МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «УНІВЕРСИТЕТ «КРОК»

Фаховий коледж Університету «КРОК»

ДИПЛОМНА РОБОТА

за темою

«Створення вебсайту для студії постпродакшену»

Студент 4 курсу групи КН-20к-1

Шевченко Дмитро Сергійович

(прізвище, ім'я та по-батькові студента)

Керівник дипломної роботи

_____

(посада керівника)

Кириченко Віктор Вікторович

(прізвище, ім'я та по-батькові керівника)

До захисту

(резолюція «До захисту»)

_____ 10.06.2024 _____

(підпис студента) (дата) (підпис викладача)

Київ, 2024 рік

Скорочення

HTML (HyperText Markup Language) - мова розмітки гіпертексту.

CSS (Cascading Style Sheets) - каскадні таблиці стилів.

SEO (Search Engine Optimization) - оптимізація для пошукових систем.

UX (User Experience) - користувацький досвід, взаємодія користувача з продуктом.

UI (User Interface) - користувацький інтерфейс, зовнішній вигляд та елементи керування програмою.

VR (Virtual Reality) - віртуальна реальність.

AR (Augmented Reality) - доповнена реальність.

SSL (Secure Sockets Layer) - захищений рівень сокетів.

API (Application Programming Interface) - програмний інтерфейс прикладного програмування.

CLI (Command-Line Interface) - інтерфейс командного рядка.

AI (Artificial Intelligence) - штучний інтелект.

DOM (Document Object Model) - об'єктна модель документа.

SSR (Server-Side Rendering) - рендеринг на стороні сервера.

Зміст

Вступ

У сучасному світі стрімкий розвиток технологій ставить перед бізнесом нові виклики та можливості. Інтернет та цифровізація проникають у всі сфери нашого життя, включаючи індустрію розваг і медіа. Студії постпродакшену, як частина цієї індустрії, потребують ефективних інструментів для презентації своїх послуг та залучення нових клієнтів. Одним із таких інструментів є вебсайт.

Вебсайт є віртуальним обличчям компанії, що дозволяє не лише представити її послуги, але й побудувати бренд, залучити нових клієнтів та підтримувати комунікацію з наявними. Створення вебсайту для студії постпродакшену SD Production є важливим кроком для підвищення її конкурентоспроможності та ефективності роботи.

На сьогоднішній день існує багато різних рішень для створення вебсайтів, включаючи використання готових платформ та фреймворків. Однак, специфіка роботи студії постпродакшену вимагає індивідуального підходу до розробки вебсайту, який би враховував особливості надання послуг, необхідність демонстрації портфоліо та зручність використання для клієнтів.

Метою даної дипломної роботи є створення функціонального та естетично привабливого вебсайту для студії постпродакшену SD Production. Для досягнення цієї мети були поставлені наступні завдання: провести аналіз спеціалізованої сфери та визначити основні вимоги до вебсайтів студій постпродакшену; дослідити сучасні технології та платформи для створення вебсайтів, вибрати найбільш підходящі для реалізації проекту; розробити концепцію та дизайн вебсайту, що відображатиме бренд SD Production; реалізувати внутрішню логіку вебсайту, забезпечивши його функціональність та зручність використання; провести тестування вебсайту та виправити виявлені помилки; оцінити отримані результати та надати рекомендації щодо подальшого розвитку вебсайту.

Дипломна робота складається з трьох основних розділів. У першому розділі проводиться аналіз спеціалізованої сфери, огляд поточного стану

індустрії постпродакшену та порівняння лідерів ринку. Другий розділ присвячений методам створення вебсайту, порівнянню технологій та вибору платформи. У третьому розділі детально описано процес створення та розробки вебсайту, включаючи концепцію, дизайн, реалізацію внутрішньої логіки та тестування.

Розробка вебсайту для студії постпродакшену SD Production має важливе значення не тільки для самої студії, але й для всієї індустрії в цілому. Вона демонструє можливості використання сучасних технологій для підвищення ефективності роботи та конкурентоспроможності на ринку. Крім того, ця робота може стати корисним прикладом для інших компаній, які планують розробку або вдосконалення своїх вебсайтів.

РОЗДІЛ 1. Аналіз спеціалізованої сфери

1.1. Історія розвитку індустрії постпродакшену

Індустрія постпродакшену є важливою складовою медіа-індустрії, що включає обробку відео- та аудіоматеріалів після їхнього знімання. Цей процес охоплює широкий спектр робіт, таких як монтаж, кольорокорекція, створення візуальних ефектів, звукорежисура та інші етапи, які покращують якість кінцевого продукту та готують його до випуску.

Історія розвитку індустрії постпродакшену налічує кілька десятиліть. З моменту появи перших кінокартин необхідність у постобробці матеріалів стала очевидною. На початку 20-го століття монтаж здійснювався вручну за допомогою фізичного різання і склеювання кіноплівки. З розвитком технологій цей процес поступово автоматизувався, що дозволило значно скоротити час та підвищити якість обробки.

Сучасні тенденції в індустрії постпродакшену значною мірою визначаються швидким розвитком цифрових технологій. Перехід від аналогових до цифрових форматів дозволив значно розширити можливості монтажу та обробки відеоматеріалів. Комп'ютерні технології дали змогу створювати складні візуальні ефекти, інтегрувати CGI (Computer-Generated Imagery) у фільми та рекламу, а також забезпечувати високий рівень деталізації та реалістичності зображень.

Однією з найважливіших тенденцій останніх років є розвиток технологій VR та AR. Ці технології знаходять широке застосування не лише в ігровій індустрії, але й у кіно та телебаченні, відкриваючи нові можливості для постпродакшену. Використання VR/AR дозволяє створювати інтерактивні та занурюючі візуальні досвіди, що стають все більш популярними серед глядачів.

Рис. 1.1. Інтегрування VR в сучасні технології

Індустрія постпродакшену також активно використовує хмарні технології, які забезпечують гнучкість та ефективність роботи. Хмарні сервіси дозволяють командам працювати над проектами віддалено, забезпечуючи доступ до потужних обчислювальних ресурсів та інструментів для обробки відео. Це особливо важливо в умовах глобалізації, коли фахівці з різних куточків світу можуть співпрацювати над одним проектом у режимі реального часу.

Окрім технічних аспектів, важливу роль у постпродакшені відіграють творчі рішення. Кольорокорекція, монтаж та звукорежисура вимагають не лише технічних знань, але й художнього підходу, що дозволяє створювати емоційно насичені та візуально привабливі продукти. Творчість і технічна майстерність йдуть пліч-о-пліч, забезпечуючи високий рівень кінцевого результату.

Загалом, індустрія постпродакшену продовжує розвиватися під впливом новітніх технологій і тенденцій. Важливими аспектами залишаються автоматизація процесів, використання штучного інтелекту для обробки матеріалів, розвиток VR/AR та хмарних сервісів. Ці фактори сприяють підвищенню ефективності роботи студій, покращенню якості кінцевих продуктів та розширенню творчих можливостей фахівців.

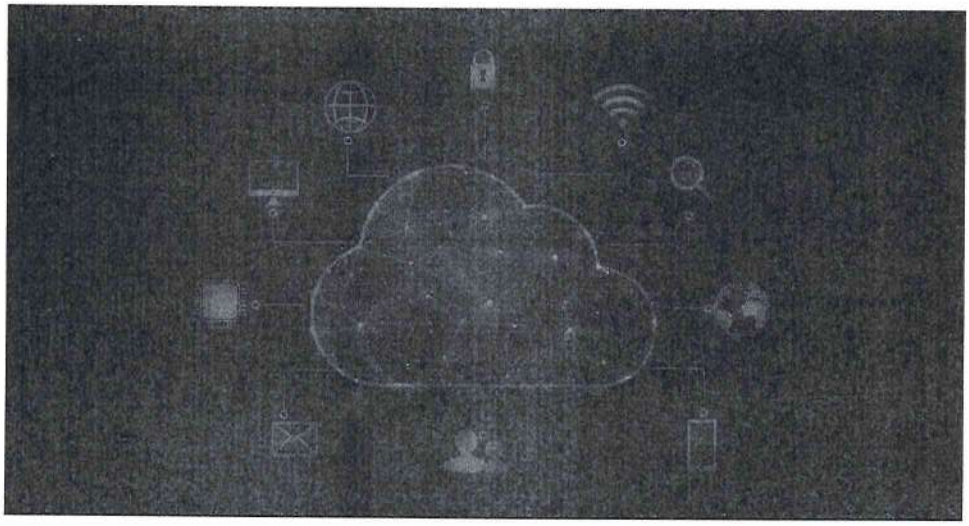

Рис. 1.2. Хмарні технології

1.2. Сучасні тенденції та технології в постпродакшені

Сучасні тенденції в індустрії постпродакшену визначаються швидким розвитком цифрових технологій та інтеграцією новітніх інноваційних рішень. Однією з ключових тенденцій є широке використання хмарних технологій, які дозволяють студіям постпродакшену працювати більш ефективно та гнучко. Хмарні сервіси надають можливість зберігати та обробляти великі обсяги даних віддалено, що особливо важливо для великих проектів з міжнародними командами.

Іншою важливою тенденцією є використання штучного інтелекту для автоматизації рутинних завдань у постпродакшені. Наприклад, AI може бути використаний для автоматичного аналізу та сортування відеоматеріалів, а також для покращення якості зображення та звуку. Це дозволяє фахівцям зосередитися на більш творчих аспектах своєї роботи.

Розвиток технологій віртуальної та доповненої реальності також відкриває нові можливості для індустрії постпродакшену. VR та AR дозволяють створювати інтерактивні візуальні досвіди, які можуть бути використані не лише в ігровій індустрії, але й у кіно та телебаченні. Ці технології дозволяють глядачам зануритися в сюжет фільму або серіалу, створюючи більш глибокий та захоплюючий досвід.

Індустрія постпродакшену також активно впроваджує технології високої роздільної здатності, такі як 4K та 8K, а також технології HDR (High Dynamic Range). Це дозволяє створювати відеопродукцію з вищою якістю зображення, більш насиченими кольорами та детальнішою картинкою. Висока роздільна здатність та HDR стають новим стандартом якості у кіно- та телеіндустрії.

Інша тенденція - це розвиток гнучких методологій та інструментів для спільної роботи. Такі платформи, як Frame.io та Wipster, дозволяють командам працювати над проектами в режимі реального часу, коментувати та редагувати матеріали віддалено. Це значно прискорює процес виробництва та підвищує якість кінцевого продукту.

Загалом, сучасні тенденції та технології в постпродакшені спрямовані на підвищення ефективності роботи, покращення якості відеоматеріалів та розширення творчих можливостей. Інтеграція нових технологій та інноваційних рішень дозволяє студіям постпродакшену залишатися конкурентоспроможними та створювати високоякісні продукти, що відповідають вимогам сучасного ринку.

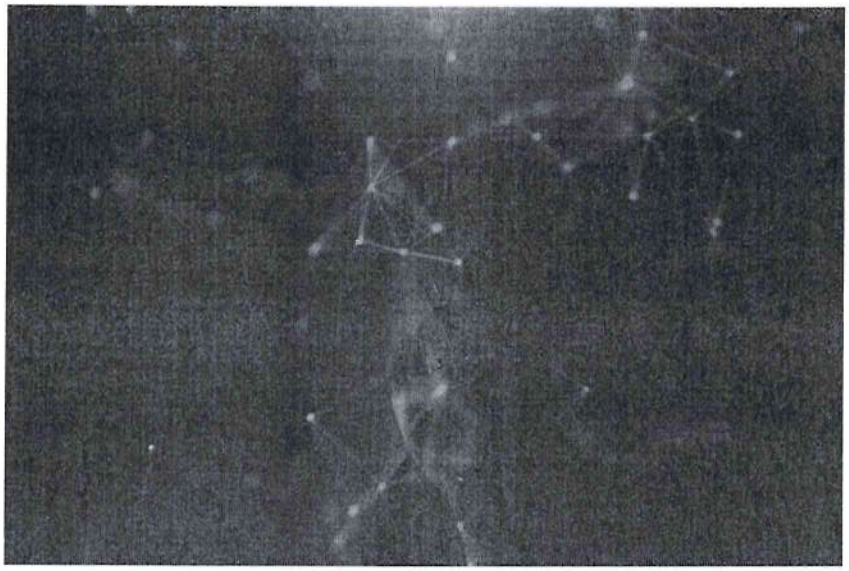

Рис. 1.3. AI технології

1.3. Вимоги до функціональності вебсайту

Функціональність вебсайту є критично важливим аспектом, що визначає його ефективність, зручність використання та здатність виконувати бізнес-завдання. Для студій постпродакшену вебсайт повинен мати такі основні функціональні можливості:

- Презентація послуг: Вебсайт повинен чітко і детально описувати всі послуги, що надаються студією. Кожна послуга має бути представлена на окремій сторінці або в окремому розділі з докладним описом, процесом виконання та перевагами для клієнта. Важливо також додати приклади робіт або кейси, які демонструють якість наданих послуг.
- Демонстрація портфоліо: Портфоліо є ключовим елементом вебсайту студії постпродакшену, оскільки дозволяє потенційним клієнтам ознайомитися з прикладами виконаних робіт. Важливо, щоб цей розділ був зручним і візуально привабливим. Портфоліо може включати відеоролики, скріншоти проектів, інтерактивні елементи та описи кожного проекту.
- Інтеграція з соціальними мережами: Соціальні мережі є важливим каналом для залучення нових клієнтів і підтримки зв'язку з існуючими. Вебсайт повинен мати кнопки для швидкого переходу на офіційні сторінки студії в соціальних мережах (Facebook, Instagram, YouTube тощо), а також можливість ділитися контентом з вебсайту у соціальних мережах.
- Блог та новини: Блог на вебсайті дозволяє демонструвати експертність студії у сфері постпродакшену, публікуючи статті про нові технології, кейс-стаді, огляди обладнання та інші матеріали. Регулярне оновлення блогу не лише підвищує SEO-рейтинги сайту, але й залучає нових відвідувачів.
- Форми зворотного зв'язку та контактна інформація: Для зручності клієнтів вебсайт повинен мати форми зворотного зв'язку, контактні телефони, електронну пошту та можливість замовлення консультацій

онлайн. Це дозволяє клієнтам легко зв'язатися зі студією для отримання додаткової інформації або замовлення послуг.

- Підтримка мультимедійного контенту: Вебсайт повинен підтримувати завантаження та відтворення різноманітного мультимедійного контенту, включаючи відео, аудіо та зображення високої якості. Це особливо важливо для демонстрації прикладів робіт студії.
- Мобільна адаптація: Вебсайт повинен мати адаптивний дизайн, що дозволить коректно відображатися на різних пристроях, включаючи настільні комп'ютери, ноутбуки, планшети та смартфони. Адаптивний дизайн забезпечує зручне використання вебсайту незалежно від розміру екрану та дозволяє охопити ширшу аудиторію.
- Оптимізація швидкості завантаження: Швидкість завантаження вебсайту є важливим фактором, який впливає на користувацький досвід та SEO-рейтинги. Вебсайт повинен бути оптимізований для швидкого завантаження, що включає оптимізацію зображень, використання кешування та мінімізацію кількості запитів до сервера.

1.4. Вимоги до UX/UI дизайну

UI та UX є критичними аспектами при створенні вебсайту для студії постпродакшену. Хороший UX/UI дизайн забезпечує не лише привабливий вигляд вебсайту, але й зручність його використання, що сприяє утриманню користувачів та покращенню їхнього досвіду.

- UI дизайн: Візуальна привабливість вебсайту є важливим фактором, що формує перше враження у користувачів. Вебсайт повинен мати якісний та професійний дизайн, що відповідає бренду студії. Використання високоякісних зображень, відеоматеріалів та графічних елементів підвищує візуальну привабливість сайту. Колірна гама, типографіка та інші елементи дизайну повинні бути узгоджені з корпоративним стилем студії, створюючи цілісний і професійний вигляд.

- UX дизайн: Зручність використання вебсайту є ключовим фактором, що впливає на задоволеність користувачів. Інтуїтивно зрозуміле меню навігації дозволяє користувачам легко знаходити необхідну інформацію та переходити між розділами сайту. Мінімізація кількості кліків для досягнення цілей, чіткі інструкції та зрозумілі позначки підвищують зручність використання вебсайту.
- Адаптивний дизайн: Адаптивність дизайну забезпечує коректне відображення вебсайту на різних пристроях, включаючи настільні комп'ютери, ноутбуки, планшети та смартфони. Це важливо для забезпечення зручного користувацького досвіду незалежно від розміру екрана. Адаптивний дизайн також позитивно впливає на SEO-рейтинги вебсайту.
- Доступність: Вебсайт повинен бути доступним для всіх користувачів, включаючи людей з обмеженими можливостями. Використання стандартів доступності, таких як WCAG (Web Content Accessibility Guidelines), дозволяє забезпечити, що вебсайт буде зручним та функціональним для всіх користувачів. Це включає текстові альтернативи для зображень, можливість навігації за допомогою клавіатури та інші елементи, що підвищують доступність.

Таким чином, функціональність вебсайту та UX/UI дизайн є ключовими факторами, що впливають на успішність вебсайту студії постпродакшену. Забезпечення високого рівня функціональності та зручності використання дозволяє створити вебсайт, який ефективно виконує бізнес-завдання та задовольняє потреби користувачів.

1.5. SEO-оптимізація

SEO-оптимізація (Search Engine Optimization) є важливим аспектом розробки вебсайту для студії постпродакшену, оскільки вона впливає на видимість сайту в пошукових системах та залучення органічного трафіку.

Ефективна SEO-оптимізація допомагає підвищити рейтинг вебсайту в пошукових системах, таких як Google, що призводить до збільшення кількості відвідувачів і потенційних клієнтів.

Одним із головних елементів SEO є використання релевантних ключових слів. Ключові слова – це терміни та фрази, які потенційні клієнти вводять у пошукові системи для пошуку послуг постпродакшену. Важливо провести дослідження ключових слів і визначити ті, що найчастіше використовуються цільовою аудиторією. Ці ключові слова повинні бути інтегровані в заголовки, описи, метатеги, URL-адреси та вміст сторінок вебсайту.

Контент на вебсайті повинен бути якісним, релевантним та інформативним. Важливо створювати унікальні описи послуг, публікувати статті в блозі, кейс-стаді та інші матеріали, які будуть цікавими для відвідувачів. Кожна сторінка повинна мати оптимізовані заголовки (H1, H2, H3), метаописи (meta descriptions) та ключові слова, які відповідають її змісту. Крім того, важливо уникати дублювання контенту, оскільки це може негативно вплинути на рейтинг у пошукових системах.

Технічний аспект SEO включає оптимізацію коду вебсайту та його структури. Вебсайт повинен мати чистий і валідний код, який швидко завантажується та правильно індексується пошуковими системами. Використання семантичної розмітки (schema markup) допомагає пошуковим системам краще розуміти вміст вебсайту. Крім того, важливо забезпечити швидкість завантаження сторінок, оскільки повільний вебсайт може негативно вплинути на рейтинг.

Вебсайт повинен бути оптимізований для мобільних пристроїв, оскільки все більше користувачів здійснюють пошуки зі смартфонів і планшетів. Адаптивний дизайн, який автоматично підлаштовується під розмір екрана, забезпечує зручність використання вебсайту на будь-якому пристрої. Google

також надає перевагу мобільним версіям вебсайтів у своєму алгоритмі ранжування, тому мобільна оптимізація є важливою складовою SEO-стратегії.

Внутрішня оптимізація включає оптимізацію внутрішніх посилань, що допомагає пошуковим системам краще індексувати сторінки вебсайту. Важливо створити логічну структуру внутрішніх посилань, яка сприяє навігації та покращує досвід користувачів. Зовнішня оптимізація включає отримання зворотних посилань (backlinks) від авторитетних ресурсів. Якісні зворотні посилання підвищують авторитет вебсайту в очах пошукових систем, що позитивно впливає на рейтинг.

Оскільки студія постпродакшену використовує багато мультимедійного контенту, важливо оптимізувати зображення та відео для пошукових систем. Використання описових імен файлів, альтернативних текстів (alt text) і стиснення зображень без втрати якості допомагає покращити SEO-показники. Відео також повинні мати оптимізовані назви, описи та теги.

Для студій постпродакшену, що обслуговують конкретний географічний регіон, важливо зосередитися на локальному SEO. Реєстрація у Google My Business, використання локальних ключових слів та розміщення контактної інформації на вебсайті допомагають залучати клієнтів з певного регіону. Відгуки клієнтів також відіграють важливу роль у локальному SEO, підвищуючи довіру до студії.

Важливо регулярно моніторити SEO-показники вебсайту за допомогою інструментів, таких як Google Analytics та Google Search Console. Аналіз даних про трафік, поведінку користувачів, рейтинги ключових слів та інші показники допомагає виявляти проблеми та вносити корективи до SEO-стратегії. Регулярний аудит вебсайту дозволяє підтримувати його оптимізованим та конкурентоспроможним.

Таким чином, SEO-оптимізація є важливим компонентом успішної роботи вебсайту студії постпродакшену. Ефективна SEO-стратегія допомагає

підвищити видимість вебсайту в пошукових системах, залучити більше відвідувачів та збільшити кількість потенційних клієнтів.

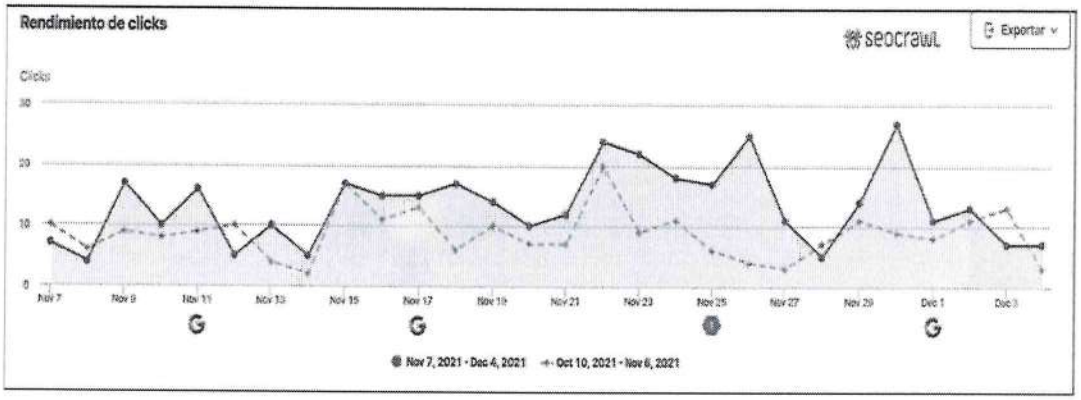

Рис. 1.4. SEO аналітика відвідуваності сайту

1.6. Порівняльний аналіз конкурентів

Для успішного створення вебсайту для студії постпродакшену важливо проаналізувати вебсайти лідерів ринку та конкурентів. Це дозволить визначити кращі практики, виявити можливі слабкі місця та вдосконалити власний сайт, враховуючи досвід інших компаній.

Перш за все, варто звернути увагу на структуру вебсайтів конкурентів. Більшість успішних студій постпродакшену мають чітко структуровані вебсайти з головною сторінкою, розділами послуг, портфоліо, блогом та контактною інформацією. Головна сторінка зазвичай містить короткий опис студії, ключові послуги та приклади робіт, що дозволяє відразу зацікавити відвідувача.

Розділи послуг повинні бути детально описані, з акцентом на перевагах та унікальності кожної послуги. Важливо, щоб ці розділи були зрозумілими та легко доступними для користувачів, що допоможе потенційним клієнтам швидко знайти необхідну інформацію. Крім того, деякі вебсайти пропонують інтерактивні елементи, такі як калькулятори вартості послуг або форми для запиту додаткової інформації, що полегшує комунікацію з клієнтами.

Портфоліо є одним з найважливіших розділів вебсайту, оскільки воно демонструє реальні приклади робіт студії. Аналізуючи портфоліо конкурентів, можна звернути увагу на формат презентації робіт, використання відео та зображень високої якості, а також інтерактивні елементи, що дозволяють користувачам детально ознайомитися з проектами. Деякі вебсайти використовують відео-презентації, які демонструють процес створення проекту від початку до кінця, що додає довіри та демонструє професіоналізм студії.

Блоги та новини на вебсайтах конкурентів також є важливим елементом. Вони не лише підвищують SEO-рейтинги сайту, але й дозволяють демонструвати експертність студії у сфері постпродакшену. Регулярне оновлення контенту, публікація статей про нові технології, кейс-стаді та інші матеріали створюють позитивний імідж компанії та залучають нових клієнтів. Більшість успішних студій публікують статті про свої останні проекти, технологічні досягнення та інтерв'ю з експертами, що додає додаткової вартості для відвідувачів.

Інтерактивність та зручність використання також є важливими аспектами. Сайти конкурентів часто мають інтерактивні елементи, такі як анімації, переходи між сторінками та інтерактивні форми, що покращують користувацький досвід. Використання сучасних веб-технологій, таких як HTML5, CSS3 JavaScript та його бібліотек, дозволяє створити динамічний та привабливий дизайн, що утримує користувачів на сайті.

1.7. Визначення сильних та слабких сторін конкурентів

Аналізуючи вебсайти конкурентів, важливо визначити їх сильні та слабкі сторони. Це дозволить зрозуміти, які аспекти можна запозичити для покращення власного сайту, а яких помилок слід уникати.

Серед сильних сторін вебсайтів лідерів ринку можна виділити якісний дизайн, зручну навігацію, детальні описи послуг та високоякісне портфоліо.

Використання сучасних технологій, таких як адаптивний дизайн, інтерактивні елементи та анімації, також є важливими складовими успіху. Вебсайти з продуманим UX/UI дизайном забезпечують зручність використання, що сприяє утриманню користувачів та покращенню їхнього досвіду.

Детальні описи послуг і приклади робіт у портфоліо дозволяють потенційним клієнтам отримати повне уявлення про можливості студії. Блоги та новини, які регулярно оновлюються, підвищують авторитет компанії та залучають нових відвідувачів. Інтерактивні елементи, такі як калькулятори вартості послуг, форми для замовлення консультацій та інтерактивні презентації проектів, роблять взаємодію з вебсайтом більш приємною та ефективною.

До слабких сторін можна віднести надмірну складність навігації, відсутність чіткої структури, недостатню кількість інформації про послуги або низьку якість контенту. Деякі вебсайти можуть мати застарілий дизайн або незручні для користувачів елементи, що відштовхує відвідувачів. Крім того, повільне завантаження сторінок і відсутність мобільної оптимізації можуть негативно вплинути на користувацький досвід.

Важливо звернути увагу на те, як конкуренти взаємодіють з відвідувачами через вебсайт. Наявність форм зворотного зв'язку, онлайн-чатів та інших засобів комунікації дозволяє швидко відповідати на запити клієнтів та покращувати їхній досвід взаємодії з компанією. Недоліки в цій сфері можуть включати відсутність інтерактивних засобів комунікації або тривалий час відповіді на запити.

Визначення сильних та слабких сторін конкурентів дозволяє створити стратегію покращення власного вебсайту, враховуючи найкращі практики та уникаючи поширених помилок. Це допоможе створити ефективний, зручний та привабливий вебсайт для студії постпродакшену SD Production, який буде конкурентоспроможним на ринку та задовольнятиме потреби користувачів.

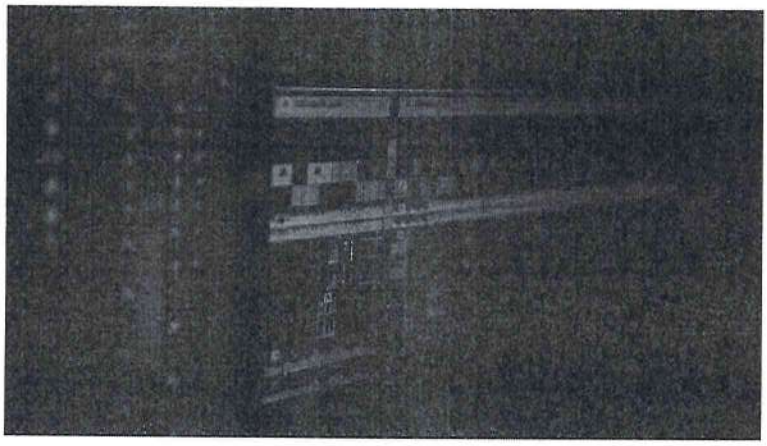

Рис. 1.5. Інтерфейс програми Adobe Premiere Pro

РОЗДІЛ 2. Методи створення вебсайту

2.1. Аналіз вимог

Процес створення вебсайту для студії постпродакшену SD Production починається з аналізу вимог. Цей етап є надзвичайно важливим, оскільки визначає основу для всіх наступних етапів розробки. Аналіз вимог передбачає збір і систематизацію інформації про цілі, які повинен виконувати вебсайт, а також про потреби і очікування користувачів. На цьому етапі важливо врахувати як бізнес-вимоги студії, так і технічні аспекти реалізації проекту.

В першу чергу, необхідно чітко визначити основні цілі вебсайту. Для студії постпродакшену такі цілі можуть включати представлення послуг, демонстрацію портфоліо, залучення нових клієнтів, підтримку зв'язку з існуючими клієнтами та підвищення впізнаваності бренду. Важливо визначити, хто є основною аудиторією вебсайту. Це можуть бути продюсери, режисери, рекламні агенції та інші потенційні клієнти, які шукають послуги постпродакшену. Розуміння потреб і поведінки цільової аудиторії допомагає створити вебсайт, що задовольняє їхні очікування.

На цьому етапі визначаються конкретні функціональні можливості, які повинен мати вебсайт. Це може включати розділи з описом послуг, портфоліо, блоги, форми зворотного зв'язку, інтеграцію з соціальними мережами, системи онлайн-замовлень та інші функції. Крім функціональних, важливо врахувати нефункціональні вимоги, такі як швидкість завантаження вебсайту, безпека, масштабованість, доступність та зручність використання на різних пристроях.

Включають вимоги до програмного забезпечення та апаратного забезпечення, які будуть використовуватися для розробки та хостингу вебсайту. Це може включати вибір платформи для розробки, бази даних, серверного середовища та інших технічних аспектів. Дослідження вебсайтів конкурентів допомагає зрозуміти, які рішення використовуються в галузі, що можна запозичити, а що слід уникати. Це дозволяє створити конкурентоспроможний вебсайт, який виділяється на тлі інших.

2.2. Планування та проектування

Після завершення аналізу вимог настає етап планування та проектування вебсайту. Цей етап передбачає розробку детального плану реалізації проекту та створення архітектури вебсайту.

Технічне завдання є документом, який детально описує всі вимоги до вебсайту, включаючи функціональні та нефункціональні вимоги, технічні характеристики, дизайн, структуру контенту та інші аспекти. Технічне завдання є основою для роботи команди розробників і забезпечує чітке розуміння проекту.

Проектний план включає визначення етапів реалізації проекту, розподіл завдань між членами команди, встановлення термінів виконання робіт та визначення ресурсів, необхідних для реалізації проекту. Важливо також передбачити можливі ризики та розробити стратегії їх мінімізації.

На цьому етапі створюється загальна структура вебсайту, включаючи визначення основних розділів і сторінок, навігаційного меню, взаємозв'язків між елементами та організації контенту. Проектування архітектури також включає визначення технологій і фреймворків, які будуть використовуватися для розробки вебсайту. У випадку SD Production було обрано React для розробки фронтенду, що дозволяє створювати динамічні і масштабовані веб-додатки.

Створення прототипів допомагає візуалізувати структуру і функціональність вебсайту. Прототипи були створені за допомогою інструменту Figma, що дозволяє тестувати користувацький інтерфейс, вносити зміни на ранніх етапах і отримувати зворотний зв'язок від зацікавлених сторін. Figma також полегшує спільну роботу над дизайном, дозволяючи різним членам команди вносити правки і пропозиції в реальному часі.

На основі прототипів розробляється дизайн користувацького інтерфейсу та користувацького досвіду. Дизайн включає визначення колірної гами, типографіки, стилю елементів, анімацій та інших візуальних аспектів. Важливо забезпечити консистентність дизайну на всіх сторінках вебсайту і створити привабливий та інтуїтивно зрозумілий інтерфейс.

Перед початком розробки важливо провести тестування прототипів і дизайну, щоб переконатися, що всі вимоги виконані, і користувачі можуть легко взаємодіяти з вебсайтом. Це може включати тестування юзабіліті, збір зворотного зв'язку від потенційних користувачів та внесення необхідних змін. Ці етапи планування та проектування є критичними для успішного створення вебсайту, оскільки вони закладають основу для подальшої розробки та забезпечують чітке розуміння всіх аспектів проекту.

Тож, ось план від створення до запуску:

- Аналіз вимог: Збір і систематизація інформації про цілі вебсайту, потреби користувачів і технічні вимоги.
- Розробка технічного завдання: Детальний опис всіх вимог до вебсайту, включаючи функціональні та нефункціональні вимоги, технічні характеристики, дизайн і структуру контенту.
- Проектування архітектури: Створення загальної структури вебсайту, визначення основних розділів і сторінок, навігаційного меню, взаємозв'язків між елементами та організації контенту.
- Створення прототипів: Розробка візуальних прототипів за допомогою інструменту Figma для тестування користувацького інтерфейсу і отримання зворотного зв'язку.
- Розробка дизайну: Визначення колірної гами, типографіки, стилю елементів, анімацій та інших візуальних аспектів, забезпечення консистентності дизайну на всіх сторінках вебсайту.

- Тестування прототипів і дизайну: Проведення тестування юзабіліті, збір зворотного зв'язку від потенційних користувачів і внесення необхідних змін.
- Розробка фронтенду: Використання React для створення динамічного і масштабованого веб-додатку.
- Інтеграція з бекендом: Зв'язок фронтенду з серверною частиною, базами даних і іншими системами.
- Тестування і налагодження: Проведення всебічного тестування вебсайту для виявлення і виправлення помилок.
- Запуск і моніторинг: Розгортання вебсайту на сервері, моніторинг його роботи і внесення необхідних коректив.

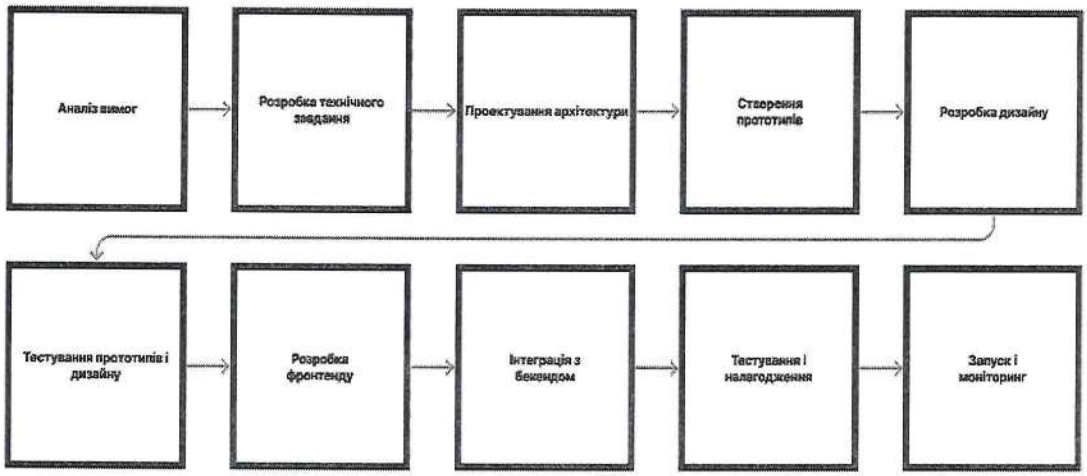

Рис. 2.1. План від створення до запуску

2.2. Розробка

Етап розробки починається після завершення планування та проектування. Основна мета цього етапу – створення функціонального та повністю робочого вебсайту згідно з технічним завданням та затвердженим дизайном. Розробка вебсайту для студії постпродакшену SD Production включає декілька ключових кроків.

Першим кроком є створення фронтенду вебсайту. Для цього було обрано React – сучасний JavaScript-фреймворк, який дозволяє створювати динамічні та інтерактивні веб-додатки. Використання React забезпечує високу продуктивність та масштабованість вебсайту. На цьому етапі розробники працюють над реалізацією всіх візуальних компонентів, таких як навігаційне меню, розділи з описом послуг, портфоліо, блог та форми зворотного зв'язку.

Наступним кроком є розробка бекенду. Бекенд відповідає за обробку даних, збереження інформації в базі даних, а також за комунікацію між фронтендом та серверною частиною вебсайту. Для бекенду може бути використаний Node.js, який дозволяє ефективно обробляти запити користувачів та забезпечує швидку роботу вебсайту. Бекенд також включає налаштування бази даних, яка може бути реалізована за допомогою MySQL або іншої системи управління базами даних.

Після цього проводиться інтеграція фронтенду з бекендом. Це включає налаштування API, які дозволяють обмінюватися даними між клієнтською та серверною частинами. Наприклад, при запиті користувача на перегляд портфоліо, фронтенд відправляє запит до бекенду, який повертає необхідні дані з бази даних.

Окрім цього, важливим аспектом є забезпечення безпеки вебсайту. Це включає використання SSL сертифікатів для захисту передачі даних, налаштування захисту від SQL-ін'єкцій, міжсайтових скриптових атак (XSS) та інших вразливостей. Безпека даних користувачів є пріоритетним завданням, тому на цьому етапі особлива увага приділяється захисту інформації.

2.3. Тестування

Тестування є критичним етапом у процесі розробки вебсайту, оскільки воно дозволяє виявити і виправити помилки перед запуском вебсайту в експлуатацію. Тестування забезпечує високу якість продукту та гарантує, що вебсайт буде працювати коректно на всіх пристроях і браузерах.

На першому етапі проводиться функціональне тестування, яке перевіряє, чи всі функції вебсайту працюють відповідно до вимог. Це включає перевірку навігації, форм зворотного зв'язку, реєстрації користувачів, завантаження та відображення контенту. Функціональне тестування допомагає виявити помилки в логіці роботи вебсайту та забезпечити його коректну роботу.

Після цього проводиться тестування юзабіліті, яке оцінює зручність використання вебсайту. Це тестування включає залучення реальних користувачів для перевірки інтерфейсу та функціональності. Користувачі виконують різні завдання на вебсайті, а їхня взаємодія аналізується для виявлення можливих проблем з юзабіліті. Результати цього тестування дозволяють внести зміни, що покращують зручність використання вебсайту.

Крім того, важливим етапом є тестування на сумісність, яке перевіряє роботу вебсайту на різних пристроях і браузерах. Це включає перевірку адаптивності дизайну, коректного відображення елементів інтерфейсу та функціональності на різних платформах, таких як десктопи, планшети і смартфони. Тестування на сумісність забезпечує однаковий досвід користувачів незалежно від використовуваного пристрою.

Далі проводиться тестування продуктивності, яке оцінює швидкість завантаження сторінок і загальну продуктивність вебсайту. Це включає вимірювання часу завантаження, перевірку роботи під навантаженням та виявлення вузьких місць, що можуть сповільнювати роботу вебсайту. Тестування продуктивності дозволяє оптимізувати вебсайт для забезпечення швидкого та стабільного досвіду користувачів.

Завершальним етапом є тестування безпеки, яке перевіряє захист вебсайту від потенційних атак і вразливостей. Це включає тестування на проникнення, перевірку захисту даних користувачів, виявлення та виправлення вразливостей. Тестування безпеки забезпечує захист вебсайту та даних користувачів від можливих загроз.

Після завершення всіх етапів тестування проводиться фінальний огляд і верифікація вебсайту. Всі виявлені помилки виправляються, а вебсайт готується до запуску. Тестування гарантує, що вебсайт відповідає всім вимогам і готовий до використання користувачами.

2.4. Запуск та підтримка

Після завершення всіх етапів розробки та тестування вебсайту настає важливий етап запуску та подальшої підтримки. Запуск вебсайту включає декілька ключових кроків, які гарантують його успішне введення в експлуатацію та забезпечення стабільної роботи.

Першим кроком є розгортання вебсайту на сервері. Для цього необхідно вибрати надійного хостинг-провайдера, який зможе забезпечити необхідні ресурси та підтримку для вебсайту. Важливо налаштувати серверне середовище, включаючи бази даних, веб-сервер та інші необхідні компоненти. Після цього здійснюється перенесення файлів вебсайту з локального середовища розробки на сервер.

Після розгортання вебсайту проводиться фінальна перевірка всіх функцій та компонентів, щоб переконатися, що вебсайт працює коректно в продуктивному середовищі. Це включає перевірку роботи баз даних, тестування інтерактивних елементів, перевірку коректності відображення контенту та інших функцій.

Наступним кроком є налаштування системи моніторингу. Моніторинг дозволяє відстежувати продуктивність вебсайту, виявляти можливі проблеми та забезпечувати оперативне реагування на них. Для цього використовуються спеціальні інструменти та сервіси, які надають детальну інформацію про стан вебсайту, його продуктивність та безпеку.

Після успішного запуску вебсайту важливо забезпечити його постійну підтримку та оновлення. Підтримка включає в себе регулярне оновлення

програмного забезпечення, виправлення можливих помилок, забезпечення безпеки та захисту даних. Важливо також проводити регулярні резервні копії даних, щоб забезпечити їх збереження у випадку непередбачених обставин.

Окрім технічної підтримки, важливо забезпечити постійне оновлення контенту на вебсайті. Це включає публікацію нових статей у блозі, оновлення портфоліо, додавання нових послуг та інший контент, який буде цікавим для користувачів. Регулярне оновлення контенту допомагає підтримувати інтерес користувачів та підвищувати рейтинг вебсайту в пошукових системах.

Підтримка вебсайту також включає роботу з користувачами. Важливо забезпечити зручні канали зворотного зв'язку, де користувачі можуть залишати свої відгуки, задавати питання та отримувати допомогу. Взаємодія з користувачами допомагає покращувати вебсайт та забезпечувати високу якість обслуговування.

2.5. Порівняння технологій та платформ для створення вебсайту

При створенні вебсайту для студії постпродакшену важливо вибрати правильні технології та платформи, які забезпечать високу продуктивність, зручність у використанні та можливість подальшого розвитку. У цій частині ми розглянемо кілька популярних технологій і платформ для розробки вебсайтів, зосереджуючи особливу увагу на React, а також на інших альтернативних технологіях.

React – це бібліотека JavaScript, розроблена компанією Facebook, яка використовується для створення інтерфейсів користувача. Основною перевагою React є його висока продуктивність завдяки використанню віртуального DOM, що дозволяє мінімізувати кількість операцій з реальним DOM і покращити швидкість роботи вебсайту. React також забезпечує компонентний підхід до розробки, що дозволяє розробникам створювати багаторазові компоненти, які можуть бути легко використані в різних частинах вебсайту.

Компонентний підхід React забезпечує високу модульність коду, що спрощує підтримку та масштабування проекту. Розробники можуть створювати незалежні компоненти, які можуть бути протестовані і використовувані повторно, що значно знижує складність розробки та дозволяє швидко вносити зміни. Крім того, React має великий набір готових бібліотек і компонентів, що дозволяє значно прискорити процес розробки.

React також підтримує SSR, що покращує SEO-оптимізацію та швидкість завантаження сторінок. Завдяки цьому вебсайт може бути індексований пошуковими системами, що покращує його видимість у пошукових результатах. Крім того, серверний рендеринг дозволяє швидше відображати сторінки для користувачів, що покращує загальний користувацький досвід.

Ще однією перевагою React є його екосистема. React має велику спільноту розробників, які постійно створюють нові інструменти та бібліотеки. Це забезпечує розробникам доступ до широкого спектру рішень для різних завдань, таких як управління станом (Redux), маршрутизація (React Router), інтеграція з графічними інтерфейсами (Material-UI) та багато іншого. Завдяки цьому розробники можуть швидко знайти і впровадити потрібні інструменти у свої проекти.

Поряд з React існує кілька інших популярних фреймворків і бібліотек для розробки вебсайтів. Одним з таких фреймворків є Angular, розроблений компанією Google. Angular використовує двостороннє зв'язування даних, що дозволяє автоматично синхронізувати модель і вид. Angular також забезпечує високу продуктивність і має потужну екосистему інструментів для розробки, проте має більшу криву навчання порівняно з React.

Vue.js – ще один популярний фреймворк для створення інтерфейсів користувача. Vue.js поєднує в собі найкращі риси React і Angular, забезпечуючи високу продуктивність і зручність у використанні. Vue.js використовує компонентний підхід і підтримує реактивне програмування, що дозволяє створювати динамічні та інтерактивні веб-додатки. Vue.js має меншу криву

навчання, ніж Angular, і забезпечує високу продуктивність, проте його екосистема менша, ніж у React.

Що стосується бекенду, Node.js є популярним вибором для створення серверних додатків. Node.js використовує асинхронну модель обробки запитів, що забезпечує високу продуктивність і можливість обробки великої кількості одночасних запитів. Node.js має великий набір бібліотек і модулів, що дозволяє розробникам швидко реалізувати необхідний функціонал. Інші альтернативи для бекенду включають Python з фреймворком Django, Ruby on Rails та PHP з фреймворком Laravel.

Для забезпечення адаптивного дизайну вебсайту важливо використовувати сучасні технології HTML5, CSS3 та фреймворки, такі як Bootstrap. HTML5 та CSS3 дозволяють створювати семантичний і стильний контент, а Bootstrap забезпечує зручні інструменти для створення адаптивного дизайну, що коректно відображається на різних пристроях і екранах.

При виборі технологій важливо враховувати також інструменти для розробки та управління проектом. Використання систем контролю версій, таких як Git, дозволяє забезпечити зручне спільне управління кодом та історією змін. Інструменти для автоматизації розгортання, такі як Docker, допомагають спростити процес налаштування серверного середовища та забезпечити стабільну роботу вебсайту в різних середовищах.

Таким чином, вибір технологій та платформ для створення вебсайту для студії постпродакшену повинен базуватися на вимогах проекту, можливостях забезпечення високої продуктивності, зручності у використанні та подальшого розвитку. React є потужним інструментом для створення фронтенду, забезпечуючи високу продуктивність та модульність коду. У поєднанні з Node.js для бекенду та сучасними інструментами для забезпечення адаптивного дизайну, ці технології забезпечують успішну реалізацію проекту.

2.6. HTML, CSS, JavaScript

Основою для будь-якого вебсайту є використання технологій HTML, CSS і JavaScript. Кожна з цих технологій відіграє важливу роль у створенні функціональних та естетично привабливих веб-сторінок.

Рис. 2.2. HTML, CSS, JavaScript

HTML – це мова розмітки, яка використовується для створення структури веб-сторінки. HTML визначає елементи, такі як заголовки, параграфи, посилання, зображення та інші компоненти, які складають вміст веб-сторінки. HTML5, остання версія HTML, надає додаткові можливості, включаючи підтримку мультимедійних елементів, семантичних тегів і покращену інтеграцію з JavaScript.

CSS – це мова стилів, яка використовується для визначення зовнішнього вигляду веб-сторінки. CSS дозволяє налаштовувати кольори, шрифти, розміри, відступи та інші стилі елементів HTML. Використання CSS забезпечує відокремлення візуального представлення від структури контенту, що полегшує підтримку та оновлення дизайну веб-сайту. CSS3, остання версія CSS, додає нові можливості, такі як анімації, переходи, градієнти і медіа-запити для адаптивного дизайну.

JavaScript – це мова програмування, яка додає інтерактивність до веб-сторінок. JavaScript дозволяє створювати динамічний контент, керувати подіями користувачів, перевіряти дані форм та взаємодіяти з сервером без перезавантаження сторінки. JavaScript є основою для багатьох сучасних веб-технологій і фреймворків, що забезпечують створення складних веб-додатків.

HTML, CSS і JavaScript працюють разом для створення повноцінного веб-досвіду. HTML забезпечує структуру, CSS – стиль, а JavaScript – функціональність. Ці технології є фундаментальними для будь-якого веб-проекту і використовуються в комбінації з іншими інструментами та фреймворками для створення сучасних веб-додатків.

2.7. React

React – це бібліотека JavaScript, розроблена компанією Facebook, яка використовується для створення інтерфейсів користувача. Основною перевагою React є його висока продуктивність завдяки використанню віртуального DOM, що дозволяє мінімізувати кількість операцій з реальним DOM і покращити швидкість роботи веб-сайту. React також забезпечує компонентний підхід до розробки, що дозволяє розробникам створювати багаторазові компоненти, які можуть бути легко використані в різних частинах веб-сайту.

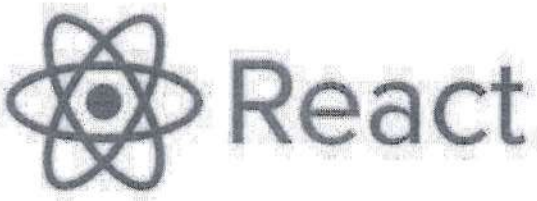

Рис. 2.3. React

Компонентний підхід React забезпечує високу модульність коду, що спрощує підтримку та масштабування проекту. Розробники можуть створювати незалежні компоненти, які можуть бути протестовані і використовувані повторно, що значно знижує складність розробки та дозволяє швидко вносити зміни. Крім того, React має великий набір готових бібліотек і компонентів, що дозволяє значно прискорити процес розробки.

React також підтримує SSR, що покращує SEO-оптимізацію та швидкість завантаження сторінок. Завдяки цьому веб-сайт може бути індексований пошуковими системами, що покращує його видимість у пошукових результатах. Крім того, серверний рендеринг дозволяє швидше відображати сторінки для користувачів, що покращує загальний користувацький досвід.

Ще однією перевагою React є його екосистема. React має велику спільноту розробників, які постійно створюють нові інструменти та бібліотеки. Це забезпечує розробникам доступ до широкого спектру рішень для різних завдань, таких як управління станом (Redux), маршрутизація (React Router), інтеграція з графічними інтерфейсами (Material-UI) та багато іншого. Завдяки цьому розробники можуть швидко знайти і впровадити потрібні інструменти у свої проекти.

Для створення повноцінного веб-сайту на базі React важливо забезпечити інтеграцію фронтенду з бекендом. У випадку SD Production бекенд був реалізований за допомогою Node.js. Node.js – це середовище виконання JavaScript, яке дозволяє створювати серверні додатки. Основною перевагою Node.js є його асинхронна модель обробки запитів, що забезпечує високу продуктивність і можливість обробки великої кількості одночасних запитів.

На веб-сайті SD Production користувачі можуть залишати свої контактні дані та запити через спеціальні форми. Ці дані надсилаються на сервер, де обробляються і зберігаються у базі даних. Node.js забезпечує швидку та ефективну обробку цих запитів, що дозволяє оперативно реагувати на запити клієнтів. Окрім цього, на веб-сайті було налаштовано інтеграцію з Telegram-ботом. Це означає, що коли користувач залишає запит на веб-сайті, відповідне повідомлення автоматично надсилається в Telegram-бот, де його можна переглянути та відповісти клієнту. Така інтеграція забезпечує зручність для адміністратора веб-сайту, оскільки всі запити клієнтів зберігаються в одному місці і можуть бути оброблені швидко і ефективно.

Таким чином, вибір технологій HTML, CSS, JavaScript і React для створення веб-сайту забезпечує високу продуктивність, зручність у використанні та можливість подальшого розвитку проекту. Комбінація цих технологій дозволяє створювати сучасні, інтерактивні та ефективні веб-додатки, які відповідають вимогам бізнесу та очікуванням користувачів.

2.8. Інші можливі технології

Окрім React, існує безліч інших технологій і фреймворків, які можна використовувати для створення вебсайту. Кожна з них має свої унікальні особливості, переваги та недоліки, що дозволяє вибрати найкраще рішення в залежності від конкретних потреб проекту.

Angular – це фреймворк JavaScript, розроблений компанією Google. Angular використовує архітектуру компонентів, подібно до React, але надає більш комплексне рішення з вбудованими інструментами для розробки. Основними перевагами Angular є двостороннє зв'язування даних, залежні ін'єкції та потужний CLI для автоматизації процесів розробки. Однак, Angular має більш круту криву навчання порівняно з React, що може бути складним для новачків.

Vue.js – це прогресивний фреймворк JavaScript для створення користувацьких інтерфейсів. Vue.js поєднує в собі найкращі риси React і Angular, забезпечуючи високу продуктивність і зручність у використанні. Vue.js використовує компонентний підхід і підтримує реактивне програмування, що дозволяє створювати динамічні та інтерактивні веб-додатки. Vue.js має меншу криву навчання, ніж Angular, і забезпечує високу продуктивність, проте його екосистема менша, ніж у React.

Svelte – це сучасний фреймворк для створення користувацьких інтерфейсів, який відрізняється від React і Vue тим, що робить основну роботу під час компіляції. Це означає, що Svelte перетворює компоненти в високоефективний імперативний код, який безпосередньо маніпулює DOM. Такий підхід дозволяє створювати дуже швидкі та легкі додатки, оскільки немає необхідності

віртуального DOM. Svelte має простий синтаксис і забезпечує високу продуктивність, але поки що має меншу спільноту і екосистему, ніж більш відомі фреймворки.

Ember.js – це фреймворк JavaScript, який дозволяє створювати масштабовані односторінкові додатки. Ember.js надає сильну архітектуру і набір вбудованих інструментів, що забезпечують високу продуктивність і зручність розробки. Ember.js використовує двостороннє зв'язування даних, подібно до Angular, і має потужний CLI для управління проектом. Незважаючи на всі переваги, Ember.js може здатися складним для новачків через свою висококонтрольовану структуру і обмеження в гнучкості.

Backbone.js – це легкий фреймворк JavaScript, який надає мінімальний набір інструментів для створення односторінкових додатків. Backbone.js використовує архітектуру MVC (Model-View-Controller) і забезпечує базову структуру для розробки додатків, включаючи маршрутизацію, моделі та колекції. Основною перевагою Backbone.js є його легкість і простота, але для більш складних проектів можуть знадобитися додаткові бібліотеки.

Bootstrap – це фреймворк для розробки адаптивного веб-дизайну, який включає набір готових компонентів CSS і JavaScript. Bootstrap дозволяє швидко створювати адаптивні і стильні вебсайти, які добре виглядають на різних пристроях. Використання Bootstrap значно прискорює процес розробки інтерфейсу користувача, але обмежує гнучкість дизайну, оскільки компоненти часто виглядають стандартно.

Next.js – це фреймворк, побудований на базі React, який надає можливості серверного рендерингу та статичної генерації. Next.js спрощує створення масштабованих і SEO-оптимізованих веб-додатків, забезпечуючи відмінну продуктивність і зручність розробки. Next.js дозволяє розробникам використовувати всі переваги React з додатковими функціями, такими як автоматичне розділення коду та інтеграція з API.

Gatsby – це статичний генератор сайтів, побудований на базі React. Gatsby дозволяє створювати швидкі і SEO-оптимізовані вебсайти, використовуючи дані з різних джерел, таких як CMS, API або файлові системи. Основною перевагою Gatsby є його продуктивність і можливість попереднього рендерингу сторінок, що забезпечує миттєве завантаження контенту для користувачів.

Таким чином, існує безліч технологій і фреймворків для створення вебсайтів, кожен з яких має свої унікальні особливості та переваги. Вибір оптимальної технології залежить від конкретних вимог проекту, рівня досвіду команди розробників та бажаних функцій вебсайту. Кожен з цих фреймворків може бути ефективно використаний для створення сучасних, інтерактивних і продуктивних веб-додатків.

2.3. Вибір платформи та обґрунтування

Вибір платформи для розробки вебсайту є важливим кроком, який визначає майбутню продуктивність, масштабованість, зручність у використанні та можливості подальшого розвитку проекту. Платформа повинна відповідати вимогам проекту, враховувати потреби цільової аудиторії та забезпечувати зручність для розробників. У випадку створення вебсайту для студії постпродакшену SD Production вибір був зроблений на користь використання комбінації технологій, включаючи HTML, CSS, JavaScript і React, а також Python для бекенду. Ось обґрунтування цього вибору.

HTML, CSS і JavaScript є фундаментальними технологіями для створення будь-якого вебсайту. HTML забезпечує структуру вебсторінок, CSS відповідає за їх зовнішній вигляд, а JavaScript додає інтерактивність та динамічність. Ці технології є стандартами в веброзробці і мають широку підтримку серед усіх веббраузерів.

HTML5 і CSS3 надають додаткові можливості, включаючи підтримку мультимедійних елементів, семантичних тегів, анімацій та адаптивного дизайну.

JavaScript дозволяє створювати складні взаємодії на стороні клієнта та інтегруватися з різними бібліотеками і фреймворками для покращення функціональності вебсайту.

React був обраний як основний фреймворк для розробки фронтенду завдяки його продуктивності, гнучкості та великій спільноті розробників. React використовує віртуальний DOM, що значно підвищує швидкість оновлення інтерфейсу користувача і забезпечує плавну роботу вебсайту. Компонентний підхід React дозволяє створювати багаторазові компоненти, які можуть бути легко інтегровані в різні частини вебсайту, що спрощує підтримку та розширення функціональності.

React також підтримує SSR, що покращує SEO-оптимізацію та швидкість завантаження сторінок. Це особливо важливо для вебсайту студії постпродакшену, оскільки забезпечує кращу видимість у пошукових системах та швидкий доступ до контенту для користувачів. Крім того, React має широку екосистему бібліотек і інструментів, таких як Redux для управління станом, React Router для маршрутизації та Material-UI для створення стильних інтерфейсів.

Для реалізації бекенду був обраний Python, оскільки він є потужною, гнучкою та легкою у використанні мовою програмування. Python широко використовується для розробки веб-додатків завдяки своїй читабельності, великій кількості бібліотек і фреймворків, а також активній спільноті розробників. У випадку SD Production Python забезпечує обробку даних, надісланих користувачами через форми на вебсайті, та інтеграцію з Telegram-ботом, який надсилає повідомлення адміністраторам вебсайту про нові запити.

Для реалізації бекенду на Python використовувався фреймворк Flask. Flask є легким і гнучким фреймворком, який дозволяє швидко створювати веб-додатки і API. Flask забезпечує мінімалістичний підхід до розробки, що

дозволяє розробникам вибирати необхідні компоненти і модулі для свого проекту. Flask також підтримує інтеграцію з різними базами даних і зовнішніми сервісами, що робить його ідеальним вибором для створення бекенду вебсайту SD Production.

На вебсайті SD Production користувачі можуть залишати свої контактні дані та запити через спеціальні форми. Ці дані надсилаються на сервер, де обробляються за допомогою Flask і зберігаються у базі даних. Flask забезпечує швидку та ефективну обробку цих запитів, що дозволяє оперативно реагувати на запити клієнтів. Окрім цього, на вебсайті було налаштовано інтеграцію з Telegram-ботом. Це означає, що коли користувач залишає запит на вебсайті, відповідне повідомлення автоматично надсилається в Telegram-бот, де його можна переглянути та відповісти клієнту. Така інтеграція забезпечує зручність для адміністратора вебсайту, оскільки всі запити клієнтів зберігаються в одному місці і можуть бути оброблені швидко і ефективно.

Вибір комбінації HTML, CSS, JavaScript, React та Python дозволяє створити сучасний, продуктивний і масштабований вебсайт, який відповідає всім вимогам проекту. Ці технології добре інтегруються між собою, забезпечуючи плавну роботу як на стороні клієнта, так і на стороні сервера. Використання цих технологій також спрощує процес розробки, оскільки дозволяє використовувати єдиний стек технологій для всього проекту.

HTML, CSS та JavaScript забезпечують створення базової структури, стилів та інтерактивності вебсайту, тоді як React додає продуктивність і гнучкість у створенні користувацьких інтерфейсів. Python забезпечує обробку серверних запитів та інтеграцію з зовнішніми сервісами, такими як Telegram-бот. Така комбінація технологій забезпечує високу продуктивність, зручність у використанні та можливість подальшого розвитку проекту.

Таким чином, вибір платформи для створення вебсайту студії постпродакшену SD Production був обґрунтований вимогами проекту,

необхідністю забезпечення високої продуктивності, зручності у використанні та можливості подальшого розвитку. Використання комбінації HTML, CSS, JavaScript, React та Python дозволяє створити сучасний, ефективний та масштабований вебсайт, який відповідає всім вимогам клієнтів та забезпечує високу якість обслуговування.

РОЗДІЛ 3. Процес створення та розробка вебсайту

3.1. Розробка концепції

Процес створення вебсайту для студії постпродакшену SD Production розпочинається з розробки концепції, яка визначає основні цілі, аудиторію, функціональність та загальну структуру майбутнього вебсайту. Концепція вебсайту є ключовим етапом, оскільки вона задає напрямок для всіх наступних кроків у процесі розробки.

Визначення цілей: Першим кроком у розробці концепції є визначення основних цілей вебсайту. Для студії постпродакшену такими цілями можуть бути:

- Представлення послуг студії.
- Демонстрація портфоліо з прикладами виконаних робіт.
- Залучення нових клієнтів та партнерів.
- Підтримка зв'язку з існуючими клієнтами.
- Підвищення впізнаваності бренду та репутації студії.

Чітке визначення цілей дозволяє створити вебсайт, який буде ефективно виконувати бізнес-завдання і задовольняти потреби користувачів.

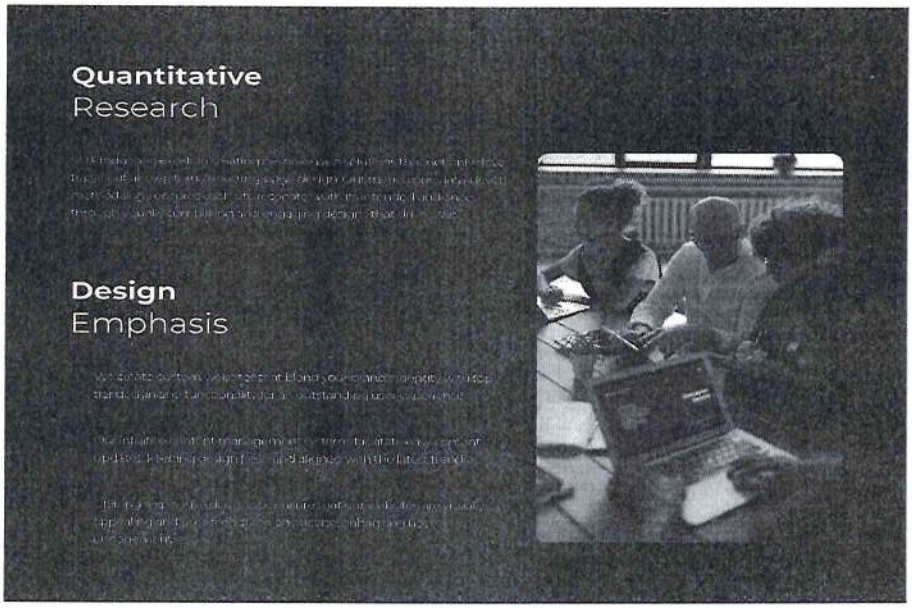

Рис. 3.1. Визначеня цілей SD Production

Аналіз цільової аудиторії: Другим важливим кроком є визначення цільової аудиторії вебсайту. Для студії постпродакшену основною аудиторією можуть бути:

- Продюсери та режисери, які шукають послуги постпродакшену для своїх проектів.
- Рекламні агенції та маркетингові компанії.
- Відеооператори та фотографи, які потребують обробки та редагування матеріалів.
- Підприємці та бізнесмени, зацікавлені у створенні рекламних відео.

Розуміння потреб та очікувань цільової аудиторії допомагає створити вебсайт, який буде привабливим і корисним для користувачів.

Рис. 3.2. Детальний аналіз цільової аудиторії

Визначення функціональності: На основі визначених цілей та аналізу аудиторії, наступним кроком є визначення функціональних можливостей вебсайту. Це може включати:

- Інформативну головну сторінку з коротким описом студії та її послуг.

- Детальні сторінки з описом кожної послуги.
- Портфоліо з прикладами виконаних робіт.
- Блог з новинами та статтями про постпродакшен.
- Форма зворотного зв'язку для отримання запитів від клієнтів.
- Інтеграція з соціальними мережами для підвищення охоплення аудиторії.
- Система онлайн-замовлень та розрахунку вартості послуг.

Структура вебсайту: Важливо також визначити загальну структуру вебсайту, яка включає основні розділи та сторінки. Основні розділи можуть включати: Головна сторінка, роботи клієнтів, послуги, команда, футер. Кожна сторінка повинна мати логічну і зрозумілу структуру, що дозволяє користувачам легко знаходити необхідну інформацію.

Візуальна концепція: Розробка візуальної концепції включає визначення стилю, кольорової гами, типографіки та інших елементів дизайну. Візуальна концепція повинна відображати бренд студії, її професіоналізм і якість послуг. Важливо забезпечити консистентність дизайну на всіх сторінках вебсайту, щоб створити привабливий і інтуїтивно зрозумілий інтерфейс.

Технічні вимоги: На завершення, розробка концепції включає визначення технічних вимог до вебсайту, таких як вибір платформи для розробки, бази даних, серверного середовища та інших технічних аспектів. Це забезпечує, що вебсайт буде відповідати всім вимогам продуктивності, безпеки та масштабованості.

Таким чином, розробка концепції є важливим етапом у створенні вебсайту для студії постпродакшену SD Production. Чітке визначення цілей, аудиторії, функціональності та візуальної концепції дозволяє створити ефективний та привабливий вебсайт, який відповідає всім вимогам клієнтів та забезпечує високу якість обслуговування.

3.2. Прототипування

Після визначення концепції вебсайту наступним кроком є прототипування, яке дозволяє візуалізувати структуру і функціональність майбутнього сайту. Прототипування є важливим етапом, оскільки воно допомагає виявити можливі проблеми на ранніх стадіях розробки і забезпечити, що всі вимоги будуть виконані.

Створення каркасів (wireframes): Перший крок у прототипуванні – створення каркасів. Каркаси – це схематичні зображення сторінок вебсайту, які показують розташування основних елементів, таких як заголовки, тексти, зображення, кнопки та форми. Каркаси не містять деталей дизайну, але допомагають зрозуміти, як буде організований контент і як користувачі будуть взаємодіяти з сайтом.

Інтерактивні прототипи: Наступним кроком є створення інтерактивних прототипів, які дозволяють моделювати поведінку вебсайту. Інтерактивні прототипи можуть включати переходи між сторінками, інтерактивні елементи, такі як випадаючі меню або модальні вікна. Прототипи створюються за допомогою інструментів для прототипування, таких як Figma, Sketch або Adobe XD. Інтерактивні прототипи дозволяють тестувати користувацький досвід і вносити зміни на ранніх етапах розробки.

Збір зворотного зв'язку: Після створення прототипів важливо отримати зворотний зв'язок від зацікавлених сторін, включаючи клієнтів, розробників та дизайнерів. Зворотний зв'язок допомагає виявити можливі проблеми та покращити прототип до початку розробки. Це може включати тестування юзабіліті з реальними користувачами, які виконують різні завдання на прототипі і надають свої враження та пропозиції.

Внесення змін: На основі отриманого зворотного зв'язку в прототип вносяться необхідні зміни. Це може включати коригування розташування елементів, зміну текстів, додавання або видалення функцій. Внесення змін на

етапі прототипування є значно дешевшим і швидшим, ніж у пізніших етапах розробки.

Підготовка до дизайну: Після затвердження прототипів вони використовуються як основа для створення остаточного дизайну вебсайту. Прототипи допомагають дизайнерам зрозуміти структуру сайту і забезпечити, що всі вимоги будуть враховані в дизайні. Прототипи також слугують важливим інструментом для комунікації між дизайнерами, розробниками та іншими членами команди.

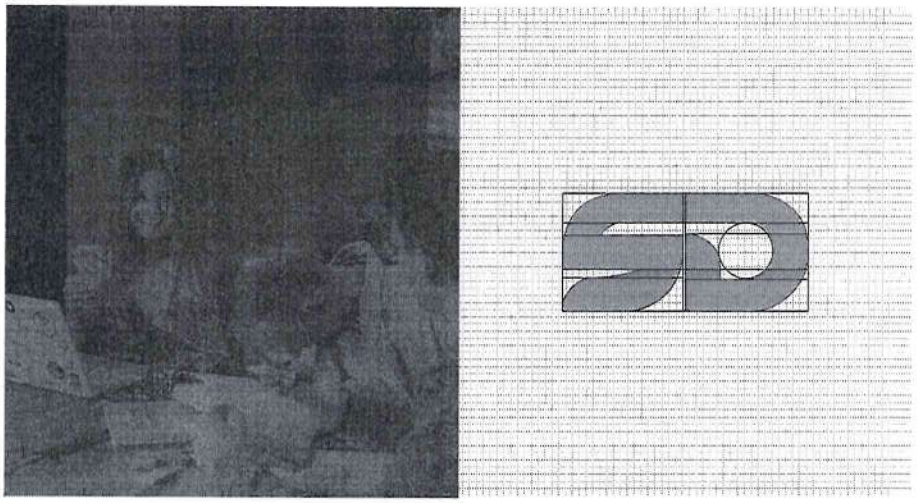

Рис. 3.2. Wireframe (каркас) дизайну

3.3. Дизайн інтерфейсу

Після завершення прототипування наступним етапом є дизайн інтерфейсу, який включає створення візуального вигляду вебсайту. Дизайн інтерфейсу є важливим етапом, оскільки він визначає, як вебсайт буде виглядати і як користувачі будуть з ним взаємодіяти.

Визначення візуального стилю: На основі затвердженої концепції розробляється візуальний стиль вебсайту. Це включає вибір кольорової гами, типографіки, стилю кнопок, іконок та інших елементів дизайну. Візуальний стиль повинен відображати бренд студії, її професіоналізм і якість послуг. Важливо забезпечити консистентність дизайну на всіх сторінках вебсайту.

Створення макетів (mockups): Наступним кроком є створення макетів сторінок, які відображають остаточний вигляд вебсайту. Макети містять всі деталі дизайну, включаючи кольори, шрифти, зображення та інші елементи. Макети створюються за допомогою інструментів для дизайну, таких як Figma, Sketch або Adobe XD. Макети дозволяють побачити, як буде виглядати кожна сторінка вебсайту в реальному вигляді.

Адаптивний дизайн: Важливо забезпечити, що дизайн вебсайту буде адаптивним, тобто коректно відображатиметься на різних пристроях і екранах, включаючи десктопи, планшети і смартфони. Для цього використовуються медіа-запити CSS, які дозволяють змінювати стилі залежно від розміру екрану. Адаптивний дизайн забезпечує зручність використання вебсайту на будь-якому пристрої і підвищує задоволення користувачів.

Тестування дизайну: Після створення макетів важливо провести тестування дизайну, щоб переконатися, що він відповідає всім вимогам і забезпечує зручність використання. Це може включати тестування юзабіліті з реальними користувачами, які оцінюють візуальний вигляд і функціональність вебсайту. Тестування дозволяє виявити можливі проблеми і внести необхідні зміни до дизайну перед початком розробки.

Підготовка до розробки: Після затвердження дизайну інтерфейсу всі макети передаються розробникам для реалізації. Важливо забезпечити тісну співпрацю між дизайнерами і розробниками, щоб гарантувати точну відповідність дизайну під час розробки. Макети слугують детальним керівництвом для розробників і допомагають уникнути непорозумінь під час реалізації проекту.

Таким чином, прототипування і дизайн інтерфейсу є критичними етапами у створенні вебсайту для студії постпродакшену SD Production. Ці етапи забезпечують, що всі вимоги будуть виконані, а вебсайт буде привабливим, зручним у використанні і відповідати високим стандартам якості.

Рис. 3.3. Варіант дизайну сайту SD

3.4. Аналіз отриманого результату

Після завершення процесу розробки вебсайту для студії постпродакшену SD Production необхідно провести детальний аналіз отриманого результату. Це дозволяє оцінити, наскільки успішно були досягнуті поставлені цілі, виявити можливі проблеми та визначити напрями для подальшого вдосконалення. Аналіз отриманого результату включає кілька важливих аспектів.

Функціональність вебсайту: Першим кроком є перевірка функціональності вебсайту. Важливо переконатися, що всі заплановані функції працюють коректно. Це включає тестування навігації, форм зворотного зв'язку, інтеграції з соціальними мережами, портфоліо, блогу та інших елементів. Функціональність вебсайту має відповідати технічному завданню і задовольняти потреби користувачів.

Користувацький досвід (UX): Оцінка користувацького досвіду є критично важливою для успіху вебсайту. Це включає аналіз зручності використання, доступності та загального враження від взаємодії з вебсайтом. Залучення реальних користувачів для проведення юзабіліті-тестів допомагає виявити слабкі місця та можливі проблеми у дизайні та функціональності. Відгуки

користувачів можуть стати цінним джерелом інформації для покращення вебсайту.

Продуктивність вебсайту: Швидкість завантаження сторінок є важливим фактором, що впливає на задоволеність користувачів і SEO-рейтинги. Аналіз продуктивності включає вимірювання часу завантаження сторінок, обробки запитів і загальної продуктивності вебсайту під навантаженням. Для цього використовуються інструменти, такі як Google PageSpeed Insights, GTmetrix та інші. Оптимізація продуктивності може включати зменшення розміру зображень, використання кешування, мінімізацію CSS та JavaScript файлів.

SEO-оптимізація: Оцінка SEO-оптимізації вебсайту дозволяє визначити, наскільки ефективно вебсайт може залучати органічний трафік з пошукових систем. Це включає аналіз ключових слів, метатегів, структури URL, внутрішньої та зовнішньої лінковки. Важливо також перевірити наявність адаптивного дизайну, який впливає на ранжування у мобільних пошукових запитах. Використання інструментів, таких як Google Search Console та SEMrush, допомагає оцінити стан SEO-оптимізації і виявити можливі напрями для покращення.

Безпека вебсайту: Безпека є важливим аспектом, який необхідно врахувати при аналізі отриманого результату. Перевірка наявності SSL сертифікатів, захисту від SQL-ін'єкцій, XSS атак та інших вразливостей забезпечує захист даних користувачів та стабільну роботу вебсайту. Регулярні аудити безпеки та використання спеціальних інструментів для моніторингу допомагають виявити та усунути потенційні загрози.

Відповідність цілям проекту: Важливо оцінити, наскільки вебсайт відповідає початковим цілям проекту. Це включає аналіз задоволеності клієнтів, залучення нових клієнтів, підвищення впізнаваності бренду та інших бізнес-цілей. Використання аналітичних інструментів, таких як Google

Analytics, допомагає отримати детальну інформацію про поведінку користувачів, конверсії та інші важливі метрики.

Зворотний зв'язок від клієнтів та користувачів: Збір та аналіз зворотного зв'язку від клієнтів та користувачів дозволяє виявити їхні враження від вебсайту, сильні та слабкі сторони. Це може включати опитування, анкети, відгуки та інші методи збору даних. Зворотний зв'язок є цінним джерелом інформації для подальшого вдосконалення вебсайту.

Напрями для покращення: На основі аналізу отриманого результату важливо визначити напрями для подальшого покращення вебсайту. Це може включати оптимізацію продуктивності, покращення юзабіліті, вдосконалення SEO-оптимізації, підвищення безпеки та інші аспекти. Регулярне вдосконалення вебсайту допомагає підтримувати його актуальність і ефективність у задоволенні потреб користувачів.

Таким чином, аналіз отриманого результату є важливим етапом у процесі створення вебсайту для студії постпродакшену SD Production. Він дозволяє оцінити, наскільки успішно були досягнуті поставлені цілі, виявити можливі проблеми та визначити напрями для подальшого вдосконалення. Це забезпечує високу якість вебсайту та задоволення потреб клієнтів і користувачів.

Висновок

У процесі створення вебсайту для студії постпродакшену SD Production було виконано значний обсяг роботи, спрямований на досягнення високих стандартів якості, продуктивності та задоволення потреб користувачів. У ході розробки були враховані всі важливі аспекти, починаючи від розробки концепції і закінчуючи фінальним аналізом отриманого результату. Висновки, зроблені на основі цього процесу, можна розділити на кілька ключових напрямків.

Однією з основних цілей створення вебсайту для студії SD Production було представлення послуг студії, демонстрація портфоліо, залучення нових клієнтів та підвищення впізнаваності бренду. Ці цілі були успішно досягнуті завдяки ретельно спланованій та реалізованій стратегії. Вебсайт містить детальний опис послуг студії, приклади виконаних робіт у розділі портфоліо, а також інтерактивні елементи, які полегшують комунікацію з клієнтами.

Розробка візуальних компонентів та дизайн інтерфейсу були виконані з урахуванням сучасних тенденцій і вимог користувачів. Використання інструментів, таких як Figma, дозволило створити привабливий та інтуїтивно зрозумілий дизайн, який забезпечує позитивний користувацький досвід. Вебсайт адаптивний і коректно відображається на різних пристроях, що підвищує зручність використання для широкого кола користувачів.

Під час розробки було приділено значну увагу продуктивності вебсайту. Оптимізація зображень, використання кешування та мінімізація CSS і JavaScript файлів забезпечили швидке завантаження сторінок, що позитивно впливає на задоволеність користувачів. Крім того, були впроваджені заходи для забезпечення безпеки вебсайту, включаючи використання SSL сертифікатів, захист від SQL-ін'єкцій та інших вразливостей.

Для забезпечення високої видимості вебсайту в пошукових системах була проведена комплексна SEO-оптимізація. Це включало аналіз і підбір ключових слів, оптимізацію метатегів, структури URL та внутрішньої лінковки. Завдяки

цьому вебсайт має високий рейтинг у пошукових системах, що сприяє залученню органічного трафіку та нових клієнтів.

Вебсайт студії SD Production був інтегрований з Telegram-ботом, що забезпечує зручність комунікації з клієнтами. Користувачі можуть залишати свої запити через спеціальні форми на вебсайті, а повідомлення автоматично надсилаються до Telegram-бота. Це дозволяє адміністраторам вебсайту оперативно реагувати на запити клієнтів, що підвищує якість обслуговування та задоволеність клієнтів.

Процес розробки вебсайту передбачав не тільки створення функціонального продукту, але й планування подальшого вдосконалення та підтримки. Регулярні оновлення контенту, проведення юзабіліті-тестів та оптимізація продуктивності забезпечують актуальність вебсайту та відповідність його сучасним вимогам. Це дозволяє підтримувати високий рівень задоволеності користувачів та ефективність вебсайту у довгостроковій перспективі.

Залучення клієнтів до процесу розробки та отримання зворотного зв'язку дозволило створити вебсайт, який максимально відповідає їхнім потребам і очікуванням. Опитування, анкети та відгуки стали важливим джерелом інформації для покращення функціональності та дизайну вебсайту. Такий підхід забезпечив високий рівень задоволеності клієнтів та позитивні відгуки про роботу студії.

Використання сучасних технологій, таких як React для фронтенду та Python для бекенду, дозволило створити продуктивний та масштабований вебсайт. Ці технології забезпечують високу гнучкість у розробці та можливість швидко вносити зміни і додавати новий функціонал. Це дозволяє студії SD Production адаптуватися до змін у ринку та забезпечувати високий рівень обслуговування клієнтів.

Особлива увага була приділена забезпеченню доступності вебсайту для користувачів з різними потребами. Використання семантичної розмітки, альтернативного тексту для зображень та дотримання стандартів доступності допомогло створити інклюзивний вебсайт. Це підвищує доступність контенту для ширшого кола користувачів і демонструє відповідальне ставлення студії до всіх відвідувачів вебсайту.

Створення вебсайту для студії SD Production мало значний позитивний вплив на бізнес. Вебсайт став ефективним інструментом для залучення нових клієнтів, підвищення впізнаваності бренду та підтримки зв'язку з існуючими клієнтами. Це сприяло зростанню замовлень на послуги студії та підвищенню її репутації на ринку постпродакшену.

Таким чином, процес створення вебсайту для студії постпродакшену SD Production був успішно реалізований завдяки ретельному плануванню, використанню сучасних технологій та тісній співпраці з клієнтами. Вебсайт відповідає високим стандартам якості, забезпечує позитивний користувацький досвід та сприяє досягненню бізнес-цілей студії. Подальше вдосконалення та підтримка вебсайту дозволять підтримувати його актуальність і ефективність у довгостроковій перспективі.

Список використаних джерел

1. Офіційний сайт React [Електронний ресурс] - Режим доступу до ресурсу: https://reactjs.org/
2. React Вікі [Електронний ресурс] - Режим доступу до ресурсу: https://en.wikipedia.org/wiki/React_(JavaScript_library)
3. Документація React [Електронний ресурс] - Режим доступу до ресурсу: https://reactjs.org/docs/getting-started.html
4. Офіційний сайт Python [Електронний ресурс] - Режим доступу до ресурсу: https://www.python.org/
5. Python Вікі [Електронний ресурс] - Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Python
6. Документація Flask [Електронний ресурс] - Режим доступу до ресурсу: https://flask.palletsprojects.com/en/2.0.x/
7. Основи React [Електронний ресурс] - Режим доступу до ресурсу: https://reactjs.org/tutorial/tutorial.html
8. Стаття про використання React у великих проектах [Електронний ресурс] - Режим доступу до ресурсу: https://medium.com/@dan_abramov/what-is-react-7f94b8d9b2b3
9. Підручник з Python [Електронний ресурс] - Режим доступу до ресурсу: https://docs.python.org/3/tutorial/
10. Стаття про переваги використання Flask для веб-розробки [Електронний ресурс] - Режим доступу до ресурсу: https://realpython.com/tutorials/flask/
11. Інтеграція Telegram-бота з Flask [Електронний ресурс] - Режим доступу до ресурсу: https://www.codementor.io/@ipsjolly/create-a-telegram-bot-using-python-and-flask-byg7x2p26
12. Огляд сучасних тенденцій у веб-розробці [Електронний ресурс] - Режим доступу до ресурсу: https://www.smashingmagazine.com/2021/01/web-development-trends-2021/

13. Керівництво з SEO-оптимізації для розробників [Електронний ресурс] - Режим доступу до ресурсу: https://developers.google.com/search/docs/beginner/seo-starter-guide
14. Документація по безпеці веб-розробки [Електронний ресурс] - Режим доступу до ресурсу: https://owasp.org/www-project-top-ten/
15. Професійний блог з веб-розробки [Електронний ресурс] - Режим доступу до ресурсу: https://css-tricks.com/

Додатки

Додаток А. Дизайн головної сторінки веб сайту

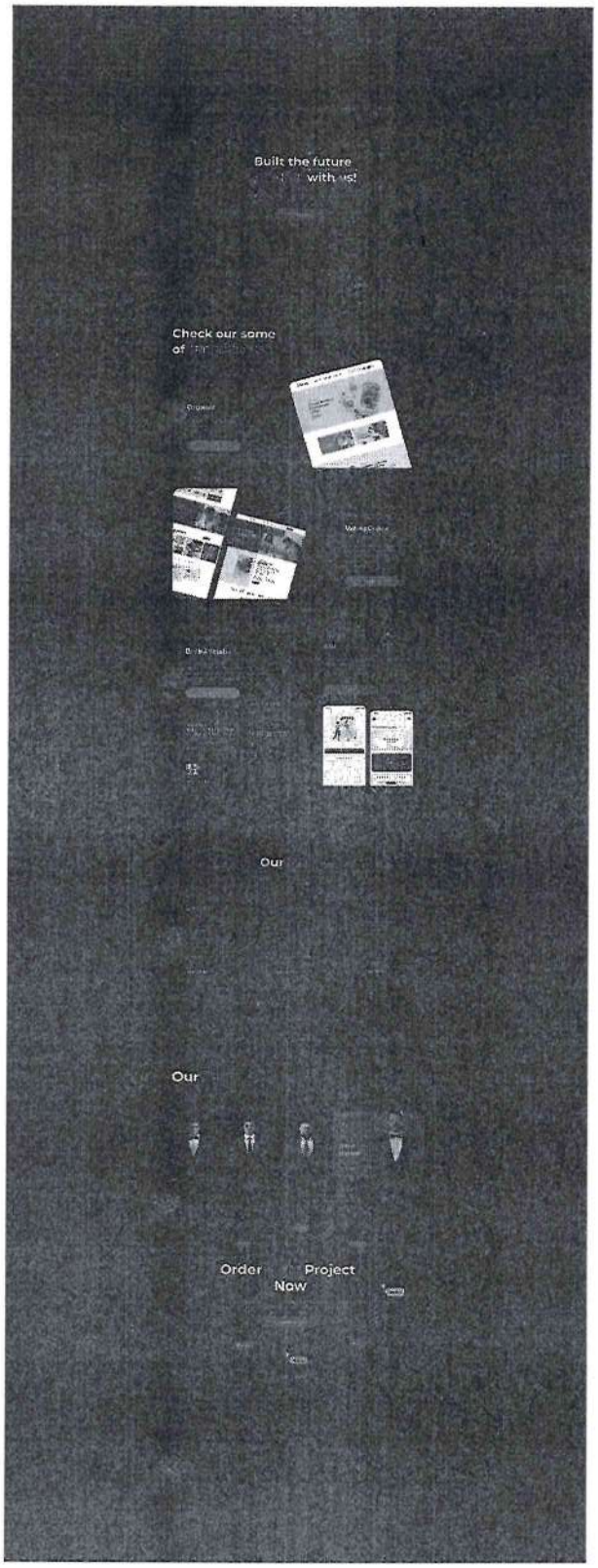

Рис. 1. Дизайн головної SD

Додаток В. Код головної сторінки вебсайту, англійської версії

```
import React, { useState, useEffect, useRef } from "react";
import ScrollTrigger from "react-scroll-trigger";
import AnimationState from "../../helperFunction/animationState/animationState";
import { HeaderEN } from "../../components/header/HeaderEN";
import { FooterEN } from "../../components/footer/FooterEN";
import { SvgHome } from "../../helperFunction/svgHome/svgHome";
import { sliderData } from "../../helperFunction/arrays/sliderData";
import { handleClickTop } from "../../helperFunction/ScrollTop";
import { ImgBWData } from "../../helperFunction/arrays/ImgBWData";
import ImageHero from "../../assets/svg/deviderLine.svg";
import { ConectUsEN } from "../../components/conectUs/conectUsEN";
import Ellipse from "../../assets/svg/Ellipse.svg";
import OrganicImg from "../../assets/img/5093ef757e18607302ebe57407fd1129.png";
import WhiteImg1 from "../../assets/img/page2.png";
import BriliniImg1 from "../../assets/img/page-3.png";
import BriliniImg2 from "../../assets/img/page-4.png";
import AbiImg1 from "../../assets/img/page4.png";
import AbiImg2 from "../../assets/img/page4-1.png";
import Prev from "../../assets/svg/left1.svg";
import Next from "../../assets/svg/right1.svg";
import Social from "../../assets/svg/teams/socials.svg";
import {
 ButtonHeader,
 HeroBackground,
 HeroBlock,
 HeroContend,
 SectionHero,
 ImgHeroDiv,
 ImgOurTeamDiv,
 ImgHero,
 ContainerPC,
 ContainerMobile,
```

Project,

ProjectsCards,

ProjectsFragment1,

ProjectsFragmentInfo1,

ButtonProject1,

ProjectImage1Container,

ProjectImage3Container,

ProjectImage1,

ProjectContainer,

ProjectsFragment2,

ProjectsFragmentInfo2,

ButtonProject2,

ProjectImageContainer1,

ProjectImage4Container,

ProjectImage21,

ProjectImage22,

ProjectsFragmentInfo3,

ProjectsFragment3,

ButtonProject3,

ProjectImage31,

ProjectImage32,

ContainerMoreProject,

ProjectContainer2,

ProjectsFragment4,

ProjectsFragmentInfo4,

ButtonProject4,

ProjectImage41,

ProjectImage42,

Services,

ServicesCards,

ToEmailLink,

ServicesFragment,

ServiceHeader1,

```
  ServiceButton,
  ServiceHeader2,
  ServiceHeader3,
  OurTeam,
  SwiperButtonContainer,
  SwiperButtonPrev,
  SwiperButtonNext,
  SwiperImgContainer,
  SliderName,
  MySwiperBox,
  SocialImg,
  SwiperWrapper,
  SwiperSlide,
  Contend,
  ImageBW,
  ImageC,
  ContainerImageBW,
  ContainerImg,
  ServicesCardsContainer,
  ServicesFragment3,
  ServiceButton3,
  ProjectContainerImg,
} from "./../styles/home.styled";
import { ModalkaEN } from "../../components/modalka/ModalkaEN";

export const HomeEN = () => {
  const [currentSlide, setCurrentSlide] = useState(0);
  const [displayImg, setDisplayImg] = useState(0);
  // const [displayImgPrev, setDisplayImgPrev] = useState(1);
  const [imgArr, setImgArr] = useState([]);
  const [openModal, setOpenModal] = useState(false);
  const [isOpenMoreProject, setIsOpenProject] = useState(false);
  const projectsRef = useRef(null);
```

```
const servicesRef = useRef(null);
const teamRef = useRef(null);
const conectRef = useRef(null);

const {
  isVisibleOne,
  isVisibleTwo,
  isVisibleThree,
  isVisibleFour,
  isVisibleFive,
  isVisibleSix,
  isVisibleSeven,
  isVisibleEight,
  handleEnterViewportOne,
  handleEnterViewportTwo,
  handleEnterViewportThree,
  handleEnterViewportFour,
  handleEnterViewportFive,
  handleEnterViewportSix,
  handleEnterViewportSeven,
  handleEnterViewportEight,
} = AnimationState();

useEffect(() => {
  if (window.location.hash) {
    const element = document.getElementById(
      window.location.hash.substring(1)
    );
    if (element) {
      element.scrollIntoView({
        behavior: "smooth",
      });
    }
```

```
  }
 }, []);

 let ImgBWDataExtended = ImgBWData.slice(5, 8);

 useEffect(() => {
  setImgArr(ImgBWDataExtended);
  // eslint-disable-next-line
 }, []);
 const [windowWidth, setWindowWidth] = useState(window.innerWidth);

 useEffect(() => {
  const handleResize = () => {
   setWindowWidth(window.innerWidth);
  };

if (windowWidth >= 768) {
   setIsOpenProject(true);
  }

  window.addEventListener("resize", handleResize);

  return () => {
   window.removeEventListener("resize", handleResize);
  };
 }, [windowWidth]);

 const addImg = () => {
  if (displayImg >= 7) {
   setDisplayImg(-1);
  }
  setDisplayImg((preveCount) => preveCount + 1);
  imgArr.shift();
```

```
  setImgArr((prevArr) => [...prevArr, ImgBWData[displayImg]]);
 };

 const nextImg = () => {
  if (displayImg <= 0) {
   setDisplayImg(8);
  }
  setDisplayImg((preveCount) => preveCount - 1);
  imgArr.pop();
  setImgArr((prevArr) => [ImgBWData[displayImg], ...prevArr]);
  console.log(displayImg);
 };

 const nextSlide = () => {
  setCurrentSlide((prevSlide) =>
   prevSlide === 0 ? sliderData.length - 1 : prevSlide - 1
  );
  nextImg();
 };

 const prevSlide = () => {
  setCurrentSlide((prevSlide) => (prevSlide + 1) % sliderData.length);
  addImg();
 };

 const hadnleOpenProject = () => {
  setIsOpenProject(true);
 };

 const handleOpenModal = () => {
  setOpenModal(true);
  document.body.style.position = "fixed";
 };
```

```
return (
 <>

  {openModal && <ModalkaEN setOpenModal={setOpenModal} />}
  <HeaderEN name={"/de"}/>
  <ScrollTrigger onEnter={handleEnterViewportOne}>
   <SectionHero className={${isVisibleOne ? "visible" : ""}}>
    <HeroBackground src={Ellipse} alt="" />
    <HeroBlock>
     <HeroContend>
      <ContainerPC>
       <h2>
        Built the future <span>together </span> with us!
       </h2>
       <img src="../img/select.svg" alt="" />
       <p>Your goals are our execution</p>
      </ContainerPC>
      <ContainerMobile>
       <h2>
        Built the <span> future </span>together
       </h2>
      </ContainerMobile>
     </HeroContend>
     <ButtonHeader onClick={handleOpenModal}>Lets go!</ButtonHeader>
    </HeroBlock>
   </SectionHero>
  </ScrollTrigger>
  <ImgHero src={ImageHero} alt="" />

<Project id="projects">
    <div ref={projectsRef}>
     <ScrollTrigger onEnter={handleEnterViewportTwo}>
```

```
<ContainerPC>
 <h2 className={${isVisibleTwo ? "visible" : ""}}>
  Check our some of <span>our projects</span>
 </h2>
</ContainerPC>
<ContainerMobile>
 <h2 className={${isVisibleTwo ? "visible" : ""}}>
  Some of our <span> projects</span>
 </h2>
</ContainerMobile>
</ScrollTrigger>
<ProjectsCards>
 <ScrollTrigger onEnter={handleEnterViewportThree}>
  <ProjectContainer
   className={${isVisibleThree ? "visible" : ""}}
  >
   <ProjectsFragment1>
    <ProjectsFragmentInfo1>
     <h4>Online Shop</h4>
     <h3>Organic</h3>
     {windowWidth > 1200 ? (
      <p>
       Online store with organic fruits, vegetables and
       berries. A England farmer wanted to sell organically
       grown produce.
      </p>
     ) : (
      <p>
       Online store with organic fruits, vegetables and
       berries.
      </p>
     )}
     <ButtonProject1
```

```
        to="/en/organicShop"
        onClick={handleClickTop}
       >
        View
       </ButtonProject1>
      </ProjectsFragmentInfo1>
     </ProjectsFragment1>
     <ProjectImage1Container>
      <ProjectImage1 src={OrganicImg} />
     </ProjectImage1Container>
    </ProjectContainer>
  </ScrollTrigger>
  <ScrollTrigger onEnter={handleEnterViewportFour}>
   <ProjectsFragment2
    className={${isVisibleFour ? "visible" : ""}}
   >
    <ProjectImageContainer1>
     <ProjectImage21 src={WhiteImg1} />
     <ProjectImage22 src={WhiteImg1} />
    </ProjectImageContainer1>
    <ProjectsFragmentInfo2>
     <h4>Business website</h4>
     <h3>WhiteCollor</h3>
     <ContainerPC>
      {windowWidth > 1200 ? (
       <p>
        Develop and implement Agilic methods in the strategic
        management of a corporation to ensure flexibility and
        adaptation to a rapidly changing business environment.
       </p>
      ) : (
       <p>
        Develop and implement Agilic methods in the strategic
```

```
            management
          </p>
        )}
      </ContainerPC>
      <ContainerMobile>
        <p>Adaptive Strategic Agile Management.</p>
      </ContainerMobile>
      <ButtonProject2 to="/en/whiteCollar" onClick={handleClickTop}>
        View
      </ButtonProject2>
    </ProjectsFragmentInfo2>
  </ProjectsFragment2>
</ScrollTrigger>
{!isOpenMoreProject ? (
  <ContainerMoreProject>
    <button onClick={hadnleOpenProject}>
      Show more
      <SvgHome />
    </button>
  </ContainerMoreProject>
) : (
  <ProjectContainer2>
    <ProjectContainerImg>
      <ScrollTrigger onEnter={handleEnterViewportFive}>
        <ProjectsFragment3
          className={${isVisibleFive ? "visible" : ""}}
        >
          <ProjectImage3Container>
            <ProjectImage31 src={BriliniImg1} />
            <ProjectImage32 src={BriliniImg2} />

</ProjectImage3Container>
          <ProjectsFragmentInfo3>
```

```
        <h4>Landing page</h4>
        <h3>Brilini Studio</h3>
        <ContainerPC>
         {windowWidth > 1200 ? (
          <p>
           Studio of interior designers specializing in the
           furnishing and design of rooms, apartments and
           houses in a modern and austere style.
          </p>
         ) : (
          <p>
           Studio of interior designers specializing in the
           furnishing and design of rooms
          </p>
         )}
        </ContainerPC>
        <ContainerMobile>
         <p>Modern austere interior design studio.</p>
        </ContainerMobile>
        <ButtonProject3
         to="/en/briliniStudio"
         onClick={handleClickTop}
        >
         View
        </ButtonProject3>
       </ProjectsFragmentInfo3>
      </ProjectsFragment3>
     </ScrollTrigger>
    </ProjectContainerImg>
    <ScrollTrigger onEnter={handleEnterViewportSix}>
     <ProjectsFragment4
      className={${isVisibleSix ? "visible" : ""}}
     >
```

```
            <ProjectImage4Container>
              <ProjectImage41 src={AbiImg1} />
              <ProjectImage42 src={AbiImg2} />
            </ProjectImage4Container>
            <ProjectsFragmentInfo4>
              <h4>Mobile App</h4>
              <h3>Abi</h3>
              {windowWidth > 1200 ? (
                <p>
                  Mobile application for creating presentations with
                  ready-made templates based on artificial intelligence.
                  The application is available on Play Market and App
                  Store.
                </p>
              ) : (
                <p>
                  Mobile application for creating presentations with
                  ready-made templates
                </p>
              )}
              <ButtonProject4 to="en/abi" onClick={handleClickTop}>
                View
              </ButtonProject4>
            </ProjectsFragmentInfo4>
          </ProjectsFragment4>
        </ScrollTrigger>
      </ProjectContainer2>
    )}
  </ProjectsCards>
</div>
</Project>

<ImgHeroDiv>
```

```
      <ImgHero src={ImageHero} alt="" />
    </ImgHeroDiv>
<Services id="services">
      <div ref={servicesRef}>
       <ScrollTrigger onEnter={handleEnterViewportSeven}>
        <h2 className={${isVisibleSeven ? "visible" : ""}}>
         Our <span>Services</span>
        </h2>
       </ScrollTrigger>
       <ScrollTrigger onEnter={handleEnterViewportEight}>
        <ServicesCards className={${isVisibleEight ? "visible" : ""}}>
         <ServicesCardsContainer>
          <ToEmailLink onClick={handleOpenModal}>
           <ServicesFragment>
            <ServiceHeader1>Design</ServiceHeader1>
            <ul>
             <li>Motion Design</li>
             <li>Logotype</li>
             <li>Video Editing</li>
             <li>3D models</li>
             <li>Advertising Banners</li>
            </ul>
            <ServiceButton>
             <p>Order service</p>
             <svg
              xmlns="http://www.w3.org/2000/svg"
              width="16"
              height="10"
              viewBox="0 0 16 10"
              fill="none"
             >
              <path
               d="M10.8689 1L15 4.76471M15 4.76471L10.8689 9M15 4.76471H1"
```

```
          stroke="#FCFCFC"
          strokeWidth="2"
          strokeLinecap="round"
          strokeLinejoin="round"
         />
        </svg>
       </ServiceButton>
      </ServicesFragment>
    </ToEmailLink>
    <ToEmailLink onClick={handleOpenModal}>
     <ServicesFragment>
      <ServiceHeader2>Development</ServiceHeader2>
      <ul>
       <li>Landing Page</li>
       <li>Corporate Website</li>
       <li>Online Shop</li>
       <li>Mobile App</li>
       <li>Redesign Website</li>
      </ul>
      <ServiceButton>
       <p>Order service</p>
       <svg
        xmlns="http://www.w3.org/2000/svg"
        width="16"
        height="10"
        viewBox="0 0 16 10"
        fill="none"
       >
        <path
         d="M10.8689 1L15 4.76471M15 4.76471L10.8689 9M15 4.76471H1"
         stroke="#FCFCFC"
         strokeWidth="2"
         strokeLinecap="round"
```

```
          strokeLinejoin="round"
         />
        </svg>
       </ServiceButton>
      </ServicesFragment>
     </ToEmailLink>
    </ServicesCardsContainer>
    <ToEmailLink onClick={handleOpenModal}>
     <ServicesFragment3>
      <ServiceHeader3>Update</ServiceHeader3>
      <ul>
       <li>SEO optimisation</li>
       <li>Multilingual</li>
      </ul>
      <ServiceButton3>
       <p>Order service</p>
       <svg
        xmlns="http://www.w3.org/2000/svg"
        width="16"
        height="10"
        viewBox="0 0 16 10"
        fill="none"
       >
        <path
         d="M10.8689 1L15 4.76471M15 4.76471L10.8689 9M15 4.76471H1"
         stroke="#FCFCFC"
         strokeWidth="2"
         strokeLinecap="round"
         strokeLinejoin="round"
        />
       </svg>
      </ServiceButton3>
    </ServicesFragment3>
```

```
            </ToEmailLink>
          </ServicesCards>
        </ScrollTrigger>
      </div>
    </Services>
<ImgOurTeamDiv>
      <ImgHero src={ImageHero} alt="" />
    </ImgOurTeamDiv>
    <div ref={teamRef}>
     <OurTeam id="team">
      <SliderName>
       <h2>
        Our <span>Team</span>
       </h2>
       <SwiperButtonContainer>
        <SwiperImgContainer>
         <SwiperButtonPrev src={Prev} onClick={nextSlide} />
        </SwiperImgContainer>
        <SwiperImgContainer>
         <SwiperButtonNext src={Next} onClick={prevSlide} />
        </SwiperImgContainer>
       </SwiperButtonContainer>
      </SliderName>

      <MySwiperBox>
       <SwiperWrapper>
        {sliderData.map((slider, index) => (
         <SwiperSlide
          key={index}
          className={index === currentSlide ? "active" : ""}
         >
          <Contend>
           <SocialImg src={Social} />
```

```
                <h2>{slider.name}</h2>
                <h3>{slider.description}</h3>
              </Contend>
              <ImageC src={slider.img} />
            </SwiperSlide>
          ))}
        </SwiperWrapper>
      </MySwiperBox>
      <ContainerImg>
        <ContainerImageBW>
          {imgArr.map((slider, index) => (
            <ImageBW key={index} src={slider.imgBW} />
          ))}
        </ContainerImageBW>
      </ContainerImg>
    </OurTeam>
  </div>
  <ImgOurTeamDiv>
    <ImgHero src={ImageHero} alt="" />
  </ImgOurTeamDiv>
  <div ref={conectRef}>
    <ConectUsEN name="ConnectUS" />
  </div>

  <FooterEN />
 </>
 );
};
```