

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ Системотехніки \_\_\_\_\_

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ другий (магістерський). \_\_\_\_\_

«Дослідження та аналіз впливу фреймворків на характеристики  
вебсторінок»

Виконав:

студент 2 курсу, групи ІТПм -22-2  
Петроченков П. М.  
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки

Тип програми освітньо- професійна  
Освітня програма Інформаційні технології  
проектування

Керівник доц. Білова Т. Г.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_  
(підпис)

Гребеннік І. В.

\_\_\_\_\_  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет	Комп'ютерних наук
Кафедра	Системотехніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	122-Комп'ютерні науки
	(код і повна назва)
Тип програми	освітньо- професійна
Освітня програма	Інформаційні технології проектування

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« 20 » листопада 2023 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студент Петроченков Павло Михайлович

(прізвище, ім'я, по-батькові)

1. Тема роботи «Дослідження та аналіз впливу фреймворків на характеристики веб сторінок»

затверджена наказом університету від «20» листопада 2023 р. № 1373Ст

2. Термін подання студентом роботи до екзаменаційної комісії «12» січня 2024 р.

3. Вихідні дані до роботи Дані статей та публікацій, книг та електронних ресурсів за темою роботи.

4. Перелік питань, що необхідно опрацювати в роботі вступ, аналіз предметної галузі, методи вирішення проблеми, проведення практичного експерименту та аналіз отриманих результатів, опис прототипу додатку, висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) ER-діаграма бази даних, структура Backend частини, Lighthouse метрики, WebPageTest метрики, , Графічний інтерфейс головної



## РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи містить: с. 70, рис. 50, таблиць 11, формул 1, дод. 3, джерел 17.

ВЕБДИЗАЙН, ВЕБРОЗРОБКА, ДОСТУПНІСТЬ, ІНТЕРНЕТ-ТЕХНОЛОГІЇ, ПРОДУКТИВНІСТЬ, ФРЕЙМВОРКИ, ANGULAR, REACT, SEO, SVELTE, VUE

Об'єктом дослідження є інтерактивні веб-сторінки, створені за допомогою різних фронтенд фреймворків.

Предметом дослідження є вплив використання різноманітних фреймворків на характеристики веб-сторінок, такі як швидкість завантаження, SEO-оптимізація, доступність та інтерактивність.

Мета дослідження: аналіз та порівняльний огляд впливу популярних веб-фреймворків на ефективність веб-сторінок з точки зору різних характеристик і визначення найкращих практик для оптимізації роботи веб-сторінок.

Методи дослідження – компаративний аналіз, тестування продуктивності, аудит доступності, аналіз SEO-оптимізації, методи статистичного аналізу.

Наукова новизна роботи полягає у розробці комплексного методологічного підходу до оцінювання веб-сторінок, заснованого на детальному аналізі впливу використання різних фреймворків, що дозволяє виявити оптимальні рішення для підвищення продуктивності, покращення SEO-показників і забезпечення високого рівня доступності веб-ресурсів.

## ABSTRACT

The explanatory note to the master's qualification work contains: p. 70, fig. 50, tables 11, form. 1, app. 3, sources 17.

ACCESSIBILITY, ANGULAR, FRAMEWORKS, INTERNET TECHNOLOGIES, PERFORMANCE, REACT, SEO, SVELTE, VUE, WEB DESIGN, WEB DEVELOPMENT

The object of research is interactive web pages created using various frontend frameworks.

The subject of the study is the impact of using a variety of frameworks on web page characteristics such as loading speed, SEO optimization, accessibility, and interactivity.

The purpose of the study: to analyze and compare the impact of popular web frameworks on the efficiency of web pages in terms of various characteristics and to identify best practices for optimizing web page performance.

Research methods – comparative analysis, performance testing, accessibility audit, SEO optimization analysis, statistical analysis methods.

The scientific novelty of the work lies in the development of a comprehensive methodological approach to the evaluation of web pages, based on a detailed analysis of the impact of using different frameworks, which allows identifying optimal solutions to enhance performance, improve SEO indicators, and ensure a high level of web resource accessibility.

Application area – web development, internet design, website optimization, user interface and interaction development, digital marketing.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень, термінів.....	7
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Опис вебсторінки та її характеристики.....	11
1.2 Аналіз існуючих інструментів оцінки характеристик та вплив фрейсворків на ці характеристики .....	20
1.3 Постановка задачі дослідження .....	23
2 Опис обраних технологій та методів проектування з реалізації системи .	25
2.1 Аналіз існуючих інструментів оцінки характеристик.....	25
2.2 Формула оцінки характеристик вебсторінки .....	27
2.3 Аналіз кожного фреймворка та його вплив на характеристики.....	29
3 Опис прийнятих проектних рішень.....	39
3.1 Вибір основного темплейту для аналіз характеристик .....	39
3.2 Аналіз Backend частини проекту .....	40
3.3 Обґрунтування Backend частини.....	41
3.4 Аналіз Fronted частини проекту та його структури.....	44
3.5 Аналіз характеристики вебсторінок за різними фреймворками...	45
3.5.1 Аналіз React .....	45
3.5.2 Аналіз Ember.....	48
3.5.3 Аналіз Vue.....	51
3.5.4 Аналіз Astro .....	53
3.5.5 Аналіз Angular .....	55
3.5.6 Аналіз NextJS.....	58
3.5.7 Аналіз Backbone .....	61
3.5.8 Аналіз Svelte .....	63
3.6 Аналіз та порівняння результатів .....	66
3.7 Аналіз перспектив покращення дослідження.....	68
Висновки.....	71
Перелік джерел посилань .....	72
Додаток А Графічні матеріали кваліфікаційної роботи.....	74
Додаток Б Інтерфейс розробленої системи.....	80
Додаток В Текст програми.....	84

,

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ, ТЕРМІНІВ

SEO– Search engine optimization.

HTML – Hypertext Markup Language.

CSS – Cascading Style Sheets.

WCAG – Web Content Accessibility Guidelines.

HTTP/HTTPS – Hypertext Transfer Protocol.

## ВСТУП

В сучасному світі інформаційні технології відіграють важливу роль у бізнесі та організаційному управлінні. Інформаційні системи стали невід'ємною складовою для виконання різноманітних інформаційних та обчислювальних завдань, що відповідають потребам як систем управління, так і їх користувачів. Вони спрощують рутинні процеси, поліпшують ефективність та роблять взаємодію з інформацією більш зручною.

Збільшення обсягу і складності інформації, яку обробляють інформаційні системи, а також зростання навантаження на них внаслідок розширення бізнесу вимагають більш ефективних та надійних рішень. Технічні збої, помилки в інтерфейсах, проблеми збереження та взаємодії з базами даних можуть серйозно позначитися на користувацькому досвіді та вплинути на бізнес-процеси.

Інформаційні системи, які надають обслуговування клієнтів або підтримують критичні для бізнесу процеси, повинні бути ефективними та надійними. Вибір правильних інструментів та технологій грає важливу роль у забезпеченні високої якості та продуктивності інформаційних систем.

Метаю роботи є ретельне дослідження та аналіз впливу JavaScript-фреймворків на функціональні характеристики вебсторінок з метою надання рекомендацій розробникам та бізнес-організаціям для досягнення оптимальних результатів у сфері веб-розробки.

Об'єктом дослідження є самі JavaScript-фреймворки і їх вплив на якість та продуктивність вебсторінок.

Предметом дослідження є вплив JavaScript-фреймворків (зокрема React, Vue.js та Angular) на функціональні характеристики вебсторінок, такі як швидкість завантаження, відзивчивість та візуальна привабливість.

Наукова новизна роботи полягає у всебічному аналізі та порівнянні характеристик вебсторінок, створених за допомогою різних сучасних фреймворків, що дозволяє визначити оптимальні підходи до розробки веб-рішень з урахуванням специфіки проекту та вимог користувачів. Робота розкриває потенціал вдосконалення продуктивності, доступності, SEO, та інших ключових параметрів вебсайтів, забезпечуючи комплексну методологію оцінки та визначення найбільш ефективних технологічних стеків. Дослідження також вносить вклад у розвиток методик оптимізації веб-додатків, що базуються на передових інженерних практиках та новітніх досягненнях у галузі веб-технологій.

Результати досліджень можуть бути корисними для широкого кола стейкхолдерів, які мають інтерес до веб-розробки та покращення характеристик вебсторінок. Результати дослідження було апробовано на V Міжнародній студентській науковій конференції "Цифровізація науки та сучасні тренди її розвитку", м. Житомир, 2023 рік [1].

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

В сучасному світі інформаційні системи стали невід'ємною частиною бізнесу та організаційного управління. Вони призначені для виконання різноманітних інформаційно-обчислювальних завдань та забезпечують потреби систем управління та їх користувачів. Функціональна структура інформаційних систем може бути представлена як сукупність функціональних підсистем, і їх функціональні особливості можуть визначатися типом діяльності організації. Наприклад, існують виробничі системи, системи маркетингу, фінансові та облікові системи, системи управління кадрами та спеціалізовані інформаційні системи, які відповідають специфіці конкретної діяльності підприємства.

Однак інформаційні системи, особливо ті, що надають обслуговування клієнтів чи підтримують критичні для бізнесу процеси, повинні бути ефективними та надійними [2]. Технічні збої, помилки у функціонуванні інтерфейсу, проблеми збереження інформації та взаємодії з базами даних можуть значно вплинути на задоволення користувачів та впливати на бізнес-процеси.

JavaScript frameworks відіграють важливу роль у покращенні функціональних характеристик вебсторінок [3]. Вони дозволяють оптимізувати завантаження сторінок, забезпечувати більшу взаємодію та покращувати візуальну привабливість. Наприклад, React, Vue.js та Angular - це деякі з популярних JavaScript-фреймворків, які допомагають розробникам покращити продуктивність та якість вебсторінок.

У цьому контексті важливо розглядати вплив різних JavaScript frameworks на функціональні характеристики вебсторінок, такі як швидкість завантаження, відзивчивість та візуальна привабливість [4]. Обрання

відповідного фреймворку може бути вирішальним фактором у досягненні успіху та задоволення потреб користувачів в інформаційних системах.

## 1.1 Опис вебсторінки та її характеристики

Для початку розглянемо термін вебсторінка, вебсторінка - це документ, який доступний через Всесвітню мережу (Інтернет) та відображається на веб-браузері користувача. Веб-сторінки складаються з тексту, зображень, відео, аудіо, гіперпосилань та інших мультимедійних компонентів, які можуть містити інформацію на різні теми, функціональність та дизайн.

Веб-сторінка може бути статичною або динамічною. Статична сторінка - це фіксований документ, який не змінюється без зміни вихідного коду. Динамічна сторінка, навпаки, змінюється відповідно до дій користувача або даних, що надходять з сервера [5]. Для створення динамічних сторінок часто використовуються різні технології, такі як JavaScript, серверні мови програмування (наприклад, PHP, Python, Ruby) та бази даних.

Властивості та характеристики вебсторінки можуть бути дуже різними і залежать від різних факторів, включаючи:

- дизайн: вигляд та оформлення сторінки, включаючи кольорову палітру, шрифти, розміщення контенту тощо;
- вміст: інформація, яка представлена на сторінці, включаючи текст, зображення, відео та інші мультимедійні елементи;
- функціональність: різні функції та можливості, які надаються користувачам, такі як пошук, реєстрація, коментування, оплата тощо;
- взаємодія: способи, якими користувачі можуть взаємодіяти зі сторінкою, включаючи кліки, наведення, введення даних тощо;

- швидкість завантаження: час, який потрібний для завантаження сторінки на пристрої користувача;
- сумісність та доступність: можливість відображення сторінки на різних браузерях і пристроях, а також доступність для людей з обмеженими можливостями;
- безпека: заходи, що приймаються для захисту вебсторінки від атак і зловмисних дій.

Загалом, вебсторінка - це інтерактивний засіб спілкування та розповсюдження інформації в інтернеті, і її характеристики залежать від цілей, потреб користувачів та технічних можливостей, використовуваних при її створенні.

Існують різні стандарти та рекомендації, які визначають характеристики та вимоги до вебсторінок для забезпечення їх якості, доступності та взаємодії з користувачами. Ось деякі з найважливіших стандартів:

- HTML: вона є мовою розмітки, яка визначає структуру та вміст вебсторінок. Він включає в себе елементи для заголовків, параграфів, списків, зображень, посилань та інших елементів;
- CSS: він визначає зовнішній вигляд вебсторінок, включаючи кольори, шрифти, розташування елементів та інші візуальні аспекти;
- WCAG: це набір рекомендацій, які визначають, як забезпечити доступність веб-контенту для людей з обмеженими можливостями. WCAG включає в себе критерії, які стосуються навігації, відображення, відмовості від користування мишкою та інших аспектів;
- HTTP/HTTPS: ці протоколи визначають, як дані передаються між вебсервером і веб-браузером. Протокол HTTPS використовує шифрування для захисту конфіденційної інформації;
- SEO: це набір практик, які спрямовані на оптимізацію вебсторінок для пошукових систем. Це включає в себе використання правильних мета-

тегів, ключових слів та інших факторів, що впливають на ранжування в результатах пошуку;

- Performance Optimization: ці рекомендації стосуються оптимізації швидкості завантаження вебсторінок, що включає в себе мінімізацію файлів, використання кешування, асинхронну загрузку та інші техніки;
- Mobile Optimization: забезпечення того, що вебсторінки коректно відображаються на мобільних пристроях, таких як смартфони та планшети;
- Cross-Browser Compatibility: вебсторінки повинні відображатися правильно на різних веб-браузерах, таких як Chrome, Firefox, Safari, Edge та інші.

Кожна рекомендація так чи інакше впливає на характеристики вебсторінки, бо саме ці стандарти допомагають визначити і забезпечити характеристики вебсторінок, що включають в себе візуальний дизайн, доступність, безпеку, швидкість та інші важливі аспекти, які впливають на сприйняття та ефективність вебсторінок користувачами.

Веб-сторінки, які знаходяться в інтернеті, мають різноманітні характеристики, що визначають їх функції, зміст та взаємодію з користувачами. Ось деякі основні характеристики вебсторінок:

- дизайн та верстка;
- навігація;
- вміст;
- SEO-оптимізація;
- швидкість завантаження;
- мобільна доступність;
- безпека;
- взаємодія та функціональність;
- аналітика та відстеження;
- сумісність та браузери.

Ці характеристики допомагають визначити якість та ефективність вебсторінок. Кращі вебсторінки враховують всі ці аспекти, щоб забезпечити задоволення користувачів та досягнути своїх цілей.

Давайте більш детально розглянемо кожну характеристику та як вона впливає на вебсторінку.

Розглянемо першу характеристику “дизайн та верстка”. Дизайн та верстка є важливою складовою характеристик вебсторінок і визначають їх зовнішній вигляд і взаємодію з користувачами. Ось докладні характеристики дизайну та верстки вебсторінок:

а) дизайн:

1) кольорова палітра: вибір кольорів для фону, тексту, кнопок і інших елементів вебсторінки впливає на її загальний настрій і сприйняття користувачами;

2) шрифти: вибір шрифтів і їх розмірів впливає на читабельність та вигляд тексту на сторінці;

3) макет: організація розміщення елементів на сторінці, включаючи заголовки, зображення, текстовий контент, кнопки та інші компоненти;

б) верстка:

1) адаптивність: забезпечення того, що вебсторінка коректно відображається на різних пристроях та розмірах екранів, включаючи комп'ютери, смартфони, планшети та інші;

2) крос-браузерна сумісність: переконання, що сторінка виглядає та працює правильно на різних веб-браузерах, таких як Chrome, Firefox, Safari, Edge та інші;

в) інтерактивність:

1) анімація та переходи: використання анімації та переходів для покращення взаємодії користувачів з вебсторінкою;

г) доступність:

1) читаємість для людей з обмеженими можливостями: забезпечення того, щоб текст та зображення були доступні для всіх користувачів, включаючи тих, хто використовує читаючі програми та інші допоміжні технології.

Загалом, дизайн та верстка вебсторінок визначають, як користувачі бачать та взаємодіють з вмістом сторінки. Дотримання вищезазначених характеристик допомагає забезпечити, що вебсторінки будуть привабливими, функціональними, доступними та ефективними для користувачів.

Тепер розглянемо “навігація”. Навігація вебсторінок - це спосіб, яким користувачі переміщаються та взаємодіють з контентом на вебсайті. Вона включає в себе:

- меню та панелі навігації: головне меню та підменю, що містять посилання на різні сторінки та розділи сайту;
- посилання та кнопки: текстові посилання і кнопки для переходу на інші сторінки та виконання дій;
- пошукові інструменти: поле пошуку та фільтри для пошуку конкретної інформації на сайті;
- схеми розташування сторінок: хлібні крихти та карта сайту, які допомагають користувачам зорієнтуватися на сайті.

Ефективна навігація забезпечує зручний доступ до інформації та полегшує користувачам взаємодію з вебсторінками.

Наступна характеристика – це “Вміст”. Вміст вебсторінок визначає, яка інформація, текст, зображення та мультимедіа містяться на сторінці. Основні аспекти характеристики вмісту вебсторінок включають:

- текстовий контент: опис продуктів, послуг, статті, новини, інструкції тощо.
- зображення та мультимедіа: фотографії, відео, аудіо, графіка, ілюстрації, інфографіка.

- заголовки та структура: використання заголовків і підзаголовків для організації контенту та покращення читабельності.

- лінки та посилання: посилання на інші сторінки вебсайту або зовнішні ресурси, які дозволяють користувачам переходити до інших сторінок або джерел інформації.

- актуальність та релевантність: забезпечення того, що вміст відповідає потребам та очікуванням цільової аудиторії.

Вміст вебсторінок визначає, наскільки корисною, цікавою та зрозумілою буде сторінка для відвідувачів, і він є ключовим елементом привертання і утримання користувачів на вебсайті.

Одна з найцікавіших як на мою думку – це “SEO-оптимізація”. SEO-оптимізація - це набір стратегій та технік, спрямованих на покращення видимості вебсторінок у пошукових системах, таких як Google, Bing, Yahoo та інші. Коротка характеристика SEO-оптимізації включає такі аспекти:

- ключові слова: вибір та використання важливих ключових слів у вмісті сторінок для покращення їх ранжування в результатах пошуку;

- метатеги: використання метатегів заголовка, опису та ключових слів для оптимізації відображення сторінки в пошукових результатах;

- структура та семантика: використання правильної структури HTML, включаючи заголовки H1, H2, H3, та семантично правильних тегів для кращого розуміння контенту пошуковими системами;

- аналітика та моніторинг: використання аналітичних інструментів для відстеження та аналізу ефективності SEO-стратегії та внесення виправлень.

SEO-оптимізація допомагає підвищити видимість вебсторінок у пошукових системах, забезпечити більше органічного трафіку та залучити цільову аудиторію.

Наступна характеристика – “швидкість завантаження”. Швидкість завантаження вебсторінок - це час, необхідний для відображення сторінки

на екрані користувача після її запити. Коротка характеристика швидкості завантаження включає наступні аспекти:

- час завантаження: це час від моменту, коли користувач натискає на посилання або вводить URL-адресу сторінки, до моменту, коли сторінка повністю завантажена і готова для взаємодії;

- вплив на користувачів: швидкість завантаження має великий вплив на користувачів, оскільки швидка сторінка забезпечує позитивний користувацький досвід, тоді як повільна сторінка може викликати роздратованість і втрату відвідувачів;

- фактори впливу: швидкість завантаження залежить від різних факторів, включаючи розмір сторінки, оптимізацію зображень, хостинг, кешування та інші.

Загалом, швидкість завантаження є важливою характеристикою, яка впливає на користувацький комфорт та успіх вебсторінок.

Мобільна доступність вебсторінок означає, що сторінки оптимізовані для відображення та взаємодії на мобільних пристроях, таких як смартфони та планшети. Коротка характеристика мобільної доступності включає такі аспекти:

- адаптивний дизайн: використання дизайну, який автоматично адаптується до різних розмірів екрану, забезпечуючи правильне відображення контенту на мобільних пристроях;

- мобільні версії сторінок: створення окремих версій сторінок, оптимізованих для мобільних пристроїв, з меншим обсягом та швидким завантаженням;

- взаємодія на сенсорних екранах: забезпечення можливості легкої навігації та взаємодії з контентом за допомогою сенсорних жестів;

- швидкість завантаження на мобільних мережах: оптимізація сторінок для швидкого завантаження на повільних мобільних мережах.

Мобільна доступність важлива, оскільки все більше користувачів використовують мобільні пристрої для доступу до вебсайтів. Забезпечення правильної мобільної доступності допомагає покращити користувацький досвід та досягти більшої аудиторії.

Безпека вебсторінок - це заходи та стратегії, спрямовані на забезпечення безпечного та надійного доступу до вебсторінок. Коротка характеристика безпеки доступності включає такі аспекти:

- HTTPS-шифрування: використання протоколу HTTPS для зашифрованого передавання даних між користувачем і сервером, забезпечуючи конфіденційність та цілісність інформації;

- аутентифікація та авторизація: використання механізмів аутентифікації та авторизації для контролю доступу до конфіденційної інформації та функціональності;

- політика паролів: застосування вимогливих політик щодо паролів користувачів, включаючи сильні паролі та регулярну зміну;

- обмеження доступу: встановлення обмежень доступу до адміністративних панелей та інших критичних ресурсів.

Безпека є важливою для захисту інформації, забезпечення довіри користувачів та уникнення можливих загроз для вебсторінок і їх відвідувачів.

Взаємодія та функціональність вебсторінок визначають способи, якими користувачі можуть взаємодіяти з вебсайтом та функціональність, яку сторінки надають. Коротка характеристика взаємодії та функціональності включає наступні аспекти:

- взаємодійні елементи: різні способи взаємодії користувачів зі сторінками, такі як кнопки, форми, поля введення, списки, чекбокси тощо;

- функціональність: список можливостей та опцій, які сторінки надають користувачам, включаючи фільтрацію, сортування, покупки, реєстрацію, коментування тощо;

- динамічний контент: використання JavaScript та інших технологій для створення динамічного контенту, який оновлюється без перезавантаження сторінки;
- зворотний зв'язок та підтримка: забезпечення можливості користувачам надсилати запити, відгуки та отримувати підтримку в разі проблем;
- адаптація до різних пристроїв: забезпечення правильної взаємодії та функціональності на різних типах пристроїв, включаючи комп'ютери, смартфони та планшети.

Взаємодія та функціональність визначають, наскільки ефективно користувачі можуть взаємодіяти з вебсторінками та використовувати їх для досягнення своїх цілей.

Аналітика та відстеження вебсторінок - це процес збору та аналізу даних про відвідування та взаємодію користувачів з вебсайтом. Коротка характеристика аналітики та відстеження включає такі аспекти:

- збір даних: використання аналітичних інструментів, таких як Google Analytics, для збору різноманітних даних про користувачів, їх поведінку та інші метрики;
- аналіз результатів: оцінка та аналіз зібраних даних для зрозуміння, як користувачі взаємодіють зі сторінками та як можна покращити їх досвід;
- звітність та сповіщення: створення звітів та сповіщень на основі аналізу даних для інформування про стан та тенденції;
- оптимізація: використання результатів аналізу для оптимізації вебсторінок та стратегії для досягнення кращих результатів.

Аналітика та відстеження є важливою частиною управління та вдосконалення вебсторінок, допомагаючи власникам сайтів розуміти, як їхні сторінки використовуються та як можна покращити їх продуктивність.

Сумісність та браузерни - це характеристика, яка визначає, наскільки добре вебсторінки працюють на різних веб-браузерах. Коротка характеристика цієї особливості включає наступні аспекти:

- крос-браузерна сумісність: здатність сторінок працювати коректно та однаково на різних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari та інші;
- тестування на різних браузерах: використання спеціалізованих інструментів та тестування для перевірки відображення та функціональності сторінок на різних браузерах;
- адаптація до різних версій браузерів: забезпечення правильної роботи сторінок на різних версіях одного браузера;
- вирішення проблем сумісності: розробка та впровадження засобів для вирішення проблем сумісності на різних браузерах.

Сумісність та браузерни важливі, оскільки різні користувачі можуть використовувати різні браузерни, і важливо, щоб вебсторінки правильно відображалися та працювали на всіх них, забезпечуючи позитивний користувацький досвід.

## 1.2 Аналіз існуючих інструментів оцінки характеристик та вплив фрейсворків на ці характеристики

На сьогоднішній день існує багато інструментів для оцінки різних характеристик вебсторінок. Ці інструменти допомагають веб-розробникам та власникам сайтів зрозуміти, наскільки їх сторінки оптимізовані і виконуються. Ось огляд кількох популярних інструментів та підходів для оцінки характеристик вебсторінок:

- PageSpeed Insights (від Google): цей інструмент оцінює швидкість завантаження вебсторінки на мобільних та настільних пристроях. Він також надає рекомендації щодо оптимізації;

– аналіз сторінки вручну: інспектування HTML, CSS та JavaScript коду сторінки за допомогою вбудованих розробницьких інструментів у браузері. Це дозволяє перевірити дизайн, структуру та інші характеристики сторінки;

– Lighthouse (від Google): Lighthouse аудитує якість та продуктивність вебсторінки, враховуючи такі аспекти, як швидкість завантаження, SEO, доступність та інше;

– Pingdom: цей інструмент вимірює час завантаження сторінки та надає докладну інформацію щодо часу завантаження кожного елемента сторінки;

– GTmetrix: GTmetrix пропонує аналіз швидкості завантаження та дозволяє виміряти час першого байта, час завантаження сторінки та інші метрики швидкості;

– WebPageTest: цей інструмент дозволяє провести тестування швидкості завантаження сторінки з різних місць у світі та на різних пристроях;

– Google Analytics: використовуючи Google Analytics, можна отримати дані про відвідуваність сторінок, взаємодію користувачів та інші аналітичні дані;

– якість коду та валідація: інструменти для аналізу якості HTML, CSS та JavaScript коду, такі як W3C Validator, допомагають виявити помилки та невідповідності стандартам;

– SEO-аудитори: використовуючи різні SEO-аудитори, можна аналізувати ключові слова, мета-теги, індексацію та інші параметри, що впливають на SEO;

– тестування на різних пристроях і браузерах: для вимірювання сумісності та адаптивності сторінок можна використовувати фізичні тести на різних пристроях та віртуальні машини для тестування на різних браузерах;

– сервіси моніторингу навігації користувачів: сервіси, які відслідковують поведінку користувачів на сторінці, допомагають оцінити ефективність навігації та інтерфейсу.

Ці інструменти можуть бути використані для оцінки різних аспектів вебсторінок, включаючи швидкість завантаження, SEO-оптимізацію, мобільну доступність та багато інших характеристик. Вибір інструментів залежить від конкретних потреб та цілей веб-розробника або власника сайту.

Існує безліч фреймворків та бібліотек, які використовуються для розробки вебсторінок та додатків. Вони надають різні можливості та функціональність для полегшення розробки та покращення користувацького досвіду. Але найчастіше використовується JavaScript фреймворки для вирішення різних проблем, пов'язаних з характеристиками вебсторінок. Ось декілька таких фреймворків і бібліотек які будуть досліджуватися в роботі:

– React: React створений для ефективного управління інтерфейсом користувача. Він допомагає створювати компоненти, які оновлюються ефективно, що поліпшує реакцію сторінки на дії користувача. Це допомагає підвищити реактивність сторінки;

– Angular: Angular надає структурований підхід до розробки односторінкових додатків (SPA). Він допомагає створювати добре організований код, що полегшує розробку та підтримку сторінок з великою кількістю функціональності;

– Vue.js: Vue.js спрощує розробку користувацьких інтерфейсів та дозволяє створювати динамічні сторінки. Він легкий і має хорошу продуктивність, що сприяє швидкій роботі сторінок;

– Svelte: Svelte робить роботу з DOM-елементами та подіями більш простою. Він полегшує розробку та взаємодію з елементами сторінки;

- Ember.js: надає структуру та конвенції для швидкого старту розробки;
- Backbone.js: Backbone.js - це легкий фреймворк для структурування веб-додатків. Надає моделі, колекції та види для організації коду;
- Astro: новий підхід у рендері компонентів який спрямований на швидке завантаження сторінок, завдяки частковій гідратації та рендерингу на стадії збірки.

Підводячи висновок можна побачити, що на даний момент існує дуже багато факторів які впливають на характеристики вебсторінок, та найбільш фундаментальну різницю дає технологія на якій буде розроблюватися ця сторінка.

### 1.3 Постановка задачі дослідження

Об'єктом дослідження є JavaScript-фреймворки та їх вплив на функціональні характеристики вебсторінок (зокрема React, Vue.js та Angular).

Предмет дослідження є вплив JavaScript-фреймворків (React, Vue.js та Angular) на функціональні характеристики вебсторінок, такі як швидкість завантаження, відзивчивість та візуальна привабливість.

У рамках цього дослідження передбачається вирішення наступних задач:

- оцінка та порівняння продуктивності вебсторінок, розроблених з використанням різних фреймворків, для визначення, як технічні аспекти фреймворків впливають на швидкість відгуку та час завантаження;

- аналіз доступності вебсайтів з метою виявлення можливих бар'єрів для користувачів з обмеженими можливостями та розробка рекомендацій для їх усунення;
- дослідження впливу фреймворків на SEO-показники, зокрема на ранжування вебсторінок у пошукових системах, та розробка стратегій для покращення видимості вебсайтів;
- оцінка впливу розміру вебсторінок та їхньої складності на загальну продуктивність та визначення методів оптимізації ваги сторінок;
- аналіз крос-браузерної сумісності та міжнародної локалізації, а також розробка підходів для поліпшення глобальної доступності веб-рішень;
- вивчення потенціалу фреймворків для масштабування та підтримки великих та складних веб-додатків, а також аналіз можливостей їх інтеграції з різними вебсервісами та API.

Метою дослідження є проведення детального аналізу впливу різних JavaScript frameworks на функціональні характеристики вебсторінок і визначити, який фреймворк може найбільше покращити швидкість завантаження, відзивчивість та візуальну привабливість.

## 2 ОПИС ОБРАНИХ ТЕХНОЛОГІЙ ТА МЕТОДІВ ПРОЕКТУВАННЯ З РЕАЛІЗАЦІЇ СИСТЕМИ

### 2.1 Аналіз існуючих інструментів оцінки характеристик

Для аналізу характеристик вебсторінок було обрано Lighthouse як основний інструмент [6]. Lighthouse - це інструмент для аудиту вебсторінок, розроблений командою Google, і він має декілька переваг, які роблять його привабливим для використання в порівнянні з іншими інструментами для оцінки характеристик вебсторінок. Наведемо головні переваги Lighthouse:

**Відкриті джерела:** Lighthouse - це вільний і відкритий інструмент, доступний для використання безкоштовно. Ви можете легко використовувати його для оцінки вебсторінок без необхідності платити за ліцензію чи підписку.

**Широкий спектр аудитів:** Lighthouse проводить аудит вебсторінок для різних аспектів, таких як продуктивність, доступність, кольори, SEO і багато інших. Це робить його універсальним інструментом для загальної оцінки якості вебсторінок.

**Інтеграція з Chrome DevTools:** Lighthouse легко доступний через вбудований інструмент Chrome DevTools, що дозволяє аналізувати сторінки безпосередньо в браузері. Це полегшує використання інструмента та надає додаткові зручності для розробників.

**Звіти і рекомендації:** Lighthouse генерує докладні звіти з оцінкою кожного аспекту сторінки і надає конкретні рекомендації для покращення. Це допомагає розробникам і веб-мастерам зрозуміти, які саме аспекти потребують покращення.

Підтримка технологій: Lighthouse підтримує останні веб-технології і стандарти, такі як PWA (прогресивні веб-додатки) і AMP (прискорений мобільний сторінки). Він також оцінює збережені офлайн ресурси і багато інших аспектів.

Зважаючи на ці переваги, Lighthouse є потужним інструментом для оцінки характеристик вебсторінок, який може бути використаний для покращення якості та ефективності вашого вебсайту.

Для більш комплексної оцінки було вирішено використовувати WebPageTest [7]. WebPageTest - це потужний інструмент для оцінки різних аспектів вебсторінок і забезпечення кращої продуктивності та швидкості їх завантаження. Ось декілька головних переваг WebPageTest:

- докладна аналітика завантаження сторінки: WebPageTest надає докладну інформацію про час завантаження сторінки на різних регіонах та пристроях. Ви можете переглядати графіки, які вказують, коли і як сторінка стає доступною для користувачів, і як швидко вона завантажується;

- аналіз впливу різних факторів: WebPageTest дозволяє вам експериментувати з різними параметрами завантаження, такими як швидкість з'єднання та пристрій, щоб визначити, як це впливає на продуктивність сторінки;

- перевірка сторінки на різних пристроях: ви можете вибрати конкретний пристрій (наприклад, мобільний телефон або планшет) і регіон для аналізу, щоб побачити, як ваша сторінка завантажується для користувачів з різних місць і на різних пристроях;

- розгорнуті рекомендації для поліпшення швидкості: WebPageTest надає конкретні рекомендації для покращення продуктивності сторінки. Ви можете отримати вказівки щодо оптимізації зображень, зменшення запитів на сервер, кешування і багато іншого;

– перевірка на безпеку: ви можете перевірити сторінку на проблеми безпеки, такі як використання незахищеного протоколу передачі даних (HTTP), кросс-сайтові атаки і багато інших.

WebPageTest допомагає розробникам і веб-мастерам отримувати докладну інформацію про продуктивність і швидкість своїх вебсторінок та приймати заходи для покращення їх якості та ефективності.

Підводячи підсумок, використання WebPageTest та Lighthouse в комплексі допомагає отримати більше повну та докладну інформацію про різні аспекти вебсторінок, такі як продуктивність, доступність, SEO та інші. Це дозволяє робити більш чітку та точну оцінку характеристик вебсторінок.

## 2.2 Формула оцінки характеристик вебсторінки

Реалізація вебсторінки на різних фреймворках може призвести до різних покращень і збільшень певних характеристик сторінки. Ось кілька головних характеристик вебсторінки, які можуть значно відрізнятись при використанні певних фреймворків:

- продуктивність і швидкість завантаження;
- адаптивність та мобільність;
- підтримка SEO;
- реактивність і взаємодія;
- використання ресурсів та швидкість завантаження.

Залежно від конкретних вимог і завдань вебсторінки, ви можете вибрати фреймворк, який найкраще підходить для досягнення певних цілей. Різні фреймворки мають свої сильні сторони та особливості, і вибір фреймворку повинен базуватися на вашій конкретній ситуації та потребах в

проекті. Для більш достовірного аналізу характеристик – головний функціонал та вебсторінки будуть однакові.

Формула для оцінки загальної характеристики вебсторінки може бути складною і залежить від багатьох факторів, таких як продуктивність, доступність, SEO і багато інших. Зазвичай така формула складається з вагових коефіцієнтів для різних характеристик і показників, які оцінюються на основі конкретних метрик.

Нехай формула для оцінки загальної характеристики сторінки буде наступна:

$$R = \sum x_a w_a,$$

де  $R$  – загальна оцінка сторінки;

$a$  – характеристика вебсторінки;

$x_a$  – оцінка характеристики вебсторінки;

$w_a$  – вагові коефіцієнт певної характеристики, які визначають важливість кожної характеристики в загальній оцінці. Вагові коефіцієнти можуть бути встановлені вручну або на основі ваших пріоритетів.

Коефіцієнти для оцінки характеристик вебсторінок можуть бути налаштовані в залежності від конкретних цілей та вимог проекту. Ось приклади коефіцієнтів для різних характеристик сторінок та їх обґрунтування:

а) продуктивність (Perfomance):

1) ваговий коефіцієнт: 0.3;

2) обґрунтування: швидкість завантаження сторінки має велике значення для користувачів, оскільки повільні сторінки можуть призвести до поганих вражень і втрати відвідувачів;

б) доступність (Accessibility):

1) ваговий коефіцієнт: 0.2;

2) обґрунтування: доступність сторінки важлива для користувачів з обмеженими можливостями та для відповідності законодавству;

в) SEO (Seo):

1) ваговий коефіцієнт: 0.2;

2) обґрунтування: гарний SEO допомагає залучати більше органічного трафіку і покращує видимість в пошукових системах;

г) Best Practice:

1) ваговий коефіцієнт: 0.2;

2) обґрунтування: забезпечення дотримання "Best Practices" допомагає уникати помилок і недоречностей, які можуть призвести до проблем з безпекою, продуктивністю і доступністю;

д) Progressive Web App (PWA):

1) ваговий коефіцієнт: 0.1;

2) обґрунтування: PWA можуть покращити взаємодію користувачів з вашим веб-додатком, забезпечуючи офлайн-доступ, повідомлення та інші корисні функції.

### 2.3 Аналіз кожного фреймворка та його вплив на характеристики

Аналіз React та його вплив на ці характеристики у порівнянні з іншими та які характеристики мають позитивний та негативний вплив та чому.

Потрібно проаналізувати усі фреймворки React, Angular, Next.js, Vue.js, Svelte, Backbone.js, Ember.js, Astro

Почнемо з React. Оцінемо як React впливає на характеристики.

Позитивний вплив React на характеристики:

- продуктивність: React може бути дуже продуктивним завдяки віртуальному DOM і можливості оптимізувати оновлення компонентів. Це дозволяє швидко рендерити зміни на сторінці;

- SEO (пошукова оптимізація): з використанням Server-Side Rendering (SSR) та підтримкою роутингу React дозволяє створювати SEO-дружні сторінки з індексованим вмістом;

- Best Practices: React сприяє дотриманню найкращих практик у веб-розробці, завдяки своїй структурі і можливостям компонентів;

- Progressive Web App (PWA)[8]: React може легко інтегрувати функції PWA, що дозволяє покращити користувацький досвід.

Негативний вплив React на характеристики:

- доступність: React не гарантує автоматичної доступності. Розробники повинні свідомо дотримуватися стандартів доступності і виконувати відповідні налаштування;

- вага сторінки: реакт-додатки можуть бути великими за розміром, особливо, якщо не виконана оптимізація. Це може вплинути на продуктивність та швидкість завантаження сторінок;

- складність навчання: навчання розробників React може займати більше часу, особливо для початківців, оскільки React використовує JSX та має свою власну концепцію компонентів;

- інтеграція з іншими технологіями: під час інтеграції React з іншими бібліотеками та технологіями можуть виникнути проблеми сумісності, що потребують додаткових зусиль.

Загалом, React має багато переваг і може бути дуже корисним для розробки сучасних веб-додатків, особливо якщо він використовується правильно і з врахуванням найкращих практик. Але розробники повинні бути обережні та використовувати його з урахуванням конкретних вимог проекту для досягнення найкращих результатів.

Аналіз впливу Angular на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином[9]:

#### Позитивний вплив Angular:

- продуктивність: Angular надає інструменти для створення ефективних та продуктивних веб-додатків. Вбудована оптимізація та швидкість рендерінгу полегшують розробку продуктивних сторінок;
- SEO (пошукова оптимізація): з використанням Angular Universal, який надає можливість Server-Side Rendering (SSR), Angular може покращити індексацію сторінок пошуковими системами;
- Best Practices: Angular пропонує структуру та конвенції, що сприяють дотриманню найкращих практик у веб-розробці;
- доступність: Angular активно підтримує стандарти доступності та надає інструменти для створення доступних веб-додатків.

#### Негативний вплив Angular:

- вага сторінки: Angular-додатки можуть бути великими за розміром, особливо якщо не виконана оптимізація. Великий обсяг коду може вплинути на швидкість завантаження;
- складність навчання: Angular може бути складним для вивчення, особливо для новачків у веб-розробці, через велику кількість концепцій та обов'язкового використання TypeScript;
- інтеграція з іншими технологіями: інтеграція Angular з іншими технологіями може бути складною та вимагати додаткових зусиль для забезпечення сумісності.

Загалом, Angular має багато позитивних аспектів, таких як продуктивність, доступність та підтримка SEO, але він може бути менш підходящим для менших проектів або розробників, які шукають менш складний фреймворк. Важливо враховувати конкретні потреби проекту та навички команди розробників при виборі між Angular та іншими технологіями.

Аналіз впливу Next.js на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином:

Позитивний вплив Next.js:

- продуктивність: Next.js надає можливість Server-Side Rendering (SSR), що поліпшує швидкість завантаження сторінок та зменшує час першого відображення (First Contentful Paint). Це сприяє покращенню продуктивності;
- SEO (пошукова оптимізація): Next.js підтримує Server-Side Rendering, що допомагає поліпшити індексацію сторінок пошуковими системами;
- доступність: Next.js дозволяє легко дотримуватися стандартів доступності та створювати доступні веб-додатки;
- Best Practices: Next.js сприяє дотриманню найкращих практик завдяки структурі проекту та інтеграції зі стандартами розробки.

Негативний вплив Next.js:

- вага сторінки: Next.js-додатки можуть бути великими за розміром, особливо якщо не виконана оптимізація. Великий обсяг коду та завантажуваних ресурсів може вплинути на швидкість завантаження сторінок;
- складність навчання: Next.js може вимагати часу та зусиль для вивчення, особливо для новачків у веб-розробці, які не мають досвіду з реакт-подібними фреймворками;
- інтеграція з іншими технологіями: інтеграція Next.js з іншими технологіями може бути складною та вимагати зусиль для забезпечення сумісності.

Загалом, Next.js відомий своєю продуктивністю, підтримкою SEO та доступності, а також полегшує використання найкращих практик у веб-розробці. Однак, важливо бути уважним щодо розміру сторінки та необхідної оптимізації для досягнення найкращих результатів.

Аналіз впливу Vue.js на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином[10]:

Позитивний вплив Vue.js:

- продуктивність: Vue.js дозволяє створювати веб-додатки швидко і ефективно завдяки легкій синтаксису та можливостям компонентів. Він також використовує віртуальний DOM для ефективного рендерінгу сторінок;

- SEO (пошукова оптимізація): з використанням Server-Side Rendering (SSR) Vue.js може покращити індексацію сторінок пошуковими системами;

- доступність: Vue.js підтримує стандарти доступності та надає інструменти для створення доступних веб-додатків;

- Best Practices: Vue.js сприяє дотриманню найкращих практик завдяки чіткої структурі та концепції компонентів.

Негативний вплив Vue.js:

- вага сторінки: Vue.js-додатки можуть бути великими за розміром, особливо якщо не виконана оптимізація. Великий обсяг коду та завантажуваних ресурсів може вплинути на швидкість завантаження;

- складність навчання: Для новачків Vue.js може виглядати складним, хоча менш складним порівняно з іншими реактивними фреймворками;

- інтеграція з іншими технологіями: інтеграція Vue.js з іншими технологіями може бути складною, особливо якщо потрібна взаємодія з іншими фреймворками або бібліотеками.

Загалом, Vue.js володіє багатьма перевагами, такими як продуктивність, доступність та підтримка SEO, а також спрощує дотримання найкращих практик у веб-розробці. Однак, аналіз розміру сторінки та оптимізація є важливими для досягнення найкращої продуктивності і швидкості завантаження сторінок.

Аналіз впливу Svelte на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином:

#### Позитивний вплив Svelte:

- простота використання: Svelte відомий своєю простотою та легкістю використання. Він дозволяє легко взаємодіяти з DOM-елементами та обробляти події;
- кросбраузерність: Svelte надає шар абстракції, що полегшує роботу з різними браузерами, спрощуючи роботу з різними API браузера;
- швидкість розробки: Svelte дозволяє швидко створювати прості веб-додатки та взаємодіяти з DOM без необхідності писати велику кількість власного JavaScript-коду.

#### Негативний вплив Svelte:

- вага сторінки: великий розмір бібліотеки Svelte може призвести до збільшення розміру сторінки, що вплине на швидкість завантаження;
- продуктивність: Svelte не завжди ефективний для створення складних веб-додатків, і його використання може призвести до менш ефективної продуктивності порівняно з сучасними фреймворками;
- складність підтримки та розширення: підтримка та розширення великих проектів на Svelte може бути складною через відсутність модульності та організації коду;
- сучасність та тривалий розвиток: Svelte не завжди надає сучасні можливості та не отримує такого активного розвитку, як сучасні фреймворки та бібліотеки.

Загалом, використання Svelte може бути корисним для простих проектів та сценаріїв, особливо коли важлива простота та кросбраузерність. Проте для складніших та сучасних веб-додатків, розглядаються інші фреймворки та бібліотеки, які надають більші можливості та продуктивність.

Аналіз впливу Backbone.js на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином:

Позитивний вплив Backbone.js:

- простота використання: Backbone.js пропонує простий та легкий для зрозуміння патерн розробки, що полегшує створення веб-додатків;
- кросбраузерність: Backbone.js не впливає на кросбраузерність, і ви можете легко створювати кросбраузерні додатки за допомогою цього фреймворку;
- модульність: Backbone.js не накладає обмежень на структуру вашого проекту, дозволяючи розробникам вибирати та інтегрувати різні бібліотеки та рішення за потреби.

Негативний вплив Backbone.js:

- вага сторінки: Backbone.js не надає інструментів для оптимізації розміру сторінки, і велика кількість коду може призвести до збільшення розміру сторінки;
- складність розробки великих проектів: Backbone.js може бути менш підходящим для великих та складних веб-додатків, оскільки він не надає велику кількість готових рішень для обробки складних завдань;
- сучасність та розвиток: Backbone.js не отримує такого активного розвитку та підтримки, як сучасні фреймворки та бібліотеки. Це може призвести до відсталості за функціональністю та засобами розробки.

Загалом, Backbone.js може бути використаний для створення простих веб-додатків, особливо якщо важливі простота та кросбраузерність. Проте для великих та складних проектів можуть бути кращі альтернативи, які надають більше можливостей та підтримки.

Аналіз впливу Ember.js на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином[11]:

Позитивний вплив Ember.js:

- продуктивність: Ember.js надає велику кількість готових рішень та конвенцій, що полегшує розробку веб-додатків і сприяє продуктивності розробників;

- SEO (пошукова оптимізація): з можливістю Server-Side Rendering (SSR), Ember.js покращує індексацію сторінок пошуковими системами;

- доступність: Ember.js активно підтримує стандарти доступності та надає інструменти для створення доступних веб-додатків;

- модульність: Ember.js використовує концепцію аддонів та модульної структури, що спрощує розширення та підтримку коду.

Негативний вплив Ember.js:

- вага сторінки: Ember.js-додатки можуть бути великими за розміром, і великий обсяг коду може призвести до збільшення розміру сторінки, що вплине на швидкість завантаження;

- складність навчання: Ember.js може бути складним для вивчення для новачків у веб-розробці через велику кількість концепцій та інші особливості фреймворку;

- інтеграція з іншими технологіями: інтеграція Ember.js з іншими технологіями може бути складною, особливо якщо потрібна взаємодія з іншими фреймворками або бібліотеками.

Загалом, Ember.js відомий своєю продуктивністю, доступністю та підтримкою SEO, але великий розмір сторінки може вимагати додаткової оптимізації. Важливо враховувати конкретні потреби проекту та навички команди розробників при виборі між Ember.js та іншими технологіями.

Аналіз впливу Astro на різні характеристики вебсторінок у порівнянні з іншими технологіями може виглядати наступним чином:

Позитивний вплив Astro:

- продуктивність: Astro надає високу продуктивність завдяки підтримці віртуального DOM та можливості Server-Side Rendering (SSR), що покращує швидкість завантаження сторінок;

- вага сторінки: Astro використовує підходи для мінімізації розміру сторінки та оптимізації завантаження ресурсів, що сприяє високій продуктивності та швидкому відображенню контенту;
- доступність: Astro надає інструменти для створення доступних веб-додатків та дотримання стандартів доступності;
- SEO (пошукова оптимізація): з можливістю Server-Side Rendering (SSR), Astro покращує індексацію сторінок пошуковими системами.

Негативний вплив Astro:

- специфічний синтаксис: Astro використовує власний синтаксис та концепції, що може вимагати часу для вивчення для розробників, які раніше не працювали з цим фреймворком;
- інтеграція з іншими технологіями: інтеграція Astro з іншими технологіями може бути складною, особливо якщо потрібна взаємодія з іншими фреймворками або бібліотеками;
- рівень підтримки та екосистема: Astro ще не має такої розгалуженої екосистеми та підтримки, як деякі інші фреймворки, що може призвести до обмеження вибору готових рішень та бібліотек.

Загалом, Astro відомий своєю продуктивністю, мінімальним розміром сторінки та підтримкою доступності та SEO. Проте важливо враховувати, що це ще відносно новий фреймворк, і він може вимагати часу для вивчення та опанування. Остаточний вибір залежить від конкретних потреб проекту та досвіду розробників.

Задля більш наочного порівняльного аналізу, ми розробили таблицю, в якій зібрані дані, отримані в результаті аналізу різних фреймворків. Переглянути цю інформацію можна в таблиці 2.1.

Таблиця 2.1 – Таблиця з оцінками характеристик для кожного фреймворку

Фреймворк	Продуктивність	Вага сторінки	Доступність	SEO	Інтеграція	Рівень підтримки	Загальна оцінка
Backbone.js	3	3	3	3	3	3	3
Svelte	3	3	3	3	4	3	3.2
Angular	3	3	3	3	4	4	3.3
Ember.js	4	3	4	4	3	3	3.6
Astro	5	5	4	5	3	3	4
Vue.js	4	4	4	4	4	4	4
Next.js	4	4	4	4	4	4	4
React	4	4	4	4	4	5	4.2

Аналізуючи дану таблицю, можна зробити висновок, що Backbone.js, хоча і визнаний за свою простоту та гнучкість в минулому, сьогодні може не задовольняти вимоги сучасних веб-додатків у повній мірі. Цей фреймворк отримав однакові середні бали по всіх категоріях, що може свідчити про його універсальність, але також і про відсутність виражених сильних сторін в порівнянні з іншими, більш сучасними фреймворками.

React, навпаки, вирізняється високим рівнем підтримки спільноти і широкими можливостями інтеграції, що робить його відмінним вибором для багатьох розробників. Висока оцінка у всіх категоріях, зокрема найвища за підтримку, підкреслює його міцну позицію на ринку веб-фреймворків.

Але треба пам'ятати, що вибір між фреймворками повинен залежати від конкретних потреб проекту, рівня підтримки, та вимог до продуктивності та SEO.

## 3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Вибір основного темплейту для аналіз характеристик

Серед типових сайтів існують декілька найрозповсюджених типів веб сторінок: лендінги, блоги, маркетплейси та інші:

- лендінги – це найпростіші сайти, часто складаються з однієї сторінки і мають за мету залучення уваги та швидке інформування відвідувачів;

- блоги вимагають більшої складності через необхідність управління контентом, коментарями та іншими функціями;

- маркетплейси є одними з найбільш складних типів вебсайтів для розробки, адже вони вимагають інтеграції різноманітних систем та компонентів. Окрім базової функціональності, як-от перегляд та купівля товарів, вони повинні мати розширену адміністративну панель для управління продуктами, користувацькими замовленнями, відстеженням інвентаризації, а також забезпечувати безпеку та обробку платежів. Усе це збільшує технічну складність та час, необхідний для розробки;

- інформаційні портали- це великі масштабні сайти, що містять різноманітну інформацію в різних форматах. Вони можуть включати новини, статті, довідкові матеріали та форуми;

- освітні платформи - це сайти, які забезпечують доступ до навчальних курсів та матеріалів, часто мають функціональність для відстеження прогресу навчання та проведення тестів онлайн.

У висновку, блоги займають золоту середину в спектрі складності вебсайтів. Вони більш складні за лендінги, оскільки вимагають функціональності для управління динамічним контентом та взаємодії з

користувачами, але менш комплексні порівняно з маркетплейсами, які потребують розширених систем управління продуктами, транзакціями та користувацькими даними. Блоги надають відмінний баланс між функціональністю та технічною складністю, роблячи їх доступними для розробників, які шукають ефективне рішення для розміщення контенту.

Було обрано Medium.com як базового шаблону для аналізу веб-характеристик є стратегічним рішенням. Завдяки своєму чистому дизайну та ефективності, Medium створює ідеальні умови для зосередження на контенті. Висока швидкість завантаження та оптимізація для SEO сприяють кращій органічній видимості та доступності на різних пристроях. Розробка простіших версій головної сторінки блогу, сторінки статті та профілю користувача з використанням різних фреймворків дозволить глибше зрозуміти вплив кожної технології на загальну ефективність та відзивчивість вебсайтів. Нижче надані рисунки екранів.

### 3.2 Аналіз Backend частини проекту

При порівнянні фреймворків фронтенду не потрібно змінювати бекенд, тому що метою є вимірювання виключно впливу клієнтської сторони на продуктивність і взаємодію з користувачем. Бекенд, який виконує серверну логіку та обробку даних, зазвичай має стандартизований API, що дозволяє різним фронтенд-фреймворкам ефективно взаємодіяти з однаковим набором даних для чесного порівняння.

### 3.3 Обґрунтування Backend частини

Для Backend частини було вирішено розробити Backend частину за допомогою такого стеку: Express.js + MongoDB[12] + Mongoose

Express.js спрощує розробку бекенду на Node.js, пропонуючи гнучку маршрутизацію та широкі можливості для обробки запитів і відповідей. MongoDB як NoSQL база даних забезпечує швидкість і гнучкість при роботі з великими обсягами даних і дозволяє легко змінювати схеми даних. Mongoose додає додатковий шар валідації та бізнес-логіки для MongoDB, забезпечуючи зручну роботу з об'єктами даних у JavaScript. Цей стек добре підходить для швидкої та ефективної розробки сучасних веб-додатків.

Для створення MongoDB бази даних, що підтримує функціональність схожу на Medium, будуть потрібні наступні схеми:

- User: зберігання інформації про користувачів, включаючи їхні імена, електронні адреси, біографії, зображення, улюблені статті та інформацію про підписки;
- Article: зберігання статей, включаючи унікальний slug, заголовок, опис, тіло статті, список тегів, автора, кількість лайків та коментарі;
- Comment: зберігання коментарів до статей з посиланням на автора та статтю, до якої вони належать;
- Tag: управління тегами статей, включаючи зв'язки зі статтями, які їх використовують.

ER-діаграма для проекту представлена на рисунку 3.1.

Кожна схема має валідацію і відносини, визначені за допомогою Mongoose, що спрощує взаємодію з MongoDB та забезпечує консистентність даних. Ця структура бази даних дозволяє ефективно реалізувати основні функції соціальної платформи для блогінгу.

ER Diagram for Blogging Platform

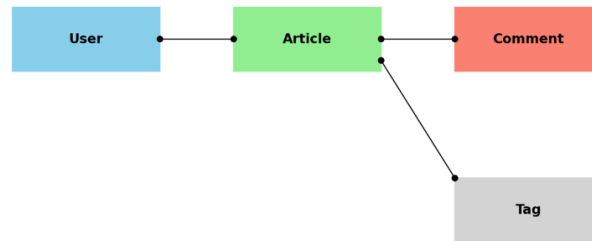


Рисунок 3.1 - ER-діаграма бази даних

Розглянемо структуру нашої Backend структури яка зображена на рисунку 3.2:

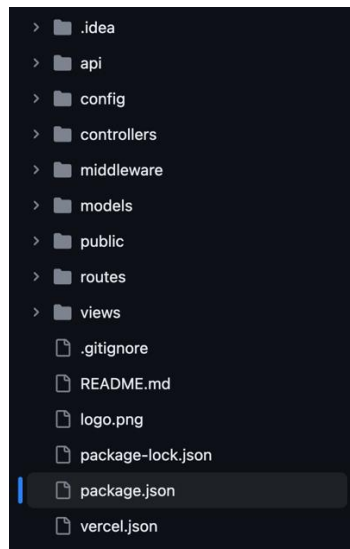


Рисунок 3.2 - Структура Backend проекту

Зображення показує структуру проекту, яка є типовою для Node.js застосунків, що використовують Express.js. Наведимо переваги та особливості такої структури:

- api: папка може містити версіоновані або груповані API ендпойнти, що спрощує управління API та його розширення;

- `config`: централізоване місце для управління конфігурацією, що сприяє безпеці та легкості змін у налаштуваннях;
- `controllers`: містить логіку обробки запитів, що дозволяє відділяти бізнес-логіку від маршрутизації, тим самим сприяючи чистоті коду;
- `middleware`: папка для зберігання проміжного ПЗ (`middleware`), яке може виконувати перехоплення запитів, авторизацію, логування тощо;
- `models`: визначає структуру даних і схеми для бази даних, що забезпечує централізований контроль над структурою даних у застосунку;
- `public`: призначена для статичних файлів, таких як зображення, стилі CSS, JavaScript файли, що доступні користувачам напряму;
- `routes`: містить визначення маршрутів для застосунку, що спрощує управління ендпойнтами та асоціацію їх з контролерами;
- `views`: для застосунків, які використовують шаблонізатор, ця папка містить шаблони сторінок;
- `.gitignore`: файл для визначення файлів та папок, які не потрібно включати в репозиторій Git;
- `README.md`: опис проекту, інструкції, документація для розробників та користувачів;
- `package.json` та `package-lock.json`: визначає залежності проекту, скрипти для запуску та іншу метаянформацію проекту.

Ця структура дозволяє організувати код застосунку згідно з принципами чистоти коду, поділу відповідальності та легкості масштабування, що робить розробку ефективною та підтримуваною.

### 3.4 Аналіз Fronted частини проекту та його структури

Структура фронтенду [13] для веб-проектів, незалежно від використовуваного фреймворку, зазвичай містить наступні компоненти:

- Components/Views: папка для UI компонентів або вьюшок, де кожен компонент або вьюшка є частиною інтерфейсу (наприклад, кнопки, форми, статті, бічні панелі) ;
- Services/Api: функції або класи, що відповідають за спілкування з API бекенду;
- Hooks: при використанні React або фреймворків, що підтримують хуки, ця папка містить перевикористовувані хуки для логіки стану та інших реактивних функцій;
- Utils/Helpers: вспомогательные утиліти, такі як форматери дат, валідатори тощо;
- Assets: статичні ресурси проекту, такі як зображення, стилі та шрифти;
- Store: у випадку Redux[14] або подібних бібліотек для управління станом, ця папка містить налаштування стору, редюсери та дії;
- Routes: визначення маршрутів для SPA[15] (Single Page Application), які вказують, який компонент завантажувати для кожного URL;
- Styles: глобальні стилі, змінні CSS та міксини для проекту.

Переваги створення за цим шаблоном:

- уніфікація: єдиний стандарт структурування проекту для всіх фреймворків спрощує взаємозамінність та розуміння проекту різними розробниками;
- підтримка: легше підтримувати та оновлювати проект, коли він слідує загальноприйнятій структурі;

- масштабування: чітке розділення компонентів та логіки сприяє легкому масштабуванню та додаванню нових функцій;
- тестування: стандартизована структура полегшує розробку тестів і забезпечує більшу покритість коду тестами.

Структура фронтенду повинна бути гнучкою, щоб можна було легко адаптувати її під специфіку використовуваного фреймворку чи бібліотеки, зберігаючи при цьому загальні принципи організації коду. Тому надалі усі frontend проекти будуть наближенні до даної структури.

### 3.5 Аналіз характеристики вебсторінок за різними фреймворками

У цьому розділі, ми проведемо детальний аналіз трьох ключових сторінок, розроблених з використанням різних фреймворків: головної сторінки, яка часто є першим місцем взаємодії відвідувачів з сайтом; сторінки статті, що демонструє здатність сайту ефективно подавати контент; та сторінки профілю, яка показує індивідуалізацію та інтерактивність. Аналізуючи ці сторінки, ми отримаємо цілісне розуміння того, як обраний фреймворк впливає на кожен аспект користувацького досвіду і загальну продуктивність вебсайту.

#### 3.5.1 Аналіз React

Проаналізуємо проект розроблений за допомогою фреймворку React. Нижче наведено метрики сторінок. Рисунок 3.3, 3.5 та 3.7 – зображують

метрики пов'язані з Lighthouse, а рисунки 3.4, 3.6 та 3.8 – зображують метрики пов'язані з WebPageTest.

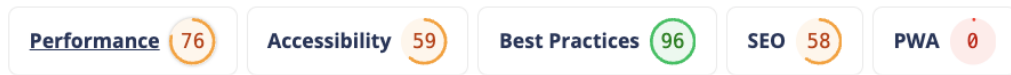


Рисунок 3.3 - Головна сторінка - Lighthouse метрики

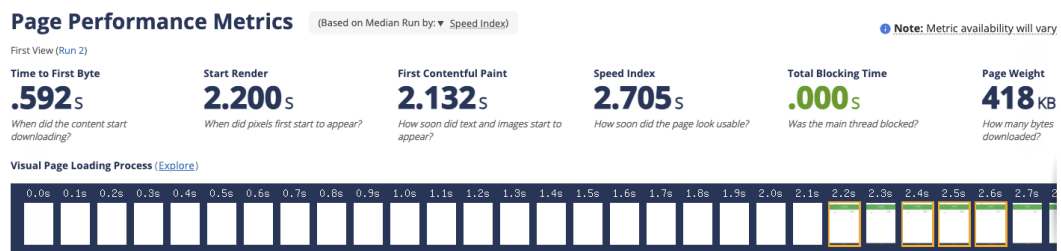


Рисунок 3.4 - Головна сторінка - WebPageTest метрики

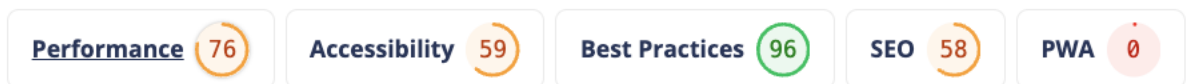


Рисунок 3.5 - Сторінка статті - Lighthouse метрики

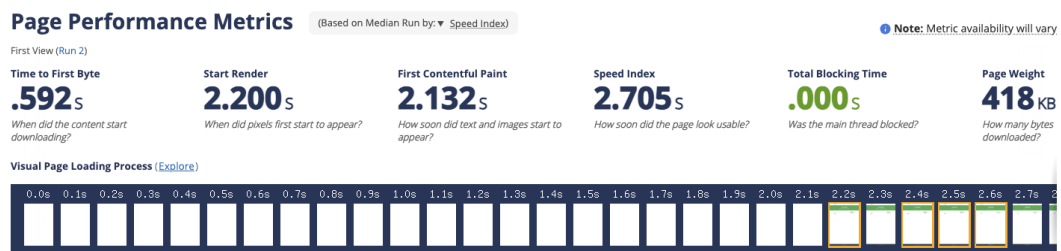


Рисунок 3.6 - Сторінка статті - WebPageTest метрики

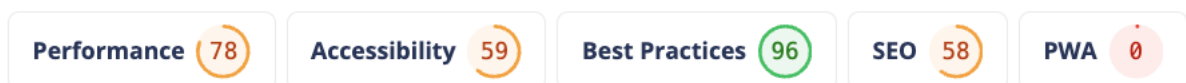


Рисунок 3.7 - Сторінка профілю - Lighthouse метрики

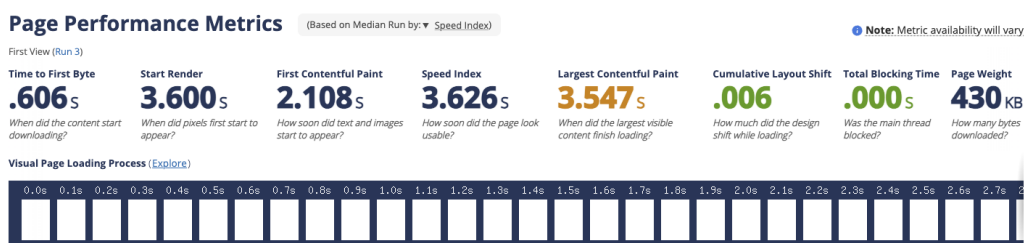


Рисунок 3.8 - Сторінка профілю - WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність: 59;
- кращі практики: 96;
- SEO: 58.

Наведемо таблицю (таблиця 3.1) з такими вебхарактеристиками та виведемо середнє значення для подальшого аналізу. У таблицю додамо такі вебхарактеристики:

- продуктивність;
- час до інтерактивності;
- індекс швидкості;
- перший змістовий розпис;
- загальний час блокування;
- вага сторінки;
- оцінка сторінки.

Проведемо аналіз усіх характеристик:

- продуктивність сторінок має відносно високі оцінки;
- кращі практики і SEO мають стабільно високі оцінки, що свідчить про дотримання рекомендованих вебстандартів;
- часи до інтерактивності є досить низькими, що є позитивним індикатором швидкості сайту;

- індекс швидкості показує хороші результати з можливістю дрібної оптимізації;
- час блокування має низькі значення, що свідчить про стабільність відображення вмісту та відсутність великих затримок;
- вага сторінки є помірною, що сприяє швидкому завантаженню.

Таблиця 3.1 - Таблиця характеристик React

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	76	76	78	76.67
Час до Інтерактивності	5.92	5.92	6.06	5.97
Індекс швидкості	2.2	2.2	3.6	2.67
Перший змістовий розпис	2.13	2.13	2.10	2.12
Загальний час блокування	418	418	430	422
Вага сторінки	418	418	430	422
Оцінка сторінки	57.8	57.8	58.2	57.93

### 3.5.2 Аналіз Ember

Проаналізуємо проект розроблений за допомогою фреймворку Ember. Нижче наведено метрики сторінок. Рисунок 3.9, 3.11 та 3.13 – зображують метрики пов'язані з Lighthouse, а рисунки 3.10, 3.12 та 3.14 – зображують метрики пов'язані з WebPageTest.



Рисунок 3.9 - Головна сторінка Lighthouse метрики

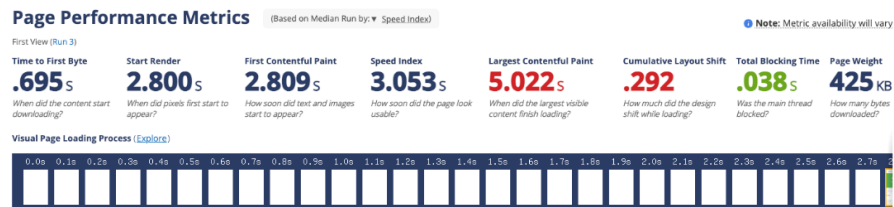


Рисунок 3.10 - Головна сторінка WebPageTest метрики

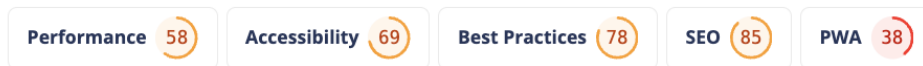


Рисунок 3.11 - Сторінка статті Lighthouse метрики

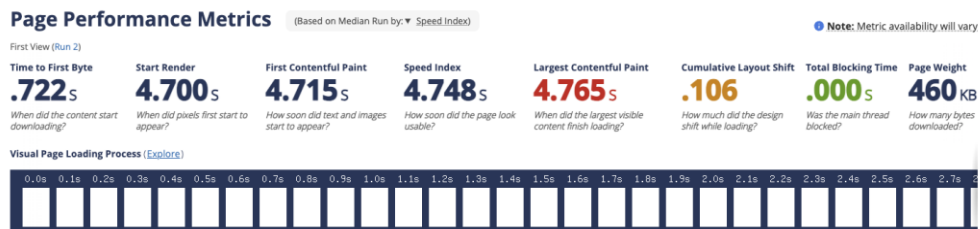


Рисунок 3.12 - Сторінка статті WebPageTest метрики



Рисунок 3.13 - Сторінка профілю Lighthouse метрики



Рисунок 3.14 - Сторінка профілю WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність:69;
- кращі практики: 78;
- SEO: 85.

Наведемо в таблиці 3.2 з інші характеристики та виведемо середнє значення для подальшого аналізу:

Таблиця 3.2 Таблиця характеристик Ember

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	43	58	57	52.57
Час до Інтерактивності	6.95	7.22	7.15	7.11
Індекс швидкості	2.8	4.7	5.6	4.37
Перший змістовий розпис	2.8	4.71	5.53	4.35
Загальний час блокування	38	0	0	12.67
Вага сторінки	425	460	437	440.67
Оцінка сторінки	62.2	65.6	65.4	64.4

Проведемо аналіз усіх характеристик:

- доступність має стабільний середній показник, але все ще існує потенціал для покращення до рекомендованих вищих значень;

- кращі практики SEO мають високі оцінки, що свідчить про добре оптимізовану веб-розробку з точки зору внутрішньої оптимізації та видимості пошуку;

- загальний час блокування є дуже маленьким;

— середня загальна оцінка свідчить про прийнятний загальний рівень, але з кількома ключовими областями для поліпшення.

### 3.5.3 Аналіз Vue

Проаналізуємо проект розроблений за допомогою фреймворку Vue. Нижче наведено метрики сторінок. Рисунок 3.15, 3.17 та 3.19 – зображують метрики пов'язані з Lighthouse, а рисунки 3.16, 3.18 та 3.20 – зображують метрики пов'язані з WebPageTest.

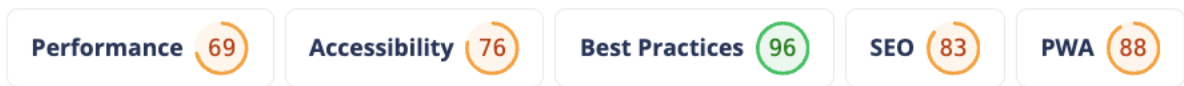


Рисунок 3.15 - Головна сторінка Lighthouse метрики

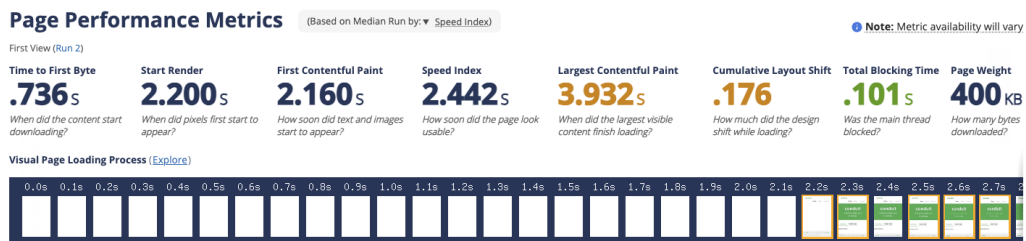


Рисунок 3.16 - Головна сторінка WebPageTest метрики

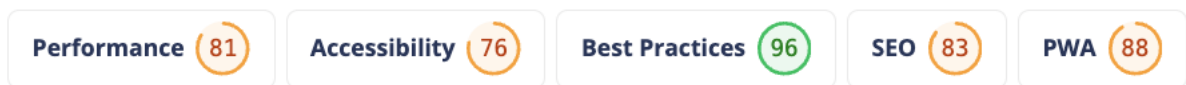


Рисунок 3.17 - Сторінка статті Lighthouse метрики

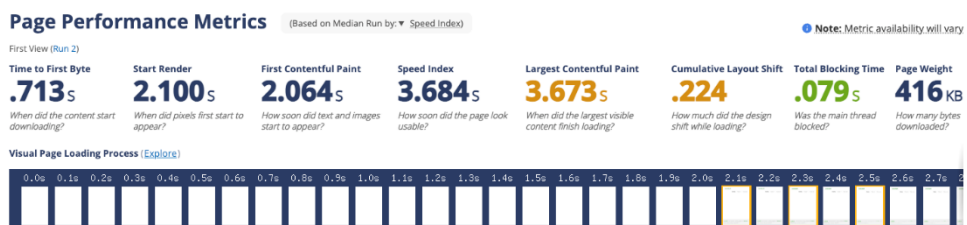


Рисунок 3.18 - Сторінка статті WebPageTest метрики

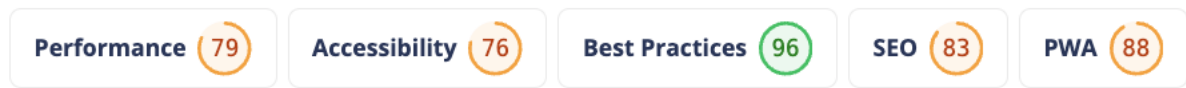


Рисунок 3.19 - Сторінка профілю Lighthouse метрики

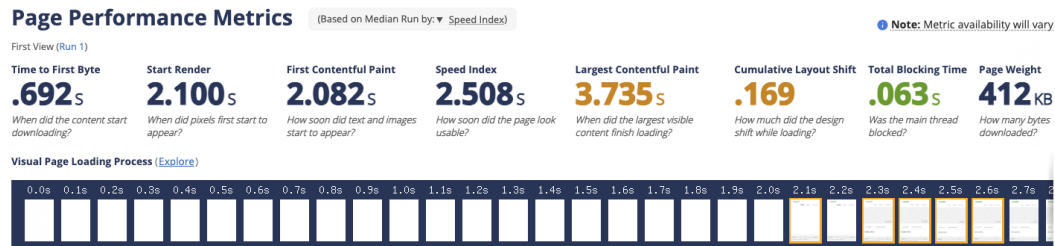


Рисунок 3.20 - Сторінка профілю WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність: 76;
- кращі практики: 96;
- SEO: 83.

Наведемо таблицю 3.3 з іншими характеристиками та виведемо середнє значення для подального аналізу:

Проведемо аналіз усіх характеристик:

- продуктивність є середнім значенням 76.33, що вказує на задовільну роботу сайту;
- доступність, кращі практики, SEO мають високі та стабільні оцінки на всіх екранах, що свідчить про добре виконання стандартів веб-розробки і оптимізації;
- загальний час блокування знаходяться в прийнятних межах, але існує потенціал для покращення;
- середня загальна оцінка в 83.87 вказує на високий рівень загальної оптимізації сайту.

Таблиця 3.3 Таблиця характеристик Vue

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	69	81	79	76.33
Час до Інтерактивності	7.36	7.13	6.92	7.14
Індекс швидкості	2.2	2.1	2.1	2.13
Перший змістовий розпис	2.16	2.06	2.08	2.1
Загальний час блокування	101	79	63	81
Вага сторінки	400	416	412	409
Оцінка сторінки	82.4	84.8	84.4	83.87

### 3.5.4 Аналіз Astro

Проаналізуємо проект розроблений за допомогою фреймворку Astro. Нижче наведено метрики сторінок. Рисунок 3.21, 3.23 та 3.25 – зображують метрики пов'язані з Lighthouse, а рисунки 3.22, 3.24 та 3.26 – зображують метрики пов'язані з WebPageTest.



Рисунок 3.21 - Головна сторінка Lighthouse метрики

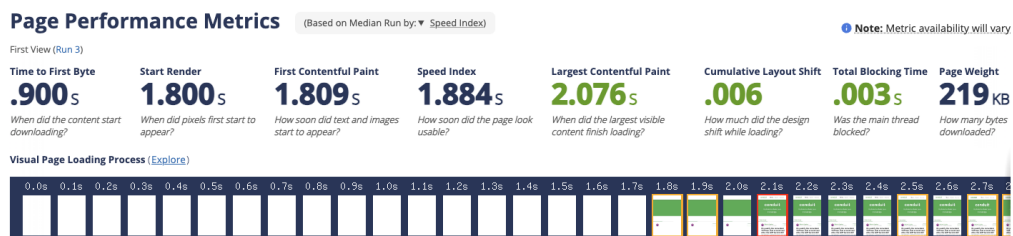


Рисунок 3.22 - Головна сторінка WebPageTest метрики



Рисунок 3.23 - Сторінка статті Lighthouse метрики

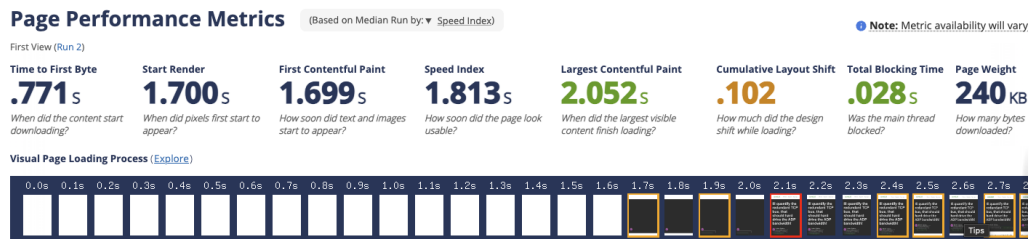


Рисунок 3.24 - Сторінка статті WebPageTest метрики

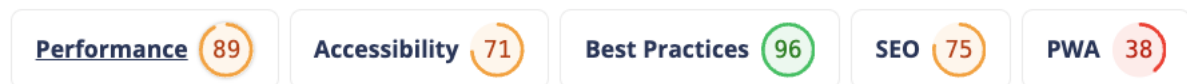


Рисунок 3.25 - Сторінка профілю Lighthouse метрики

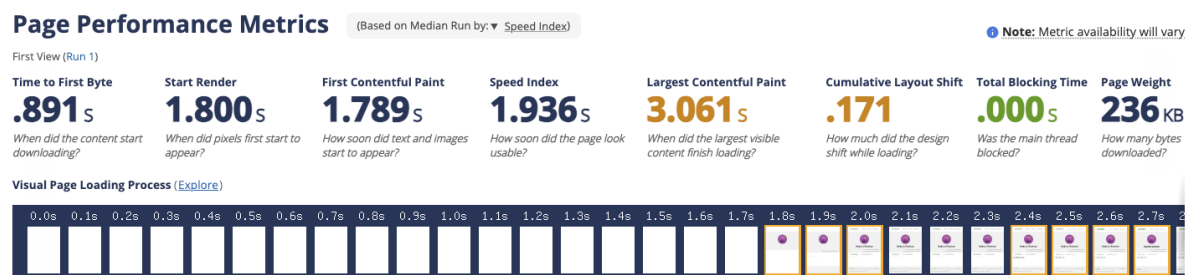


Рисунок 3.26 - Сторінка профілю WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність: 71;
- кращі практики: 96;
- SEO: 82.

Наведемо таблицю 3.4 з іншими характеристиками та виведемо середнє значення для подальшого аналізу:

Таблиця 3.4 - Таблиця характеристик Astro

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	97	98	89	94.67
Час до Інтерактивності	9	7.71	8.91	8.54
Індекс швидкості	1.8	1.7	1.8	1.77
Перший змістовий розпис	1.8	1.7	1.78	1.76
Загальний час блокування	3	28	0	10.33
Вага сторінки	219	240	236	231.67
Оцінка сторінки	76.8	77	73.8	75.87

Проведемо аналіз усіх характеристик:

- продуктивність показує дуже високі результати на екранах;
- кращі практики SEO мають високі оцінки, особливо кращі практики, що свідчить про дотримання високих стандартів розробки;
- вага сторінки є досить низькою, що сприяє швидкому завантаженню;
- середня загальна оцінка показує дуже хороший рівень оптимізації сайту.

### 3.5.5 Аналіз Angular

Проаналізуємо проект розроблений за допомогою фреймворку Angular. Нижче наведено метрики сторінок. Рисунок 3.27, 3.29 та 3.31 –

зображують метрики пов'язані з Lighthouse, а рисунки 3.28, 3.30 та 3.32 – зображують метрики пов'язані з WebPageTest.



Рисунок 3.27 - Головна сторінка Lighthouse метрики

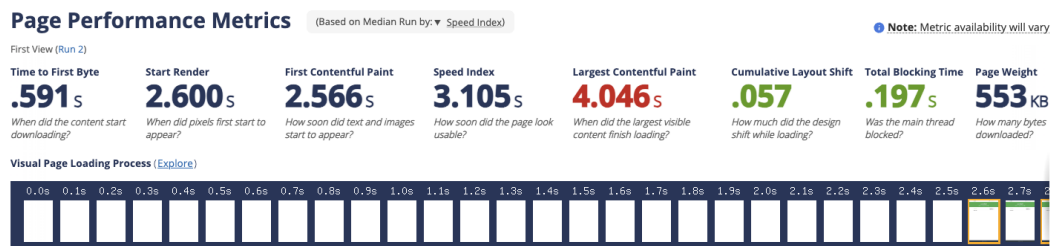


Рисунок 3.28 - Головна сторінка WebPageTest метрики

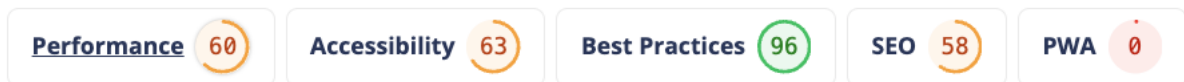


Рисунок 3.29 - Сторінка статті Lighthouse метрики

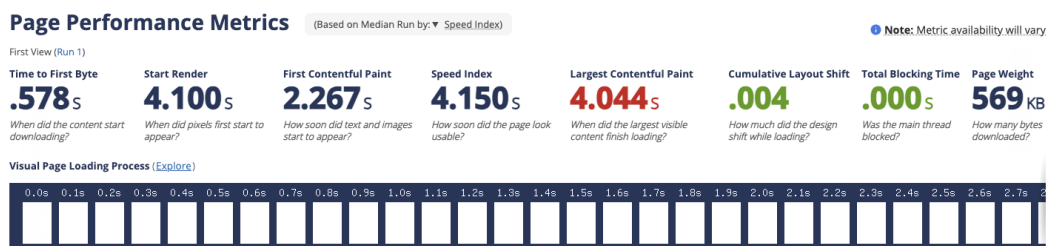


Рисунок 3.30 - Сторінка статті WebPageTest метрики

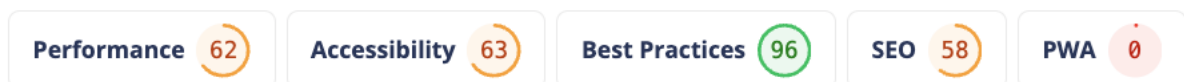


Рисунок 3.31 - Сторінка профілю Lighthouse метрики

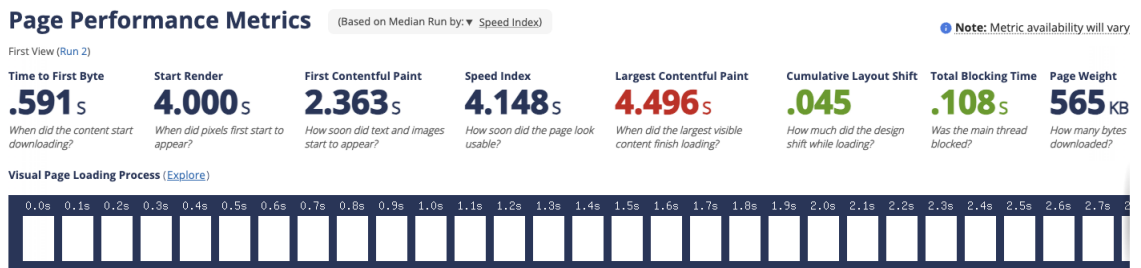


Рисунок 3.32 - Сторінка профілю WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність: 63;
- кращі практики: 96;
- SEO: 58.

Наведемо таблицю 3.5 з іншими характеристиками та виведемо середнє значення для подальшого аналізу.

Таблиця 3.5 - Таблиця характеристик Angular

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	61	60	62	61
Час до Інтерактивності	5.91	5.78	5.92	5.87
Індекс швидкості	2.6	4.1	4	3.57
Перший змістовий розпис	2.56	2.26	2.36	2.39
Загальний час блокування	197	0	108	101.67
Вага сторінки	553	569	565	562.33
Оцінка сторінки	55.6	55.4	55.8	55.6

Проведемо аналіз усіх характеристик:

– продуктивність, доступність, кращі практики, і SEO мають стабільні оцінки, з високими показниками для кращих практик. Однак, показники продуктивності і SEO потребують покращення;

– часи до інтерактивності є прийнятними, хоча оптимально прагнути до менших значень;

– загальний час блокування і вага сторінки мають середні показники, які також можуть бути оптимізовані.

### 3.5.6 Аналіз NextJS

Проаналізуємо проект розроблений за допомогою фреймворку NextJS. Нижче наведено метрики сторінок. Рисунок 3.33, 3.35 та 3.37 – зображують метрики пов'язані з Lighthouse, а рисунки 3.34, 3.36 та 3.38 – зображують метрики пов'язані з WebPageTest.

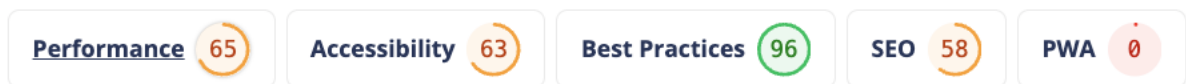


Рисунок 3.33 - Головна сторінка Lighthouse метрики

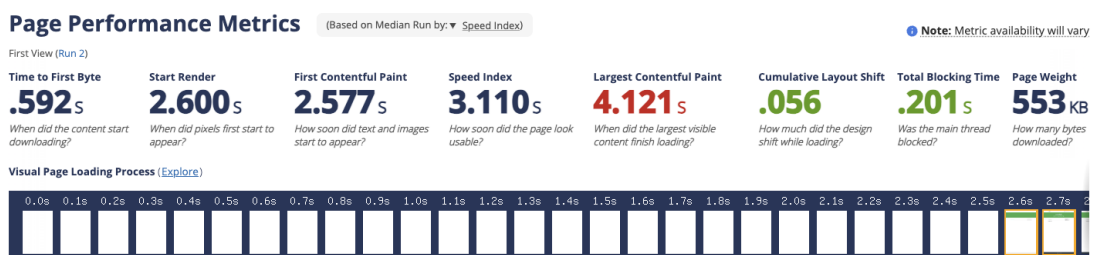


Рисунок 3.34 - Головна сторінка WebPageTest метрики

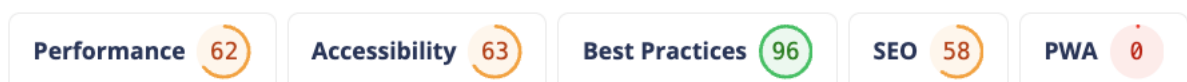


Рисунок 3.35 - Сторінка статті Lighthouse метрики

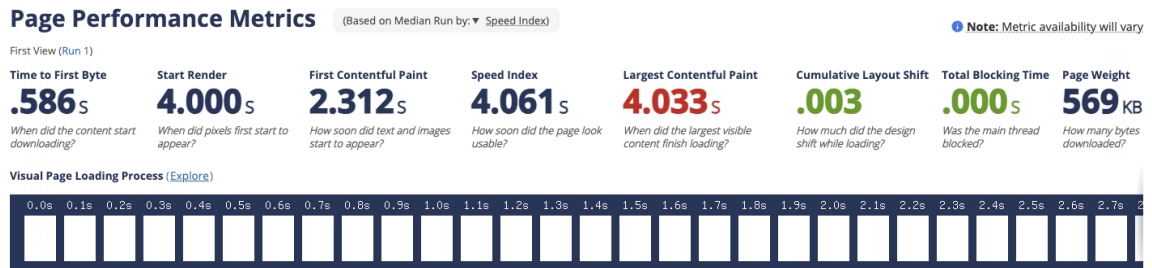


Рисунок 3.36 - Сторінка статті WebPageTest метрики

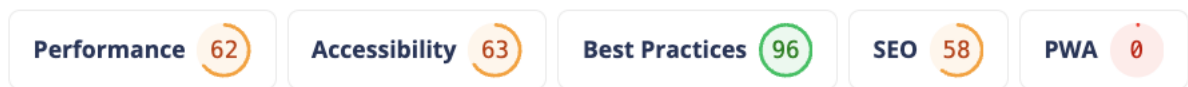


Рисунок 3.37 - Сторінка профілю Lighthouse метрики

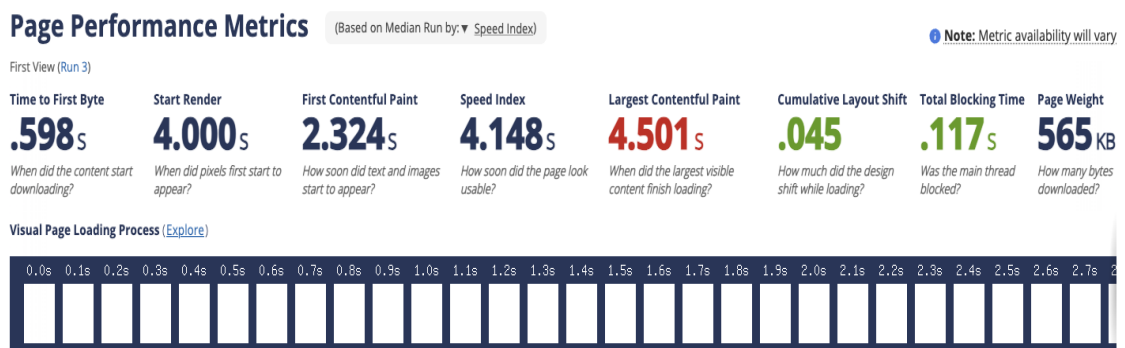


Рисунок 3.38 - Сторінка профілю WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність: 63;
- кращі практики: 96;
- SEO: 58.

Наведемо таблицю 3.6 з іншими характеристиками та виведемо середнє значення для подальшого аналізу.

Таблиця 3.6 - Таблиця характеристик NextJS

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	65	62	62	63
Час до Інтерактивності	5.92	5.86	5.96	5.92
Індекс швидкості	2.6	4.1	4	3.57
Перший змістовий розпис	2.57	2.26	2.32	2.38
Загальний час блокування	201	0	117	106
Вага сторінки	553	569	565	562.33
Оцінка сторінки	56.4	55.8	55.8	56

Проведемо аналіз усіх характеристик:

- продуктивність має низькі оцінки на всіх сторінках, що вказує на потребу оптимізації швидкості реагування;
- доступність також має середній рівень і потребує покращення для кращого досвіду користувачів з обмеженими можливостями;
- кращі практикита SEO показують високі оцінки, що вказує на дотримання встановлених стандартів і хорошу оптимізацію для пошукових систем;
- вага сторінки є вищою за бажану, що може впливати на швидкість завантаження;
- середня загальна оцінка 56.0 вказує на середній рівень оптимізації вебсайту з можливостями для покращення в різних аспектах, зокрема в продуктивності та доступності.

### 3.5.7 Аналіз Backbone

Проаналізуємо проект розроблений за допомогою фреймворку Backbone. Нижче наведено метрики сторінок. Рисунок 3.39, 3.41 та 3.43 – зображують метрики пов'язані з Lighthouse, а рисунки 3.40, 3.42 та 3.44 – зображують метрики пов'язані з WebPageTest.

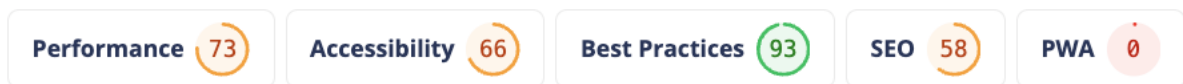


Рисунок 3.39 - Головна сторінка Lighthouse метрики

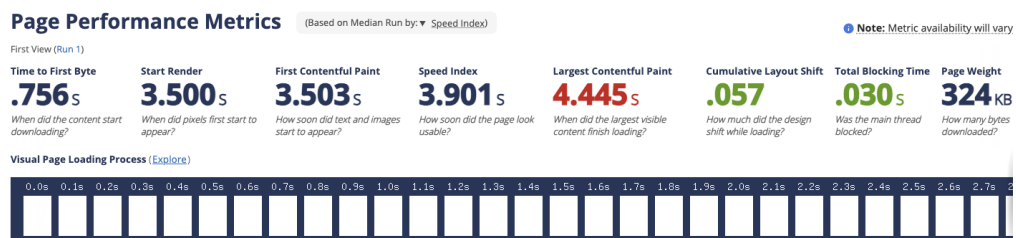


Рисунок 3.40 - Головна сторінка WebPageTest метрики

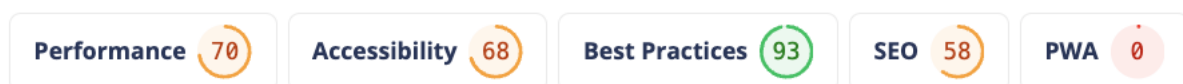


Рисунок 3.41 - Сторінка статті Lighthouse метрики

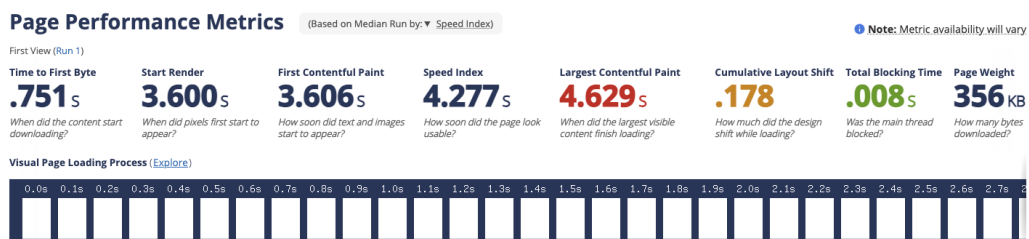


Рисунок 3.42 - Сторінка статті WebPageTest метрики



Рисунок 3.43 - Сторінка профілю Lighthouse метрики

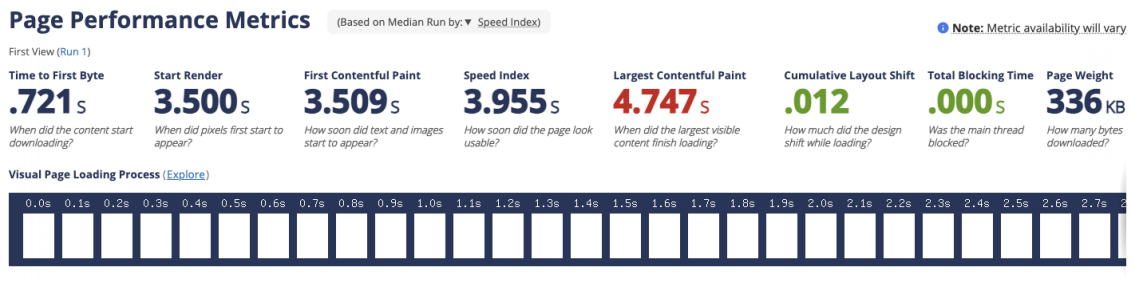


Рисунок 3.44 - Сторінка профілю WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність: 66;
- кращі практики: 93;
- SEO: 58.

Наведемо таблицю 3.7 з іншими характеристиками та виведемо середнє значення для подальшого аналізу.

Проведемо аналіз усіх характеристик:

- продуктивність сторінок показує відносно середні результати, що вказує на потребу в подальшій оптимізації;
- доступність має показники нижче оптимальних і вимагає покращення для кращої взаємодії з користувачами з обмеженими можливостями;
- кращі практикимають високі оцінки, що свідчить про дотримання рекомендованих методів розробки;

– час до інтерактивності, індекс швидкості вказують на помірну швидкість завантаження.

Таблиця 3.7 Таблиця характеристик Backbone

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	73	70	71	71.33
Час до Інтерактивності	7.56	7.51	7.21	7.43
Індекс швидкості	3.5	3.6	3.5	3.53
Перший змістовий розпис	3.5	3.6	3.5	3.53
Загальний час блокування	30	8	0	12.67
Вага сторінки	324	356	336	338.67
Оцінка сторінки	58	57.8	57.6	57.8

### 3.5.8 Аналіз Svelte

Проаналізуємо проект розроблений за допомогою фреймворку Svelte. Нижче наведено метрики сторінок. Рисунок 3.45, 3.47 та 3.49 – зображують метрики пов'язані з Lighthouse, а рисунки 3.46, 3.48 та 3.50 – зображують метрики пов'язані з WebPageTest.

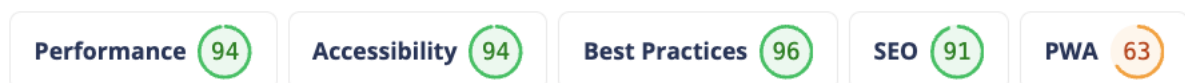


Рисунок 3.45 - Головна сторінка Lighthouse метрики

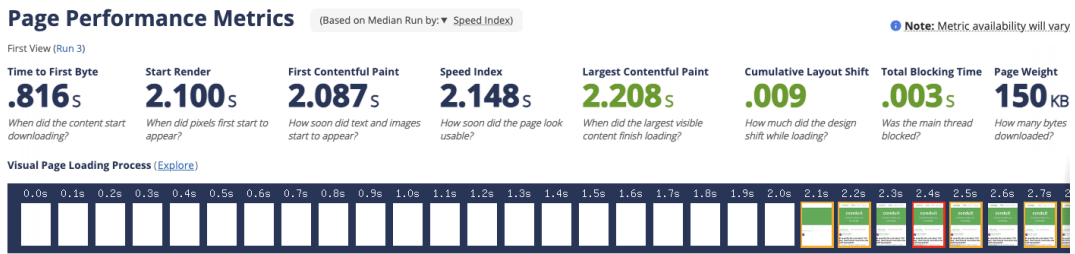


Рисунок 3.46 - Головна сторінка WebPageTest метрики

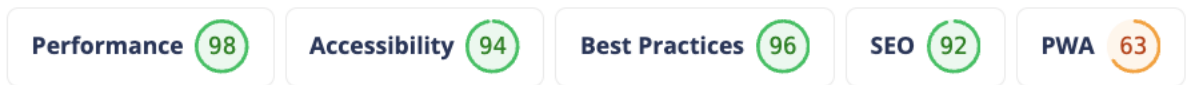


Рисунок 3.47 - Сторінка статті Lighthouse метрики

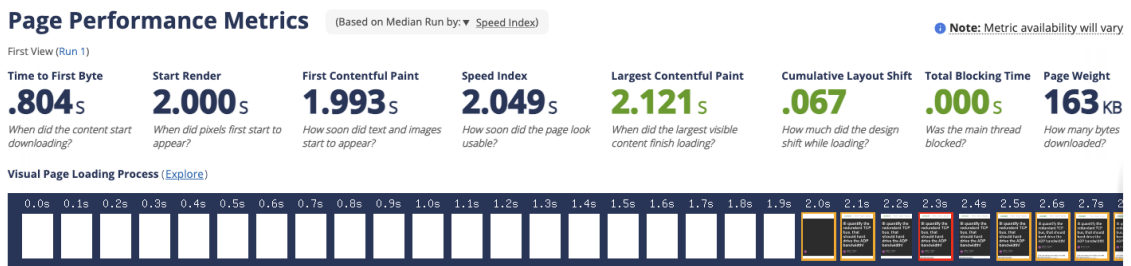


Рисунок 3.48 - Сторінка профілю WebPageTest метрики



Рисунок 3.49 - Сторінка профілю Lighthouse метрики

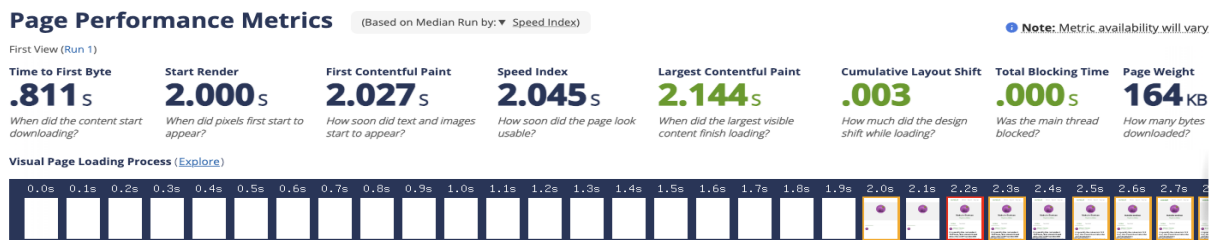


Рисунок 3.50 - Сторінка статті WebPageTest метрики

Відокремимо характеристики на які впливає фреймворк але вони не змінюються між сторінками:

- доступність:94;
- кращі практики: 96;
- SEO: 91.

Наведемо таблицю 3.8 з іншими характеристиками та виведемо середнє значення для подальшого аналізу:

Таблиця 3.8 - Таблиця характеристик Svelte

	Головна сторінка	Сторінка статті	Сторінка профілю	Середнє
Продуктивність	94	98	95	95.67
Час до Інтерактивності	8.16	8.04	8.11	8.1
Індекс швидкості	2.1	2	2	2.03
Перший змістовий розпис	2.08	1.99	2.02	2.03
Загальний час блокування	3	0	0	1
Вага сторінки	150	163	164	159
Оцінка сторінки	93.75	95	94.25	94.33

Проведемо аналіз усіх характеристик:

- продуктивність, доступність, кращі практики та SEO мають високі оцінки, що свідчить про загалом відмінну оптимізацію сторінок;
- час до інтерактивності та індекс швидкості відображають гарну реактивність сторінок;

– час блокування є дуже низькими, що забезпечує стабільність візуального вмісту та відсутність перешкод для користувача під час завантажень.

### 3.6 Аналіз та порівняння результатів

Зробимо загальну таблицю 3.9 для порівняння кожного фреймворку та проаналізуємо її.

Таблиця 3.9 - Таблиця порівняння усіх фреймворків

	Доступність	Кращі практики	SEO	Продуктивність	Час до Інтерактивності	Індекс швидкості	Перший змістовий розпис	Загальний час блокування	Вага сторінки	Оцінка сторінки
React	59	96	58	76.67	5.97	2.67	2.12	422	422	57.93
Ember	69	78	85	52.57	7.11	4.37	4.35	12.67	440.67	64.4
Vue	76	96	83	76.33	7.14	2.13	2.1	81	409	83.87
Astro	71	96	82	94.67	8.54	1.77	1.76	10.33	231.67	75.87
Angular	63	96	58	61	5.87	3.57	2.39	101.67	562.33	55.6
NextJS	63	96	58	63	5.92	3.57	2.38	106	562.33	56
Blackbone.js	66	93	58	71.33	7.43	3.53	3.53	12.67	338.67	57.8
Svelte	94	96	91	95.67	8.1	2.03	2.03	1	159	94.33

Для більш наглядного та суцільного порівняння створимо нову таблицю з коефіцієнтами від нуля до п'яти, де п'ять – це максимальна можливе значення та ноль – це мінімальне можливе значення.

Зробимо загальну таблицю 3.10 для порівняння кожного фреймворку та проаналізуємо її.

Таблиця 3.10 - Таблиця порівняння усіх фреймворків з коефіцієнтами

	Доступність	Кращі практики	SEO	Продуктивність	Час до Інтерактивності	Індекс швидкості	Перший змістовий розпис	Загальний час блокування	Вага сторінки	Оцінка сторінки	Зашальна оцінка фреймворку
React	2	4.5	3	3.5	4.5	4	4	2	3	3	33.5
Ember	3	3	4	2.5	3.5	3	2.5	4.5	3	3.5	32.5
Vue	3.2	4.5	4	3.5	3.5	4.5	4.5	3.5	3	4	38.2
Astro	3	4.5	4	4.5	2.5	5	5	4	5	3.5	41
Angular	2.5	4.5	3	3	4.5	4	4	3	2	3	33.5
NextJS	2.5	4.5	3	3	3.5	4	4	3	2	3	32.5
Backbone.js	2.5	4	3	3.5	3.5	4	3	4.5	3.5	3	34.5
Svelte	4.5	4.5	4.5	4.5	3	5	4.5	5	5	4.5	45

Аналізуючи дані з таблиці, можна зробити наступні висновки:

- Svelte виокремлюється як найсильніший фреймворк із загальною оцінкою 45. Він показує високі результати у всіх категоріях, особливо у доступності, кращих практиках, SEO, і продуктивності, демонструючи відмінний баланс між швидкістю та оптимізацією;
- Astro також показує дуже сильні результати із загальною оцінкою 41, особливо у категоріях кращих практик, індексу швидкості, і

першого змістовного розпису, що вказує на ефективність у візуальному відображенні та швидкості відгуку;

- Vue має високу загальну оцінку 38.2, з сильними показниками у SEO, продуктивності, та індексі швидкості, що вказує на хорошу оптимізацію та швидку відгуку сторінок;

- React та Angular мають однакову загальну оцінку 33.5. Обидва фреймворки показують збалансовані результати, з сильними сторонами у часі до інтерактивності та кращих практиках, але з місцем для покращення у доступності та SEO;

- Ember, NextJS, і Backbone.js мають відносно нижчі загальні оцінки (32.5, 32.5, та 34.5 відповідно), що підкреслює потребу в подальших оптимізаціях, особливо в доступності та SEO для Ember і NextJS, а у продуктивності та часі до інтерактивності для Backbone.js.

Загалом, таблиця 3.10 відображає, що, хоча деякі фреймворки виокремлюються в певних аспектах, важливо враховувати баланс всіх характеристик при виборі технології для конкретного проекту. Наприклад, якщо важливою є швидкість завантаження та час до інтерактивності, то Svelte або Astro можуть бути найкращим вибором. Однак, якщо фокус на доступності та SEO, то Vue може бути більш відповідним варіантом. Зважаючи на це, розробники повинні враховувати специфіку проекту та цілі, щоб обрати найбільш підходящий фреймворк.

### 3.7 Аналіз перспектив покращення дослідження

Аналізуючи перспективи покращення дослідження веб-рішень, що базуються на різних фреймворках, можна визначити декілька ключових напрямків для розширення:

- розширення бази фреймворків: включення більшої кількості фреймворків може дати більш повне уявлення про різноманітність доступних опцій та їхні потенційні переваги й недоліки;
- глибше тестування продуктивності: замість обмеження аналізу загальними показниками, можна здійснити детальне тестування з використанням різних інструментів та сценаріїв навантаження, щоб виявити конкретні точки можливих покращень;
- користувацький досвід (UX)[16]: оцінка того, як фреймворки впливають на UX, може включати тестування з реальними користувачами, щоб зрозуміти, як функціональність та продуктивність впливають на задоволеність користувачів;
- оптимізація залежно від типу контенту: різні типи контенту (текст, зображення, відео) можуть вимагати різних методів оптимізації. Аналіз, що враховує цю різноманітність, може дати конкретніші рекомендації для покращення;
- безпека [17]: це важливий аспект, який іноді може бути відсунутий на другий план. Оцінка безпеки фреймворків може допомогти зменшити потенційні ризики;
- сумісність із старими браузерами: деякі фреймворки краще працюють із новітніми технологіями браузерів, тоді як інші забезпечують більшу сумісність із старими версіями. Оцінка цього аспекту може бути корисною для проектів, де це є пріоритетом;
- міжнароднізація та локалізація: дослідження можливостей локалізації дозволить оцінити, наскільки легко фреймворки дозволяють адаптувати веб-рішення під різні ринки та культури;
- вплив на SEO: окрім загальних оцінок SEO, можна здійснити детальний аналіз структури сторінок, їхньої семантики та інших факторів, які впливають на ранжування у пошукових системах;

– соціальна інтеграція: аналіз того, як різні фреймворки сприяють або ускладнюють інтеграцію з соціальними мережами та платформами, може бути корисним для стратегії цифрового маркетингу;

– масштабованість та підтримка: розгляд потенціалу фреймворків для масштабування проектів та доступність підтримки і ресурсів для розробників.

Кожен з цих напрямків може бути розглянутий окремо або в комплексі, щоб створити більш глибоке та всебічне дослідження, яке дозволить оцінити повний потенціал фреймворків і виявити найбільш ефективні шляхи для оптимізації веб-рішень.

## ВИСНОВКИ

У світі, насиченому інформаційними технологіями, роль інформаційних систем стає ключовою для бізнесу та управління. Сучасні вимоги до обробки великого обсягу складної інформації підкреслюють потребу у високоефективних та надійних рішеннях.

Ця робота спрямована на дослідження впливу JavaScript-фреймворків, зокрема React, Vue.js та Angular, на функціональні характеристики вебсторінок. Мета полягає в наданні рекомендацій для розробників та бізнес-організацій з метою досягнення оптимальних результатів у веб-розробці.

Дослідження охоплює важливі аспекти, такі як швидкість завантаження, відзивчивість та візуальна привабливість вебсторінок. Результати можуть бути корисними для розробників, які шукають ефективні та продуктивні інструменти для веб-розробки, а також для бізнесу, який має інтерес до покращення характеристик своїх веб-додатків.

Надані рекомендації та висновки щодо вибору JavaScript-фреймворку можуть сприяти покращенню якості та продуктивності вебсторінок, що, в свою чергу, сприятиме успішному функціонуванню бізнес-процесів та задоволенню потреб користувачів.

У ході роботи було проведено аналіз кількох ключових технологій веб-розробки, зокрема Angular, Next.js, Vue.js, Svelte, Backbone.js, Ember.js та Astro. Розглянуті аспекти включали продуктивність, SEO, доступність, вагу сторінок, складність навчання, інтеграцію з іншими технологіями, модульність та підтримку.

## ПЕРЕЛИК ДЖЕРЕЛ ПОСИЛАНЬ

1. Петроченков П.М. Підхід до оцінювання характеристик вебсторінок. Цифровізація науки та сучасні тренди її розвитку : матеріали V Міжнародної студентської наукової конференції, м. Житомир, 24 листопада, 2023 рік / ГО «Молодіжна наукова ліга». Вінниця: ТОВ «УКРЛОГОС Груп», 2023. С. 313-314.
2. Learning JavaScript Design Patterns, Addy Osmani, 2010, 236 с.
3. JavaScript: The Good Parts Publication, Douglas Crockford, 2008, 172 с.
4. You Don't Know JS, Kyle Simpson, 2018, 88 с.
5. JavaScript: The Definitive Guide, David Flanagan, 2020, 706 с.
6. Quick-start guide on using Lighthouse URL <https://developers.google.com/web/tools/lighthouse/> (дата звернення 15.11.23)
7. Using WebPageTest: Web Performance Testing for Novices and Power Users, Rick Viscomi, 2015, 214 с.
8. Building Progressive Web Apps: Bringing the Power of Native to the Browser, Tal Ater, 2017, 288 с.
9. Angular: Up and Running: Learning Angular, Step by Step, Shyam Seshadri, 2018, 300 с.
10. Vue.js: Up and Running: Building Accessible and Performant Web, Callum Macrae, 2018, 164 с.
11. Building Web Apps with Ember.js Write Ambitious JavaScript, Jesse Cravens, Thomas Brady, 2014, 186 с.
12. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage 3rd Edition, Kristina Chodorow, 2019, 426 с.
13. Building Microservices: Designing Fine-Grained Systems, Sam Newman, 2021, 500 с.

14. Learning React: Functional Web Development with React and Redux 1st Edition, Alex Banks, 2017, 320 c.
15. Web Development with Node and Express: Leveraging the JavaScript Stack 2nd Edition, Ethan Brown, 2018, 326 c.
16. UX Research: Practical Techniques for Designing Better Products 1st Edition, Brad Nunnally, 2016, 256 c.
17. Defensive Security Handbook: Best Practices for Securing Infrastructure 1st Edition, Lee Brotherston, 2017, 274 c.