ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп’ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**

з дисципліни

«Web-дизайн та web-програмування»

на тему:

Розробка веб-сайту «healthcare»

Студента І курсу групи ІПЗс-11

спеціальності

121 – Інженерія програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сищук В.Г.

(підпис)

Керівник: асистент А.В. Вознюк

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

**Луцьк 2023**

**Луцький національний технічний університет**

***Факультет комп’ютерних та інформаційних технологій***

***Кафедра інженерії програмного забезпечення***

Ступінь вищої освіти: *бакалавр*

Спеціальність: *121 «Інженерія програмного забезпечення»*

Дисципліна: *«Web-дизайн та web-програмування»*

Курс *І* Група *ІПЗс-11* Семестр *І*

***ЗАВДАННЯ***

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сищук Владислав Геннадійович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім’я, по батькові)

1. Тема роботи Розробка веб-сайту «healthcare» .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи: *асистент Вознюк Анастасія Вадимівна*

2. Строк подання студентом роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до роботи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перелік графічного матеріалу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Дата видачі завдання 04.11.2023

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ з/п** | **Назва етапів курсової роботи** | **Строк**  **виконання**  **етапів роботи** | **Примітка** |
| 2 | Узгодження постановки задачі з керівником | 04.11.2023 |  |
| 3 | Пошук та вивчення літератури з питань курсової роботи | 18.11.2023 |  |
| 4 | Вибір макета сайту | 18.11.2023 |  |
| 5 | Узгодження макета сайту та функціоналу з керівником | 18.11.2023 |  |
| 6 | Розробка сайту та розміщення сайту на хостингу | 16.12.2023 |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 | Підготовка пояснювальної записки | 16.12.2023 |  |
| 13 | Здача курсової роботи на перевірку | 23.12.2023 |  |
| 14 | Захист | 29.12.2023 |  |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Сищук Владислав  (прізвище на ініціали) |
| Керівник роботи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Анастасія Вознюк |

**АНОТАЦІЯ**

Сищук В.Г. Розробка веб-сайту «healthcare». – Курсова робота з дисципліни «Web-дизайн та web-програмування» студента спеціальності 121 «Інженерія програмного забезпечення». – Луцький національний технічний університет. – Луцьк, 2023. – с.

Пояснювальна записка до курсової роботи містить: 24 стор., 2 рис., 3 лістингів, 10 джерел.

Мета курсової роботи полягає в розробці сайту для медичного закладу.

Пояснювальна записка до курсової роботи складається з двох розділів: аналіз предметної області, розробка веб-сайту. В розділі «Аналіз предметної області» проаналізовано програми для розробки веб-сайту, етапи розробки, веб-хостинги для розміщення продукту. У розділі «Розробка веб-сайту «healthcare» обрано для реалізації програмний продукт, описано етапи створення сайту, етапи були наведені приклади коду певних функцій сайту, тестування сайту і відповідність вимогам.

Ключові слова: Css, Scss, HTML , swiper, front end, Javascript, Sass.

**ЗМІСТ**

**[ВСТУП](#_Toc154766290)** [6](#_Toc154766290)

**[РОЗДІЛ 1.](#_Toc154766291)** [7](#_Toc154766291)

**[АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.](#_Toc154766292)** [7](#_Toc154766292)

[1.1 Огляд та аналіз сучасних Front End технологій. Вибір засобів розробки для розробки веб-сайту «healthcare» 7](#_Toc154766293)

[1.2 Постановка завдання на курсову роботу 10](#_Toc154766294)

**[РОЗДІЛ 2.](#_Toc154766295)** [11](#_Toc154766295)

**[РОЗРОБКА ВЕБ-САЙТУ «HEALTHCARE».](#_Toc154766296)** [11](#_Toc154766296)

[2.1 Обґрунтування вибору технологій і засобів вирішення поставленого завдання 11](#_Toc154766297)

[2.2 Етапи розробки веб-сайту «healthcare» 14](#_Toc154766298)

[2.3 Практична реалізація веб-сайту «healthcare» 14](#_Toc154766299)

[2.4 Тестування веб-сайту «healthcare» 20](#_Toc154766300)

**[ВИСНОВКИ](#_Toc154766301)** [23](#_Toc154766301)

**[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛА](#_Toc154766302)** [24](#_Toc154766302)

# **ВСТУП**

Розробка веб-сайту «healthcare» - це важливий крок у напрямку покращення доступу до медичних послуг і підвищення якості охорони здоров'я. У сучасному світі, де технології швидко розвиваються, віртуальна присутність у сфері охорони здоров'я є необхідністю, а створення веб-сайту «healthcare» відкриває безліч можливостей для поліпшення системи надання медичних послуг.

Цей веб-сайт буде спрямований на надання користувачам зручного та ефективного інтерфейсу для отримання інформації про різноманітні аспекти охорони здоров'я. Зокрема, він надасть можливість швидкого доступу до медичної інформації, консультацій та онлайн-прийому лікаря, що є критичними у випадках невідкладних ситуацій.

Для реалізації цієї роботи поставлено наступні завдання:

* обрати макет сайту у програмі Figma;
* обрати технологій що будуть використовуватися при розробці;
* створити html розмітку сайту та оформити його за допосогою scss стилів;
* реалізувати форму зворотнього зв’яку;
* адаптувати сайт під пізні пристрої
* протестувати сайт та виправити знайдені помилки.

Спроектований веб-сайт буде інтегрований з функціоналом для запису на прийом до лікаря, а також нагадувань про важливі медичні процедури та прийом ліків. Це не тільки підвищить ефективність надання послуг, але і сприятиме збереженню часу для як пацієнтів, так і медичних працівників.

# **РОЗДІЛ 1**

# **АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.**

## Огляд та аналіз сучасних Front End технологій. Вибір засобів розробки для розробки веб-сайту «healthcare»

В наш час сфера веб-розробки прогресує неперервно, відкриваючи перед розробниками безліч інструментів та технологій для створення високоякісних веб-додатків. Зокрема, фронтенд-розробникам необхідно майстерно володіти різноманітними технологіями для успішної імплементації стильних та функціональних веб-сайтів. Основою залишаються HTML, CSS та JavaScript, що визначають структуру, дизайн і функціонал веб-сторінок.

JavaScript Frameworks та бібліотеки:

React - це бібліотека для створення інтерфейсів користувача, розроблена Facebook. Він використовує компонентний підхід до побудови UI, дозволяючи розбивати інтерфейс на невеликі, повторно використовувані частини. Реакт також має велику спільноту та широку підтримку.

Vue.js - це прогресивний JavaScript-фреймворк для створення інтерфейсів користувача.Vue відомий своєю простотою в освоєнні, легкістю інтеграції та швидкістю виконання. Він також часто використовується для розробки односторінкових додатків.

Angular - це повноцінний фреймворк для розробки веб-додатків, розроблений Google. Angular надає великий набір інструментів та функціональних можливостей для розробки складних додатків. Він використовує TypeScript і пропонує велику кількість готових рішень.

Сучасні стандарти JavaScript та інші інструменти:

Webpack - це інструмент для збірки та пакування ресурсів веб-додатків.Він дозволяє ефективно управляти залежностями, оптимізувати ресурси та автоматизувати рутинні задачі.

Інші технології:

* CSS Frameworks. Bootstrap, Tailwind CSS, і інші CSS-фреймворки полегшують стилізацію та розмітку веб-сайтів.Вони пропонують готові компоненти та стилі, що полегшує розробку та робить сайти більш адаптивними;
* GraphQL. Це мова запитів для вашого API, що дозволяє клієнтам отримувати тільки ту інформацію, яка їм потрібна. GraphQL дозволяє ефективно обмінюватися даними між клієнтом та сервером, забезпечуючи гнучкість та швидкість розробки;
* Progressive Web Apps (PWA). PWA - це веб-додатки, які поєднують переваги веб-сайтів та мобільних додатків. PWA надають зручність використання, навіть у відсутності Інтернет-з'єднання, і можуть працювати на різних платформах.

Ці технології складають основу для сучасного Front End розробки. Індустрія швидко розвивається, тому розробники повинні слідкувати за новинами та трендами, щоб залишатися в курсі останніх досягнень.

У 2023 році сучасні Front End технології характеризуються такими тенденціями:

Перехід на JavaScript. JavaScript є мовою програмування, яка все більше використовується для створення Front End інтерфейсів. Це пов'язано з тим, що JavaScript є багатофункціональною мовою, яка дозволяє створювати складні інтерфейси з використанням різних фреймворків і бібліотек.

Зростання популярності React. React є фреймворком JavaScript, який все більше використовується для створення Front End інтерфейсів. React є популярним вибором для створення складних інтерфейсів, оскільки він дозволяє легко створювати компоненти, які можна повторно використовувати.

Розвиток веб-прикладань на основі React Native. React Native є фреймворком для створення мобільних додатків на основі React. React Native дозволяє створювати мобільні додатки, які виглядають і функціонують як оригінальні додатки для iOS і Android.

Зростання популярності TypeScript. TypeScript є розширеною версією JavaScript, яка додає підтримку типів. TypeScript є популярним вибором для створення великих і складних Front End проектів, оскільки він дозволяє покращити якість коду і полегшити його обслуговування.

Аналіз сучасних Front End технологій

Перехід на JavaScript є однією з найважливіших тенденцій у світі Front End технологій. JavaScript є потужною мовою програмування, яка дозволяє створювати складні інтерфейси з використанням різних фреймворків і бібліотек.

React є одним з найпопулярніших фреймворків JavaScript для створення Front End інтерфейсів. React є популярним вибором для створення складних інтерфейсів, оскільки він дозволяє легко створювати компоненти, які можна повторно використовувати.

Розвиток веб-прикладань на основі React Native також є важливою тенденцією. React Native дозволяє створювати мобільні додатки, які виглядають і функціонують як оригінальні додатки для iOS і Android.

Зростання популярності TypeScript є ще однією важливою тенденцією. TypeScript є розширеною версією JavaScript, яка додає підтримку типів. TypeScript є популярним вибором для створення великих і складних Front End проектів, оскільки він дозволяє покращити якість коду і полегшити його обслуговування.

Штучний інтелект (ШІ) – це передова технологія у сфері веб-розробки, здатна ефективно обробляти обширні обсяги даних, приймати обґрунтовані рішення та робити точні прогнози. Із завдань, які раніше були виключною компетенцією людських програмістів, штучний інтелект вносить революцію у сферу створення, оптимізації та завершення коду.

Figma – це інструмент для колективного дизайну веб-інтерфейсів, який дозволяє користувачам спільно працювати над створенням живих та інтерактивних прототипів. Відзначившись в 2016 році, Figma швидко завоювала популярність серед фахівців у сфері веб-дизайну та спільнот онлайн. Завдяки Figma користувачі можуть ефективно взаємодіяти та обмінюватися шаблонами, дизайнами та віджетами з мільйонами користувачів у всьому світі.

## 1.2 Постановка завдання на курсову роботу

Враховуючи обрані технології та загальні цілі розробки веб-сайту новин про технології, було поставлено конкретні завдання, які дозволять досягти успішного виконання курсової роботи. Зазначені цілі відображають основні напрямки та вимоги проекту:

- вибір макету для сайту у програмі Figma – Обрання макету сайту у програмі Figma є важливим етапом у веб-розробці, оскільки це визначає основну структуру та зовнішній вигляд веб-сайту. Перед тим, як розпочати написання коду, було проведено детальне планування та вибір оптимального макету веб-сайту новин про технології;

- створення html розмітки сайту та створення scss стилів – ці кроки представляють собою ключові етапи веб-розробки, які взаємодіють між собою для формування структури та зовнішнього вигляду веб-сторінок;

- написання адаптиву для мобільних пристроїв – створення веб-сайту таким чином, щоб він ефективно та естетично виглядав на різних розмірах екранів, зокрема на мобільних пристроях. Цей підхід забезпечує оптимальний користувацький досвід незалежно від того, чи відбувається перегляд веб-сайту на смартфоні, планшеті чи настільному комп'ютері;

- реалізація форми зворотнього зв’яку – створення веб-форми, яка дозволяє користувачам надсилати повідомлення чи коментарі, прямо адміністратору. Цей елемент важливий для взаємодії з відвідувачами та забезпечення каналу зв'язку між користувачами та веб-сайтом;

- тестування та виправлення помилок – проведення систематичних тестів для виявлення та виправлення будь-яких помилок та недоліків у функціоналі та дизайні інтернет-магазину.

Це ключові завдання для успішної розробки веб-сайту новин про технології. Вони є найважливішими аспектами у процесі розробки даного веб-сайту.

# **РОЗДІЛ 2**

# **РОЗРОБКА** **ВЕБ-САЙТУ** **«HEALTHCARE».**

## 2.1 Обґрунтування вибору технологій і засобів вирішення поставленого завдання

Під час створення веб-сайту «healthcare», вибір відповідного середовища розробки визначається як критично важливий етап для досягнення ефективності та продуктивності. На цьому етапі було вирішено скористатися інтегрованим редактором коду Visual Studio Code, оскільки він пропонує ряд переваг, які суттєво спрощують та оптимізують процес розробки.

Visual Studio Code, розроблене Microsoft, є інтегрованим середовищем розробки та творчим майданчиком, що дозволяє розробникам ефективно створювати, редагувати та налагоджувати код перед подальшим його публікацією. Цей текстовий редактор має безліч настроюваних функцій у вигляді плагінів, які дозволяють створювати індивідуалізоване середовище розробки.

Visual Studio Code є легким та потужним, легко встановлюється на будь-яку платформу. Забезпечуючи вбудовану підтримку для Node.js, TypeScript, та JavaScript, а також розширеної екосистеми для інших мов програмування, таких як C++, Java, C#, PHP, Go та Python, він виходить за рамки звичайного текстового редактора.

Visual Studio Code входить в той же ряд із відомими інструментами, такими як Atom, Sublime та Text Wrangler, проте відрізняється своїми вищими можливостями та надійністю. Дозволяючи розробникам писати, редагувати та налагоджувати код в одному місці, VS Code підтримує багато мов програмування та пропонує вбудовану багатомовну підтримку.

Розробники можуть сподіватися на Visual Studio Code як на надійний інструмент для всіх аспектів розробки, особливо якщо використовувати його у поєднанні з відповідними інструментами. Завдяки вбудованій підтримці JavaScript, Node.js та TypeScript, а також можливості додавати підтримку інших мов, таких як Python, C#, PHP, Java та інших, Visual Studio Code є невід'ємною частиною процесу розробки, забезпечуючи неймовірну зручність та функціональність на всіх платформах [10].

У наш час цифрових технологій забезпечення ефективного та зручного механізму отримання зворотного зв'язку на веб-сайті визначається як один з ключових аспектів взаємодії з користувачами. Цікавою альтернативою традиційній електронній пошті є використання месенджера Telegram для обробки повідомлень від відвідувачів.

Telegram, як месенджер, володіє надзвичайно швидким та миттєвим способом передачі повідомлень. На відміну від традиційної електронної пошти, де інтервали часу можуть бути великими, використання Telegram дозволяє негайно взаємодіяти. Користувачі можуть отримати відповіді та відправляти повідомлення без зайвої затримки, що особливо важливо в ситуаціях, де час важливий фактор.

Однією з ключових переваг Telegram є його мобільність та універсальність. Мобільні додатки Telegram доступні для різних операційних систем, таких як iOS та Android, забезпечуючи зручність для користувачів різних пристроїв. Завдяки месенджеру, відвідувачі можуть взаємодіяти з веб-сайтом з будь-якого місця та пристрою, забезпечуючи гнучкість та доступність комунікації.

Також Telegram дозволяє взаємодіяти не тільки текстовими повідомленнями, але й різноманітним мультимедійним контентом. Можливість надсилати зображення, відео, аудіозаписи та інші формати робить комунікацію більш інформативною. Це сприяє кращому розумінню питань та дозволяє детальніше вирішувати проблеми, що виникають. Загальна взаємодія з користувачами через Telegram може бути більш ефективною завдяки високому рівню візуальної комунікації.

Для комфортної роботи з макетом проекту використовувалася Figma – інструмент для розробки дизайну та створення макетів веб-сайтів.

Figma відзначається високою ефективністю порівняно з іншими інструментами, такими як Avocode та Photoshop, завдяки великій кількості розширень та введенню нового режиму "dev-mode". Плагіни полегшують інтеграцію зовнішніх сервісів та автоматизують певні процеси, зробивши роботу більш швидкою та ефективною. Режим "dev-mode" спрощує взаємодію з розробниками, дозволяючи перевіряти та тестувати інтерфейс безпосередньо в середовищі розробки. Це значно прискорює процес внесення змін та узгодження макету з кодом.

Прекрасним інструментом для розробки є ChatGPT – мовна модель, розроблена компанією OpenAI, яка може генерувати текст, враховуючи контекст та конкретне завдання.

В рамках даної курсової роботи ChatGPT використовувався для реалізації функціоналу, пов'язаного з відправкою електронного листа замовлення. Це дозволило автоматизувати процес створення автоматичних повідомлень, спрямованих на поліпшення взаємодії з користувачами та оптимізацію обробки замовлень.

Однією з ключових областей впливу ChatGPT у веб-розробці є можливість покращення взаємодії з користувачем. Веб-сайти та додатки тепер можуть надавати більш природні та інтуїтивно зрозумілі консультації чи відповіді на запитання, спрощуючи користування та підвищуючи задоволення від використання.

ChatGPT може використовуватися для автоматичного створення контенту для веб-сайтів, включаючи статті, огляди та інші текстові матеріали. Це дозволяє заощаджувати час на написання та оптимізувати SEO-показники.

Інтелектуальні алгоритми ChatGPT можуть бути використані для аналізу поведінки користувачів та прогнозування їхніх потреб, що дозволяє персоналізувати веб-сайти, рекомендувати контент або товари, відповідати на індивідуальні запитання, забезпечуючи більш особистий досвід взаємодії. Застосування ChatGPT у веб-розробці відкриває широкі можливості для покращення користувацького досвіду, оптимізації роботи веб-сайтів та використання штучного інтелекту в різних сферах веб-індустрії.

## 2.2 Етапи розробки веб-сайту «healthcare»

Етапи розробки веб-сайту новин про технології включають деталізовані кроки, що описують весь процес створення веб-додатку, починаючи від вибору дизайну та закінчуючи розгортанням готового проекту на хостингу.

1. Вибір макету:

* вибір макету у Figma веб-сайту новин про технології;
* створення структури, дизайну та елементів функціоналу.

1. Розробка базової верстки:

* створення основної структури сторінок за допомогою HTML;
* використання CSS для надання стилізації та естетичного вигляду;
* використання JavaScript для впровадження інтерактивності та реалізації потрібного функціоналу.

1. Тестування та виправка помилок:

* виконання тестування для перевірки функціональності та коректності роботи сайту;
* виправлення помилок та переписування тих частин коду що конфліктують одна з одною, або не працюють так як задумувалося.

1. Розгортання на хостингу:

* вибір середовища для розгортання сайту;
* розміщення сайту.

## 2.3 Практична реалізація веб-сайту «healthcare»

Секція герой на сайті є ключовим елементом веб-дизайну, який захоплює увагу відвідувачів та створює перший враження про продукт. В даній секції присутня форма зворотнього зв’язку для подальшої комунікації лікарів та пацієнтів.

Зазвичай розташована у верхній частині головної сторінки, ця секція повинна вражати естетикою та візуальною привабливістю. Вона часто містить велике, високоякісне зображення чи відео, що ілюструє сутність бренду або передає ключовий повідомлення. Текстовий блок містить короткий та чіткий заголовок, який привертає увагу, разом із коротким описом чи запрошенням до дії. Щоб забезпечити ефективність, секція герой добре скоординована за кольорами та шрифтами, враховуючи відмінності відображення на різних пристроях. Ця секція створює враження, спонукає користувачів до дослідження сайту та визначає тон для подальшого взаємодії з контентом.

На даній секції розміщене модальне вікно (рис. 2.1), за допомогою якого клієнти можуть залишати свої контактні дані для того щоб з ними в подальшому зв’язалися лікарі для початку процедури лікування.

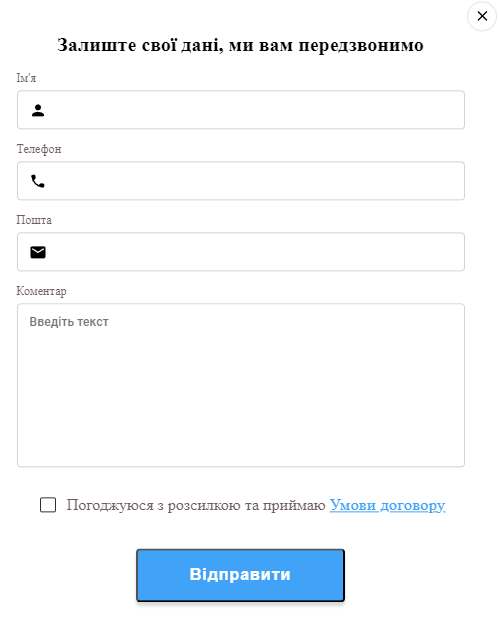


Рисунок 2.1 – Модальне вікно.

Для сайту було створено форму зворотнього зв’язку яка отримує електронну пошту користувача та передає її з допомогою телеграм бота адміністратору для подальших дій (ліст. 2.1).

Лістинг 2.1 – реалізація форми зворотнього зв’язку з допомогою javaScript

const forms = document.querySelectorAll("#subscribe");

forms.forEach(form => {

form.addEventListener("submit", (e) => {

e.preventDefault();

var subscribe\_\_email = form.querySelector("#email").value;

var subscribe = `New subscriber Email:<b>${subscribe\_\_email}</b>`

var token = "6270391808:AAEeDST2kRZacIxzisJz-GBovp8C8uRHcY0";

var chat\_id = '1728443213';

var url = `https://api.telegram.org/bot${token}/sendMessage?chat\_id=${chat\_id}&text=${subscribe}&parse\_mode=html`;

let api = new XMLHttpRequest();

api.open("GET", url, true);

api.send();

console.log("Subscribe successfully sent!");

console.log(subscribe\_\_email)

form.reset();

});

});

кінець лістингу 2.1

Для програмної реалізації модального вікна було створено функції що вимикають та вмикають можливість прокручувати сторінку. (лістинг 2.2)

Лістинг 2.2 – програмна реалізація модального вікна на javaScript

(() => {

const forms = document.querySelectorAll("#subscribe");

const refs = {

openModalBtns: document.querySelectorAll("[data-modal-open]"),

closeModalBtn: document.querySelector("[data-modal-close]"),

modal: document.querySelector("[data-modal]"),

};

function disableScroll() {

document.body.style.setProperty('overflow', 'hidden', 'important');

}

function enableScroll() {

document.body.style.removeProperty('overflow');

}

function openModal() {

// Get the form associated with the clicked button

const form = this.closest("form");

if (form.checkValidity()) {

refs.modal.classList.remove("is-hidden");

disableScroll();

} else {

form.reportValidity();

}

}

function closeModal() {

refs.modal.classList.add("is-hidden");

enableScroll();

}

refs.openModalBtns.forEach(element => {

element.addEventListener("click", openModal);

});

refs.closeModalBtn.addEventListener("click", closeModal);

})();

Кінець лістингу 2.2

За допомогою штучного інтелекту для секцій service та … вдалось зробити swiper. Swiper - це потужна та гнучка JavaScript-бібліотека для створення сучасних та інтерактивних слайдерів. Вона широко використовується для створення каруселей зображень, галерей, презентацій та інших компонентів, де може бути використана функціональність перегортання.

Слайдер, реалізований за допомогою JS бібліотеки "Swiper", становить важливий аспект для поліпшення інтеракції з контентом на веб-сайті, дозволяючи елегантно презентувати змінюючийся матеріал без зайвого коду. Використання технології Swiper гарантує чудову адаптивність та гнучкість, забезпечуючи плавні переходи між слайдами та покращуючи користувацький досвід.

Для секції service був розроблений слайдер ручного типу, за допомогою цього можна самостійно переглядати потрібний слайд, або переглянути

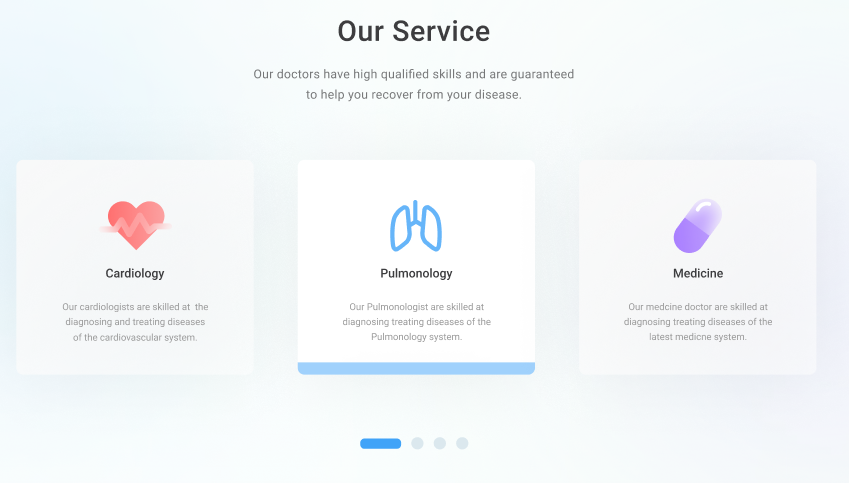


Рисунок 2.2 – Слайдер.

Для секції comments був розроблений слайдер для показу відгуків про закляд, щоб відвідувачі могли переглянути що про нього думають різні люди. Під час розробки було вирішено зробити слайдер автоматичним, використовуючи параметр "autoplay". Ця функція надає динаміку та привабливість презентації на веб-сайті, оскільки слайди автоматично переглядаються через визначений часовий інтервал. Використання параметра "autoplay" є важливим елементом для автоматичного відтворення слайдів без активної участі користувача.

Цей підхід особливо корисний для веб-сайтів, що акцентують візуальний зміст, портфоліо чи товари, де параметр "autoplay" забезпечує зручний та ефективний спосіб презентації контенту. Користувачам не потрібно взаємодіяти зі слайдером для перегляду слайдів, що робить презентацію більш динамічною та привабливою, залучаючи увагу аудиторії та підвищуючи привабливість веб-сайту.

Також вдалось зробити меню burger – це іконка, яка використовується для представлення мобільного меню на веб-сайтах. Ця іконка зазвичай складається з трьох горизонтальних ліній, що нагадують бургер, тому її так і називають.

Коли ви бачите іконку burger, це зазвичай означає, що на сторінці використовується адаптивний дизайн, і коли екран вузьший (наприклад, на мобільному пристрої), основне меню приховується, замінюючись іконкою бургера, це допомагаю компактніше розміщати певні елементи та надає сайту зручності. Коли користувач торкається чи клікає на цю іконку, з'являється випадаюче меню або бічне меню, яке містить основні посилання чи категорії сторінок. (лістинг 2.3)

1. (() => {
2. const menuBtnRef = document.querySelector("[data-menu-button]");
3. const mobileMenuRef = document.querySelector("[data-menu]");
4. const body = document.querySelector("body");
5. menuBtnRef.addEventListener("click", () => {
6. const expanded = menuBtnRef.getAttribute("aria-expanded") === "true" || **false**;
7. menuBtnRef.classList.toggle("is-open");
8. menuBtnRef.setAttribute("aria-expanded", !expanded);
9. mobileMenuRef.classList.toggle("is-open");
10. body.classList.toggle("no-scroll");
11. });
12. })();

Лістинг – 2.3 burger.

## 2.4 Тестування веб-сайту «healthcare»

SEO оптимізація для веб-сайту «healthcare» - це процес покращення видимості та релевантності веб-сайту в результатах пошуку для певних ключових слів та фраз, пов'язаних із галуззю охорони здоров'я.

SEO оптимізація для веб-сайтів «healthcare» має свої особливості, пов'язані з особливостями цієї галузі. Зокрема, для медичних веб-сайтів важливо:

Забезпечити високу якість контенту. Контент медичного веб-сайту повинен бути точним, актуальним та корисним для користувачів. Він повинен відповідати на запитання, які можуть виникнути у потенційних клієнтів, і надавати їм необхідну інформацію про послуги, які надає компанія.

Використовувати правильні ключові слова та фрази. При виборі ключових слів для медичних веб-сайтів важливо враховувати, що користувачі можуть використовувати різні пошукові запити для пошуку медичних послуг. Наприклад, користувач може шукати "лікаря загальної практики в Києві", "хірургію в Харкові" або "стоматологію в Одесі".

Створити структуру сайту, зручну для пошуку. Структура медичного веб-сайту повинна бути чіткою та логічною. Користувачі повинні легко знаходити потрібну інформацію на сайті.

Додати зовнішні посилання на сайт. Зовнішні посилання з авторитетних веб-сайтів допомагають покращити рейтинг медичного веб-сайту в результатах пошуку.

Ось деякі конкретні поради щодо SEO оптимізації для веб-сайтів «healthcare»:

Створіть унікальні заголовки та мета-опис для кожної сторінки веб-сайту. У заголовках і мета-описах повинні бути використані ключові слова та фрази, за якими користувачі можуть шукати ваш веб-сайт.

Підпишіть зображення на веб-сайті. Підписи до зображень повинні містити ключові слова та фрази, пов'язані з зображеннями.

Створюйте корисний і інформативний контент. Контент медичного веб-сайту повинен бути корисним для користувачів і відповідати на їхні запитання.

Розміщуйте інформацію про ваші медичні послуги на авторитетних веб-сайтах. Це допоможе покращити рейтинг вашого веб-сайту в результатах пошуку.

SEO оптимізація для медичних веб-сайтів - це складний і тривалий процес. Однак, якщо ви будете дотримуватися цих порад, ви зможете покращити видимість вашого веб-сайту в результатах пошуку і залучити більше потенційних клієнтів.

Тестування веб-сайту «healthcare» - це процес перевірки функціональності, продуктивності, безпеки та юзабіліті веб-сайту медичного закладу перед його запуском або після внесення змін. Воно допомагає виявити помилки, гарантує правильне функціонування веб-сайту і підвищує якість досвіду користувача.

Типи тестування веб-сайтів «healthcare»

Функціональне тестування - це перевірка того, чи працюють всі функції веб-сайту відповідно до вимог. Цей тип тестування зазвичай виконується QA-інженерами, які використовують спеціальні інструменти та методи.

Тестування продуктивності - це перевірка того, як веб-сайт працює в умовах навантаження. Цей тип тестування зазвичай виконується інженерами-програмістами, які використовують спеціальне обладнання та програмне забезпечення.

Тестування безпеки - це перевірка того, наскільки веб-сайт захищений від злому. Цей тип тестування зазвичай виконується секторами безпеки, які використовують спеціальні інструменти та методи.

Тестування юзабіліті - це перевірка того, наскільки легко і зручно використовувати веб-сайт. Цей тип тестування зазвичай виконується користувачами або дослідниками юзабіліті, які використовують різні методи, такі як опитування, інтерв'ю та спостереження.

Важливість тестування веб-сайтів «healthcare»

Тестування веб-сайтів "healthcare" є важливим процесом, який допомагає гарантувати, що веб-сайт працює належним чином і забезпечує позитивний досвід користувачів. Тестування може допомогти виявити помилки та проблеми, які можуть привести до втрати клієнтів або навіть фінансових збитків.

Ось деякі конкретні приклади того, як тестування веб-сайтів "healthcare" може допомогти:

Функціональне тестування може допомогти виявити помилки в таких функціях, як бронювання зустрічей, замовлення ліків або онлайн-консультація з лікарем.

Тестування продуктивності може допомогти гарантувати, що веб-сайт може обробляти велику кількість користувачів одночасно.

Тестування безпеки може допомогти захистити веб-сайт від хакерських атак.

Тестування юзабіліті може допомогти переконатися, що веб-сайт легко і зручно використовувати для людей з різними здібностями та обмеженнями.

Рекомендації щодо тестування веб-сайтів «healthcare»

Ось кілька рекомендацій щодо тестування веб-сайтів «healthcare»:

Включіть в процес тестування різні типи користувачів, включаючи пацієнтів, медичних працівників і адміністраторів.

Враховуйте специфічні вимоги медичної галузі під час розробки тестових випадків.

Використовуйте спеціальні інструменти та методи тестування, які підходять для медичної галузі.

Тестування веб-сайтів «healthcare» - це важливий процес, який допомагає забезпечити якісний і безпечний досвід для пацієнтів і медичних працівників

# **ВИСНОВКИ**

Розробка веб-сайту «healthcare» є ключовим напрямком у вдосконаленні системи надання медичних послуг та забезпеченні зручного доступу до інформації для користувачів. З урахуванням специфічних вимог і особливостей у сфері охорони здоров'я, важливо зазначити кілька ключових висновків:

Зазначені цілі відображають основні напрямки та вимоги проекту:

- вибір макету для сайту у програмі Figma – Обрання макету сайту у програмі Figma є важливим етапом у веб-розробці, оскільки це визначає основну структуру та зовнішній вигляд веб-сайту. Перед тим, як розпочати написання коду, було проведено детальне планування та вибір оптимального макету веб-сайту новин про технології;

- створення html розмітки сайту та створення scss стилів – ці кроки представляють собою ключові етапи веб-розробки, які взаємодіють між собою для формування структури та зовнішнього вигляду веб-сторінок;

- написання адаптиву для мобільних пристроїв – створення веб-сайту таким чином, щоб він ефективно та естетично виглядав на різних розмірах екранів, зокрема на мобільних пристроях. Цей підхід забезпечує оптимальний користувацький досвід незалежно від того, чи відбувається перегляд веб-сайту на смартфоні, планшеті чи настільному комп'ютері;

- реалізація форми зворотнього зв’яку – створення веб-форми, яка дозволяє користувачам надсилати повідомлення чи коментарі, прямо адміністратору. Цей елемент важливий для взаємодії з відвідувачами та забезпечення каналу зв'язку між користувачами та веб-сайтом;

- тестування та виправлення помилок – проведення систематичних тестів для виявлення та виправлення будь-яких помилок та недоліків у функціоналі та дизайні інтернет-магазину.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛА**

1. List of frontend technologies every developer should know in 2023. URL: https://merge.rocks/blog/list-of-frontend-technologies-every-developer-should-know-in-2023 (дата звернення: 21.12.2023).
2. What Is CSS? (And Why It’s Important) URL: https://www.indeed.com/career-advice/career-development/what-is-css (дата звернення: 21.12.2023).
3. Top Front End Technologies You Must Know [2023] URL: <https://www.interviewbit.com/blog/front-end-technologies/> (дата звернення: 21.12.2023).
4. Sass (style sheet language) URL: https://en.wikipedia.org/wiki/Sass\_(style\_sheet\_language) (дата звернення: 21.12.2023).
5. JavaScript URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
6. Building modern sliders with HTML, CSS, and Swiper. URL: https://blog.logrocket.com/building-modern-sliders-html-css-swiper/(дата звернення: 21.12.2023).
7. Pros and Cons of Using AI for Web Development: Stumbling Blocks You Have to Know About. URL: ttps://litslink.com/blog/using-ai-for-web-development
8. What is Figma? URL: https://www.nobledesktop.com/learn/figma/what-is-figma (дата звернення: 21.12.2023).
9. The Power of Figma as a Design Tool URL: https://www.toptal.com/designers/ui/figma-design-tool(дата звернення: 21.12.2023).
10. Visual Studio Code URL: https://code.visualstudio.com/docs/editor/whyvscode (дата звернення: 21.12.2023).