

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

БРИЖАК ВЛАДИСЛАВ АНАТОЛІЙОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент
_____ О. В. Зелінська
« ____ » _____ 20__ р.

**РОЗРОБКА САЙТУ ПІДБОРУ ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ
КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

О. В. Зелінська, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

Вінниця – 2025

АНОТАЦІЯ

Брижак В.А. Розробка сайту підбору інструментів реалізації користувацького інтерфейсу. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2025.

У кваліфікаційній (бакалаврській) роботі, яка присвячена розробці сайту для підбору інструментів реалізації користувацького інтерфейсу, за допомогою технологій: HTML, CSS та JavaScript розроблено вебсайт, який допоможе користувачам обрати необхідні інструменти реалізації користувацького інтерфейсу.

Ключові слова: вебсайт, фронтенд, веброзробка, HTML, CSS, JavaScript, фреймворки React та Angular.

Рис. 24, Табл. 1, Джерел 40.

ABSTRACT

Bryzhak V.A. Development of a website for selecting tools for implementing a user interface. Specialty 122 "Computer Science", educational program "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2025.

In the qualification (bachelor's) work, which is devoted to the study of the concept of a website, the basic concepts and principles of developing Internet pages. Using technologies: HTML, CSS and JavaScript, a website was developed that will help users choose the necessary tools for implementing a user interface.

Keywords: website, frontend, web development, HTML, CSS, JavaScript, React and Angular frameworks.

Fig. 24, Table. 1, Source 40.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 Основні теоретичні положення та постановка задачі	7
1.1 Дослідження предметної області та огляд існуючих аналогів.....	7
1.2 Основні етапи у створенні таких сайтів	19
1.3 Постановка задачі.....	23
РОЗДІЛ 2 Аналіз та вибір програмних інструментів для реалізації сайту	24
2.1. Інструменти для розробки дизайну інтернет сторінок.....	24
2.2 Інструменти для верстки та розробки необхідного функціоналу	27
2.3 Оцінювання та вибір середовища для розміщення проєкту.	39
РОЗДІЛ 3 Реалізація сайту для підбору інструментів реалізації користувацького інтерфейсу	41
3.1. Реалізація сайту для підбору інструментів реалізації користувацького інтерфейсу.	41
3.2. Структура сайту та особливості функціоналу.....	46
3.3. Проведення тестування реалізованого сайту.....	50
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	53
ДОДАТКИ	57

ВСТУП

Сьогодні важко уявити сучасних людей, які не користуються Інтернетом. Адже мережа, це вже не просто місце, де можна знайти нову інформацію, це безмежні можливості для спілкування, бо тут знаходять давніх знайомих та знайомляться з новими людьми. Також за допомогою Інтернету можна знайти роботу і навіть заробляти в мережі, тут підприємці можуть продавати свій товар та знаходити нових покупців і постачальників. Ще люди проводять вільний свій час в інтернеті за прослуховуванням музики, переглядом фільмів та фото. На сотнях тисяч сайтів з новинами щодня публікуються мільйони статей і публікацій. А загалом із перелічених фактів можна зробити висновок, що інтернет та вебсайти стали інструментами для комунікації, бізнесу, освіти та розваг.

Раніше сайти сприймалися як додаток до основного бізнесу. Сьогодні – це його невід’ємна частина, а іноді й основне місце для взаємодії з клієнтами. На даний момент у світі, певно, не знайдеться людини, яка не відвідувала хоча б одну інтернет сторінку щодня.

Вміння ефективно підбирати та відбирати інформацію в інтернеті стає одним із найважливіших рис сучасної людини. В епоху, коли вебсайти почали замінювати бібліотеки, газети, дошки оголошень, довідники, архіви та багато інших інформаційних джерел, надзвичайно важливо вміти орієнтуватися в цифровому просторі.

У сучасних умовах щоденного збільшення кількості активних користувачів мережі Інтернет зростає попит на зручні й функціональні вебсторінки. Проте тенденція до простоти в дизайні та навігації залишається одним із головних факторів, який вказує на створення успішного продукту. Важливим етапом у процесі розробки сайту - є вибір оптимальних інструментів та технологій для реалізації. Оскільки вебсайти більше не є чимось дивовижно не досяжним і дорогим, в плані розробки, то бар’єр входу для розробників значно знизився. Тепер для того, щоб створити власний односторінковий сайт потрібно вміти

впевнено користуватися можливостями комп'ютера, мати доступ до Інтернету та велике бажання. Багато спеціально створених конструкторів сайтів надають великий набір готових макетів та окремих частин сторінки, але жоден такий онлайн інструмент не дасть повного контролю над розробкою.

Веброзробникам, особливо початківцям, часто складно визначити, які інструменти краще застосувати в тому чи іншому випадку — HTML/CSS/JavaScript, React або Angular. У зв'язку з цим виникає потреба у зручному інструменті, який дозволить швидко оцінити доцільність використання конкретної технології у відповідності до бажань користувача.

Даний проєкт створений для того, щоб з легкістю на основі дизайну сайту аналогу або текстового опису отримати рекомендацію щодо вибору необхідного інструментарію для подальшої розробки.

Наступним кроком, після отримання усіх необхідних вхідних даних, йде розробка макету сайту, а далі реалізації за допомогою обраних програмних інструментів. Одним із основних кроків, які допомогли у розробці дизайну цього сайту став порівняльний аналіз та дослідження двох сайтів аналогів – ICStudio та IAMPM. Подальші кроки втілено за допомогою IDE Visual Studio Code, HTML, CSS, JS та декількох потрібних бібліотек.

Метою даної роботи є створення інтернет сайту, який дозволить, для кожного користувачу, на основі заданих параметрів запропонувати оптимальні технології та інструменти для реалізації користувацького інтерфейсу.

Завданням кваліфікаційної роботи є:

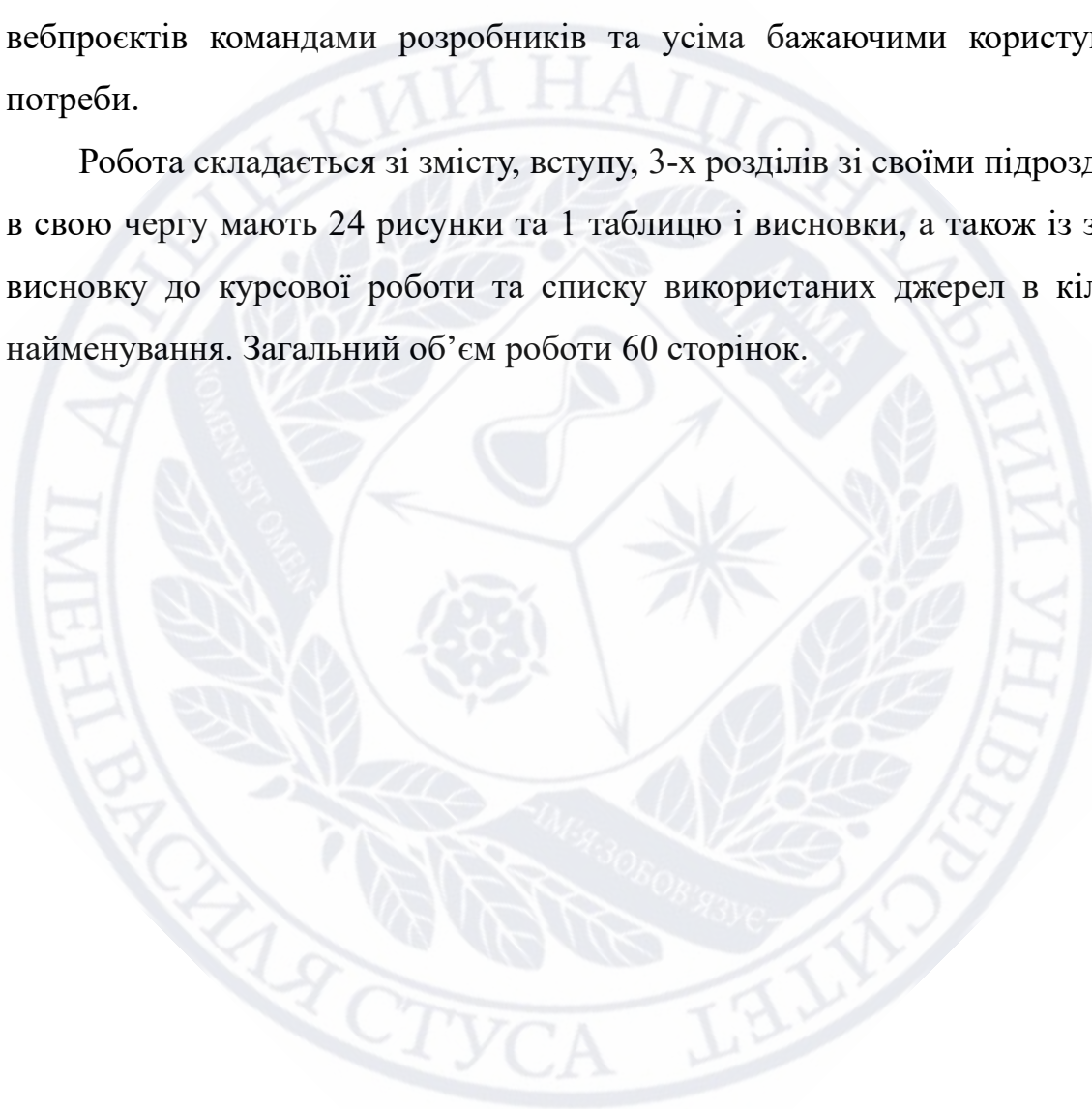
- Дослідження предметної області, історії створення і розвитку сайтів, а також їх архітектури.
- Розглянути та проаналізувати існуючі аналоги, зробити постановку задачі.
- Проаналізувати та вибрати програмні інструменти для реалізації проєкту.
- Розробити сайт з усім необхідним функціоналом. Зробити висновок.

Об'єктом даного дослідження є процес вибору інструментарію для створення користувацького інтерфейсу сайту.

Предмет дослідження - методи та інструменти реалізації фронтенд частини сторінки, а також принципи підбору відповідно до заданих умов.

Теоретичне та практичне значення полягає у створенні інструменту, який може використовуватися студентами, у навчальному процесі, при плануванні вебпроектів командами розробників та усіма бажаючими користувачами за потреби.

Робота складається зі змісту, вступу, 3-х розділів зі своїми підрозділами, які в свою чергу мають 24 рисунки та 1 таблицю і висновки, а також із загального висновку до курсової роботи та списку використаних джерел в кількості 40 найменування. Загальний об'єм роботи 60 сторінок.



РОЗДІЛ 1

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Дослідження предметної області та огляд існуючих аналогів

В першу чергу, перед тим, як починати більш детально розбиратися в темі розробки користувацького інтерфейсу потрібно сповна усвідомити, що собою являє вебсайт. Вебсайтом називається одна або декілька пов'язаних між собою за змістом інтернет сторінок, які відкриваються за допомогою браузері. Складність сайтів може бути абсолютно різною, починаючи від максимально простих дошок оголошень та різних форумів, закінчуючи багатofункціональними і сторінками із великою кількістю анімацій [1].

З кожним днем вплив технологій на життя людей спонукає кожного з нас все частіше відвідувати різні інтернет сторінки або користуватися певними додатками. Інколи навіть не замислюючись можна користуватися сайтом, коли просто відкрив якийсь додаток, бо дуже часто, особливо в останні декілька років, додатки та сайти тісно пов'язані між собою, навіть можуть бути один в одному. З вище сказаного, можна зрозуміти, що вплив вебсайтів на життя повсякденне життя людей з кожним днем стає все більшим.

Сайти мають декілька переваг відносно додатків, наприклад для розробників набагато простіше написати адаптивний сайт, який буде правильно відображатися як на персональних комп'ютерах так і на мобільних телефонах. В той самий час, для аналогічних можливостей потрібно буде робити два окремих додатки. Така можливість зберігає час для розробників, адже їм не потрібно писати окремий код для того, щоб їхній сайт відкривався на різних браузерах, (не важливо якому це може бути Internet Explorer, Chrome або Firefox) відповідно одна розробка покриває і різні операційні системи, різні пристрої та навіть браузери. То ж для того, щоб скористатися інтернет сторінкою користувача потрібно мати лише який-небудь пристрій та доступ до Інтернету.

Виглядає так, що дуже легко сплутати вебдодаток та вебсайт, справді, тут є досить багато спільного. Проте також потрібно сказати і про розбіжності між ними:

- Вебсайт – це одна або певна кількість зв’язаних між собою HTML файлів,

Що відкриваються за допомогою Інтернету та браузера. В той самий час вебдодаток – це програмне забезпечення або програма, що відкривається в браузері.

- Вебдодаток може часто буває складовою частиною вебсайту або ж окремим цільним додатком.

- Вебсайти служать користувачу як місце для отримання певної інформації, такої як різноманітні тексти або музика чи відео. Вебдодаток надає більше можливостей для взаємодії з користувачем, проводити певні маніпуляції із якимись даними, тощо.

- Вебдодаток здебільшого передбачає аутентифікацію. Коли для сайту це не є обов’язковим, але така можливість також є [2].

Провівши багато часу в пошуках аналогів, було виявлено, що їх зовсім немає. В основному за запитом браузер видає сторінки зі статтями, в яких йдеться про те, який тип сайтів підходить в тих чи інших ситуаціях або які є інструменти та технології для розробки. Проте це все ж не можна вважати прямим конкурентом. Варто зауважити, що аналіз навіть таких аналогів було проведено, задля збору ще більшої кількості інформації та полегшення подальшої роботи [3-4]. Під час пошуків було знайдено дві сторінки, які варті уваги, а саме «ICStudio» та «IAMPM». Перший з них – це сайт вебстудії розробки програмного забезпечення, в них є декілька дуже цікавих статей по даній тематиці. «IAMPM» - пропонує статті, вебінари та спеціалізовані проекти в галузі IT, та веб розробці включно. Розпочнемо із перегляду «ICStudio», а після того перейдемо до іншої сторінки, що дасть можливість в подальшому кроці зробити певний висновок.

ICStudio - це українська IT-компанія, яка спеціалізується безпосередньо на розробці інтернет сторінок для користувачів. Їх спеціалізація стосовно інструментарію – це фреймворки Laravel та October CMS. Вони надають безліч послуг, але головними вважаються створення корпоративних сайтів та розробка

CRM систем для них, а так інтеграція платіжних сервісів до уже готових сайтів. Варто зауважити, що також на їх сторінці дуже багато статей присвячених деталям та комплексним підходам до веброзробки. Перейшовши за посиланням, потрапляємо на головну сторінку, одразу впадає в око кольорова палітра, переважають білий, чорний та жовтий кольори, в поєднанні дуже гармонійні і приємні для користувача. Зрозуміла навігація, проста і ефективна. Увагу користувача намагаються привернути саме на навігацію, де їхні головні матеріали.

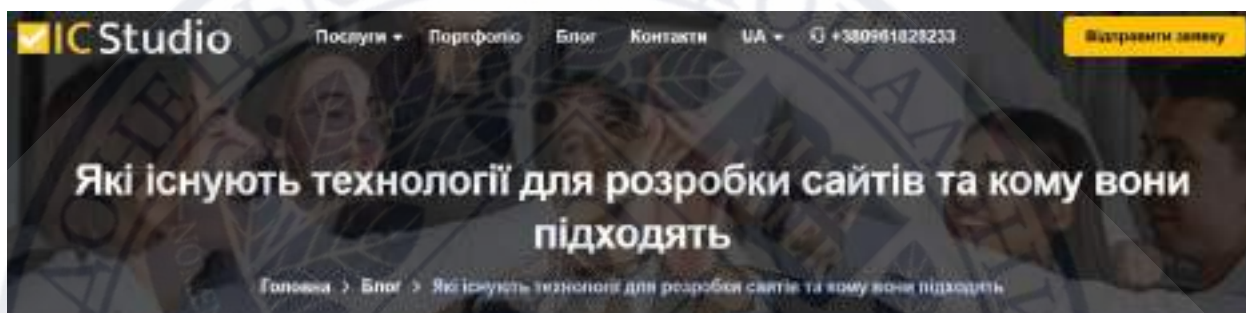


Рисунок 1.1. Сторінка блогу на сайті “ICStudio”

Відкривши їхній блог про технології розробки сайтів, одразу в очі кидається добре оформлений заголовок та цікаве фонове зображення. Прогорнувши трохи вниз, зустрічаємось із великою кількістю текстової інформації. Шрифти досить приємні для читання, але розмір та колір тексту змушують сильно вглядатися, щоб могли чітко і розбірливо переглядати сторінку.

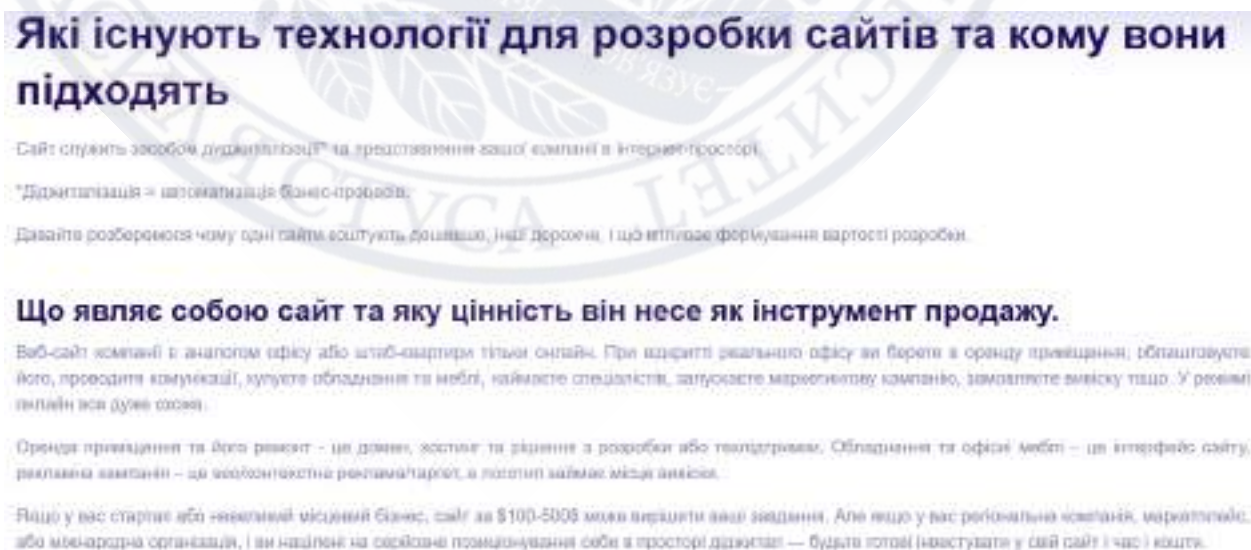


Рисунок 1.2. Вигляд текстової частини сторінки “ ICStudio ”

Прогорнувши блог вниз, натрапляємо на частину із прикладом їхньої роботи, що є дуже важливим елементом для кращого сприйняття поданої інформації. Варто відзначити, що представлені тільки фотоматеріалами із невеликими пояснювальними підписами.

Які існують рішення створення сайтів і як вибрати відповідне

Види сайту та цілі.

Ландінг:

Це односторонній сайт, що містить у собі загальну інформацію про компанію, методи роботи та експерти дати згоду. Зазвичай мають простий функціонал, який змінює технічних доробів у майбутньому. Такі сайти не складні у розробці, їх можна зробити навіть на онлайн-конструкторах (Tilda, Wix, LP Generator).



Рисунок 1.3 Вигляд частини із прикладом лендінгу на сторінці “ ICStudio ”

Якщо оцінювати в загальному, то сторінка виглядає дуже цікаво, функціоналу не дуже багато, але все коректно працює. Варто зауважити, що присутня реклама, але на сьогоднішній день для багатьох сайтів вона може бути чи не єдиним способом прибутку. Щодо вибору мови, то доступна англійська та українська, дуже тішить відсутність російської. Проте на зображеннях їхніх робіт присутні розроблені російськомовні сторінки.

Якщо ж перейти на сайт конкурента, то варто зауважити, надзвичайно приємний візуальний вигляд сайту. «IAMPМ» – це українська онлайн-платформа для навчання ІТ-освіті. Вони прагнуть розвивати людей різного віку та спеціалізації в ІТ-просторі. Надають близько 40-ка різноманітних теоретичних та практичних курсів, зокрема Full-stack, Frontend та Backend розробка, та деякі інші, які дотичні до даної тематики.

На головній сторінці все дуже просто і зрозуміло, в хедері – логотип та назва сайту, зручна та проста навігація, кнопка зміни мови на сторінці, яка дає можливість обрати потрібний користувачу варіант. Кольорової гама вдало підібрана, переважають білі, чорні та сині кольори, які непогано гармонують між собою. Варто зазначити, що першим в списку навігації сторінки йде вкладка «Про нас», це дуже зручно, бо часто користувач відкривши сайт, хоче ознайомитися із спеціалізацією даної платформи, дізнатися більше про розробників та їхній досвід і вміння.

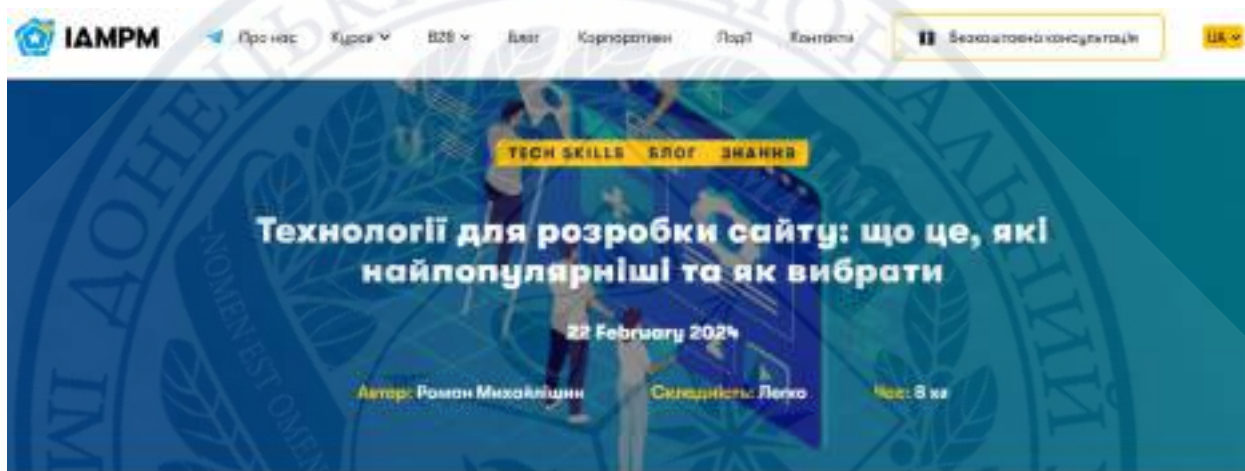


Рисунок 1.4. Сторінка блогу на сайті “IAMPRM”

Відкривши блог про найпопулярніші технології розробки сайтів, одразу зустрічаємо гарне фонове зображення та кольори текстової інформації на ньому. Також одразу вказано автора роботи, її складність та приблизний час, який буде витрачено на те, що ознайомитися із даною статтею. Дуже сучасно розроблений дизайн, добре привертає увагу читача, це безумовно плюс, але варто зазначити, що знову ж, як і в попередньому аналізі, текстова частина, могла б виглядати краще.

Навіщо менеджеру розбиратися в технологіях розробки



Розуміння основних принципів і можливостей різних технологій дає змогу РМ-у ухвалювати обґрунтовані рішення: **на чому писати сайт**, скільки часу на це виділяти, який бюджет закладати. Усе разом допомагає краще керувати ресурсами проекту, включно з найманням ІТ-фахівців, розподілом завдань і винесенням оцінок.

Рисунок 1.5. Текстова частина сайту разом із фотоелементом “ІАМРМ”

Прогорнувши частину тестового наповнення блогу перед читачем постає цікавий фотоелемент, який показує поєднання багатьох технологій в розробці інтернет сторінки. Однак, пояснень або певних посилань на дане зображення в тексті немає. Варто було б у такому більше звертати увагу читача на такий важливий елемент. Щодо шрифту та кольору, текстового наповнення, то він є кращим ніж у раніше розглянутому прикладі, виділені важливі частини, для того щоб безпосередньо вказувати на їх важливість в даній тематиці. Проте розмір тексту можна було би зробити трохи більшим, на телефоні читати даний блог може бути проблематично, особливо для людей, які мають проблеми із зором.

Фреймворки, бібліотеки та бази даних для розробки вебсайтів



Фреймворки — програмні платформи з набором певних інструментів і бібліотек, які потрібні для спрощення роботи за допомогою обраної мови. Вони включають інтерфейси з найчастішими функціями. Серед найпопулярніших:

Рисунок 1.6. Покрокова інструкція приготування на сайті “IAMPM ”

Секція в якій йдеться про новітні фреймворки для розробки сайтів також має багато цікавої, важливої та гарно поданої інформації. Зображення на початку секції є інформативним і одразу показує користувача, про що буде йти мова в наступних декількох абзацах тексту.

Коли робиться порівняльний аналіз сайту, то обов’язково потрібно порівнювати швидкодію сторінок. Для цього можна скористатися онлайн-сервісом PageSpeed Insight. Дуже крутий інструмент, він не тільки дає оцінку, а ще й пропонує вирішення проблем, які можуть виникнути, також можна перевірити оптимізацію телефонної версії сайту. Тож приступаємо до даного аналізу, розпочнемо із сайту “IAMPM”.

Варто зазначити, що порівнювати результати потрібно окремо для десктопної версії сторінки та окремо для мобільної. Перевірка різних параметрів для двох різновидів відрізняється, а також вигляд сторінки від зміни розмірів пристрою перегляду, здебільшого, зазнає суттєвих змін. Для цього беремо URL-

адресу і вставляємо її в пошукову стрічку, зачекавши декілька секунд, отримуємо оцінки із чотирьох критеріїв:

1. Ефективність – показник, який поєднує в собі декілька важливих критеріїв, а саме: час першого відображення контенту, найбільший час відображення контенту, загальний час блокування сторінки, кумулятивне зміщення макету, індекс швидкості. Ці критерії мають бути якомога меншими, для того, щоб рівень ефективності був максимальним.
2. Доступність - рівень доступності на даному сайті такий низький через те, що майже усі фото не мають атрибута “alt”, посилання на соціальні мережі не мають безпосередньо кінцевої адреси посилання, тобто вони є просто заглушками.
3. Оптимальні методи – показник, який має максимальне значення, це вказує на те, що не використовуються застарілі методи, також застосовується протокол HTTPS, який набагато краще захищений, в інструментах розробника в браузері відсутні помилки.
4. Оптимізація пошукових систем – даний параметр немає максимального значення тільки через те, що у фотоматеріалів немає атрибута “alt”, який широко використовується у пошукових системах браузера.

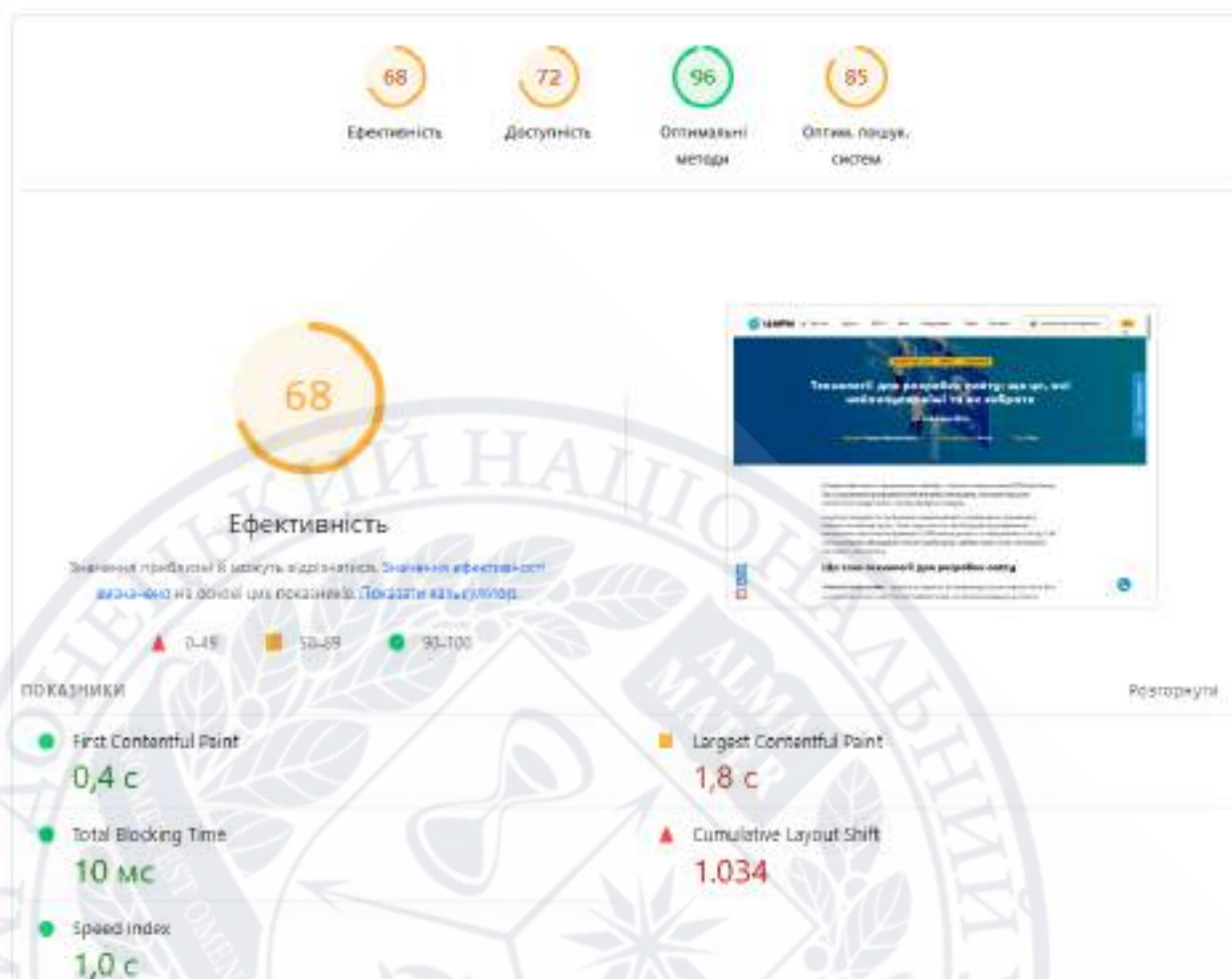


Рисунок 1.7. Аналіз комп'ютерної версії сайту “IAMPRI” за допомогою сервісу PageSpeed Insight

Отримані результати аналізу, виявилися дещо гіршими, ніж були очікування, якщо взяти середнє значення за усіма показниками, то отримаємо оцінку - 80 балів. Це середній показник, зазвичай провідні інтернет сторінки мають показники більші за 90, з кожного параметру. Дана інформація вказує на те, що є великий простір можливостей для покращення швидкодії та стабільності роботи сайту. Також їхня сторінка не найкращим чином оптимізована для пошукових систем, а це може впливати на відображення сайту при запиті користувача, що відповідно зменшить кількість відвідувачів. Однак показник «Оптимальні методи» має хороший результат, тобто сторінка швидше за все захищена протоколом HTTPS та в її розробці були задіяні сучасні підходи до реалізації. Наступним кроком проведемо аналіз мобільної версії сторінки.

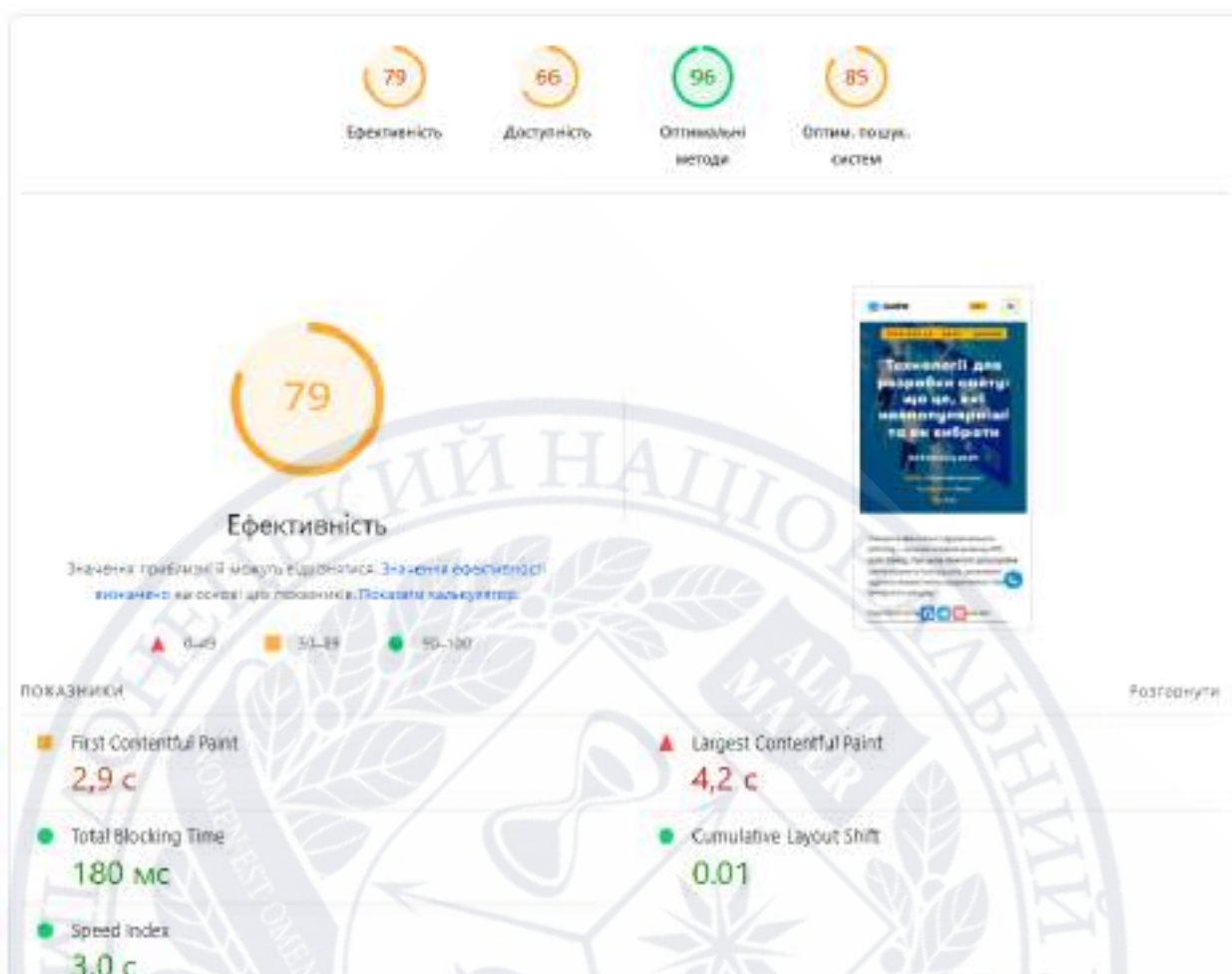


Рисунок 1.8. Аналіз мобільної версії сайту “IAMP” за допомогою сервісу PageSpeed Insight

Результати аналізу, показали покращення загального результату, але в певних категорія, показники погіршились, тому отримаємо оцінку – 81 бал. Це досі середній показник, який вимагає покращень в трьох із чотирьох проаналізованих категоріях. Варто зазначити, що при переході на мобільну версію сторінки, дуже суттєво змінюється уся її побудова. Зміни в розмітці сайту, а також в положенні певних елементів, що може давати як позитивний, так і негативний ефект.

Переглянемо аналогічні кроки із сайтом «ICStudio», після чого буде можливість порівняти їхні результати, та зробити певний висновок за отриманими показниками.

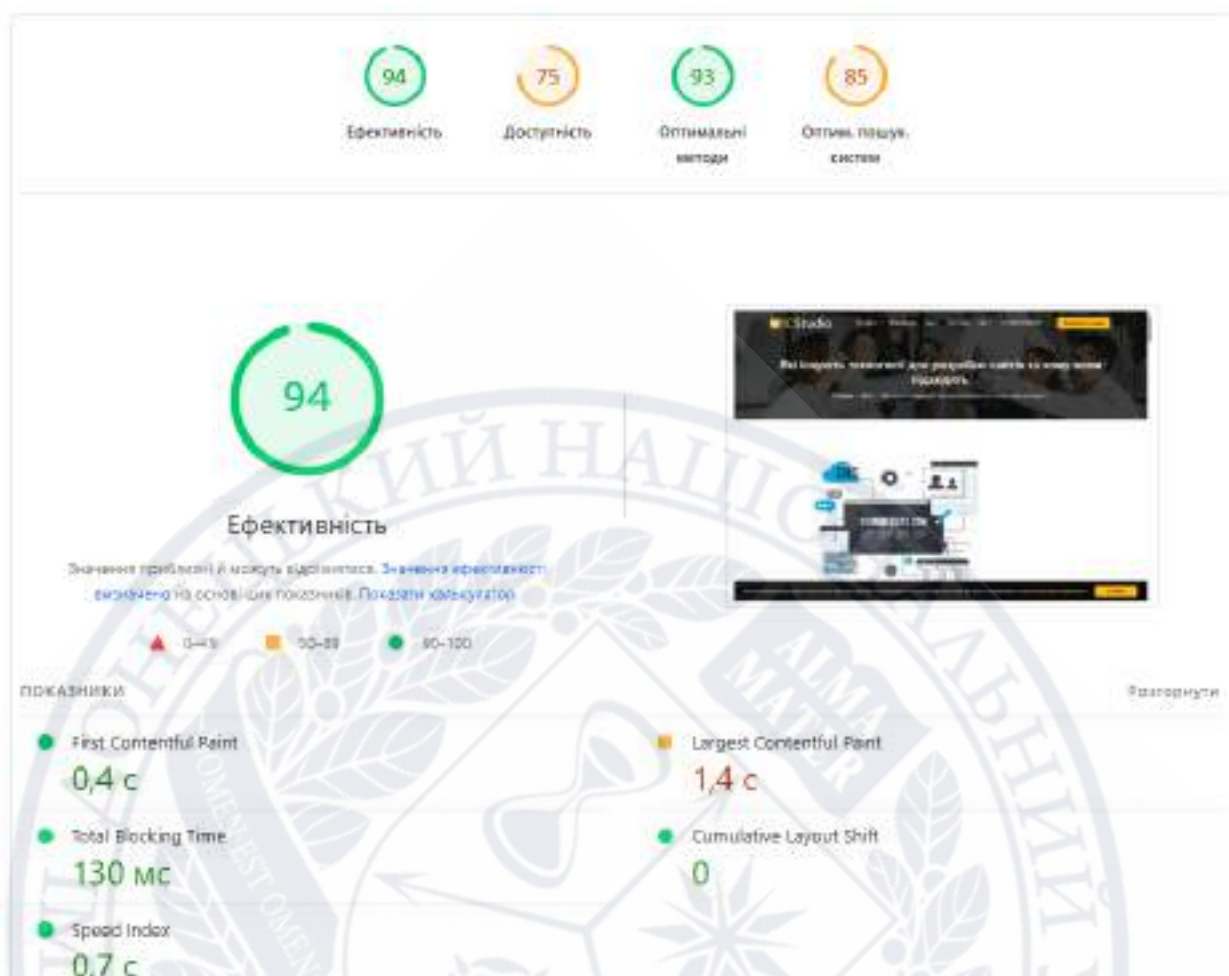


Рисунок. 1.9. Аналіз комп'ютерної версії сайту "ICStudio" за допомогою сервісу PageSpeed Insight

Отримані результати аналізу, виявилися доволі непоганими в двох із трьох категоріях. Показники «Ефективності» та «Оптимізовані методи» мають високі оцінки, що позитивно впливає на загальний результат аналізу. Отримавши середнє значення за усіма показниками маємо оцінку - 87 балів. Це результат близький до тих, які відповідають високому рівню якості розробки інтернет сторінок, але дві категорії все ж мають деякі недоліки, через що даний сайт потребує певних допрацювань. При подальших оновленнях їм варто було би звернути увагу на оптимізацію пошукових систем, адже це дуже важливий показник у просуванні власної сторінки. Наступним кроком проведемо аналіз мобільної версії сторінки.

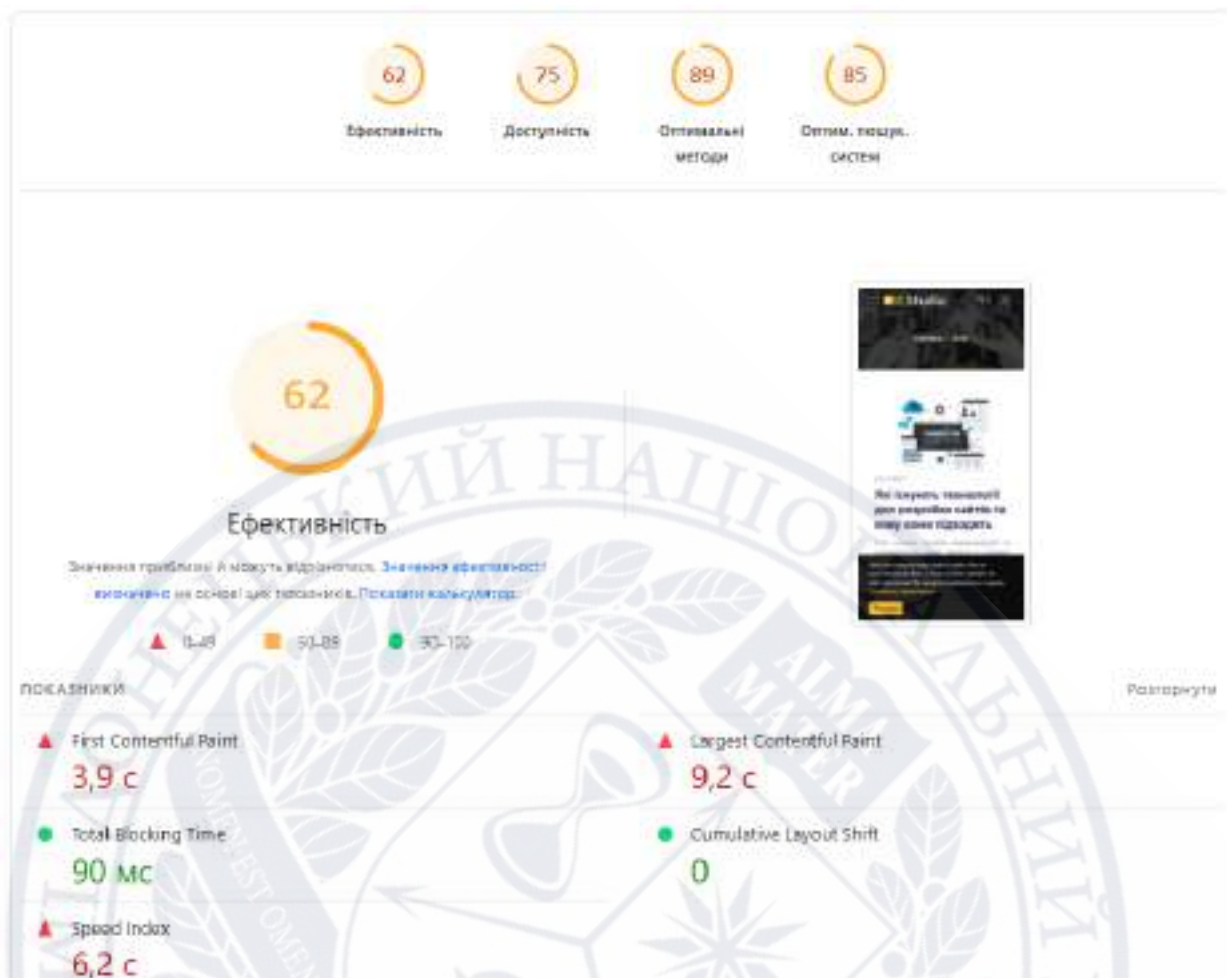


Рисунок 1.10. Аналіз комп'ютерної версії сайту “ICStudio” за допомогою сервісу PageSpeed Insight

Результати аналізу, показали значне погіршення загального результату, в декількох категорія, тому отримаємо оцінку – 78 бал. Це вказує на те, що при розробці уваги на мобільну версію сторінки було набагато менше, в підсумку це відобразилося в результатах. Суттєве погіршення в категорії «Ефективність», більш ніж 30 балів, говорить про не стабільність даної версії сайту.

Порівнюючи результати, отримані з аналізу сайту «ICStudio» та «AIMPM», можна розгледіти певний підхід в першому та другому аналозі. Якщо «AIMPM» обрали стратегію в якій змогли розробити дві версії із практично ідентичними показниками, то «ICStudio» в свою чергу зосередились на комп'ютерній версії свого сайту і мають тут досить хороші результати. В цей час показники мобільної сторінки є значно гіршими. Який із підходів обирати? Варто дуже добре обдумувати даний етап розробки, адже він є дуже важливим.

1.2. Основні етапи у створенні таких сайтів

Коли ми розробляємо вебсайт - ми створюємо інформаційну одиницю в глобальній мережі. І вже декілька десятиліть цю можливість має кожен, хто має потребу та підключення до інтернету [5]. Сайти можуть мати як інформаційний, так і комерційний характер, що допомагає в просуванні бізнесу чи особистого бренду, пошуку і залученню нових клієнтів, а це в свою чергу допомагає збільшити заробіток. А якщо брати в загальному то поділ web-сторінок досить великий, наприклад: новинні портали, соціальні мережі, освітні ресурси, маркетплейси, медичні портали та багато інших [6].

Звісно, у реальності межа між інформаційними та комерційними вебсайтами може бути доволі розмитою. Багато сайтів зараз поєднують у собі обидві функції. І це дуже важливо для власників цих сайтів, щоб ефективно досягати своїх цілей-чи то поширення знань, чи то збільшення продажів і прибутків.

За даними Internet Live Stats, які збирають дані з різних джерел і оновлюються в реальному часі, на серпень 2022 року в Інтернеті було близько 1.9 мільярда веб-сайтів [7]. Більшість цих веб-сайтів вже не використовуються і не відвідуються, але вони досі існують і були у підрахунку.

Хто займається створенням веб-сайтів в Інтернеті? Це досить цікаве запитання. Розробкою сайту може займатися одна людина — так званий Full Stack розробник. Він відповідає як за візуальну частину (інтерфейс), так і за логіку, функціональність і серверну частину проєкту. Такий підхід стає дедалі популярнішим, адже дозволяє оптимізувати ресурси [8].

Однак у більшості компаній процес розробки все ще розділений на дві спеціалізації: Frontend-розробник, який відповідає за зовнішній вигляд сайту, та Backend-розробник, який працює з серверною частиною та внутрішньою логікою. Таким чином, сьогодні створити веб-сайт може практично будь-хто — бізнес, урядова установа, організація чи навіть окрема людина — залежно від потреб і ресурсів.

Що можна зробити на веб-сайті? Більшість веб сайтів містить велику кількість інформацію, яку відвідувачі мають змогу прочитати. Майже обов'язково на кожному сайті присутні гіперпосилання, ви за бажанням, можете переходити за ними, натискаючи на них, щоб знайти більше інформації або виконати якусь певну дію. Також є можливість відтворювати відео, слухати музику, купувати, спілкуватися, завантажувати файли на власний комп'ютер або ж навпаки, в Інтернет, віднедавна з'явилась можливість у вільному доступі користуватися засобами, які використовують штучний інтелект та багато інших функціональних можливостей на веб-сайтах [9],[10].

Розробка архітектури сайту є одним із найважливіших етапів створення нового проекту, оскільки містить в собі блок робіт, під час яких продумується внутрішня і зовнішня структура онлайн-ресурсу. Саме під час розробки архітектури сайту обирається тип зв'язку між усіма його елементами, відповідно до вимог замовника, а також правил юзабіліті та очікувань цільової аудиторії.

Відкриваючи вебсторінку в Інтернеті користувач отримує результат роботи тих чи інших технологій. На сьогоднішній день, їх налічується дуже багато. Коли заходить мова про технології для створення веб-сторінок, то першим ділом згадують мову гіпертекстової розмітки HTML. HTML описує структуру вебсторінки за допомогою семантичних елементів, які називаються тегами, браузер під час завантаження сайту перетворює їх у відповідні графічні об'єкти. Через декілька років, після великої кількості масштабних оновлень, мова HTML отримала функціонал, який дозволив використовувати різні механізми для додавання стилів до елементів сторінки. Постійні покращення HTML розв'язали руки вебдизайнерам в плані створення зовнішнього вигляду сайту, але для розробників написання коду ставало значно все складніше, складніши і для написання, і для редагування. Спершу, існувала велика проблема із запуском сайту в різних браузерах, оскільки сторінка дуже часто відображалася по-різному, тому основною ціллю стало вирішення даного питання [11]. Організація W3C, що є головною міжнародною організацією, яка впроваджує новітні технологічні стандарти для мережі веб, як оптимальний варіант обрала

технологію CSS, тобто каскадні таблиці стилів. Однією із переваг CSS була можливість використовувати різні таблиці стилів до одного документу або сторінки [12]. CSS файл - це набір правил з форматування, оформлення та позиціонування, вони застосовуються або окремих до елементів сторінки, або загалом до усієї сторінки. Поділ на класи або за тегом, дав можливість легше впорядковувати стилі. CSS ідеальний інструмент для розробки візуального вигляду сторінки. З появою нових побажань від користувачів сайтів, з'явилася необхідність того, щоб сторінки були динамічнішими, одним із варіантів того, як вирішити цю проблему було додавання анімацій для деяких елементів. Мінімальні інструменти для створення анімації також є в CSS. Розробникам цього функціоналу було мало, спеціально для цього створили мову програмування для інтернет сторінок та назвали JavaScript. Вона являється динамічною, об'єктно-орієнтованою прототипною мовою програмування. З самого початку JavaScript дуже добре інтегрувалася у веб-переглядачі [13],[14]. Також для даної мови програмування написали декілька дуже зручних у використанні та дуже оптимізованих фреймворків, про які я детально розповім дещо пізніше. Отже, розробка сайту – дуже непростий і клопіткий процес, якщо було прийнято рішення писати власну сторінку самостійно. Зараз в мережі Інтернет, безліч різноманітних студій веб-дизайну, кожна з них пропонує швидко, але іноді не завжди якісне рішення, тобто розробка сторінки, також це може бути досить дороговартісна послуга. Проте вихід є, для того, що сайт був унікальний, просто потрібно зробити його власноруч, а в разі потреби завжди буде можливість внести потрібні корективи. Етапи створення сайту, показані нижче [15].

Основні етапи розробки інтернет сторінки:

1. Визначення вхідних даних для майбутнього сайту;
2. Обрати дизайн зовнішнього вигляду, а також визначити бажаний функціонал сайту;
3. Формування структури сайту зокрема – розділів і меню;
4. Створення дизайн-макета головної сторінки сайту;

5. Html-верстка, дизайн і створення внутрішніх сторінок;
6. Розробка додаткових сторінок, оптимізація зображень;
7. Стилiзація частин сайту відповідно до макету;
8. Програмування: визначення завдань програмування, перевірка правильності виконання функціональних можливостей;
9. Зовнішнє та внутрішнє тестування.
10. Завантаження проєкту на сервіс хостингу [16].

Усі сайти можна поділити на певні групи за функціоналом або за призначенням чи за контентом, кожна сторінка може бути різною за своєю спеціалізацією та призначенням. Під час пошуку найкращих, переглянуто декілька десятків сайтів і виділив декілька груп сайтів. Ось деякі із них:

1. Портали з ресурсами: Ці сайти пропонують велику колекцію інформації про інструменти, бібліотеки, плагіни та шаблони як для візуального дизайну так і для розробки, включаючи розділи за типом інструменту, технологіями, категоріями стилів, призначенням та іншими параметрами.
2. Платформи для спільноти: Такі сайти об'єднують дизайнерів, розробників та користувачів, які зацікавлені в даній тематиці, а також інших фахівців для обміну напрацюваннями, порадами, прикладами робіт, навчальними матеріалами та іншою корисною інформацією.
3. Блоги та медіа: Розробники та експерти діляться своїм досвідом, оглядами інструментів, експериментами, ідеями та сучасними трендами у сфері, часто розповідаючи про особисті враження та історії свого професійного розвитку.
4. Навчальні платформи: Сайти, які пропонують навчальні матеріали, відеоуроки, онлайн-курси, вебінари та інші ресурси для тих, хто бажає опанувати нові інструменти та вдосконалити свої навички.

То ж розробка сторінки одного із вище перелічених типів, може дещо відрізнятися, однак основні етапи все ж залишатимуться тими самими. Попри те, сторінок, які активно підтримуються та мають сучасний дизайн, високий рівень

швидкодії та підтримуються за допомогою сучасних технологій, не дуже багато, зазвичай вони тільки в лідерів в даній сфері. Одним із таких є обрані для порівняльного аналізу, два сайти, які мають приблизно однакову структуру, функціонал та виділяються на фоні конкурентів, а саме: «IAMPM» та «ICStudio».

1.3 Постановка задачі.

Для того, щоб провести постановку задачі потрібно виокремити конкретні задачі, які мають бути виконані. Основні завдання, які необхідно вирішити в рамках розробки сайту:

1. Створити структуру сайту, що складатиметься з:
 - Головної сторінки з функціоналом аналізу за посиланням.
 - Сторінка із функціоналом аналізу за текстом.
 - Сторінка «Про нас».
2. Написати код, який відповідатиме за обробку отриманих даних від користувача, та зможе за допомогою різних підходів провести аналіз та сформулювати рекомендацію щодо використання тих чи інших інструментів та технологій.
3. Забезпечити адаптивну верстку, що дозволить комфортне користування сайтом як на персональному комп'ютері, так і на мобільних пристроях з шириною екрану від 360 пікселів.
4. Реалізувати приємний для візуального перегляду користувацький інтерфейс для обох версій сайту.
5. Реалізувати просту і зрозумілу систему навігації та можливість звернутися до розробника в разі непередбачуваних ситуацій або за допомогою.

Результат, який очікується — готовий до використання сайт, який слугуватиме допомогою для розробників і початківців у сфері розробки користувацького інтерфейсу, зокрема обрати найкращі інструменти для роботи над відповідними завданнями.

РОЗДІЛ 2

АНАЛІЗ ТА ВИБІР ПРОГРАМНИХ ІНСТРУМЕНТІВ ДЛЯ РЕАЛІЗАЦІЇ САЙТУ

2.1. Інструменти для розробки дизайну інтернет сторінок

З кожним роком індустрія вебдизайну розвивається шаленими кроками. Щодня, сотні тисяч людей працюють над розробками макетів, з'являються нові підходи, цікаві ідеї та новітні інструменти. Кожен із них, має свої особливості та спеціалізацію, щоправда інколи краще, щоб в одному інструменті було поєднано усе необхідне для розробки. Зазвичай, маючи особисті розмови із вебдизайнерами, мова йде саме про один із провідних інструментів – Figma. Але варто зауважити, що це не єдиний із доступних та функціонально непоганих наповнених виборів для розробки макетів. Canva, Sketch, Adobe Illustrator та Adobe Photoshop – усі вони також можуть використовуватися для тих чи інших завдань у роботі із дизайном. Проте в кожного із перелічених інструментів є свої доволі суттєві мінуси. Прикладом може бути Canva – дуже цікавий інструмент, якщо потрібно стилізувати якісь окремі частини сторінки, є можливість перенести напрацювання у Figma. Однак те, що вони не вийшли із ринку росії, стає вирішальним моментом у виборі. Для продуктів компанії Adobe – все набагато простіше, їхні програми надають широкий спектр можливостей, які недоступні в інших інструментах, але місячна оплата на підписку є занадто дорогою для студентів, які не користуються цими продуктами на постійній основі та не заробляють виконуючи роботи за допомогою даних інструментів. Можливість пробного періоду на 1 тиждень надає можливість ознайомитися із інтерфейсом та функціоналом, але для повного освоєння продукту може знадобитися значно більше. Тож було прийнято рішення зосередитися на одному інструменті, детально ознайомитися із його можливостями та усіма аспектами, про які варто згадати. Цим продуктом стала Figma, вона є лідером в даній сфері.

Даний інструмент максимально орієнтований для користування як дизайнерами, так і верстальниками. Тобто важливо розуміти, що це значно пришвидшує загальний процес розробки, та зменшує шанс того, що потрібні будуть доопрацювання зі сторони дизайнера або веб-розробника.

Figma - це сервіс для дизайну і прототипування, що дозволяє дизайнеру створювати векторні макети, макети і інтерфейси користувача для вебсайтів, мобільних додатків та інших цифрових продуктів.

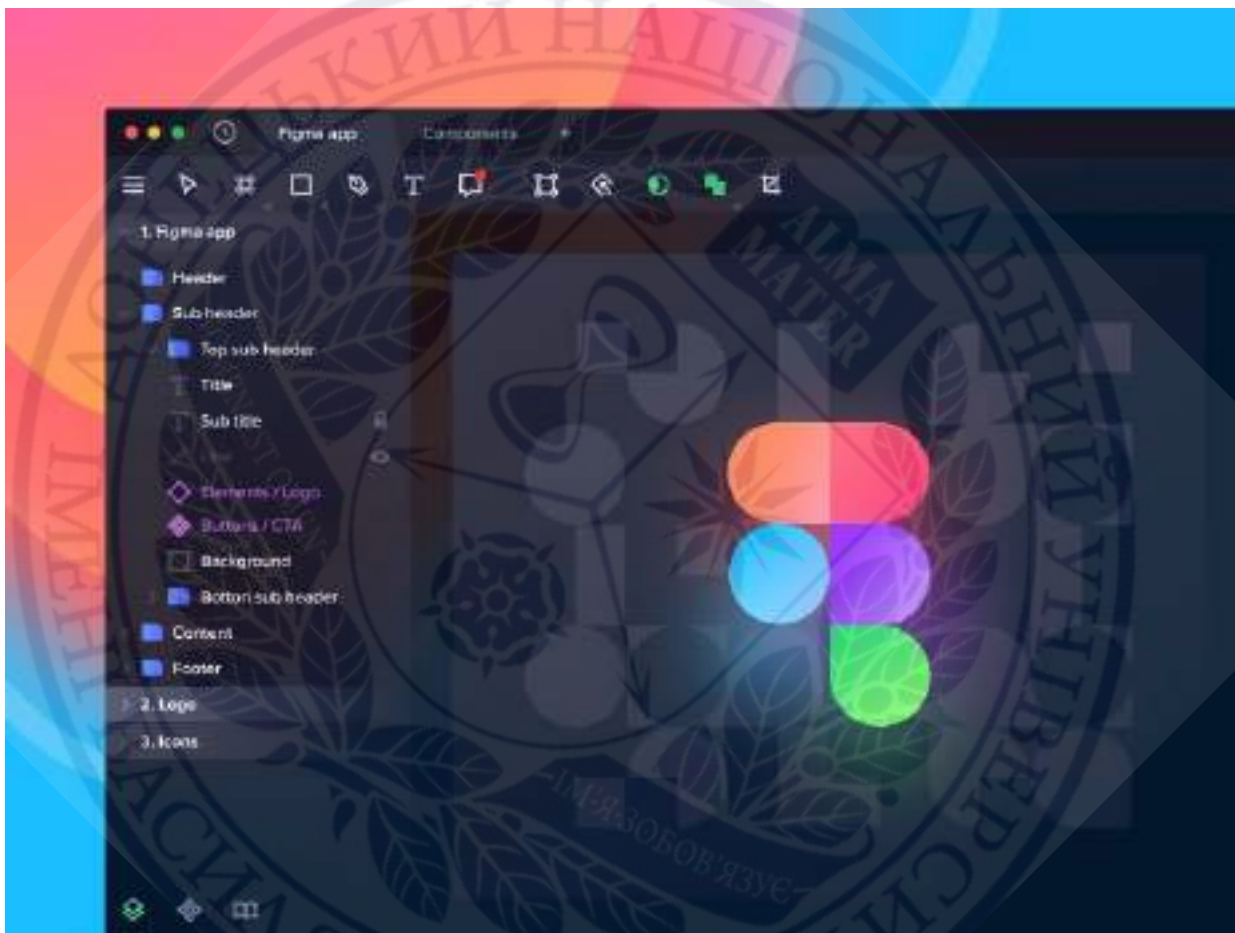


Рисунок 2.1. Робочий екран Figma

У Figma дуже простий і зрозумілий інтерфейс, є можливість спільно із іншими розробниками взаємодіяти в реальному часі через вбудований чат. Крім того, Figma підтримує функції відкритості для інтеграції з іншими інструментами та розширеннями, що робить його популярним інструментом для дизайну у багатьох командних середовищах[17].

У Figma є можливість зберігати документи на власному пристрої, а також усі макети зберігаються у хмарі. Це надає чудову можливість для зручної

колективної роботи, макети можна відкривати за посиланням, що є дуже зручно, варто просто зберегти URL і без проблем працювати із даним файлом.

Figma має як браузерний додаток так і десктопний, можна завантажити програму на будь-яку операційну систему, та продовжувати працювати. Десктопна версія гадає можливість працювати в офлайн режимі, натомість одразу ж при підключенні до мережі Інтернет, зміни будуть синхронізовані [18].

Основні переваги Figma, які варто відзначити:

1. Прототипування. Figma дозволяє зробити клікабельну версію сайту або мобільного додатка, створюючи безліч екранів. За допомогою цього користувачі можуть протестувати сайт. Згодом після отримання потрібних даних, розробники можуть налаштувати будь-яку функцію: кнопки, спливаючі меню, анімовані та модальні вікна. Коли прототип готовий, ви можете увімкнути режим презентації, тоді можна натискати на кнопки, переходити між розділами сайту, перевіряти як працюють пункти меню.

2. Компоненти. У роботі над великими проектами є дуже корисною є функція Figma «Компоненти». Компоненти - це повторювані UI елементи в проектах. Компоненти у Figma можуть бути основними і залежними. Якщо ви намалювали кнопку, зробили її компонентом, а потім скопіювали, то перша кнопка буде основним компонентом, а друга залежним. Зміни, які ви внесете до основного компонента, будуть застосовані і до залежного компонента.

3. Спільне редагування у Figma. Вносячи зміни члени команди можуть залишати коментарі або обговорювати будь-який елемент у проекті. Це дуже зручно, коли команда працює над проектом не одночасно, а по-різну, тоді для максимальної зручності можна залишати коментарі в яких будуть прописані зміни або навпаки будуть задаватися питання.

4. Контроль версій. У Figma можна відстежувати історію дій кожного з учасників команди, переглядати та відновлювати попередні версії документа. Сервіс автоматично зберігає версії при закритті вкладки проекту.

5. Присутня можливість безкоштовного завантаження інструменту для особистих цілей (Некомерційна діяльність) [19].

Самостійно працюючи, в своєму обліковому записі, ви матимете доступ до абсолютно усього функціоналу абсолютно безкоштовно. Усе ще потрібно – просто зареєструватися і одразу можна починати працювати. Доступ до перегляду документа іншим користувачам за допомогою посилання — також безкоштовно.

Потрібно зауважити, що є і деякі обмеження. Редактор створений насамперед для вебдизайну, в ньому не найприємніше працювати з поліграфією. У Figma немає СМУК палітри та перемикання розмірності сітки на см/мм. Неприємне обмеження це те, що у Figma не має можливості прямого імпорту файлів із Photoshop. Однак, якщо перевести макет у формат Sketch, проблем із імпортом більше не буде [20].

Підіб'ємо підсумок:

Figma являється дуже добре продуманим інструментом, який спеціально зроблений дизайнерами для дизайнерів. Можна визначити такі головні переваги редактора:

- Можливість спільної роботи над одним документом — створювати і редагувати в режимі реального часу.
- Документи зберігаються у хмарному сховищі – макети не потрібно завантажувати, якщо немає такої потреби, можна з легкістю показати колегам чи замовнику.
- Кросплатформенність – можливість працювати з редактором на Windows, Mac, Linux.
- Можливість створювати інтерактивні прототипи, векторні сітки, головні та залежні компоненти.
- Зручність подальшої роботи з макетом безпосередньо верстальнику [21] .

2.2 Інструменти для верстки та розробки необхідного функціоналу

Першим вибором буде середовище програмування, тобто редактор коду, яким широко використовується як Front end так і Back end розробниками – це Visual Studio Code. Надзвичайно зручна IDE, яка являється лідером у

використанні саме веб-розробниками. Даний інструмент дозволить організації створити ідеальне середовище розробки для своїх працівників [22].

Серед особливостей, які присутні в даній IDE можна відзначити:

- Поліпшення якості коду та виявлення проблем на ранній стадії;
- Створюйте, керуйте, захищайте та діліться програмними компонентами;
- Інструменти для різних видів тестування продукту;
- Система контролю версій із приватними репозиторіями, не дасть зробити фатальну для проєкту помилку;
- Visual Studio Code розроблений у 2015 році і компанією Microsoft;
- Повна інтеграція з Make, Ant, Gulp, Jake, Rake та MSBuild;
- Забезпечує інтерфейс перетягування, підсвічування синтаксису, підсвічування помилок, користувацькі гарячі клавіші;
- Теми, що настраюються, інтеграція з Git, автозаповнення, спільна робота і попередній перегляд в реальному часі;
- Підтримка різноманітних плагінів, декілька найпопулярніших, таких як GitHub Copilot, Batch Rename, Gatito Theme та Nightwatch;
- Забезпечує розширюваність, зіставлення дужок, автоматичний відступ, бох-вибір;
- Пропонує кілька мов програмування, таких як JavaScript, HTML, C, C#, CSS та TypeScript, та безліч їхніх фреймворків;
- Підтримується на платформах - Linux, macOS та Windows;
- Цінова політика виглядає наступним чином - плани розпочинаються з 45 доларів на місяць. Присутня пробна версія, яка включає довічний безкоштовний базовий план з майже усіма доступними функціями;

З вище перелічених вище фактів, зрозуміло, що даний програмний інструмент є мало не ідеальним для розробки сайтів, включає всі необхідні

мови програмування та особливості для швидкої та комфортної роботи [23], [24].

Наступним редактором коду, який заслуговує уваги є – Dreamweaver.

Dreamweaver – популярний інструмент IDE для веб-розробки. Цей інструмент допомагає вам створювати, публікувати веб-сайти та керувати ними. Сайт створений за допомогою DreamWeaver можна завантажити на будь-який веб-сервер.

Серед особливостей, які присутні в даній IDE можна відзначити:

- Динамічні веб-сайти можна швидко розробити за допомогою Dreamweaver, також ви можете створити веб-сайт, який підходить для будь-якого розміру екрана.
- Цей інструмент допоможе вам налаштувати робочий простір так, як вам подобається.
- Присутній убудований валідатор HTML для перевірки вашого коду.
- Легко інтегрується з Fireworks та Flash.
- Пропонує готові шаблони для блогів, електронної комерції, інформаційних бюлетенів та портфоліо.
- Dreamweaver була заснована в 1997 році та розроблена Macromedia.
- Забезпечує інтерфейс перетягування, підсвічування синтаксису, підсвічування помилок, користувацькі гарячі клавіші, інтеграцію Git та попередній перегляд у реальному часі.
- Підтримка плагінів, таких як PVI та Code View.
- Забезпечує швидке та гнучке кодування, плавне редагування в режимі Live View, підтримку кількох моніторів, оновлений та сучасний інтерфейс користувача.
- Пропонує кілька мов програмування, таких як HTML, XML, CSS та JavaScript.

- Підтримується на платформах - Linux, macOS та Windows;
- Цінова політика наступна – плани розпочинаються з 26.99

доларів на місяць присутня безкоштовна пробна версія на 7 днів, тому дана IDE не користується такою популярністю, як Visual Studio Code [25].

Після аналізу поданої вище інформації про найкращі IDE для фронтенд-розробки, складемо таблицю, на основі якої буде прийматися рішення, щодо вибору інтегрованого середовища програмування для даної роботи.

Таблиця 2.1 Порівняння інтегрованих середовищ програмування

Критерій	Інтегроване середовище програмування (IDE)	
	Visual Studio Code (VSC)	DreamWeaver
Підтримується на усіх платформах	Так	Так
Інтегрованість Git в середовищі програмування	Так	Часткова
Підтримувані технології для фронтенд-розробки	JavaScript, HTML, Python, C, C#, CSS та TypeScript, та інші.	HTML, XML, CSS та JavaScript
Безкоштовна версія	Так	Ні
Ціна максимальної версії за місяць користування	3,75\$ - місячна підписка(45\$ - річна підписка), але є можливість використання безкоштовної версії, яка включає майже весь функціонал.	26,99\$ - місячна підписка, є пробна версія на 1 тиждень.

Отже, переглянувши порівняння надане в Таблиці 1, найкращим варіантом для вибору інтегрованого середовища програмування (IDE), було обрано саме Visual Studio Code.

Оскільки розробка сайту неможлива без HTML та CSS, оскільки це дві основоположні технології. Також JavaScript щороку стає все більш незамінною мовою програмування для веб-розробки, бо вона найбільш оптимізована і зручна для роботи із браузерами. Але розвиток технологій на цьому не зупинився, оновлення та розробка нових фреймворків та бібліотек, тому вирішив трохи

глибше дослідити дану тематику. Порівняти в чому різниця між двома фундаментальними підходами в розробці сайтів - Multiple-page Application та Single-page Application. Від вибору, який підхід для певного завдання підходить більше, розробник робить невеликий підсумок і обирає як йому краще реалізовувати поставлене завдання. Використати “чистий JS” або популярні фреймворки, та як React.JS, Vue.JS та Angular.JS [26].

Отож, в чому відмінність між Multiple-page Application та Single-page Application? Візьмемо довільний вебсайт, для прикладу візьмемо колекцією фотоелементів на тематику осінні пейзажі, розкладом тренувань тощо. Для кожного сайту притаманно мати декілька сторінок, якщо це не односторінковий сайт, наприклад: домашня, профіль, сторінка колекції та сторінка одного елемента колекції.

Multiple-page Application - декілька років тому була б головним претендентом як технологія для реалізації, тому що включає декілька окремих HTML-сторінок.

- Архітектура клієнт-сервер
- Вся логіка живе на сервері
- На кожен запит сервер надсилає готовий HTML-документ
- Перезавантаження сторінки при кожному запиті
- Погана інтерактивність
- Відмінне SEO [27]



Рис. 2.2. Візуальне представлення підходу Multiple-page Application

Single-page Application – більш сучасний підхід, який полягає в тому, що при переході на посилання іншої вкладки сторінки, не відбувається перехід на інші HTML-сторінки. Користувацький інтерфейс, замість того, щоб отримати новий набір файлів з яких відбувається відображення, перемальовується на клієнті, на одній і тій самій сторінці, без перезавантаження.

- Архітектура клієнт-сервер
- При завантаженні сайту сервер завжди віддає стартову HTML-сторінку index.html
- Кожен наступний запит на сервер отримує лише дані у JSON-форматі
- Оновлення інтерфейсу відбувається динамічно на клієнті
- Завантаження першої сторінки може бути досить повільним (лікується)
- Логіка, не пов'язана із безпекою, живе на клієнті

- Слабке SEO (лікується)

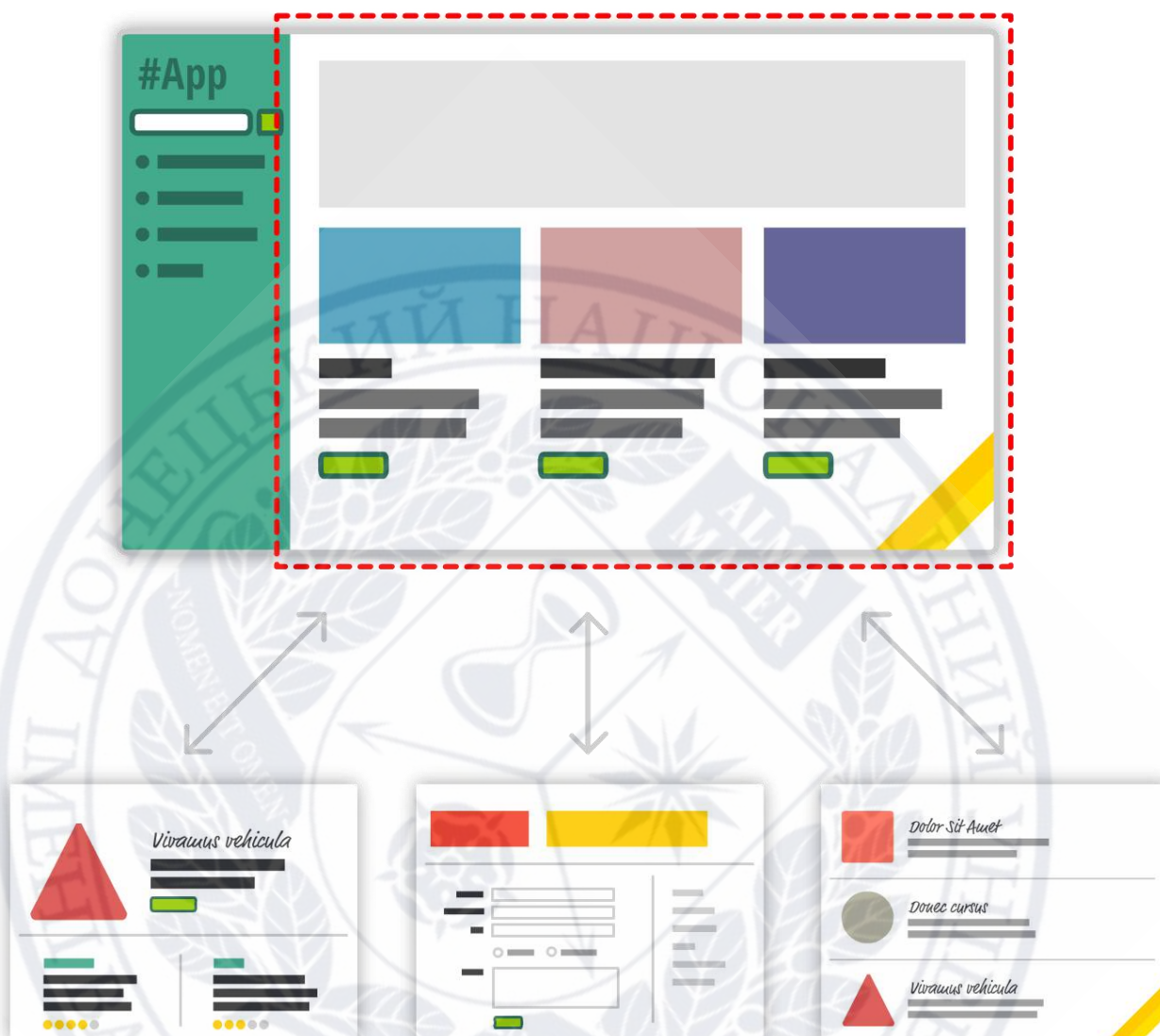


Рисунок 2.3. Візуальне представлення підходу Single-page Application

Хоч і Multiple-page Application виглядає дещо застарілим, але він до зараз дуже широко використовується, тоді, коли на сайті не потрібно велика кількість інтерактивності, тоді даний підхід дуже навіть актуальний. Single-page Application – важчий в розробці та підтримуванні, і як наслідок більш дорожчий в усіх можливих планах [28].

JavaScript та декілька бібліотек – все, що потрібно для розробки сайту. Але для створення динамічно змінювальних частин сторінки, функціоналу самої мови програмування деколи не вистачає, для цього було розроблено топові

фреймворки – React.JS, Vue та Angular. Для кращого розуміння, які переваги пропонує кожен із вищеперечислених програмних платформ, було вирішено більш детально розповісти про них, та можливо, трохи порівняти їх між собою [29],[30].

React.JS — це іноваційна бібліотека, що після двох років від випуску стала однією із найпопулярніших для сучасної веброзробки на JavaScript. React.js — це фреймворк JavaScript, який надає доступ до вихідного коду, розроблений Meta Platforms. Швидке та ефективне написання користувацьких інтерфейсів із використанням невеликої кількості коду, значно меншої ніж при написанні звичайним JavaScript. У React компоненти відповідають за розробку частин сайтів або додатків, їх можна багаторазово використовувати, та навіть потрібно розглядати як незалежні блоки, адже вони пишуться окремо та можуть використовуватися абсолютно різними варіаціями. Компоненти являються окремими частинами інтерфейсу. В зборі такі частинки складають цілісний та сучасний інтерфейс для користувача. Якщо зазвичай ми беремо до розгляду інтерфейс сторінки як єдине ціле, то React.JS має свій цікавий підхід. Розробникам надається можливість розробляти окремі компоненти, які в подальшому будуть використовувати знову і знову. Утворюється щось подібне на будівельні блоки, які призначені для того, щоб скласти цілісний користувацький інтерфейс. Платформа React JS поєднує і швидкість, і ефективність JavaScript, що є його головними перевагами. Також дуже ефективно маніпулювання DOM-деревом, швидкий рендеринг сторінки. Це функціональні компоненти, які отримують ззовні певні вкладені властивості (props) і повертають в результаті розмітку JSX (JavaScript XML). React.Js здобув широку популярність завдяки можливості швидкій, ефективній та масштабованій розробці вебсайтів. Сьогодні його застосовують багато тисяч різноманітних проєктів – від всесвітньо відомих компаній до нових стартапів. Серед них: Facebook, Instagram, Netflix, Reddit, Uber, Airbnb, SoundCloud та інші. Однією з ключових особливостей React є використання віртуального DOM. Оскільки віртуальний DOM працює набагато швидше, ніж стандартний, це дозволяє

суттєво покращити продуктивність сторінок і знижує навантаження на браузер [31] .

Над оновленнями ReactJS працює велика кількість розробників з різних країн, що допомагає постійному розвитку фреймворка. Щодо недоліків React, то це - не велика кількість документації, але з кожним днем з'являється все більше і більше матеріалів, синтаксис є досить складним та заплутаним для новачків.

На відміну від React, Angular, розроблений усім відомою компанією-гігантом Google. Наразі не можна вважати Angular бібліотекою для роботи з DOM, з недавніх пір це вже абсолютно точно повноцінний фреймворк. Він керується наступним MVVM (Model-View-ViewModel) принципом побудови застосунків або сайтів та має такі можливості, як:

- Angular пропонує CLI-інструмент, який дозволяє швидко генерувати структуру файлів для різних компонентів, створювати прт-бібліотеки, та запускати різні тести.
- У даному фреймворку доступний великий набір бібліотек для різноманітних завдань — Router(для маршрутизації), інструменти для покращення обробки форм, анімації та багато інших.
- Під час ініціалізації нового проєкту є широкий вибір різноманітних налаштувань, також є можливість вибору які з доступних бібліотек будуть додано до нього.
- Angular також підтримує інтегровані засоби для лінтування коду.
- Для написання та запуску Unit і End to End тестування можна використовувати інструменти Karma та Jasmine [32].

Angular з'явився трохи згодом після написання React, хоча базується на фреймворку AngularJS, який є попередником для обох. Angular у порівнянні з React, вирізняється набагато складнішою структурою. Окрім компонентів, даний фреймворк містить додаткові структурні елементи, кожен з яких виконує свою окрему функцію:

- Пайпи — це допоміжні інструменти, вони широко використовуються для редагування візуального вигляду даних у HTML-шаблонах.

- Директиви — класи, що додають функціонал іншим елементам сторінки. Зазвичай прописуються як атрибути елементів розмітки.

- Компоненти — основна одиниця інтерфейсу. Поєднують в собі стилі, HTML-файл, а також логіку написану на JavaScript та TypeScript.

- Сервіси — головне призначення централізованого зберігання та обробка даних. Повторне використання пайпами, директивами, компонентами й навіть іншими сервісами дає значне полегшення в написанні коду.

- Модулі — дозволяють легко та логічно поєднувати усі вищезгадані елементи у функціональні блоки.

Згаданий раніше при розгляді React, JSX, дозволяє розробнику вписати будь-який валідний JavaScript код напряму в розмітку. В той час Angular маніпулює DOM-деревом вбудованими директивами [33]. Як і в будь-чому в Angular є свої мінуси: вхідний поріг вимагає від розробника великий обсяг потрібних знань, які будуть мінімальним набором для того, щоб почати писати на Angular, тобто поріг входу є досить високим в порівнянні з іншими фреймворками. Відсутність української мови в переліку мов офіційної документації. Виділено саме ці проблеми, є ще й інші, але їх можна більш-менш просто владнати [34].

Vue - Наймолодший з усіх перелічених найбільш популярних бібліотек/фреймворків. Версія 2.6.x це — актуальна версія, її широко використовують розробники різних рівнів на абсолютно різного рівня проектах, але вже дуже активно розробляються нові версії, які проходять етап тестування, тому можливо, найближчим часом статус LTS версії буде змінено. Vue можна назвати таким собі поєднанням ідеї Angular та React. З одного боку, у базовому вигляді він вважається бібліотекою для маніпуляцій з DOM, точно так само як і React, але з іншого боку — це повноцінний фреймворк, з власним Vue CLI та доволі великою кількістю бібліотек навколо нього. Однією із особливостей для

Vue часто використовується підхід Single File Component. Суть полягає в тому, щоб в одному файлі зберігати усе: стилі, розмітка і логіка компонента. Є деякі вагання в тому, що це дуже зручно у використанні, але такий підхід має право на існування.

Якщо поглянути на можливості даного фреймворку, то можна знайти не один пункт, який співпадає і з React, і з Angular:

- Елементами DOM-дерева маніпулюємо за допомогою директив, прямого аналога Angular.

- Vue2.6 застосовується декларативний підхід до логіки компонента, тобто створюється об'єкт, що має відповідні властивості, методи та конфігурацію нашого компонента. З Vue 3, з'явилась підтримка Composition API, що наблизило структуру компонентів дуже подібним на тіло функціональних React компонентів.

- У Vue зміна стану реалізується шляхом безпосереднього мутування властивостей компонента.

До плюсів варто віднести високу продуктивність збірки, її збірку, а також роботу сторінки в загальному. Вхідний поріг – це доволі середній рівень знань, щось між Angular і React, але трохи ближче до React. Вивчити та почати писати код на Vue набагато легше тим, хто має певні знання про React.JS [35].

Серед очевидних мінусів – бібліотека дуже рідко оновлюється, бо як такої команди розробників, яка б постійно працювала над Vue – немає. Творець бібліотеки та небайдуже ком'юніті – основні рушії оновлень фреймворку. Масштабні оновлення Vue супроводжуються значними змінами, які найчастіше не є сумісними з попередніми версіями. Це вносить багато головного болю для розробників, бо значно ускладнює підтримку сайтів і додатків. У версії 2.6 можна зустріти помилки, які не тільки важко вирішити, але й неможливо знайти рішення в Інтернеті, а це свідчить про досить слабку офіційну документацію. Практично відсутня підтримка Typescript.

Проаналізувавши три топові фреймворки для веброботників, можна підвести невеликі, суб'єктивні, підсумки:

• React — неймовірно комфортний фреймворк, який часто оновлюється, має достатню кількість документації, легкий в розумінні та написанні коду, надзвичайно потужний, головне писати на ньому «правильно». Завдяки можливості використовувати великий спектр доступних інструментів, обов'язково Typescript, а також стандартизованих й класно задокументованих бібліотек, цей підхід ідеально підходить для реалізації складних проєктів. Однак, за грамотного використання усіх ресурсів, чіткої структури компонентів та файлів, можна з високою ефективністю виконувати й доволі прості задачі.

• Angular — доволі зручний у використанні, проте потрібно дуже уважно поставитися до наявної документації, повністю її дослідити. Він активно підтримується, отримує регулярні оновлення, часто додається новий функціонал, але і не забувають про вдосконалення комфортності розробки. Це потужний фреймворк, який дуже добре масштабується. Включає в себе різні інструменти, від Typescript до тестування за допомогою Unit тестів E2E/Integration тестів. Усе, що було перелічено, вже є у фреймворку, що значно полегшує життя розробникам, адже не потрібно шукати інформацію і можливості підключення тестів для проєкту. В основному краще підходить для масштабних, складних проєктів, але також можливе використання для реалізації простих задач за потреб та гарно підібраного підходу..

• Vue — щодо комфорту написання коду дуже різні відгуки, але все ж-таки дещо більше негативних, проблема помилок, які постійно потрібно гуглити дуже збільшують час розробки. Проте відгуки про швидкість того, як все компілювалось і в кінцевому результаті працювало просто неймовірно, думаю, якщо сповна оволодіти даним фреймворком, то можна досягати дуже крутих результатів [36],[37].

Оскільки для даної роботи не було нагальної потреби використовувати якийсь із перелічених фреймворків/бібліотек, вирішено, що розробка сайту

“АналізаторСайтів” буде за допомогою “чистого JS” та декількох бібліотек для стилізації та полегшеної роботи із анімацією та запитами.

2.3 Оцінювання та вибір середовища для розміщення проєкту.

Для розробки даного сайту, також обов’язково потрібно продумати місце розміщення нашого проєкту. Власне це і є останній крок вибір хостинг провайдеру, на серверах якого буде безперервна робота розробленого сайту. До вибору серверу провайдера, варто поставитися дуже уважно, бо деякі провайдери можуть не підтримувати ті чи інші інструменти та мови програмування. Тому важливо, обрати саме той хостинг, який якнайкраще підійде для даного проєкту. Перед початком вибору, потрібно детально вивчити наявні програмні інструменти на «ринку» пропозицій самих сервісів. Серед провідні компанії, що надають послуги в даній сфері варто відзначити Heroku, GitHub Pages – розгалуження GitHub та Vercel. Усі вони чудово підходять для виконання поставлених задач, проте в кожному з перелічених інструментів є свої переваги та недоліки.

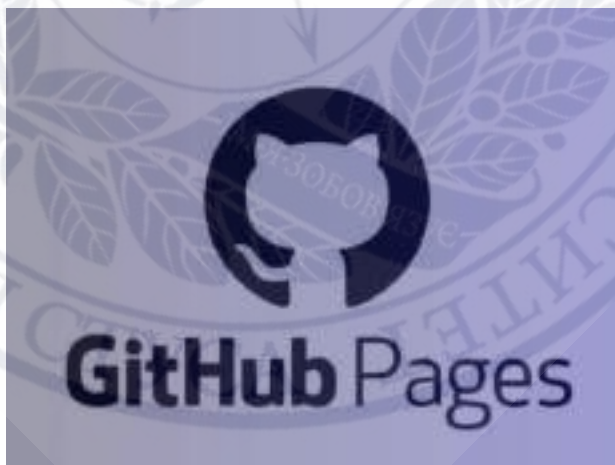


Рисунок 2.4 – Логотип GitHub Pages.

Для даного проєкту було обрано саме GitHub Pages, оскільки він дуже простий у використанні та не потребує виконання додаткових кроків. Власне, для того, щоб розмістити розроблений сайт на хостингу застосунків, потрібно просто

відправити свій проєкт на github. Після того, провести декілька хвилин за налаштуваннями і все буде готове до роботи [38-39].



РОЗДІЛ 3

РЕАЛІЗАЦІЯ САЙТУ ДЛЯ ПІДБОРУ ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

3.1. Реалізація сайту для підбору інструментів реалізації користувацького інтерфейсу.

Враховавши усі аспекти, які перераховані у попередніх розділах, як вже було сказано, для розробки макету було обрано Figma. Верстка сторінки розпочалася із десктопної версії сайту, а надалі для адаптивності дещо змінювався вигляд сторінки залежно від ширини екрану пристроїв на яких переглядатиметься сайт. В підсумку, отримано ось такий результат:

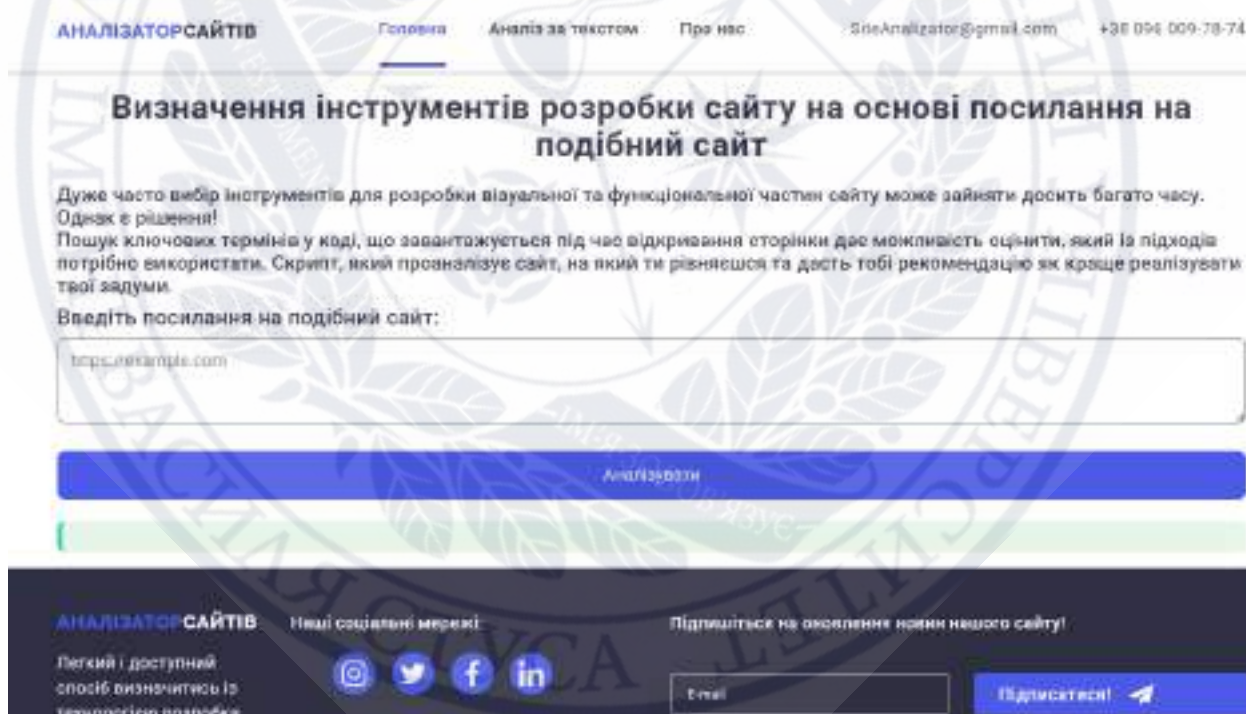


Рисунок 3.1. Візуальний вигляд головної сторінки

Для оформлення текстових елементів обрано шрифт Roboto. Для різних типів текстових елементів підібрано відповідну жирність. Також обов'язково для хорошої читабельності, важливо підібрати контрастні кольори шрифтів, а саме білий та чорний, на мою думку вони непогано гармонують. В хедері присутні

посилання на сторінки, для зручного переходу, а також підтримки сервісу, власне найнеобхідніші речі.

АНАЛІЗАТОР САЙТІВ Головна Аналіз за текстом Про нас SiteAnalizator@gmail.com +38 096 009-76-74

Аналіз за описом характеристик майбутнього сайту

Дуже часто вибір інструментів для розробки візуальної та функціональної частин сайту може зайняти досить багато часу. Однак є рішення!

Просто напиши, що ти хочеш бачити на своїй сторінці та дай їй певні характеристики, програма проаналізує отриманий текст і видасть певний результат. Ключові слова будуть головним орієнтиром для скрипту, але рекомендація відбувається не за одним словом, а за більшою кількістю із однієї із категорій. Аналіз опису сторінки, яку ти уявляєш в майбутньому, у висновку дасть одну технологію, якою буде легше та найкраще реалізувати твої задуми.

Введіть опис сайту або функціоналу...

Аналізувати текст

АНАЛІЗАТОР САЙТІВ Наші соціальні мережі: Підпишіться на оновлення новин нашого сайту!

Легкий і доступний спосіб вквітнитися із технологією розробки вашого сайту!

Instagram Twitter Facebook LinkedIn

Email **Підписатися!**

Рисунок 3.2. Візуальний вигляд сторінки аналізу за текстовим описом

Як видно на Рисунку 3.2, реалізована центральна частина сторінки, тут є текстове пояснення того, що робить скрипт. Форма в яку записується опис потрібного користувачу сайту. Далі йде кнопка та невелике поле куди виводиться рекомендація щодо використання того чи іншого інструменту або технології.

Також внизу розміщений футтер, де невеликий текстовий слоган, а також набір соціальний мереж та підписка на оновлення даного сайту.

Про нашу сторінку

Вітаємо на нашому сайті! Цей ресурс був створений з метою допомогти користувачам швидко визначити, яка технологія найкраще підходить для реалізації вебсайту — класичний підхід (HTML/CSS/JavaScript) чи сучасні фреймворки (React/Angular).

Розробник цього сайту — веб-ентузіаст, який поєднує простоту, зручність і аналітичний підхід до розробки інтерфейсів. Основний акцент зроблено на зручності користування, швидкому аналізі та розширюваності функціоналу.

Сайт реалізовано з використанням [Node.js](#) для серверної частини, [Cheerio](#) для аналізу DOM-структури сторінок, та чистого HTML/CSS/JS для клієнтського інтерфейсу.

Дякуємо, що користуєтеся нашим інструментом! Ми відкриті до зворотного зв'язку та вдосконалення проекту.

Про сайт

Цей сайт створений для того, щоб допомогти користувачам обрати найкращу технологію для розробки вебпроєкту. Залежно від складності сайту або макету, система рекомендує або використання фреймворків (React/Angular), або базовий стек HTML/CSS/JavaScript.

Про розробника



Ім'я: Брижак Владислав

Спеціалізація: Frontend-розробник

Ціль: Розробка корисних інструментів для автоматизації аналізу інтерфейсів

Рисунок 3.3. Візуальний вигляд сторінки «Про нас»

Сторінка «Про нас» як і було заплановано, включає в себе текстову інформацію про сторінку, важливі частини виділено, для більшого звернення уваги користувача. Також частина про розробника, тут є фото та трохи інформації про спеціалізацію та ціль девелопера. Внизу сторінки, в футтері розміщено невеликий список партнерів, із працюючими посиланнями на них. За бажання користувач зможе із легкістю перейти за ними та ознайомитися з інформацією на їхніх сайтах. Для зручності форму та додано кнопку, які розміщена праворуч, вони слугують для того, щоб залишити свою електронну адресу, щоб отримувати повідомлення про оновлення даного сайту.



Рисунок 3.4. Візуальний вигляд мобільного меню

Оскільки сайт розроблений також і для мобільних пристроїв, на екранах малих розмірів дуже складно користуватися навігацією. Для того, щоб вирішити цю проблему було розроблено мобільне меню [40]. Натиснувши на нього, у користувача з'являється модальне вікно, що показане на Рисунку 3.4. Тут зібрана уся потрібна інформація, що може знадобитися. Зверху спеціально розроблена кнопка, яка закриває дане меню.

Внизу модального вікна також присутні контактні дані розробника та посилання на сторінки партнерів.

Переглянувши сторінки та їх користувацький інтерфейс, можна переходити до перевірки роботи скриптів. Першим кроком потрібно обрати будь-який сайт,

який будемо подавати, як приклад того, що ми хочемо реалізувати в майбутньому, нехай для прикладу це буде сторінка стрімінгового сервісу Мегого. Отже беремо посилання на їхній сайт, та вставляємо в форму на головній сторінці.

Натиснувши кнопку, отримуємо пораду, яка показана на Рисунку 4.5.

Визначення інструментів розробки сайту на основі посилання на подібний сайт

Дуже часто вибір інструментів для розробки візуальної та функціональної частини сайту може зайняти досить багато часу. Однак є рішення!

Пошук ключових термінів у коді, що завантажується під час відкривання сторінки дає можливість оцінити, який із підходів потрібно використати. Скрипт, який проаналізує сайт, на який ти рванешся та дасть тобі рекомендацію як краще реалізувати твої задуми.

Введіть посилання на подібний сайт:

Аналізувати

Рекомендується використати React або Angular — складна DOM-структура, багато інтерактивних елементів.

Рисунок 3.5. Результат роботи коду, на прикладі сайту «Мегого»

Код відпрацював коректно та вказав на те, що кращим вибором буде скористатися новітніми фреймворками. Перейдімо до тестування іншого скрипту, який аналізує на основі тексту.

Аналіз за описом характеристик майбутнього сайту

Дуже часто вибір інструментів для розробки візуальної та функціональної частини сайту може зайняти досить багато часу. Однак є рішення!

Просто напиши, що ти хочеш бачити на своїй сторінці та дай їй певні характеристики, програма проаналізує отриманий текст і видасть певний результат. Ключові слова будуть головним орієнтиром для скрипту, але рекомендація відбувається не за одним словом, а за більшою кількістю із однієї із категорій. Аналіз опису сторінки, яку ти уявляєш в майбутньому, у висновку дасть одну технологію, якою буде легше та найкраще реалізувати твої задуми.

Сайт повинен вирізнятися простим і зрозумілим інтерфейсом, має бути хороша навігація, інтуїтивно зрозуміла навіть для користувачів. Його має розроблено з використанням усіх стандартів, також важливе швидке завантаження та коректне відображення на різних пристроях.

Аналізувати текст

Рекомендується HTML/CSS/JavaScript — простий або статичний сайт.

Рисунок 3.6. Результат роботи скрипту, на прикладі даного тексту

Код відпрацював правильно, було знайдено декілька ключових слів, що вказують на те, що набір HTML/CSS/JS буде оптимальним варіантом при

розробці подібного сайту. Для обох скриптів прописані виняткові результати, тобто коли переданий пустий рядок або не коректно введені дані, наприклад неправильне посилання.

3.2. Структура сайту та особливості функціоналу

Правильна структура сайту є важливою складовою для розробки успішного продукту. Вона визначає не тільки простоту в навігації та оцінці зовнішнього вигляду сторінки користувачем, а й відповідає за ефективність індексації даного сайту пошуковими системами. Логічно спроектована структура дозволить швидко знаходити необхідну інформацію, посприє покращенню взаємодії відвідувача із продуктом та збільшить загальний рівень юзабіліті. Отже цей етап є надзвичайно важливим, адже саме він буде вирішальним в досягненні поставлених задач, як для сторінки так і для усього проекту.

Структура сайту дуже тісно пов'язана із шляхом користувача по сторінці, можна сказати, вони впливають один із одного, тож для розробленого сайту, на Рисунку 3.7 показано актуальний шлях відвідувача сторінки.

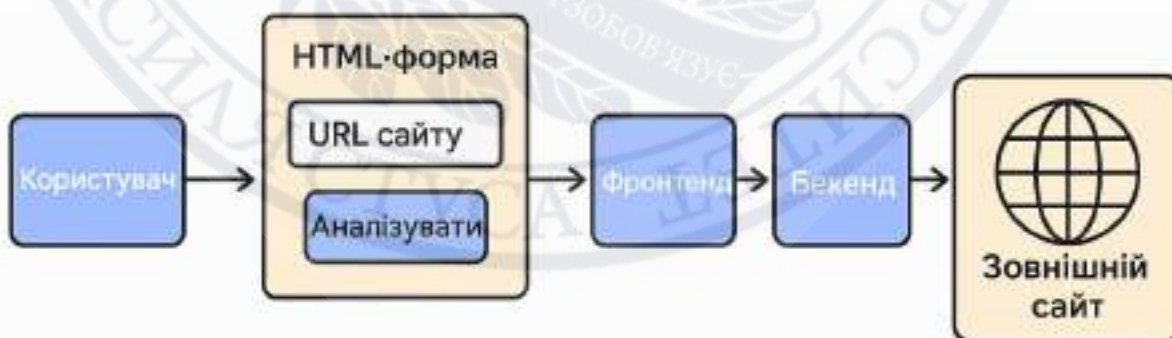


Рисунок 3.7. Шлях користувача по головній сторінці «АналізаторСайтів»

Зважаючи на переваги та недоліки всіх сторінок з ідентичною тематикою, які досліджувались в першому розділі, потрібно спроектувати і продумати логіку для сайту, який власне і буде реалізовуватись.

Почати хотілось би з простого, з вибору назви. Хотілось би, щоб вона асоціювалась з тематикою інтернет сторінки, тож було прийнято рішення обрати саме таку назву – «АналізаторСайтів». Оскільки розробка сайту була з розрахунком на українську аудиторію, тому і назва розроблена нашої державною мовою. Дана назва поєднує два слова, які є ключовими для запланованого функціоналу сайту, тому покращить сприйняття користувачами нашої сторінки і зробить легшим запам'ятовування.

Стосовно кількості сторінок на сайті, та їх функціоналу. Під час виконання роботи, планується розробити три головні розділи:

- Головна сторінка.
- Сторінка з аналізом за текстом.
- Сторінка «Про нас».

На кожній сторінці буде реалізовано простий і зрозумілий хеддер, де знаходитиметься блок навігації та контактна інформація. Також футтер, де розміщений невеликий текстовий слоган, також набір соціальних мереж та підписка на оновлення даного сайту.

На головній сторінці буде міститись коротка інформація про те що можна зробити на сторінці та безпосередньо поле із можливістю введення потрібно для користувача посилання. З головної сторінки буде можливість на будь-яку іншу сторінку сайту. Головним функціоналом сторінки буде вікно, в яке вводиться посилання на сайт схожий за візуальним виглядом та функціоналом на потрібний користувачу. Кнопка, яка активує скрипт, що проаналізує отримане посилання та невеличке поле, куди виводиться результат, тобто порада щодо використання того чи іншого інструменту. Із даним скриптом можна ознайомитися в Додатку А.

Сторінка з аналізом за текстом буде майже ідентична головній сторінці, проте є важлива відмінність, а саме замість посилання користувач має вписати текстовий опис того сайту, який він хоче розробити. Після натискання на кнопку,

в дію вступить інший код, він пройде отриманий текст та віднайде потрібно для висновку ключові слова. В результаті користувач так само отримає певну рекомендацію.

Сторінка «Про нас» несе тільки ознайомчу функцію. Тут буде розміщена текстова інформація про, чим користувач може скористатися та що зможе отримати на даній сторінці. Також обов'язково інформація про розробника.

Обов'язковою вимогою до розробки сайту є його мобільна версія. Тому було прийнято рішення про верстку сторінки таким чином, щоб вона повністю функціонувала та мала привабливий зовнішній вигляд як на комп'ютерах так і на смартфонах.

Під час розробки сайту, було максимальне намагання дотримуватись таких критеріїв створення:

- Адаптивність сайту;
- Кроссбраузерність сайту;
- Легка навігація;
- Високі показники швидкодії;
- Простий і зрозумілий код.

Адаптивність та кроссбраузерність, надають можливість користувачам легко користуватися сайтом з будь-яких пристроїв, та з будь-якого браузера. Легкість навігації вже давно є одним із головних критеріїв хороших вебсторінок.

Показники швидкодії розробленого сайту буде представлено дещо пізніше. Чому код повинен бути простим і зрозумілим і водночас правильно працювати? Усі ці критерії полегшують життя розробнику, який буде в подальшому підтримувати, оновлювати чи просто переглядати код, що був розроблений. Для того, щоб все було зрозуміло і просто потрібно вибудувати хорошу структуру проєкту саме в інтегрованому середовищі програмування. Завантаживши даний проєкт розробник одразу побачить, що файли розміщені по папкам, а основні файли винесені окремо. Такий підхід полегшує пошук потрібних файлів та прибирає можливість взаємодії із непотрібними розробнику частинами проєкту.

Файлова структура розробленого сайту має наступний вигляд. Зараз дещо детальніше поясню про те, за що відповідають деякі файли і каталоги.

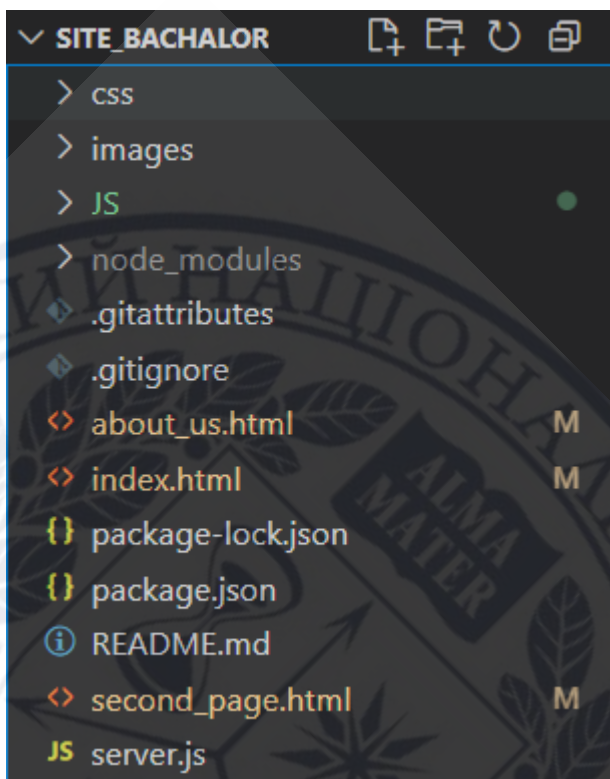


Рисунок 3.8. Файлова структура проєкту

Опис файлів та каталогів:

- node_modules – папка, в якій зберігаються залежності, які використовує node.js для проєкту;
- css – папка в якій зберігаються усі необхідні стилі для візуального оформлення сайту;
- js – папка з JS файлами для реалізації сайту;
- images – папка с завантаженими фото;
- .gitignore – файл в якому описуються каталоги та файли які не будуть завантажуватись до репозиторію;
- index.html – файл, який першим завантажується та відкривається браузером;
- second_page.html – файл, який відповідає за рендер сторінки «Аналіз за текстом»;

- about_us.html – файл, який відповідає за рендер сторінки «Про нас»;
- server.js – серверна частина програми;
- package.json – опис проекту, версія, список залежностей для роботи;
- package-lock.json – зберігаються версії підключених пакетів;

3.3. Проведення тестування реалізованого сайту.

Завершивши роботу над сайтом, перевірено характеристики швидкодії за допомогою сервісу “PageSpeed Insight”.



Рисунок 3.9. Аналіз комп'ютерної версії сайту “АналізаторСайтів” за допомогою сервісу PageSpeed Insight

Отримані після аналізу результати показують, що була виконана непогана робота, характеристики високо рівня, тільки категорію «Доступність» можна

покращити. Сервіс пояснює це тим, що не занадто велика контрастність кольорів, хоча, ніби, все виглядає нормально, також декілька проблем із формами, хоча все працює нормально. Загалом отриманий середній бал – 93 бали. Хороший результат, проте можливі подальші доопрацювання зможуть дати суттєве покращення. Перейдімо до аналізу мобільної версії сторінки.

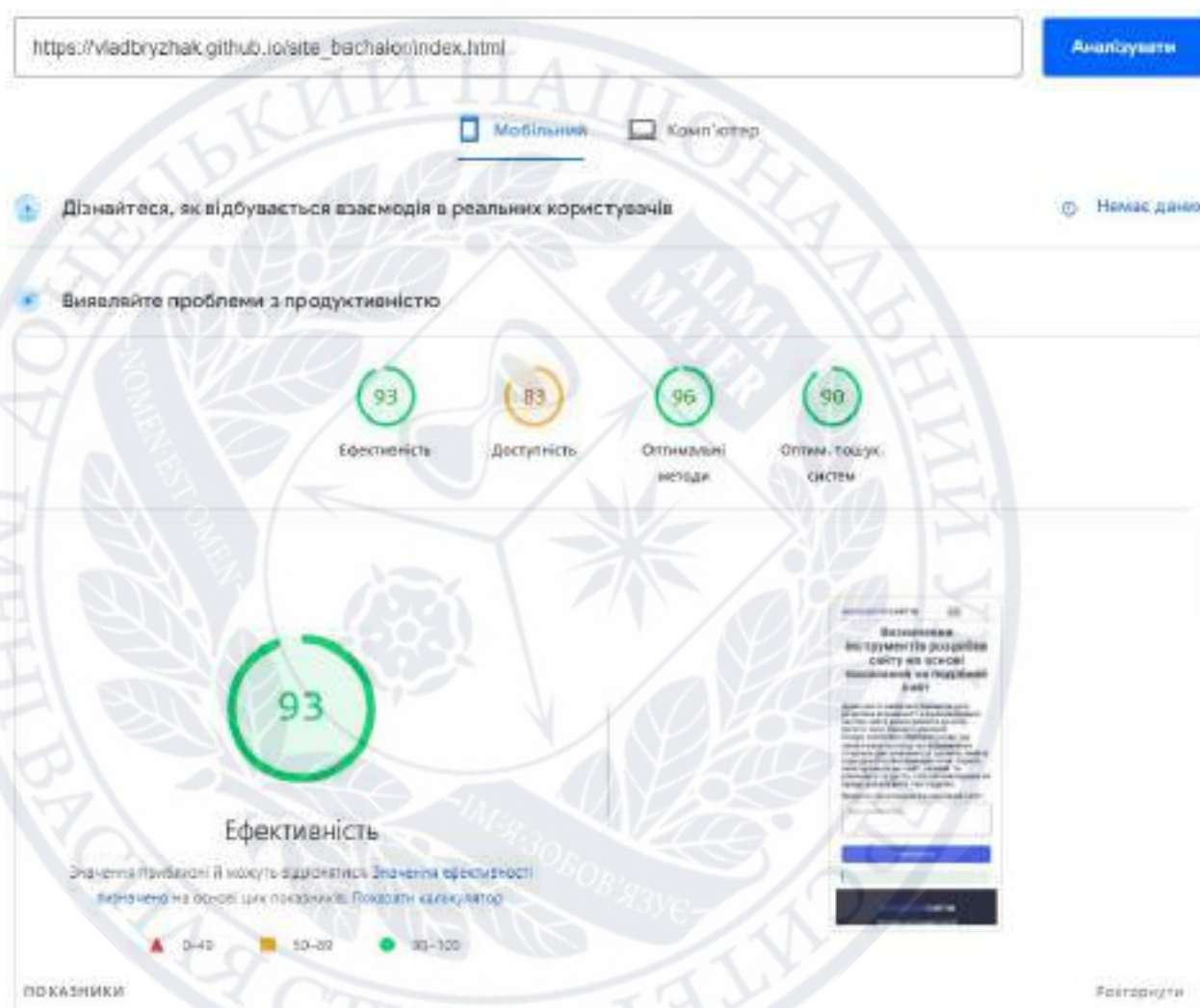


Рисунок 3.10. Аналіз сайту “АналізаторСайтів”, для мобільної за допомогою сервісу PageSpeed

В цілому, характеристики високо рівня, хоча і дещо гірші в порівнянні із комп’ютерною версією. Загалом отриманий середній бал – 90,5 балів. Для мобільної версії дуже хороший результат, оскільки було повна зміна стилів, в усіх категорія маємо непогані показники.

ВИСНОВКИ

Кожен другий у XXI столітті користується тими чи іншими веб-сайтами. Зовнішній вигляд, швидкодія та кросплатформеність є ключовими факторами під час розробки сайтів. На сьогоднішній день власний вебсайт став загальнодоступним практично для кожного, хто має потребу і наполегливість.

Метою даної роботи було – розробка сайту, який дозволив би кожному користувачу із легкістю підібрати оптимальні технології та інструменти для реалізації користувацького інтерфейсу на основі заданих параметрів.

Досліджено історію створення веб-сайтів. З дня створення першої інтернет сторінки пройшло дуже багато часу, змін і нових технологій. В минулому примітивні сайти, які не мали ніякого функціоналу як такого, переросли у просто неймовірні сайти.

Аналіз сторінок аналогів показав певні тенденції та довів важливість та актуальність даного проєкту. За допомогою сервісу “PageSpeed Insight”, зроблено висновки, які були враховані, під час розробки сайту «АналізаторСайтів».

Далі було приступлено до аналізу найбільш популярних і сучасних програмних інструментів. Як інструмент для вебдизайну, для даного проєкту обрано Figma, наступним кроком - найбільш зручне IDE, а саме Visual Studio Code. Було проаналізовано плюси і мінуси найбільш використовуваних у веброзробці фреймворків. В ході роботи над даним проєктом було розглянуто популярні та актуальні технології розробки веб-сайтів, а саме Multiple-page Application та Single-page Application. В результаті обрано саме такі, що найбільш зручним та доступним способом реалізують початковий задум. Врахувавши усю знайдену і опрацьовану інформацію, було розроблено сайт «АналізаторСайтів».

Завершивши роботу над вебсторінкою, перевірено швидкодію за допомогою сервісу “PageSpeed Insight”, та проаналізовано отримані результати. Застосунок має зрозумілий інтерфейс, яким легко можуть користуватись навіть не обізнані у роботі вебсайтів користувачі.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. What is a website?. Computer Hope. URL:<https://www.computerhope.com/jargon/w/website.htm> (Дата звернення: 12.03.2025).
2. Що таке вебсайт та веб-застосунок: різниця понять URL: <https://blog.ithillel.ua/articles/website-and-web-application> (Дата звернення: 11.03.2025).
3. User experience: Process and Methodology URL: <https://uiuxtrend.com/user-experience-ux-process/> (Дата звернення: 20.04.2025).
4. Що таке IT? URL: [Що таке IT? « Іннети](#) (Дата звернення: 16.03.2025).
5. Структура сайту: основні види, вимоги та приклади URL: <https://wedex.com.ua/blog/struktura-sajtu-osnovni-vydy-vymogy-ta-pryklady/> (Дата звернення: 16.03.2025).
6. 28 років тому запустили перший у світі веб-сайт. Ось як виглядав би theБабель у 1990 році. Перший у світі веб-сайт запустили 28 років тому. URL:<https://babel.ua/texts/23307-28-rokiv-tomu-zapustili-pershiy-u-sviti-veb-sayt-mi-pokazali-yak-viglyadav-bi-thebabel-v-1990-roci> (Дата звернення: 18.03.2025).
7. Internet Live Stats: Total Number of websites. URL: <https://www.statheap.app/reports/internet-live-stats-total-number-of-websites> (Дата звернення: 18.03.2025).
8. Roles and Responsibilities of a Product Manager URL: [What is the Role & Responsibilities of Product Manager](#) (Дата звернення: 18.03.2025).
9. Ryan L. Modern Front-end Architecture / Lanciaux Ryan., 2021, с75.
10. What Does Online Service Mean? URL: [What is an Online Service? - Definition from Techopedia](#) (Дата звернення: 22.03.2025).
11. Що таке HTML та як за допомогою нього увійти в IT. URL: <https://mc.today/uk/shho-take-html-ta-yak-za-dopomogoyu-nogo-uvijti-do-it/> (Дата звернення: 28.03.2025).

12. Що таке розробка сайтів та чому це важливо? URL: <https://webbookstudio.com/ua/articles/what-is-website-development-and-why-it-is-important/>? (Дата звернення: 28.03.2025).
13. Які існують технології для розробки сайтів та кому вони підходять? URL: <https://icstudio.online/post/tehnologii-dlya-rozrobky-sajtov> (Дата звернення: 18.04.2025).
14. Зелінська О.В., Морозюк А.А. Основні принципи веб-аналітики та її вплив на розробку веб-сайтів, 2023 URL: [Основні принципи веб-аналітики та її вплив на розробку веб-сайтів | Прикладні інформаційні технології](#) С. 30-32.
15. Основні тренди веб-дизайну в 2022 році. URL: <https://webdevandseo.com.ua/the-main-trends-of-web-design-in-2021/> (Дата звернення: 07.04.2025).
16. Як побудувати структуру сайту: від А до Я з прикладами URL: <https://sendpulse.ua/blog/website-structure-with-examples> (Дата звернення: 07.04.2025).
17. Інструменти які ми використовуємо для розробки сайтів. URL: <https://www.websvit.com/blog/2015/10/29/instrumenty-yaki-my-vykorystovujemo-dlya-rozrobky-sajtiv/> (Дата звернення: 03.05.2025).
18. Інструменти веб-дизайну. URL: <https://lemon.school/blog/instrumenty-veb-dyzajnu> (Дата звернення: 03.05.2025).
19. Що таке Figma і навіщо вона потрібна? URL: <https://dan-it.com.ua/uk/blog/chto-takoe-figma-i-zachem-ona-nuzhna/> (Дата звернення: 03.05.2025).
20. Зелінська О.В., Поліщук Д.О. Сучасні засоби веб розробки – Прикладні інформаційні технології, 2023 URL: [Сучасні засоби веб розробки | Прикладні інформаційні технології](#) С. 55-56.
21. 10 Essential Web Development Tools for Streamlined Projects in 2025 URL: <https://www.netguru.com/blog/web-development-tools/>? (Дата звернення: 06.05.2025).

22. 10 кращих редакторів коду для програмістів. URL: <https://mate.academy/blog/front-end-and-js/top-10-text-editors/> (Дата звернення: 10.05.2025).
23. Знайомство з Visual Studio Code URL: <https://romul.name/blog/znayomstvo-z-visual-studio-code/> (Дата звернення: 06.05.2025).
24. Adobe Dreamweaver: Всі мінуси. URL: <https://designtalk.club/adobe-dreamweaver-vsi-minusy-chastyna-1/> (Дата звернення: 06.05.2025).
25. Top Web Development Tools in 2025. URL: <https://www.browserstack.com/guide/web-development-tools> (Дата звернення: 10.05.2025).
26. Що таке Single Page Application? URL: <https://asabix.com.ua/what-is-a-single-page-application/> (Дата звернення: 10.05.2025).
27. Односторінкові (spa) і багатосторінкові (pwa) веб-додатки URL: <https://embo-studio.ua/uk/blog/odnostorinkovi-spa-i-bagatostorinkovi-pwa/> (Дата звернення: 10.05.2025).
28. Multiple-page Application та Single-page Application. URL: <https://www.edu.goit.global/uk/learn/11043865/31183/31218/training?blockId=20317186> (Дата звернення: 10.05.2025).
29. Найпопулярніші мови програмування у 2024 році. URL: https://dan-it.com.ua/uk/blog/samyje-populjarnye-jazyki-programmirovaniya-v-2024-godu/#2_JavaScript (Дата звернення: 10.05.2025).
30. Що таке JavaScript URL: <https://javascript.tutorial.in.ua/what-is-javascript/> (Дата звернення: 10.05.2025).
31. Angular vs React vs Vue: Core Differences URL: <https://www.browserstack.com/guide/angular-vs-react-vs-vue> (Дата звернення: 10.05.2025).
32. Zelinska O., Sichko T. Design and implementation of the portfolio site "portfolion" Polish journal of science. 2023. № 58. URL: https://www.slideshare.net/Polish_journal_of_science/polish-journal-of-science-58-2023 С. 66-73.

33. Порівнюємо React, Angular і Vue – найпопулярніші бібліотеки й фреймворки у 2022 році URL: <https://dou.ua/forums/topic/39933/> (Дата звернення: 10.05.2025) React,Vue,Angular. Що краще? URL: https://markup-ua.com/react-vue-angular-shho-krashhe/#google_vignette (Дата звернення: 10.05.2025).
34. Vue, Angular чи React: який фреймворк обрати початківцю? URL: <https://mate.academy/blog/front-end-and-js/vue-angular-or-react-what-to-choose-for-a-novice/> (Дата звернення: 10.05.2025)
35. Огляди фреймворків JavaScript. Що, для чого і коли використовувати? URL: <https://dou.ua/forums/topic/34739/> (Дата звернення: 10.05.2025).
36. Адаптивні вебсайти. Надзвичайно швидко. URL: <https://www.adobe.com/ua/products/dreamweaver.html> (Дата звернення: 29.04.2025).
37. Технології для розробки сайту: що це, які найпопулярніші та як вибрати? URL: <https://iamppm.club.ua/blog/tehnologiyi-dlya-rozrobki-sajtu-shho-cze-yaki-najpopulyarnishi-ta-yak-vibrati> (Дата звернення: 18.04.2025).
38. What is GitHub Pages? URL: <https://docs.github.com/ru/pages/getting-started-with-github-pages/what-is-github-pages> (Дата звернення: 10.05.2025).
39. Веб аналітика URL: [Веб аналітика - налаштування, аналіз показників | IT-Kitchen](#) (Дата звернення: 11.04.2025).
40. Інструменти для аналітики сайту URL: <https://hostiq.ua/blog/ukr/35-tools-for-web-analytics/> (Дата звернення: 11.04.2025).

ДОДАТКИ



Додаток А

Скрипт, який відповідає за обробку отриманої інформації від користувача та вивід результату на сторінку

```
const express = require('express');
const cors = require('cors');
const axios = require('axios');
const cheerio = require('cheerio');

const app = express();
app.use(cors());
app.use(express.json());
// Аналіз сайту за URL
app.post('/analyze', async (req, res) => {
  const { url } = req.body;
  if (!url || !url.startsWith('http')) {
    return res.status(400).json({ result: 'Будь ласка, введіть коректне посилання (http/https).' });
  }
  try {
    const response = await axios.get(url);
    const $ = cheerio.load(response.data);
    const scriptCount = $('script').length;
    const inputCount = $('input, textarea, select').length;
    const buttonCount = $('button').length;
    const divCount = $('div').length;
    const componentsWithClasses = $('[class]').length;
    const reactHints = $('[data-reactroot], [id^="__next"]').length;
```

```

const isSimple = scriptCount < 5 && inputCount < 3 && buttonCount < 3 &&
divCount < 50 && componentsWithClasses < 40;

const isReactLike = reactHints > 0 || scriptCount > 10 || componentsWithClasses >
100 || inputCount + buttonCount > 10;

if (isReactLike) {

  return res.json({ result: 'Рекомендується використати React або Angular —
складна DOM-структура, багато інтерактивних елементів.' });

  } else if (isSimple) {

    return res.json({ result: 'Рекомендується використати HTML/CSS/JavaScript —
сайт має просту структуру.' });

    } else {

      return res.json({ result: 'Рекомендується використати React або HTML/CSS/JS
залежно від ваших потреб — сайт має середню складність.' });

    }

  } catch (error) {

    console.error('Помилка:', error.message);

    return res.status(500).json({ result: 'Не вдалося проаналізувати сайт. Перевірте
URL або спробуйте пізніше.' });

  }

});

// Аналіз тексту користувача
app.post('/analyze-text', (req, res) => {

  const { text } = req.body;

  if (!text || text.trim().length === 0) {

    return res.status(400).json({ result: 'Будь ласка, введіть текст для аналізу.' });

  }

  const lowerText = text.toLowerCase();

  const frontendKeywords = ['сайт', 'інтерфейс', 'макет', 'html', 'css', 'простий'];

  const reactKeywords = ['динамічний', 'spa', 'react', 'компонент', 'стан', 'redux',
'анімація', 'routing'];

  const angularKeywords = ['angular', 'модуль', 'типізація', 'rxjs', 'typescript'];

```

```

const countMatches = (keywords) =>
  keywords.reduce((count, word) => count + (lowerText.includes(word) ? 1 : 0), 0);
const reactScore = countMatches(reactKeywords);
const angularScore = countMatches(angularKeywords);
const frontendScore = countMatches(frontendKeywords);

let result;

if (reactScore > angularScore && reactScore > frontendScore) {
  result = 'Рекомендується React — сучасний підхід для складного функціоналу.';
} else if (angularScore > reactScore && angularScore > frontendScore) {
  result = 'Рекомендується Angular — підходить для масштабних проєктів.';
} else if (frontendScore >= reactScore && frontendScore >= angularScore) {
  result = 'Рекомендується HTML/CSS/JavaScript — простий або статичний сайт.';
} else {
  result = 'Немає достатньо даних. Уточніть опис.';
}

res.json({ result });
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Сервер запущено: http://localhost:${PORT}`);
});

```

