку налагодження, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають функціональність (див. рис. 1.7).

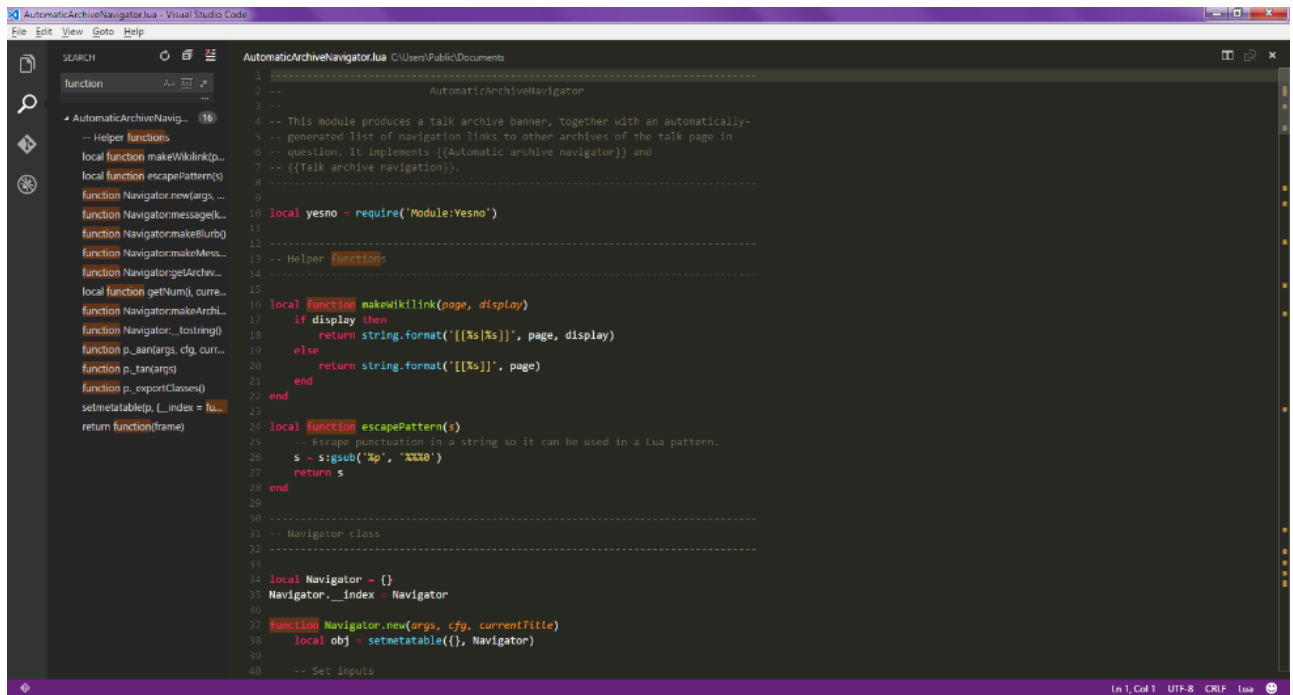


Рисунок 1.7 – Інтерфейс Visual Studio Code

Вебудівельні інструменти – це програмні засоби, які дозволяють створювати вебсайти шляхом перетягування та розміщення елементів на сторінці, без необхідності програмування або глибоких технічних знань. Вони надають інтуїтивно зрозумілий інтерфейс та спрощують процес розробки. Ось декілька важливих аспектів використання веббудівельних інструментів для розробки сайту.

Готові шаблони: веббудівельні інструменти надають велику кількість готових шаблонів, які ви можете обрати відповідно до вашого стилю та потреб. Ці шаблони зазвичай включають готовий дизайн, макет сторінок, зразки контенту та інші елементи, які можна налаштувати під власні потреби.

Драг-анд-дроп інтерфейс: одна з ключових особливостей веббудівельних інструментів – це можливість перетягувати та розміщувати елементи на сторінці за допомогою простого драг-анд-дроп інтерфейсу. Ви можете додавати тексти, зображення, кнопки, форми, слайдери та багато іншого, просто перетягуючи їх на бажане місце на сторінці.

Налаштування та редагування елементів: веббудівельні інструменти дозволяють легко налаштовувати та редагувати елементи на сторінці. Ви

можете змінювати кольори, шрифти, розміри, фони та інші властивості елементів безпосередньо у веббудівельному інтерфейсі. Це дозволяє вам швидко налаштовувати дизайн та вигляд вашого сайту.

Мобільна адаптація: багато веббудівельних інструментів надають можливість перегляду та налаштування сайту для мобільних пристроїв. Ви можете побачити, як ваш сайт виглядає на різних розмірах екранів та виконати необхідні зміни для оптимального відображення на мобільних пристроях.

Хостинг та публікація: веббудівельні інструменти часто надають можливість безпосередньо розміщувати ваш сайт в Інтернеті. Вони можуть надати хостингові послуги, реєстрацію доменного імені та інші зручності, щоб ваш сайт був доступний онлайн.

Wix: Wix є одним з найбільш популярних конструкторів вебсайтів, який пропонує широкий спектр шаблонів та функціональних можливостей. Ви можете використовувати перетягування та опускання, щоб налаштувати дизайн вашого сайту та додати різноманітні функції (див. рис. 1.8).

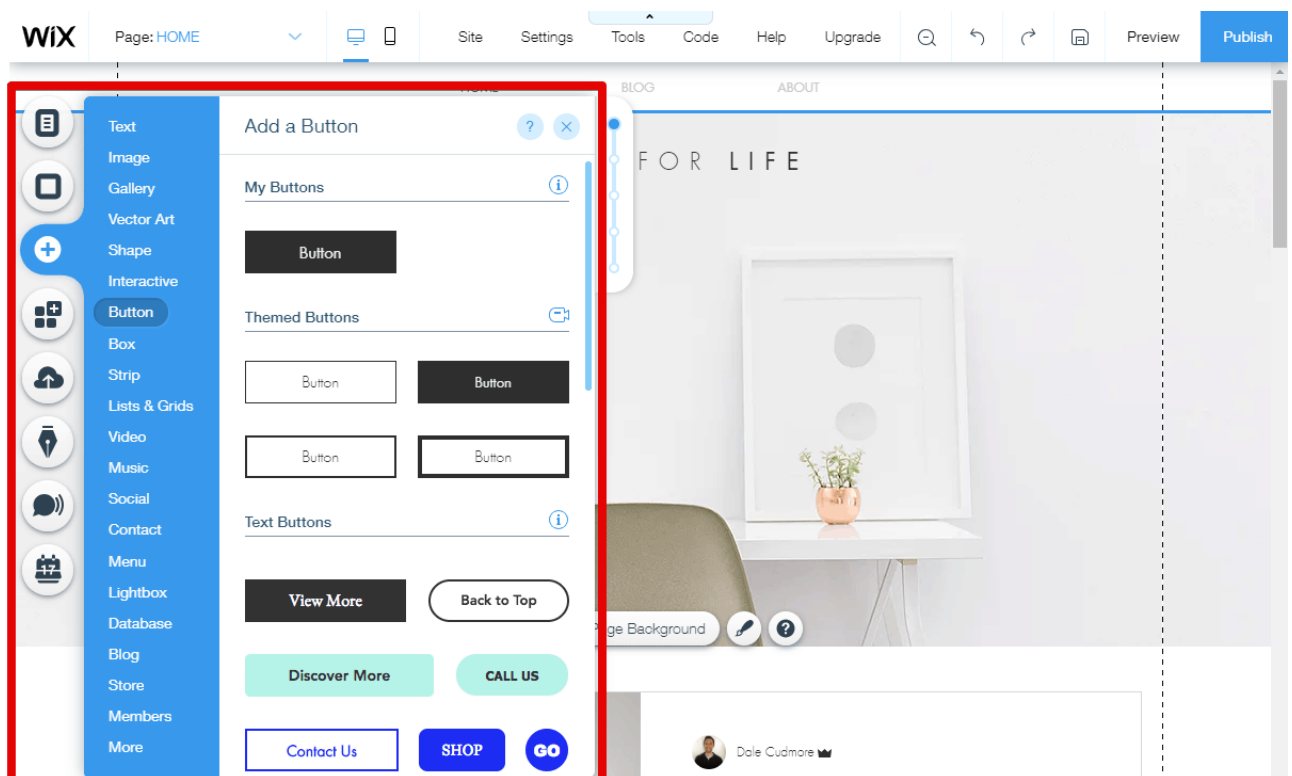


Рисунок 1.8 – Інтерфейс “Wix”

Squarespace: Squarespace є іншим популярним вебсервісом, який надає користувачам широкі можливості для створення професійних вебсайтів. Вони пропонують чисті та сучасні шаблони, а також інтуїтивно зрозумілий редактор (див. рис. 1.9).

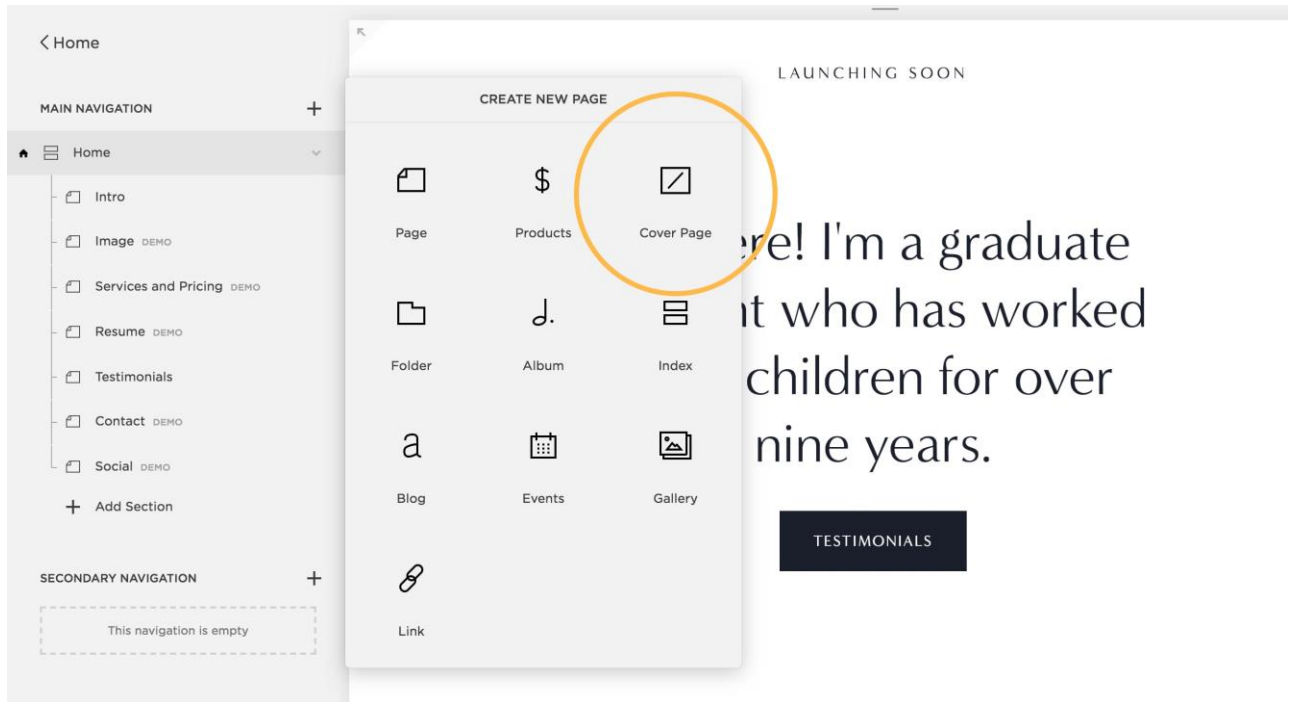


Рисунок 1.9 – Інтерфейс “Squarespace”

Веббудівельні інструменти та ручне кодування мають свої плюси і мінуси. Виконаємо порівняння.

Плюси веббудівельних інструментів:

- простота використання: веббудівельні інструменти мають інтуїтивно зрозумілий інтерфейс, що дозволяє швидко створювати вебсторінки без необхідності в програмуванні або глибоких технічних знаннях;
- швидкість розробки: завдяки готовим шаблонам та можливості перетягування та редагування елементів, веббудівельні інструменти дозволяють створювати сайти швидше, зекономивши час на проектування та розробку;
- менше технічних проблем: веббудівельні інструменти зазвичай надають вбудовану підтримку та оновлення, що дозволяє уникнути

деяких технічних проблем, таких як сумісність з різними браузерами або оптимізація для мобільних пристроїв;

- зручне управління контентом: веббудівельні інструменти часто мають вбудовані системи управління контентом (CMS), що полегшує додавання та редагування вмісту на сайті.

Мінуси веббудівельних інструментів:

- обмежена гнучкість: веббудівельні інструменти можуть бути обмеженими у термінах дизайну та функціональності (ви можете стикнутися з обмеженнями в налаштуванні певних елементів або додаванні власного коду);
- висока залежність: використання веббудівельних інструментів означає, що ви залежите від їх платформи та сервісів (якщо вони припиняють підтримку або мають технічні проблеми, це може вплинути на ваш сайт);
- складніша настройка SEO: деякі веббудівельні інструменти можуть мати обмежені можливості налаштування для оптимізації пошукової системи (SEO), що може вплинути на видимість вашого сайту у пошукових результатах.

Плюси ручного кодування:

- гнучкість та повна контроль: ручне кодування дозволяє повністю контролювати дизайн та функціональність вашого сайту (ви можете створити унікальні рішення та настроїти кожен елемент сайту під свої потреби);
- більша можливість для оптимізації: коли ви кодуєте власний сайт, ви можете докладно налаштувати його для оптимізації пошукової системи, що допоможе поліпшити видимість у пошукових результатах;
- розвиток технічних навичок: ручне кодування дає вам можливість розвивати власні технічні навички веброзробки та розуміння вебтехнологій.

Мінуси ручного кодування:

- вимагає великої технічної експертизи: ручне кодування вимагає глибоких знань HTML, CSS, JavaScript та інших вебтехнологій (це може бути складним для початківців або тих, хто не має достатньо часу для вивчення цих навичок);
- більше часу на розробку: ручне кодування може зайняти більше часу порівняно з використанням веббудівельних інструментів (ви повинні писати код для кожної сторінки та функціональності, що може затягнути процес розробки);
- складніше управління контентом: без використання системи управління контентом (CMS) ручне кодування може зробити процес оновлення та редагування вмісту менш зручним для не-технічних користувачів.

Загалом, веббудівельні інструменти підходять для швидкого розробки простих сайтів без глибоких технічних знань, тоді як ручне кодування надає повну гнучкість та контроль для складніших проектів, але вимагає більшого часового та технічного зусилля.

2 РОЗРОБКА ПРОЕКТУ ВЕБСАЙТУ

2.1 Технічне завдання

2.1.1 Найменування і область застосування

Проект, що буде розроблятися матиме назву “Сайт портфоліо фотографа”. Вебсайт може бути використаний будь-ким, якщо користувач зацікавлений у послугах фотографа.

2.1.2 Призначення розробки

Даний проект призначений для вирішення наступних завдань:

- ознайомлення звичайних користувачів з прикладами робіт фотографа;
- можливість зв'язку з фотографом для подальшого обговорення та надання послуг.

2.1.3 Технічні вимоги до програмного продукту

На початку роботи над системою були сформульовані наступні вимоги:

- зручний інтерфейс (зрозумілий та простий у використанні);
- коректна (безперебійна) робота сайту;
- сторінки з необхідною для користувача інформацією.

2.2 Етапи створення вебсайтів

Основні етапи розробки вебсайтів:

- проектування сайту (аналіз вимог, розробка технічного завдання,

- проектування інтерфейсів);
- розробка креативної концепції сайту;
- створення дизайн-концепції сайту;
- створення макетів сторінок;
- верстка сторінок і шаблонів;
- тестування.

2.3 Проектування макету

На основі технічного завдання був спроектований макет сайту за допомогою онлайн-сервісу Figma.

На рисунку 2.1 наведений макет головної сторінки портфоліо. На сторінці розташований блок з основною інформацією про фотографа та видами його праці.

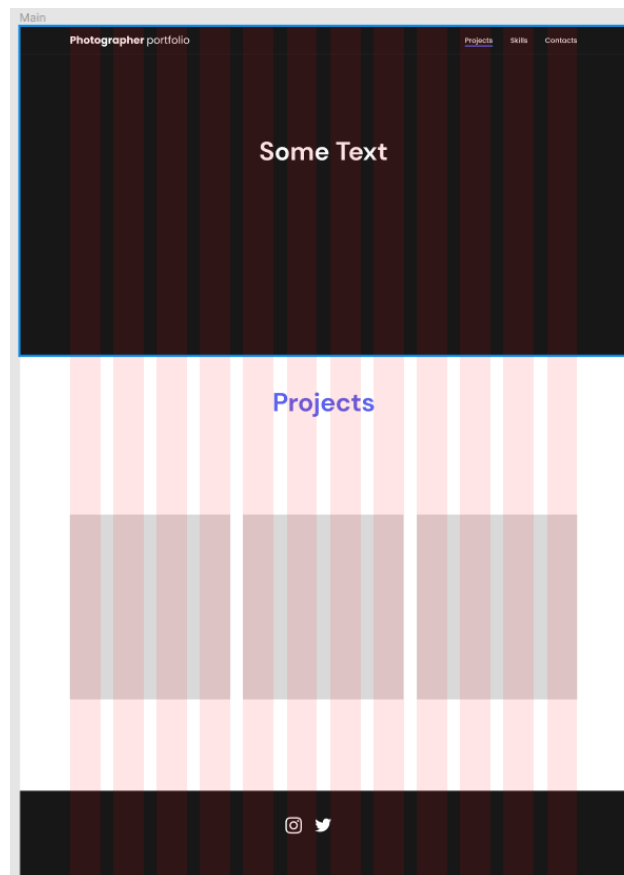


Рисунок 2.1 – Макет головної сторінки

На рисунку 2.2 наведений макет сторінки з фотографіями. Сторінка містить блоки з прикладами робіт фотографа.

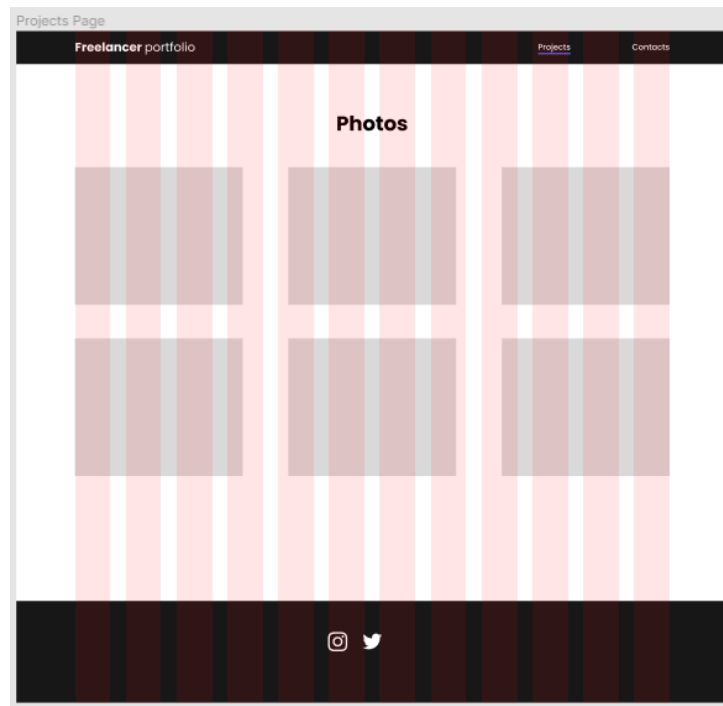


Рисунок 2.2 – Макет сторінки з прикладами фотографій

На рисунку 2.3 наведений макет сторінки з контактними даними для зворотнього зв'язку.

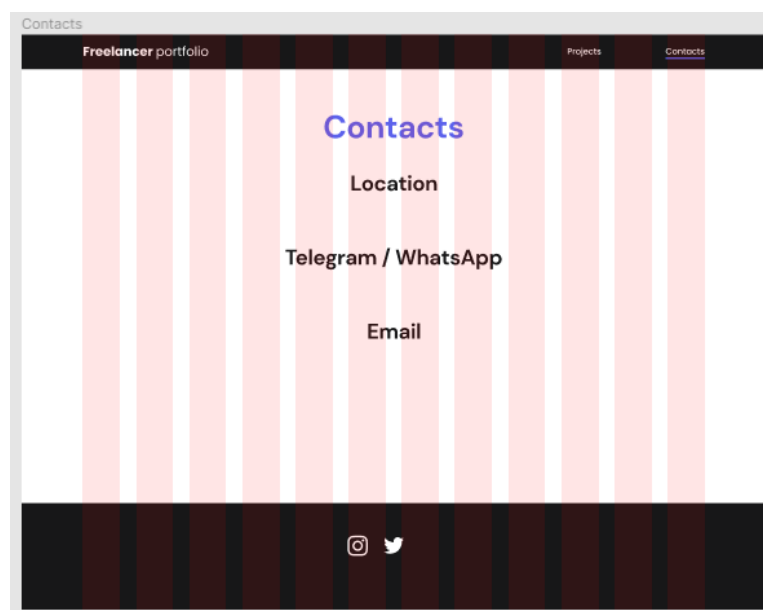


Рисунок 2.3 – Макет сторінки з контактними даними

2.4 Мови розробки

2.4.1 HTML

Для розробки каркасу сайту використовував HTML (HyperText Markup Language). Це мова розмітки гіпертекстових сторінок яка описує структуру вебсторінки і включені сигнали для зовнішнього вигляду документу. Браузери отримують HTML-документ з локальної пам'яті або вебсервера. Елементи є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку [8]. HTML надає засоби для створення структурованих документів, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи (див. рис. 2.4).

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Кваст</title>
6   </head>
7   <body>
8     <div class="page-header">
9       <div class="header-top">
10        <div class="logo">
11          
12        </div>
13        <ul>
14          <li><a href="#">История</a> </li>
15          <li> <a href="#">Производство</a> </li>
16          <li> <a href="#">Ассортимент</a> </li>
17          <li> <a href="#">Где купить</a> </li>
18        </ul>
19      </div>
20      <br>
21      <p>Крафтовый квас<br>
22      всему голова!</p>
23    </div>
24  </body>
25 </html>
```

Рисунок 2.4 – Приклад коду на HTML

2.4.2 CSS

Для стилізації сайту використовував CSS (Cascade Style Sheets). Це мова стилів сторінок, яка використовується для опису їх зовнішнього вигляду, самі

сторінки написані за допомогою HTML. CSS має порівняно простий синтаксис і використовує небагато англійських слів для найменування різних складових стилю [9]. Перевагами CSS є прискорення завантаження сторінок і зменшення обсягів інформації, навантаження на сервер та канал передачі (досягається за рахунок кешування (запам'ятовування) інформації про стилі в браузері і використання їх для всіх сторінок, а не завантажувати для кожної); інформація про стиль усього сайту або його частин, може міститися в одному файлі (.css), що дозволяє швидко редагувати сторінки (див. рис. 2.5).

```
.gallery {  
  position: relative;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  flex-wrap: wrap;  
  max-width: 900px;  
  width: 100%;  
  margin-left: auto;  
  margin-right: auto;  
}  
  
.gallery__card {  
  display: flex;  
  width: 300px;  
}  
  
.gallery__img {  
  max-width: 100%;  
  max-height: 100%;  
}  
  
.gallery__card:hover .gallery__img {  
  transform: scale(1.01);  
  cursor: zoom-in;  
}
```

Рисунок 2.5 – Приклад коду на CSS

2.4.3 JavaScript

Для додання сайту інтерактивності використовував JavaScript (JS). Це об'єктно-орієнтовна мова програмування, використовується для створення сценарію вебсторінок, надання можливості клієнту взаємодіяти з користувачем, обмінюватися даними з сервером, змінювати структуру та вигляд сторінки [10]. Щодо переваг, користувачі цієї мови мають доступ до великої кількості

бібліотек з відкритим кодом, легко вивчається новачками, рідко змінюється, універсальність та популярність. Недоліками є проблема роботи із типами даних (проявляється у тому, що усі помилки можна побачити лише після компіляції та запуску програми); не реалізовані принципи о'єктно-орієнтованого програмування; можливості мови досить обмежені, для їх розширення необхідно підключати або створювати сторонні бібліотеки та фреймворки (див. рис. 2.6).

```

1  import { useEffect, useRef } from "react";
2  import sun from "./sun.svg";
3  import moon from "./moon.svg";
4  import "./style.css";
5  import { useLocalStorage } from "../../utils/useLocalStorage";
6  import detectDarkMode from "../../utils/detectDarkMode";
7
8  const BtnDarkMode = () => {
9    const [darkMode, setDarkMode] = useLocalStorage("darkMode", detectDarkMode());
10   const btnRef = useRef(null);
11
12   useEffect(() => {
13     if (darkMode === "dark") {
14       document.body.classList.add("dark");
15       btnRef.current.classList.add("dark-mode-btn--active");
16     } else {
17       document.body.classList.remove("dark");
18       btnRef.current.classList.remove("dark-mode-btn--active");
19     }
20   }, [darkMode]);
21
22   const toggleDarkMode = () => {
23     console.log("toggleDarkMode Fired!");
24     setDarkMode((currentValue) => {
25       return currentValue === "light" ? "dark" : "light";
26     });
27   };
28
29   return (
30     <button ref={btnRef} className="dark-mode-btn" onClick={toggleDarkMode}>
31       <img src={sun} alt="Light mode" className="dark-mode-btn__icon" />
32       <img src={moon} alt="Dark mode" className="dark-mode-btn__icon" />
33     </button>
34   );
35 };
36
37 export default BtnDarkMode;
38

```

Рисунок 2.6 – Приклад коду JavaScript

2.4.4 Бібліотека React.js

React – це відкрита JavaScript бібліотека для розробки інтерфейсів користувача. Вона дозволяє створювати компоненти і будувати інтерактивні

вебдодатки. Основною ідеєю React є розбиття вебінтерфейсу на невеликі, повторно використовувані компоненти, які можуть змінюватися в залежності від стану додатку [11].

Розглянемо основні принципи і можливості React.

Virtual DOM: React використовує віртуальний DOM (Document Object Model) для оптимізації швидкості та ефективності оновлення вебсторінки. Він створює віртуальне представлення реального DOM та оновлює його тільки при зміні стану, що зменшує навантаження на браузер та покращує продуктивність додатків.

Компонентний підхід: React дозволяє розбити інтерфейс на незалежні компоненти. Кожен компонент має свою внутрішню логіку, стиль та стан. Це дозволяє розробникам зосередитися на конкретному компоненті без необхідності втручатися в решту додатку.

Стан компонентів: React дозволяє керувати станом компонентів, що дозволяє динамічно змінювати вміст і зовнішній вигляд. Зміни стану викликають перерендеринг компонента та оновлення відповідних частин інтерфейсу.

JSX синтаксис: React використовує JSX – розширений синтаксис JavaScript, який дозволяє писати HTML-подібний код безпосередньо в JavaScript файлів. Це робить код більш зрозумілим та легким для розробки та збереження.

Реактивність:** React дозволяє стежити за змінами стану компонентів та автоматично оновлювати відображення при зміні даних. Це спрощує роботу з динамічними даними та покращує взаємодію з користувачем.

Розширені можливості: разом з React часто використовуються додаткові бібліотеки, такі як React Router для маршрутизації, Redux для керування станом додатку, або Material-UI для готових компонентів та дизайну. Це дозволяє розширити можливості React та спростити розробку.

Загалом, React є потужним інструментом для розробки вебдодатків, який дозволяє побудувати модульний, швидкий та ефективний інтерфейс

користувача. Він широко використовується в індустрії та має активну спільноту розробників, що робить його популярним вибором для багатьох проектів.

2.5 Програмні засоби

2.5.1 Проектування

Для розробки інтерфейсів я використовував векторний онлайн-сервіс Figma. Даний редактор підходить для створення як простих дизайн-систем, так і складних проектів (наприклад мобільні додатки). Перевагою цього сервісу є її хмарність; зручний та легкий у використанні інтерфейс; широкі можливості для інтеграції з корпоративним месенджером Slack і інструмент для високорівневого прототипування Framer. Щодо недоліків, неможливо працювати без інтернет з'єднання; не підтримує варіативні шрифти; не завжди коректно імпортує файли Sketch та не підтримує кольоровий простір A3 (див. рис. 2.7).

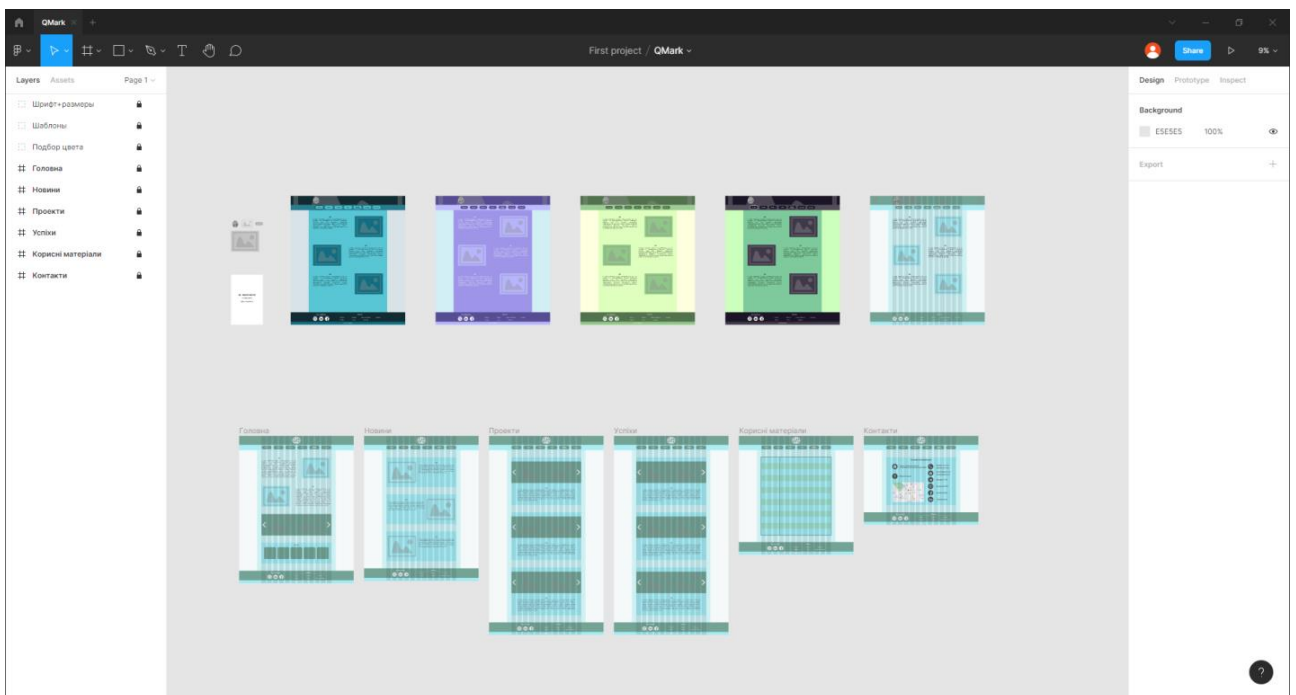


Рисунок 2.7 – Інтерфейс Figma

2.5.2 Дизайн

Для вирішення питання з підбором кольору для вебсайту, використав додаток “colorhunt.co”, на якому люди складають свої палітри кольорів та пропонують їх іншим користувачам (див. рис. 2.8).

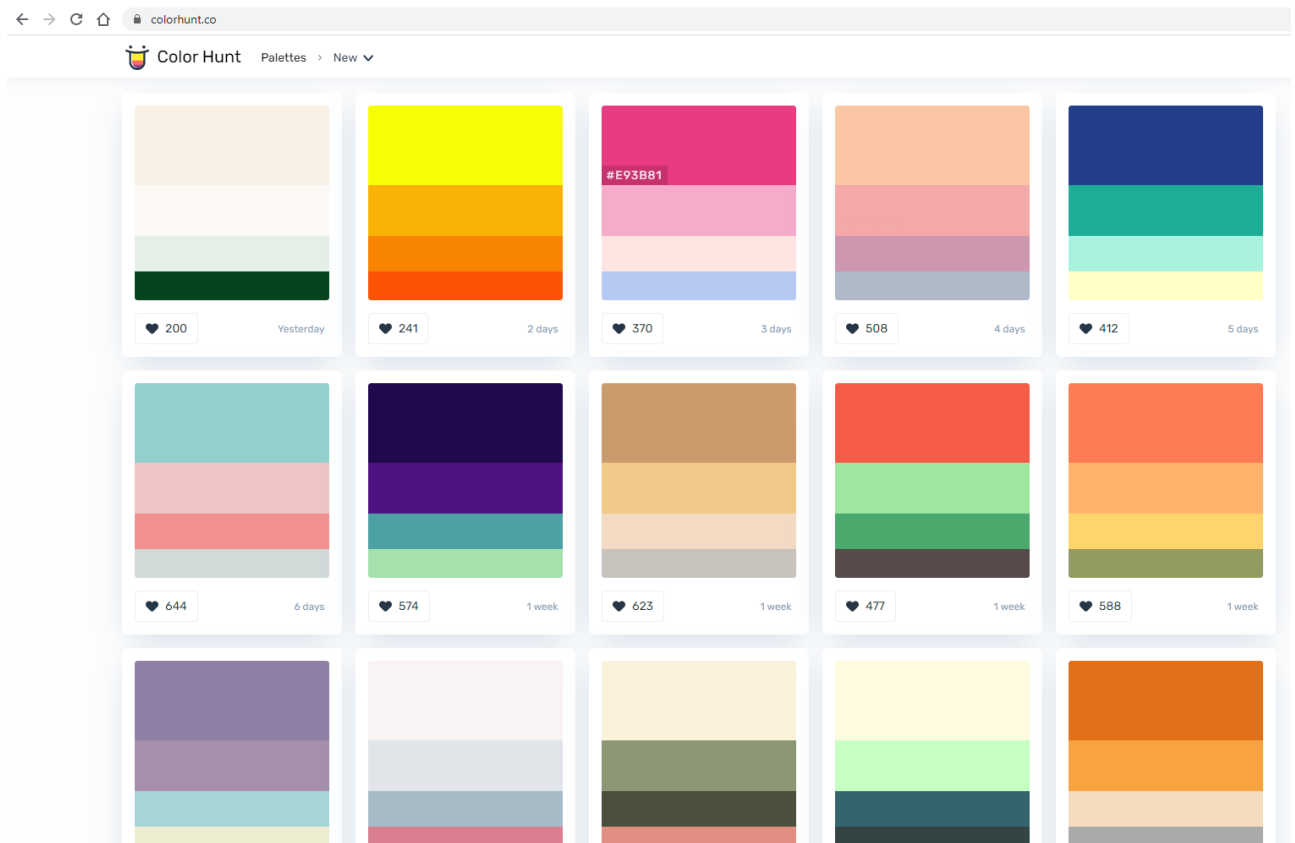


Рисунок 2.8 – Інтерфейс сайту “colorhunt.co”

Adobe Photoshop – це професійний графічний редактор, який розробляється компанією Adobe Systems. Він є одним з найпопулярніших програмних засобів для редагування та обробки зображень і використовується в різних сферах.

Розглянемо основні можливості та застосування Adobe Photoshop.

Ретушування та виправлення зображень: Photoshop надає широкі можливості для виправлення помилок, дефектів та недоліків на фотографіях. Ви можете видалити плями, змінити кольори, виправити червоні очі, видалити

нежадані об'єкти або особи з фотографії.

Створення та редагування графічних елементів: Photoshop дозволяє створювати нові графічні елементи з нуля або редагувати існуючі. Ви можете малювати, малювати вектори, застосовувати текстури, ефекти та шари, щоб створити професійні графічні дизайни, логотипи, банери та ілюстрації.

Фотомонтаж та складання зображень: Photoshop дозволяє об'єднувати різні елементи зображень, щоб створити складні композиції. Ви можете об'єднувати фотографії, створювати колажі, застосовувати спеціальні ефекти, змінювати фони та створювати фантастичні маніпуляції зображеннями.

Дизайн вебсайтів та інтерфейсів: Photoshop використовується для створення макетів та дизайну вебсайтів, мобільних додатків та інтерфейсів. Ви можете створювати візуальні компоненти, розташування елементів, стилізацію кнопок та інші елементи дизайну, щоб побудувати привабливі та ефективні інтерфейси.

Графічний дизайн та реклама: Photoshop є основним інструментом для графічного дизайну та створення рекламних матеріалів. Ви можете створювати банери, постери, рекламні афіші, проспекти, упаковку продуктів та інші графічні елементи для друку або використання в онлайн-рекламі.

Фотографічний мистецтво та ретушування: Photoshop широко використовується в фотографічній індустрії для ретушування фотографій та створення художніх ефектів. Ви можете додавати фільтри, регулювати кольори, створювати ефекти освітлення, відтінку та контрасту, щоб покращити зображення та надати їм унікального стилю.

3D-графіка та анімація: Photoshop має деякі можливості для роботи з 3D-графікою. Ви можете створювати та редагувати 3D-моделі, створювати текстури, освітлення та матеріали. Також, Photoshop має інтеграцію з іншими програмами Adobe, такими як Adobe After Effects, для створення анімації та спеціальних ефектів.

Adobe Photoshop є потужним та розширюваним інструментом, який забезпечує широкий спектр можливостей для редагування та обробки

зображень у різних сферах, включаючи дизайн, фотографію, веброзробку та рекламу. Він є незамінним інструментом для професіоналів та ентузіастів, які працюють з графічними зображеннями (див. рис. 2.9).

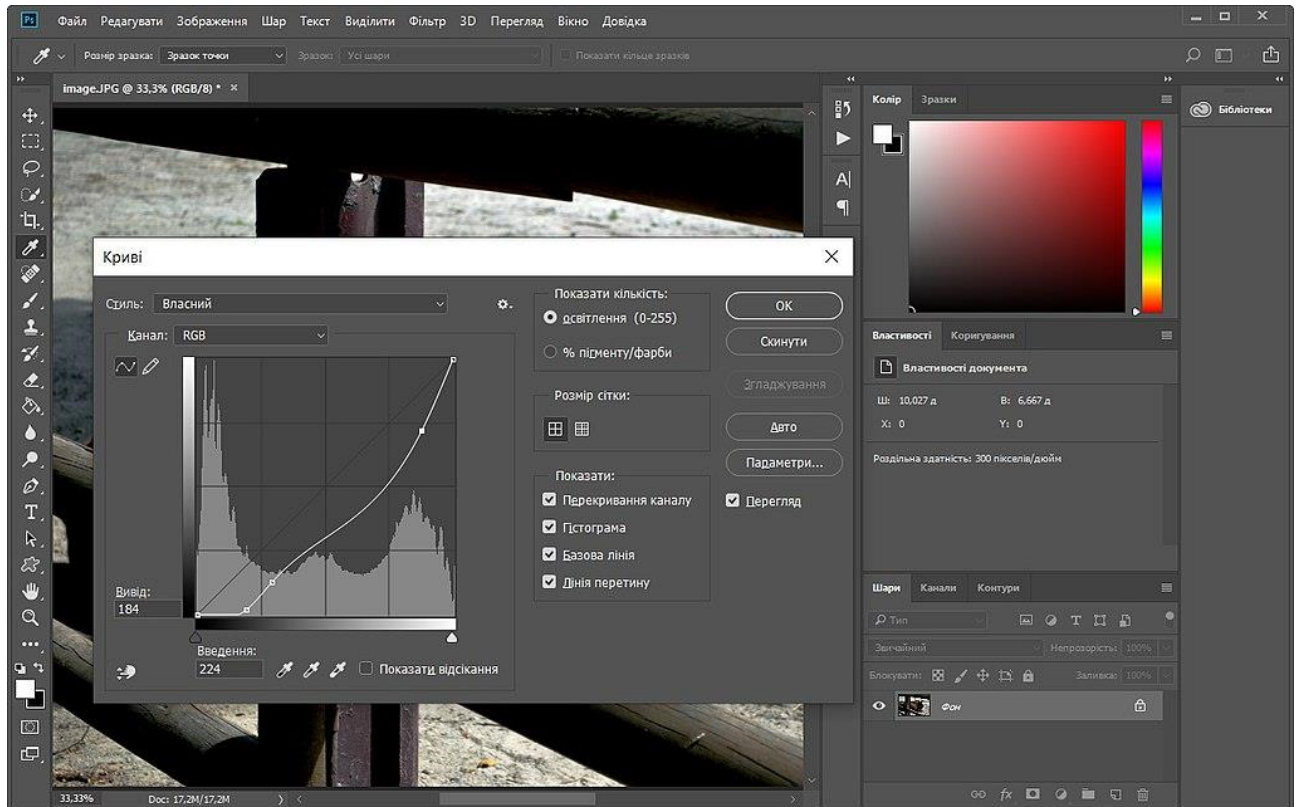


Рисунок 2.9 – Інтерфейс Adobe Photoshop

Pexels.com – це онлайн-платформа, яка надає безкоштовні фотографії та відео високої якості для використання у комерційних та особистих проектах. Вона пропонує велику колекцію безкоштовних зображень, які можуть бути використані без обмежень у різних сферах діяльності.

Розглянемо основні особливості сайту Pexels.com.

Безкоштовні фотографії та відео: Pexels.com надає доступ до великого архіву безкоштовних фотографій та відео, які можна використовувати безкоштовно у комерційних та особистих проектах. Це робить його ідеальним ресурсом для вебдизайнерів, маркетологів, блогерів та інших професіоналів.

Розмаїття категорій та тем: сайт пропонує широкий вибір категорій, що охоплюють різні теми, включаючи природу, людей, їжу, технології, міста,

спорт та багато іншого. Користувачі можуть легко знайти відповідні зображення для своїх потреб.

Висока якість зображень: всі фотографії та відео, доступні на Pexels.com, високої якості та професійно виглядають. Це дозволяє користувачам отримувати візуально привабливий контент для своїх проєктів.

Простий пошук та навігація: сайт має зручний інтерфейс зі зручною системою пошуку та навігації. Користувачі можуть швидко знайти потрібні зображення, використовуючи ключові слова, фільтри, популярні категорії та рекомендації.

Pexels.com є корисним та зручним ресурсом для тих, хто шукає візуальний контент для своїх проєктів (див. рис. 2.10).

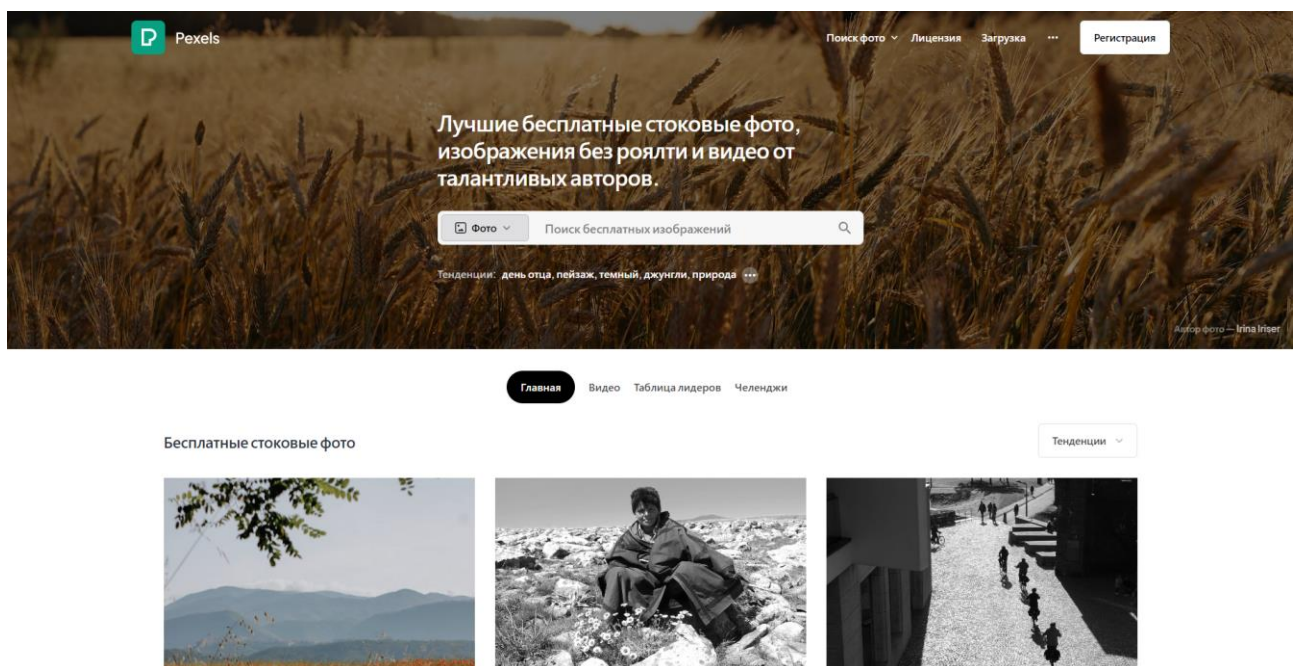


Рисунок 2.10 – Интерфейс сайта “pexels.com”

3 РЕАЛІЗАЦІЯ ВЕБСАЙТУ

3.1 Приклади реалізації вебсайту

Приклад головної сторінки, яка складається з меню навігації, контент зони та підвалу (див. рис. 3.1).

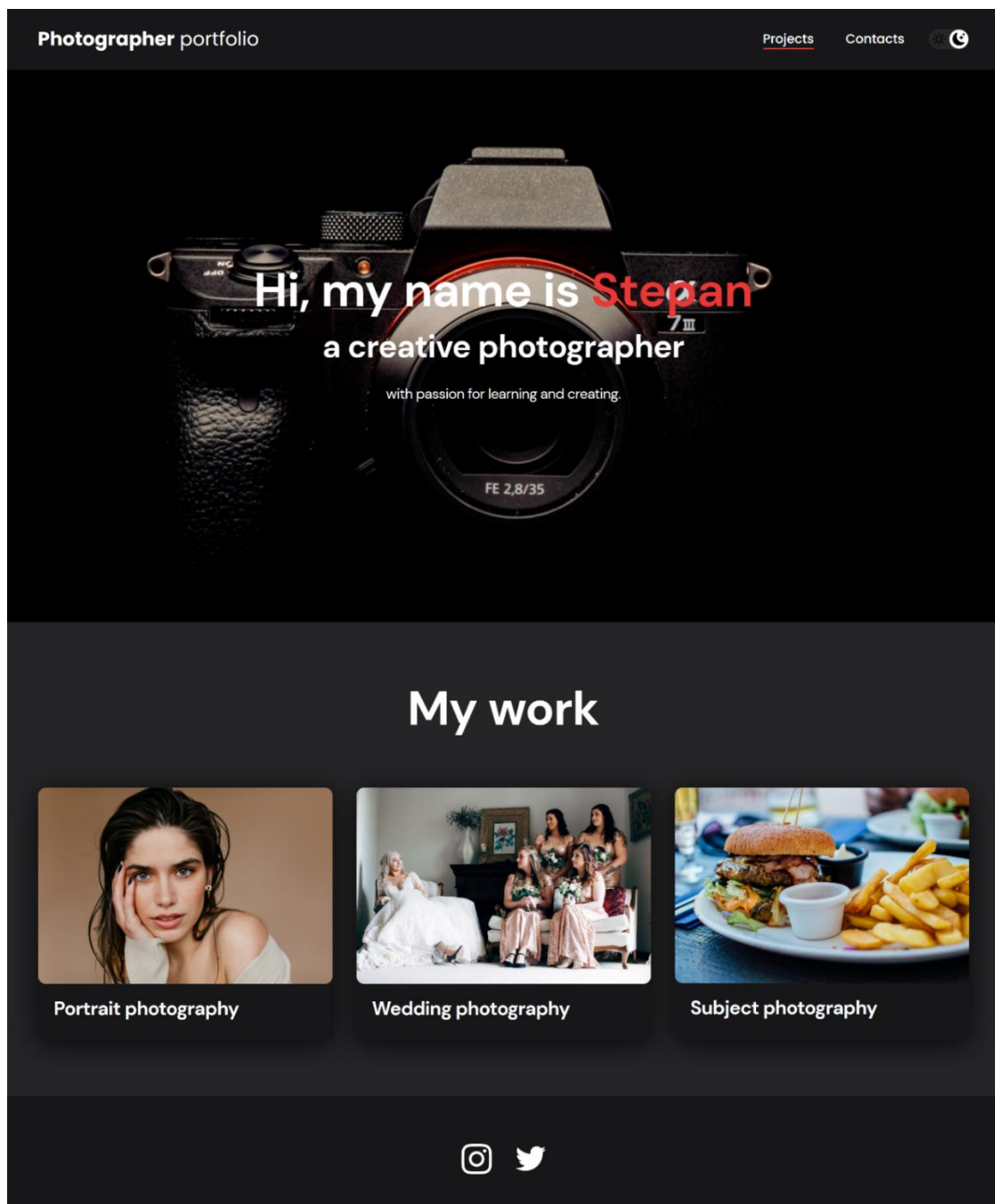


Рисунок 3.1 – Приклад сторінки “Галерея художників”

Приклад коду сторінки наведено на рисунку 3.2.

```
import Header from "../././components/header/Header";

import port from "../././img/projects/1.jpg";
import wedd from "../././img/projects/8.jpg";
import subj from "../././img/projects/15.jpg";

const Home = () => {
  return (
    <>
    <Header />
    <main className="section">
      <div className="container">
        <h2 className="title-1">My work</h2>
        <ul className="projects">
          <li className="project">
            <a href="/Project_p">
              <img src={port} alt="Project img" className="project__img" />
              <h3 className="project__title">Portrait photography</h3>
            </a>
          </li>
          <li className="project">
            <a href="/project_w">
              <img src={wedd} alt="Project img" className="project__img" />
              <h3 className="project__title">Wedding photography</h3>
            </a>
          </li>
          <li className="project">
            <a href="/Project_s">
              <img src={subj} alt="Project img" className="project__img" />
              <h3 className="project__title">Subject photography</h3>
            </a>
          </li>
        </ul>
      </div>
    </main>
  </>
);
};

export default Home;
```

Рисунок 3.2 – Приклад коду сторінки “Галерея художників”

Приклад сторінки з прикладами робіт фотографа. Відрізняється контент зоною з прикладами фотографій (див. рис. 3.3).

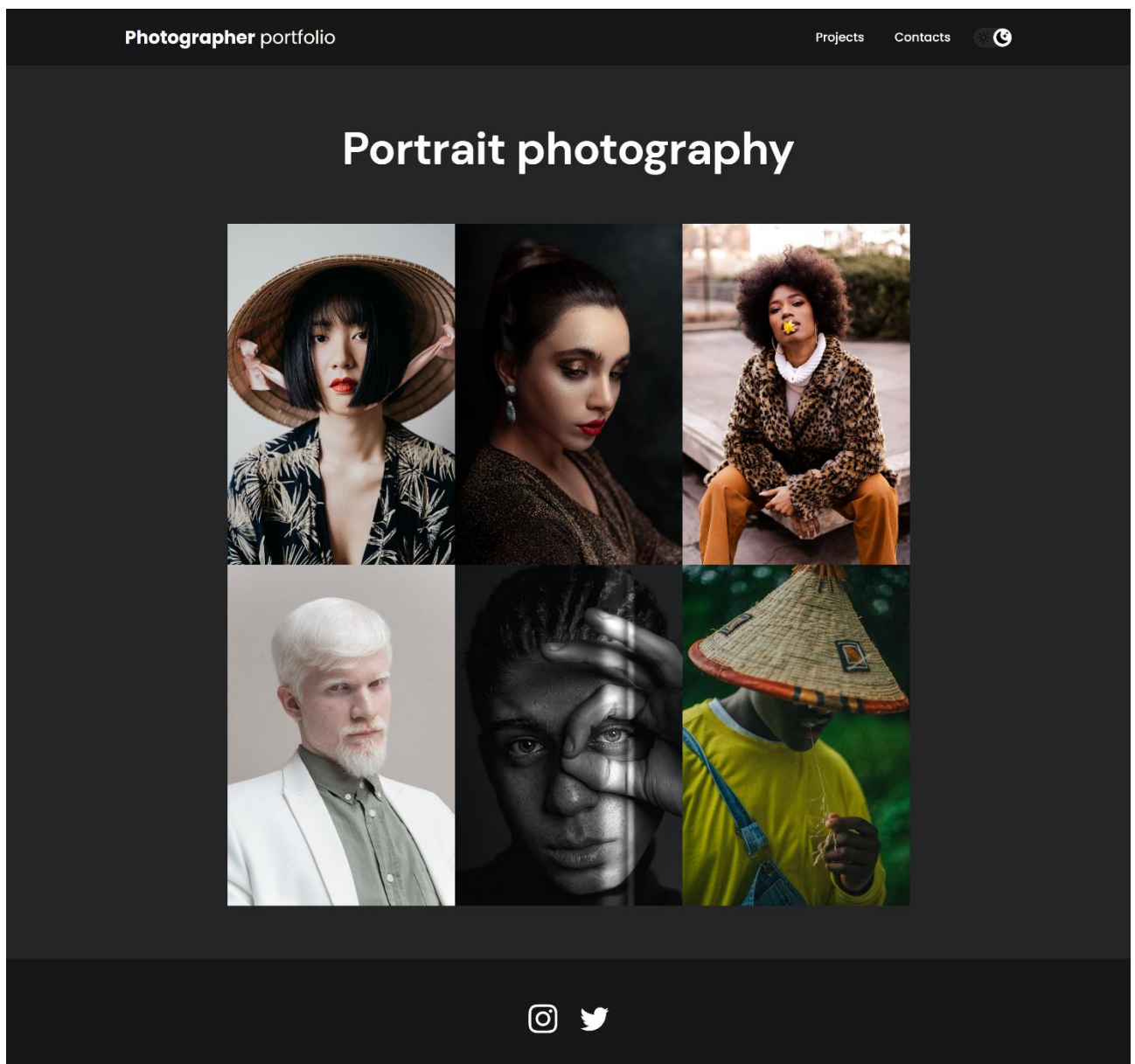


Рисунок 3.3 – Приклад сторінки з прикладами робіт фотографа

Приклад коду сторінки наведено на рисунку 3.4.

```
import portrait01 from "../../img/projects/2.jpg";
import portrait02 from "../../img/projects/3.jpg";
import portrait03 from "../../img/projects/4.jpg";
import portrait04 from "../../img/projects/5.jpg";
import portrait05 from "../../img/projects/6.jpg";
```

```

import portrait06 from "../../img/projects/7.jpg";

import "../../utils/slider/slider.js";

const Project_p = () => {
  return (
    <main className="section">
      <h1 className="title-1">Portrait photography</h1>
      <div className="container">
        <div className="gallery">
          <div className="gallery__card">
            <img className="gallery__img" src={portrait01} alt="pj" />
          </div>
          <div className="gallery__card">
            <img className="gallery__img" src={portrait02} alt="pj" />
          </div>
          <div className="gallery__card">
            <img className="gallery__img" src={portrait03} alt="pj" />
          </div>
          <div className="gallery__card">
            <img className="gallery__img" src={portrait04} alt="pj" />
          </div>
          <div className="gallery__card">
            <img className="gallery__img" src={portrait05} alt="pj" />
          </div>
          <div className="gallery__card">
            <img className="gallery__img" src={portrait06} alt="pj" />
          </div>
        </div>

        <div className="slider hidden">
          <div className="slider__container">
            <button className="slider__btn slider__btn-left">PREV</button>
            <button className="slider__btn slider__btn-right">NEXT</button>
          </div>
          <button className="slider__btn-close">CLOSE</button>
        </div>
      </div>
    </main>
  );
};

export default Project_p;

```

Рисунок 3.4 – Приклад коду сторінки

На сторінці з прикладами робіт фотографа, за допомогою мови JavaScript реалізовано галерею фотографій зі слайдером (див. рис. 3.5, 3.6).

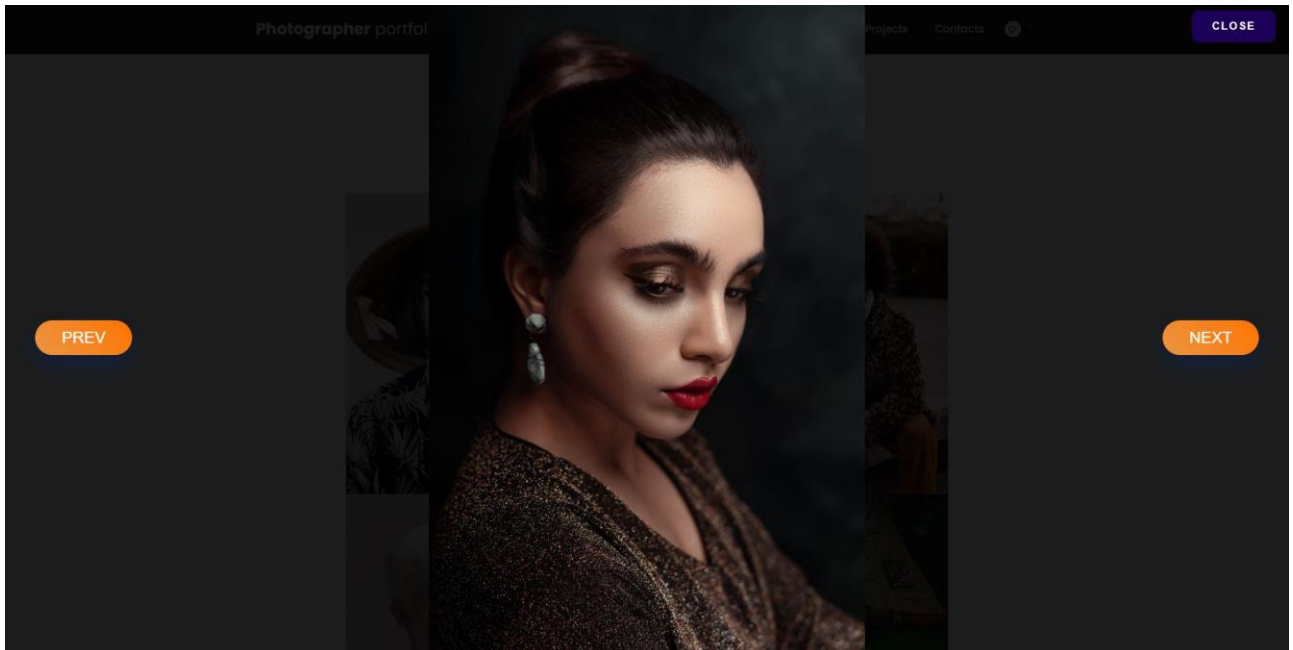


Рисунок 3.5 – Реалізація слайдера

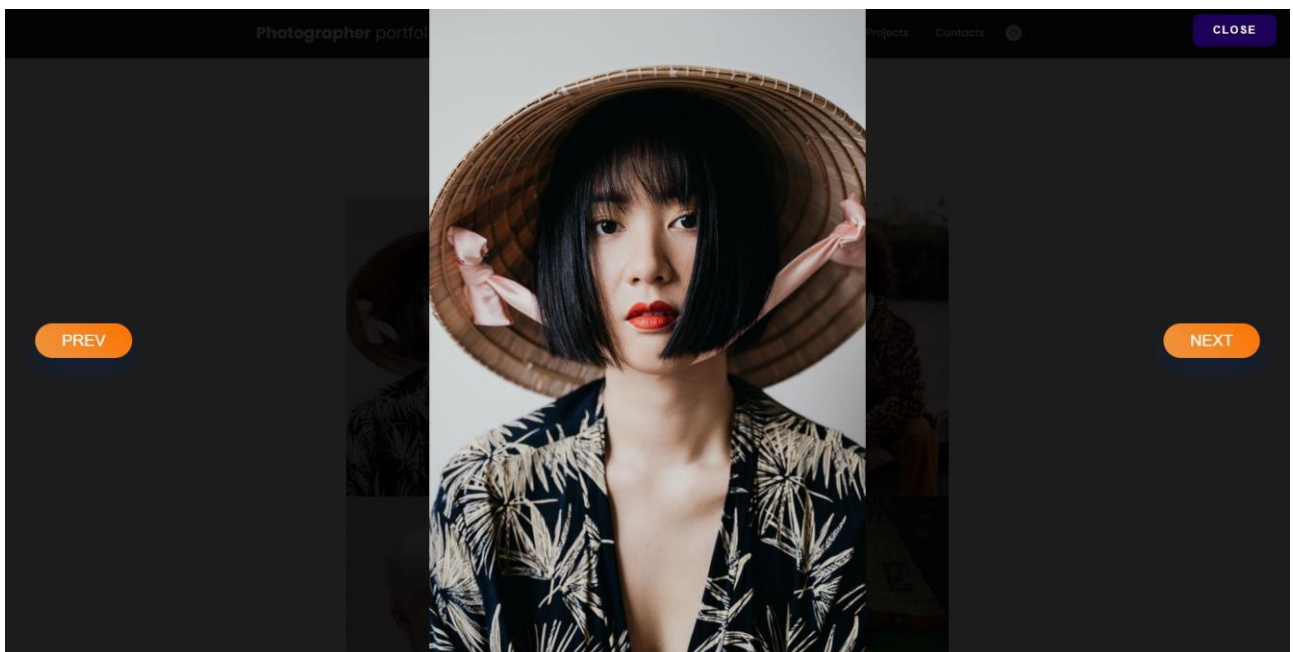


Рисунок 3.6 – Реалізація слайдера

Приклад коду реалізації слайдера наведено на рисунку 3.7.


```

import "./style.css";

const cards = document.querySelectorAll(".gallery__card"),
  pictures = document.querySelectorAll(".gallery__img"),
  slider = document.querySelector(".slider"),
  sliderContainer = document.querySelector(".slider__container"),
  sliderBtnLeft = document.querySelector(".slider__btn-left"),
  sliderBtnRight = document.querySelector(".slider__btn-right"),
  sliderClose = document.querySelector(".slider__btn-close");

let cardIndex = 0,
  pictureFull = null,
  newPictureFull = null;

sliderBtnLeft.addEventListener("click", () => changePicture("left"));
sliderBtnRight.addEventListener("click", () => changePicture("right"));
sliderClose.addEventListener("click", () => closeSlider());

cards.forEach((item, index) => {
  item.addEventListener("click", () => {
    cardIndex = index;
    showPicture();
  });
});

const showPicture = () => {
  pictureFull = pictures[cardIndex].cloneNode();
  pictureFull.style.width = "100%";
  sliderContainer.append(pictureFull);
  slider.classList.remove("hidden");
};

const changePicture = (dir) => {
  if (dir === "left") {
    cardIndex > 0 ? cardIndex-- : (cardIndex = cards.length - 1);
  } else if (dir === "right") {
    cardIndex < cards.length - 1 ? cardIndex++ : (cardIndex = 0);
  } else {
    return;
  }

  newPictureFull = pictures[cardIndex].cloneNode();
  newPictureFull.style.width = "100%";
  pictureFull.replaceWith(newPictureFull);
  pictureFull = newPictureFull;
};

const closeSlider = () => {
  pictureFull && pictureFull.remove();
  newPictureFull && newPictureFull.remove();
  slider.classList.add("hidden");
};

```

Рисунок 3.7 – Приклад коду реалізації

Приклад сторінки з контактними даними для зворотного зв'язку з організацією. Контент зона складається з блоку в якому розташована інформація про місцезположення, адресу, номер телефону та соціальні мережі (див. рис. 3.8). Також на сайті реалізовано зміна кольорової теми за допомогою мови JavaScript (див. рис. 3.9).

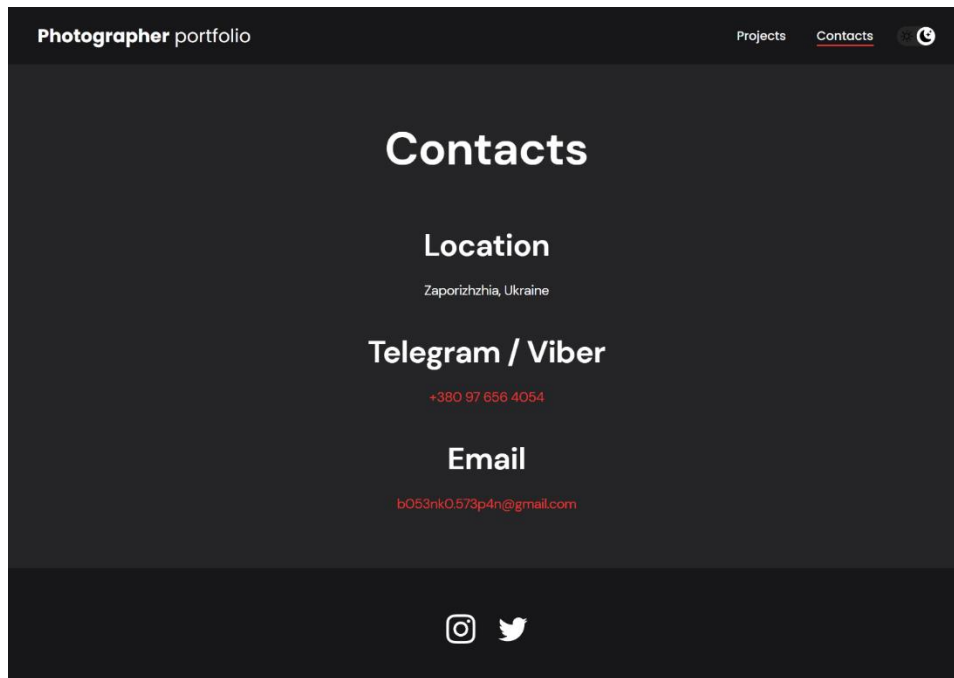


Рисунок 3.8 – Приклад сторінки з контактами

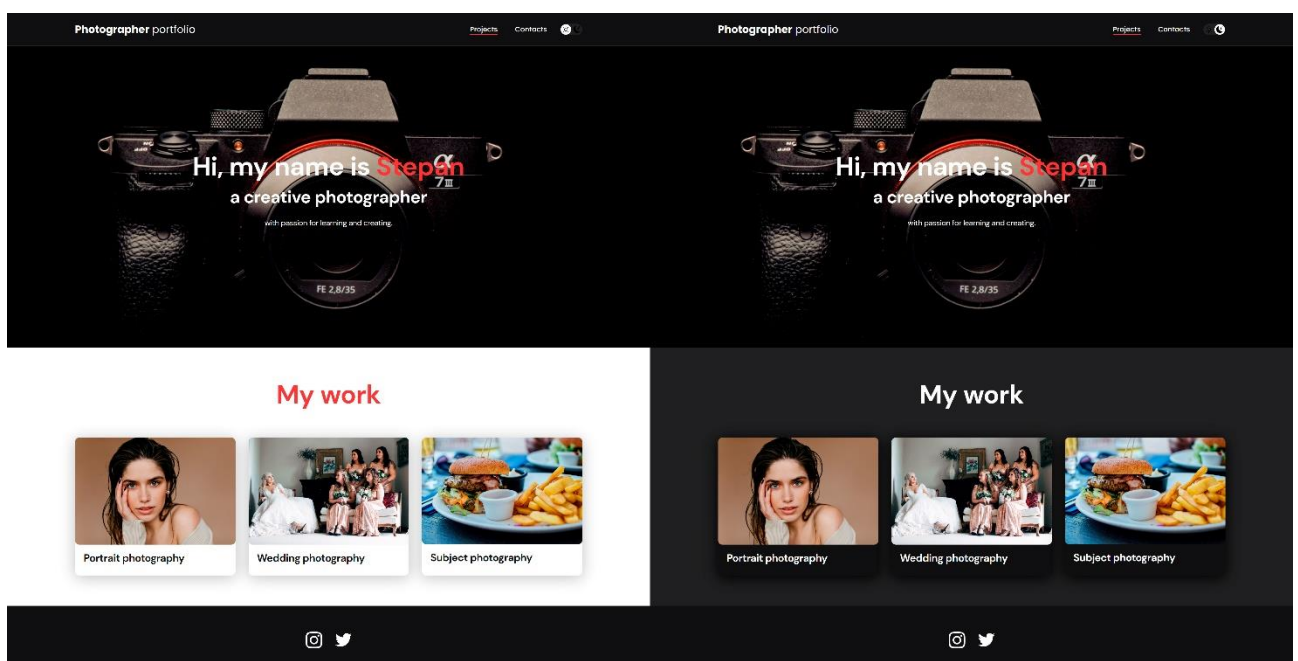


Рисунок 3.9 – Приклад зміни кольорової теми

Приклад коду наведено на рисунку 3.10.

```

import { useEffect, useRef } from "react";
import sun from "./sun.svg";
import moon from "./moon.svg";
import "./style.css";
import { useLocalStorage } from "../../utils/useLocalStorage";
import detectDarkMode from "../../utils/detectDarkMode";

const BtnDarkMode = () => {
  const [darkMode, setDarkMode] = useLocalStorage("darkMode", detectDarkMode());
  const btnRef = useRef(null);

  useEffect(() => {
    if (darkMode === "dark") {
      document.body.classList.add("dark");
      btnRef.current.classList.add("dark-mode-btn--active");
    } else {
      document.body.classList.remove("dark");
      btnRef.current.classList.remove("dark-mode-btn--active");
    }
  }, [darkMode]);

  const toggleDarkMode = () => {
    console.log("toggleDarkMode Fired!");
    setDarkMode((currentValue) => {
      return currentValue === "light" ? "dark" : "light";
    });
  };

  return (
    <button ref={btnRef} className="dark-mode-btn" onClick={toggleDarkMode}>
      <img src={sun} alt="Light mode" className="dark-mode-btn__icon" />
      <img src={moon} alt="Dark mode" className="dark-mode-btn__icon" />
    </button>
  );
};

export default BtnDarkMode;

```

Рисунок 3.10 – Приклад коду

Також за допомогою JavaScript реалізовано функцію опізнання кольорової теми на пристрої і використання такої ж теми на сайті (див. рис. 3.11).

```
const detectDarkMode = () => {  
  if (  
    window.matchMedia &&  
    window.matchMedia("(prefers-color-scheme: dark)").matches  
  ) {  
    return "dark";  
  }  
  return "light";  
};  
export default detectDarkMode;
```

Рисунок 3.11 – Приклад коду

ВИСНОВКИ

У дипломному проекті було розглянуто створення вебсайтів. Було розглянуто вимоги до сучасних вебсайтів та засоби для їх створення. Було сформоване технічне завдання та технічні вимоги до розроблюваного застосунку. Був створений вебсайт портфоліо фотографа.

У ході роботи були виконані такі завдання:

- визначено поняття вебсайту;
- проаналізовано сучасні вебсайти;
- описано інструменти для реалізації вебсайтів;
- розроблено проект вебсайту;
- розроблено макет вебсайту;
- створений вебсайт портфоліо фотографа;
- виконано тестування вебсайту.

Розроблений вебсайт відповідає сформованому технічному завданню та технічним вимогам.

ПЕРЕЛІК ПОСИЛАНЬ

1. Навчальні матеріали на відеосервісі “YouTube”. URL: <https://www.youtube.com/> (дата звернення: 02.03.2023).
2. Пошук теоретичних відомостей за допомогою нейронної мережі “ChatGPT”. URL: <https://chat.openai.com/> (дата звернення: 02.03.2023).
3. Планування та проектування веб-сайту. URL: <https://internetdevels.com/uk/blog/planuvannya-ta-proektuvannya-veb-saytu> (дата звернення: 15.03.2023).
4. Етапи розробки веб-сайту. URL: <https://webdeluxe.com.ua/blog/etapy-rozrobki-veb-sajtu> (дата звернення: 02.03.2023).
5. Дизайн веб-сайту та його розробка. URL: <https://webcapital.com.ua/blog/dizajn-veb-sajtu-ta-yogo-rozrobka> (дата звернення: 10.03.2023).
6. Вибір технологій для веб-сайту. URL: <https://codefront.com.ua/vybir-tehnologij-dlya-veb-sajtu> (дата звернення: 05.03.2023).
7. Front-end розробка веб-сайту. URL: <https://webakademia.com.ua/blog/front-end-rozrobka-veb-saytu> (дата звернення: 05.03.2023).
8. Основи html. URL: <https://webakademia.com.ua/blog/osnovi-html> (дата звернення: 15.04.2023).
9. Основи css. URL: <https://webdeluxe.com.ua/blog/osnovi-css> (дата звернення: 18.04.2023).
10. Вступ до javascript. URL: <https://internetdevels.com/uk/blog/vstup-do-javascript> (дата звернення: 25.04.2023).
11. React.js: вступ до основ. URL: <https://webcapital.com.ua/blog/react-js-vstup-do-osnov> (дата звернення: 25.04.2023).