

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Пармас В.К.

" ____ " _____ 2022 р.

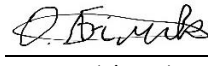
КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

«Розробка сайту-візитки для інформаційної підтримки підприємства»

Виконав: студент групи 2147ст



(підпис)

Білик О.А.

Керівник роботи:

(посада, науковий ступінь вчене звання)

(підпис)

Пармас В.К.

м. Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"
Гарант освітньої програми

к.т.н, доц. Партас В.К.
" ____ " _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"

Студенту Білику Олександр Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка сайту-візитки для інформаційного підприємства

Керівник роботи
Партас Віктор Кирилович

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем,

4. Перелік питань, що належать до розробки (найменування розділів): Вступ, аналіз предметної області та розробка концепції сайту-візитки підприємства, розробка проектних рішень, реалізація проектних рішень, охорона праці, висновки, список використаних джерел, Додаток А, Додаток Б

5. Перелік презентаційних матеріалів: Початок презентації, Мета, Конкретна проблема підприємства, Актуальність сайту-візитки, Діаграма роботи з сайтом-візиткою, Взаємодія з користувачем, Переваги використання технічного рішення з використанням технологій, Демонстрація створених моделей даних, Список створених полів для моделі даних, Створення даних згідно моделі даних, Висновок, Кінець презентації

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Партас В.К.	14.02. 2022	21.03.2022
2	Партас В.К.	21.03.2022	21.04.2022
3	Партас В.К.	21.04.2022	01.06.2022
4	Гурець Н.В.	01.06.2022	07.06.2022

7. Дата видачі завдання 14 лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області	21 березня	
2	Розробка проектних рішень	21 квітня	
3	Реалізація проектних рішень	1 червня	
4	Вирішення питань охорони праці	7 червня	
5	Виконання графічних документів	13 червня	
6	Оформлення пояснювальної записки	19 червня	
7	Подання дипломної роботи (ДР) на кафедральний захист	20 червня	
8	Подання ДР рецензенту	20 червня	
9	Подання ДР на державний захист	27 червня	
10			

Студент


(підпис)

Білик О.А.
(ПІБ)

Керівник роботи

(підпис)

Партас В.К.
(ПІБ)

АНОТАЦІЯ

Дана дипломна робота присвячена дослідженню функціонуванню незалежного автосервісного підприємства, проведенню аналізу конкурентів, для створення конткурентноспроможного підприємства та збільшення клієнтів.

Метою роботи є підвищення ефективності підприємства за допомогою структуризації інформації, послуг та контактних даних, для відображення їх на Інтернет-ресурсі, для скорочення витрат на неефективні способи представлення діяльності підприємства в інформаційному просторі, тим самим підвищення рівня залучення та опрацювання клієнтів, підвищення продуктивності та рентабельності підприємства, як кінцевий результат створення сайту-візитки.

В процесі виконання дипломного завдання, виконано дослідження способів розміщення подібних конкуруючих підприємств на Інтернет-ресурсі, відображення необхідної інформації для кінцевого користувача, підбір ефективних та сучасних технологій, для реалізації поставлених завдань, розглянута структура управління підприємства, для залучення нових працівників з необхідними навичками.

В якості використовуваних методів рішень були застосовані: онлайн-сервіс для розробки інтерфейсів і прототипування, «хмарні»- та WEB-технології з відкритим вихідним кодом та безголова система управління контентом.

Розроблена ІС підприємства з залученням клієнтів на основі WEB-технологій. ІС здатна надавати актуальну інформацію на Інтернет-ресурсі та залучення нових клієнтів.

Дана дипломна робота виконана на 84 сторінках друкованого тексту, містить 18 рисунків, на яких зображені різноманітні схеми та має 2 додатки. Список літератури в кількості 15 найменувань.

Робота виконана українською мовою.

ANNOTATION

This diploma project investigates the functioning of the independent car service company, analysis of competitors, to create a competitive company and increase customers.

The aim of the work is to increase the efficiency of the enterprise by structuring information, services and contact data to display them on the Internet, to reduce the cost of poor ways to promote the enterprise, thereby increasing customer engagement and processing, increasing productivity and profitability of the enterprise. creating a business card site.

In the process of completing the thesis, a study of ways to place such competing companies on the Internet, display the necessary information for the end user, selection of effective and modern technologies for the implementation of tasks, considered the management structure of the company to attract new employees with the necessary skills.

The following methods were used: online service for interface development and prototyping, "cloud" - and open source WEB-technologies and headless content management system.

Developed IS of enterprise with the involvement of customers based on WEB-technologies. IS is able to provide up-to-date information on the Internet resource and attract new orders.

The work is done in the state Ukrainian language, consists of 84 pages contains 2 appendices, 15 sources of information used.

Зміст

1. ЗАГАЛЬНА ЧАСТИНА	7
1.1 Аналіз предметної області та розробка концепції сайту-візитки підприємства.....	9
1.2 Загальна інформації різновидності підприємства	10
1.3 Проблеми представлення діяльності підприємства в інформаційному просторі	11
1.4 Актуальність сайту-візитки	12
1.5 Аналіз існуючих рішень	14
1.6 Переваги створення сайту-візитки та використання сучасних рішень	16
1.7 Концепція рішення	19
1.8 Технічне завдання	20
2. Розробка проекту програми та ПЗ	21
2.1 Загально-системні рішення	21
2.2 Рішення в ПЗ.....	23
2.2.1 Вибір технологій для виконання задач	23
2.2.2 Порівняння технологій	26
2.2.3 Запровадження SEO-оптимізації сайту-візитки.....	28
2.2.4 Можливість редагувати контент.....	31
2.2.5 Структура проєкту	34
2.3 Інформаційне забезпечення	35
2.4 Організаційне забезпечення	35
3. Реалізація ІС підприємства	36
3.1 Реалізація модулів рішення	39
4 ОХОРОНА ПРАЦІ	40
4.1 Вступ	40
4.2 Основні законодавчі і нормативні документи, які регламентують Охорону праці України.....	42
4.3 Санітарно-гігієнічні вимоги до організації та обладнання робочих місць з ПК	43
4.4 Перелік шкідливих та небезпечних факторів, які супроводжують роботу програміста	43
4.5 Висновки	43
ВИСНОВКИ	44
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	45
Додаток А – Розробка технічного завдання	46
Додаток Б – Код програмних засобів	48

ВСТУП

Сучаний сервіс зазнає вагомих змін та удосконалень, для задоволення потреб населення в повсякденному житті. Цей процес стає дедалі більше клієнтоорієнтованим, щоб мати перевагу поміж конкурентів. Що водночас додає можливість привнести підприємствам більше автоматизовані рішення та оновлення бізнес-процесів.

Сервіс – це сукупність організацій до обов'язків яких входить обслуговування населення в різних сферах повсякденного життя. У зв'язку з розвитком технологічного процесу та якості життя людей, відбувається розвиток надання в сферах послугу.

Все більше різних сфер надання послуг частіше приходять висновку, що в умовах стрімкого глобального прогресу потрібно шукати та швидко змінювати підходи в своїй роботі з клієнтами, щоб бути конкурентноспроможним. Потрібно адаптуватися до актуальних вимог населення. До цього відноситься використання інформаційних технологій та залучання нових працівників з відповідними навичками та знаннями.

Збільшення кількості сервісів у ринкових умовах вимагає від них оперативних дій в різних аспектах підприємства, як в покращенні якості послуг, так і нові маркетингові рішення. В Україні досить потужно проходить діджиталізація бізнесів. До речі, цей процес може продовжуватися роками постійно покращуючи рішення.

Метою дипломного проекту є підвищення конкурентноспроможності автосервісу за допомогою програмного продукту, що дозволить залучати більше клієнтів та зростати в збільшенні масштабів сервісу за допомогою створення додаткового способу поширення інформації про сервіс. Також автоматизація цього процесу, для подальшого його підтримання та оновлення, що призведе до скорочення витрат і актуальності інформації.

Для досягнення даної мети необхідно вирішити ряд завдань стовно цілей підприємства:

- виконати аналіз предметної області звернувши увагу на проблеми представлення діяльності підприємств в інформаційному просторі;
 - розробити технічне завдання;
 - розробити проектні рішення;
 - розробити ІС підприємства;
- провести необхідні дослідження з охорони праці.

1. ЗАГАЛЬНА ЧАСТИНА

1.1 Аналіз предметної області та розробка концепції сайту-візитки підприємства

Основним завданням автосервісу є представлення послуг по плановому технічному обслуговуванню, поточному та капітальному ремонтах, виявленню та виправленню поломок, відновульного ремонту автомобілів та встановлення додаткового обладнання.

В свою чергу залучення нових клієнтів відбувається за допомогою розповсюдження листівок та рекомендації від постійних клієнтів.

Нагальним є питання впровадження інформаційних технологій в процеси залучення клієнтів через розповсюдження інформації про підприємство з переліком послуг, графіком роботи, адресою, та написанням корисних статей для просування по пошуковій видачі, для тих, хто спробував вирішити проблему власноруч, щоб потім користувач став клієнтом автосервісу через знаходження інформації підприємства на персоналізованому сайті.

З цією метою необхідно створити сучасний сайт-візитку з описом основної інформації про підприємство:

- назва підприємства;
- послуги сервісу;
- пропозиції нашим клієнтам;
- профільні корисні статті;
- робочий графік;
- контактні дані;
- інтерактивна карта для зручного візуального представлення адреси.

Що отримає підприємство завдяки створенню сайту-візитки? Насамперед, сайт-візитка – це стартовий проект, головною метою якого є ефективний та сучасний пошук нових клієнтів за допомогою мережі Інтернет.

Саме реалізування можливості автоматичного просування сайту в пошуковій видачі, базуючись на корисному контенті для користувачів, які власноруч будуть

пробувати вирішити проблему з транспортним засобом, а потім вже звертатися до нас, для більш швидкого та високого рівня обслуговування транспортного засобу, тому для цього разом з корисним матеріалом будуть додані заклики та посилання, для переходу на основну сторінку сайту.

Сайт повинен мати можливість динамічного редагування контенту з адмін-панелі для подальшого оновлення інформації без навичків програмування для оптимізації витрат .

1.2 Загальна інформації різновидності підприємства

Проаналізувавши інформацію про наявні різновидності поданого підприємства, можна дійти висновку, що автосервіси різняться між собою, проте є декілька чітко сформованих видів підприємств:

- Гаражні автосервіси. Здебільшого до цього відносяться невеликі компанії, які здебільшого мають одного механіка, який працює, як фізична особа-підприємиць. Переважно це чоловіки, які мають тісне коло знайомств, до кого звертаються через рекомендації. В такому випадку це підприємство може обслуговувати лише обмежену кількість клієнтів. Перелік послуг досить незначний.
- Незалежні автосервіси. Широкий спектр послуг, майстерність виконання роботи, безпека на підприємстві, може мати можливість запису на попередню консультацію. До цього необхідно мати лояльні відносини з клієнтами.
- Дилерські автосервіси. Цей вид автосервісу дуже схожий з незалежним, ле за однієї відмінності – надання офіційної гарантії на всі види робіт, обслуговування авто на найвищому рівні з використанням офіційних та якісних запчастин в разі поломки. В дилерський автосервіс слід звертатися суто, коли авто знаходиться на гарантії.

- Спеціалізовані автосервіси. Виступають як технічні центри вузького напрямлення, Також надають гарантію на виконану роботу та високу майстерність.

Саме тому для конкретного виду сервіса потрібно підбирати персональне рішення кожному підприємству. Від цього залежить наскільки доцільними будуть витрати різного напрямлення та яка ефективність такої зробленої комплексної роботи, як створення абсолютно нової вагової складової підприємства.

В данному випадку, ми маємо вид автосервісу, як незалежний. Для незалежного виду автосервісу можна створювати досить гнучкі та сучасні рішення з використанням сучасних технологій та підходів в створенні потрібного рішення поставленої задачі. Також є доцільним не тільки викорисовувати сучасні технології, а розробити саму концепцію, як підтримувати кінцевий результат в актуальності на довгий час вперед та зменшити витрати в подальшому оновленню, вдосконаленню і розширенню програмного рішення.

1.3 Проблеми представлення діяльності підприємства в інформаційному просторі

Розвиток інноваційних технологій та засобів комунікації створив потужний поштовх для формування нового напрямку в сучасному світі бізнесу – діджиталізації бізнес-процесів підприємства. Аналогова інформація перетворюється на цифрові дані. Більшість засобів масової інформації стають цифровими з переважною часткою електронної складової, яка задає основну тенденцію бізнес-процесів. Електронні форми спілкування підвищують ефективність спілкування між продавцями і покупцями та забезпечують сучасні перспективи для перебудови бізнес-процесів.

Активне впровадження різноманітних цифрових технологій в економічну діяльність є фактором підвищення її ефективності, включаючи бізнес-процеси, створені завдяки використанню електронних комунікаційних технологій, та нові підходи в організації обміну інформацією.

В даному підприємстві залучення нових клієнтів відбувалося за допомоги роздавання візиток та подальші рекомендації від попередніх клієнтів. Саме з цього виникав ряд недоліків, які обмежували розвиток підприємства:

- Рекламні листівки видавалися в дуже великій кількості на одну людину, яка доносила до найближчого смітника, щоб позбутися листівки.
- Поштова скриньки були завалені різною рекламою, тому доцільне призначення листівок було дуже незначним, тому що власники поштових скриньок звикли до такої реклами та дуже часто не переглядали інформацію на листівці.
- Видавалися листівки не цільовій аудиторії.
- Рекомендації попередніх клієнтів були досить вагомим вкладом в бренд автосервісу. Проте такий спосіб просування було неможливо прорахувати появу нових клієнтів.

Враховуючи описані вище недоліки стає зрозумілим необхідна потреба в новому та сучасному рішенні в залученні клієнтів та мати переваги перед конкурентами.

Саме для вирішення цієї нагальної проблеми керівники підприємства дійшли висновку в необхідності діджиталізації бізнесу, а саме в створенні сайту-візитки.

1.4 Актуальність сайту-візитки

Сайт-візитка – це головне, що необхідно бізнесу, адже власне інтернет-представництво – це запорука того, що про бізнес довідається досить значна кількість потенційних клієнтів.

Паперова візитна картка, за якою дізнаються про компанію, є обличчям фірми, але за обміркуваннями та недоліками «класичної» візитки ще більш якісним вектором успіху буде сайт-візитка.

Географічно підприємство розташоване в одному місті (регіоні або країні), тому вона не може охопити потенційну аудиторію за межами сфери свого впливу, але Інтернет стирає територіальні обмеження. З використання сайту-візитку з'являються значні переваги:

- Створюючи веб-сайт, без проблем можливо вийти на глобальний ринок, на цілий населений пункт в разі надання сервіс послуг, як в нашому випадку.
- 24 години на добу і 7 днів на тиждень представництво вашої компанії буде відкрите і досягне з будь-якої точки планети.
- Від організації не потрібно додаткових витрат на відкриття нових офісів за межами свого регіону.
- Можливості для залучення і утримання клієнтів виходять на новий рівень.

Таким чином, один веб-сайт забезпечує глобальне охоплення аудиторії і дозволяє залучати все нових та нових клієнтів.

Стосовно сайту-візитки вагома актуальність полягає в тому, що значно простіше виконувати SEO просування. SEO просування – це процес оптимізації сайту для більш ефективного його відображення в пошуковій видачі.

Для сайту-візитки, де знаходиться переважно матеріал стосовно підприємства, ключові тези та послуги підприємства, контактні дані та ще іншої ключової інформації SEO просування буде значно ефективнішим насамперед саме в стислій інформації та корисних матеріалів для відвідувачів цього сайту. Та SEO просування сайту-візитки займе значно менше часу, ніж якийсь звичасний багатосторінковий сайт з купою інформації, яка часто не дуже цікава відвідувачу цього інтернет-ресурсу.

Просування Інтернет ресурсу в пошукових системах – найважливіший маркетинговий етап реклами для організацій. Від того, наскільки грамотно виконана оптимізація, безпосередньо залежить ефективність всієї рекламної кампанії і успіх підприємства.

1.5 Аналіз існуючих рішень

В даний час просто створити власний сайт-візитку під силу навіть новачку - потрібно знайти безкоштовний сервіс з шаблонами і вибрати вподобаний дизайн. Тобто виконання самої послуги створення веб-сайту стає все більш доступнішою, що є досить очікуваним зважаючи на стрімкий розвиток технологій, для прискорення створення сайту, зменшення витрат на програмістів тощо. Тим не менш, до його створення потрібно підходити дуже відповідально, адже сайт-візитка - це майбутнє обличчя бізнесу. Дизайн повинен бути приємним, це збільшує ваші шанси на велику кількість відвідувачів, а значить і замовників, а контент - цікавим і унікальним.

Постає проблема в тому, що з'являється досить значна частка некваліфікованих виконавців на ринку праці в сфері інформаційних технологій, які зловживають довірою підприємства, через необізнаність в сфері інформаційних технологій. Це приводить до того, що робота буде виконана непрофесійно з недотриманням стандартів актуального та конкурентноспроможного сайту.

Наприклад, одні з базових вимог до верстки сайту - це перевірки на написання валідного коду HTML та CSS за допомогою валідаторів Online (W3C) [5] [6].

Подані вище за посиланням валідатор (W3C) – це служба перевірки розмітки консорціуму World Wide Web, який дозволяє користувачам Інтернету перевіряти документи HTML та XHTML до HTML5 на предмет правильної розмітки з визначення типу документа. Також це відноситься до

вірного написання CSS. Перевірка розмітки — важливий крок до забезпечення технічної якості веб-сторінок.

Також ці сервіси роблять перевірку на семантику верстки. Семантика верстки, або семантичний HTML-код — це підхід до створення веб-сторінок на мові HTML, заснований на використанні HTML-тегів, те з чого складається верстка, відповідно до їх семантики (призначення), а також передбачає логічну та послідовну ієрархію веб-сторінки. Що в свою чергу призводить до кращих результатів в выдачі по пошуковим запитам користувачів.

Стандарти, які можна віднести до створення актуального та конкурентоспроможного сайту:

- Валідне написання HTML та CSS.
- Використання семантичних тегів HTML.
- Наявність мобільної (адаптивної) версії сайту.
- Швидке завантаження та відкриття сайту.
- Находження сайту в «зеленій» зоні сервісу PageSpeed Insights API [7].
- Сучасний дизайн веб-сайту зі зрозумілим відображенням інформації для користувачів.

Опираючись на описані вимоги, можна дійти висновку, що створення актуального та конкурентоспроможного сайту-візитки підприємства потребує доволі значних зусиль та пошук кваліфікованих виконавців, щоб виділятися серед конкурентів.

Сервісів для тестування ресурсу згідно з параметрами SEO, конверсії і юзабіліті багато. Під час вибору потрібно орієнтуватися на точність їхніх показників. Повільне завантаження сьогодні — наслідок проблем із кодом. Усувати їх повинні програмісти. На помилки в коді вказують спеціальні speed test-сервіси [10].

Проаналізувавши конкурентів в Інтернет просторі, можна відзначити можливі переваги відносно інших підприємств, які також мають власні веб-

сайти, для просування бізнесу в цілому, інформації про сервіс та послуги, та залучення нових клієнтів за допомогою створенного веб-сайту.

Один із конкурентів в сфері послуг є підприємство «Автоклініка» [8], яке відноситься до незалежного автосервісу та має досить схожий спектр послуг. Аналізуючи веб-сайт даного підприємства, відразу можна побачити недоліки, які потрібно уникнути при створенні сайту-візитки для нашого підприємства. Даний веб-сайт відноситься багатосторінкових сайтів. До переліку недоліків можна віднести:

- Користувальницький інтерфейс є дуже простим та застарілим з великими порожніми проміжками між контенту та відсутність структури виділення основного тексту поміж графічного контенту.
- Відсутність зручної мобільної (адаптивної) версії сайту на різних девайс, де відбувається нагромадження елементів поверх інших елементів.
- Досить повільне відкриття сайту понад 5 секунд, незважаючи на досить незначну кількість контенту.
- Находження сайту в «жовтогарячій» зоні сервісу PageSpeed Insights API [7]. Це показник середньої зони, що означає наявність проблем та неоптимізованих рішень.
- Наявність 10 помилок валідності HTML коду за перевірки в сервіс-валідаторі [5].
- Візуальне представлення місцезнаходження підприємства знаходиться на одній з внутрішніх сторінок сайту. Це призводить до того, що користувач може не знайти швидко потрібну інформації про підприємство та закрити веб-сторінку.

Схожий результат був в іншого підприємства – автосервіс «У Сергія» [9].

1.7 Переваги створення сайту-візитки та використання сучасних рішень

Сайт-візитка - це електронна візитна картка компанії або окремої людини, з описом діяльності. Більшість сайтів-візиток, що зустрічаються в Інтернеті, складаються з 1-3 окремих сторінок.

Основний елемент сайту-візитки - це головна сторінка, яка повинна містити всю необхідну інформацію про компанію чи людину. Зокрема, її подають у вигляді слайдера з текстовим наповненням. Важливий елемент головної сторінки - логотип компанії, який відображає сферу діяльності, зрозумілий і якомога лаконічніший.

Значною перевагою сайту-візитки - це питання корисності та життєздатності, де при незначному обсягу та розмірі всього веб-сайту буде менше часових витратах на вирішення цих питань.

Наприклад, досягнути швидкого завантаження сайту-візитки значно легше, ніж для багастрінкових веб-сайтів, чи довгих односторонікових сайтів (лендингів). В свою чергу швидкість завантаження впливає на юзабіліті, конверсію, ранжування сайту — тобто на ключові показники:

- Зручність. Оптимальна швидкість завантаження сайту — 2 секунди. Якщо веб-ресурс вантажиться довше зазначеного проміжку часу — планомірно зростає кількість відмов від візиту на нього.
- Ранжування. Google радить скорочувати швидкість завантаження в рамках просування за допомогою SEO. Боти, які відвідують сторінки, не можуть їх повноцінно проіндексувати. Завантаження сайту тривалістю понад 3 секунди вже є причиною для зниження рангу в пошуковій системі.
- Трафік і конверсія. Користувачі Інтернету не люблять чекати. Низька швидкість завантаження призводить до відмов від перегляду сайту. Знижується трафік, а з ним і конверсія, показник якої залежить від загальної кількості відвідувань.

Так як сайт-візитка має зручність в компактному виведенні інформації та незначній кількості сторінок, то є досить значна можливість в невикористанні бази даних. В результаті мінімізуються ресурси, що витрачаються на веб-розробку. А керування різними платформами здійснюється централізовано з одного інтерфейсу, що зручно. При цьому вміст налаштовується гнучко для кожного окремого каналу.

Особливості, які має мати сайт-візитка:

- Короткий зміст та лаконічна презентація, що допомагає відвідувачу цього Інтернет-ресурсу швидко знайти чи натрапити на інформацію стосовно підприємства;
- Зручний та продуманий UX дизайн;
- Адаптивність сайту під різні пристрої;
- Висока швидкість завантаження сайту через відсутність неефективного контенту для користувачів, застарілих технологій та неоптимізованих рішень;
- Можливість редагувати контент на сайті та створити новий зручним способом без залучення технічного спеціаліста;
- Відсутність бази даних для зменшення витрат на розробку та зручність управління здійснюється централізовано з одного інтерфейсу;
- Стабільність роботи ресурсу.

Так звана модернізація нашого рішення для підприємства повинна мати для цього можливості, щоб це не стало причиною переробки зробленої роботи через досить незначний проміжок в часу опираючись на стрімкий розвиток інформаційних технологій та конкурентоспроможності в візуальному, технологічному та комунікаційному спектрах ефективності поданного вдосконалення підприємства:

- Розробка концепції, де подальша взаємодія з вмістом сайту-візитки буде відбуватися без залучення технічного спеціаліста, в тому разі якщо це не переробка самої структури функціоналу. Для подальшого роботи над вмістом буде раціонально створити нову роль відподальності в підприємстві, для підтримки сайту та взаємодії з потенціальними клієнтами-користувачами сайту. В кінцевому результаті нову описану роль буде виконувати співробітник-адміністратор, де на досить незначну зайнятість та рівень навичок можлива оплата праці буде в пів ставки.

- Використання програмних технологій, які є актуальними, мають підтримку розробників цих технологій, щоб технологія не застаріла і не пришлося повністю робити заміну та переробляти і переписувати програмний продукт.
- Використання технологій, які мають велику спільноту підтримки та дискусій на різних інтернет-ресурсах, щоб бути впевненим, що в разі виникнення проблеми з використанням технології, чи наявності якоїсь поломки, так званого бага, в сторонній програмі якоїсь версії, яку ми використовуємо для рішення задач, то завдяки великій спільноті з'являється досить велика можливість вирішити це значно швидше, зекономивши витрати часу на створення власного виправлення для технології, що призводить до зменшення витрат підприємства.

1.8 Концепція рішення

Способом залучення клієнтів є представлення діяльності підприємства в інформаційному просторі. Створення сучасного сайту-візитки з наявністю інформації про підприємство дозволяє залучати нових клієнтів через стрімкий попит людей в пошуку підприємств для вирішення їх проблем в Інтернет-просторі.

Потреба майбутніх користувачів веб-сайту полягає в задоволенні їх потреб та сподівань, що саме вони хотіли б побачити, перейшовши на веб-сторінку профільного сайту підприємства. Та як саме вони можуть потрапити на сайт-візитку.

Ідея вмісту сайту-візитки полягає в наданні стислої інформації про підприємство, а саме назва, контактні дані, адреса та заголовок повинні відобразитися на видному місці для користувача, тому що це перше, що може він шукати. Не менш важливим є додаткова інформація про перелік послуг та опис роботи підприємства, робочий графік.

В такому випадку веб-сайт повинен відкриватися швидко, мати зрозумілий та естетично оформлений головний екран, тобо першу секцію веб-сайту, яку відразу бачить користувач. Користувач повинен зрозуміти відразу, що це саме те що він пробував знайти, при запиті вирішити його проблему – знайти автосервіс та вирішити поломки з його транспортним засобом.

Ідея представлення діяльності підприємства в інформаційному просторі, окрім точного запиту пошуку атосервісу користувачем в пошукових-сервісах, можливо досягти створення блоку корисних статей для вирішення тієї чи іншої проблеми пов'язаної з транспортним засобом. Внаслідок такого переходу користувача на корисну статтю, повинно бути посилення головну сторінку та інформація про автосервіс, щоб користувач вирішив радше скористатися послугами підприємства, ніж чинити поломку власноруч.

Висновком може бути, що розроблене програмне рішення здатне залучити нових клієнтів та зекономити витрати підприємства на нефективні методи представлення підприємства, які були раніше. Розроблена концепція дозволяє вирішити запити клієнтів в пошуку вирішення їх проблем з транспортними засобами.

1.9 Технічне завдання

Для реалізації технічного рішення, надано розроблене технічне завдання, яке можна знайти за посиланням [Додаток Б].

2. Розробка проекту програми та ПЗ

2.1 Загально-системні рішення

З метою виявлення системних вимог заснований на використанні варіантів використання системи (Use Cases), які охоплюють як функціональні, так і не функціональні вимоги, які специфічні для конкретного варіанту використання.

Діаграма варіантів використання в наочному вигляді демонструє замовнику сформульовані, на основі його вимог, варіанти використання майбутньої програми.

Для адміністратора важливо мати можливість працювати з вмістом на веб-сайті, щоб постійно показувати користувачам актуальну інформацію. Таким чином, якщо визначити всі можливі варіанти використання системи користувачами, або іншими зовнішніми процесами, то ми отримаємо функціональні вимоги до неї.

Для визначення вимог до налаштування можливості редагування вмістуна сайті, звернемо увагу на діаграму прецедентів (рис 2.1).

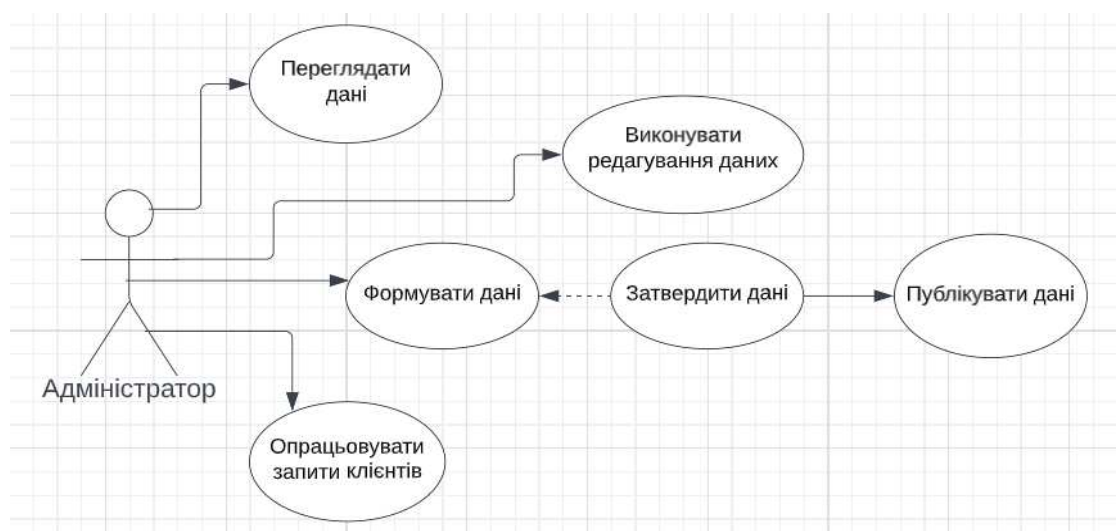


Рисунок 2.1 Діаграма прецедентів

Адміністратор. Він (актор) виконує певні функції в системі. Use-case діаграма дозволяє в подальшому нам визначити архітектуру програми та взаємодію з даними.

Варіанти використання адміністратора включають:

- Перегляд даних;
- Редагування даних;
- Формування даних;
- Затвердження даних;
- Публікація даних;
- Опрацювання запитів клієнтів.

Для взаємодії користувача з даними приведемо приклад з використанням діаграми потоків даних, де графічно є можливість візуалізації цього процесу.

На діаграмі потоків даних (рис 2.2) показана взаємодія користувача та отримання даних з системи управління вмістом.

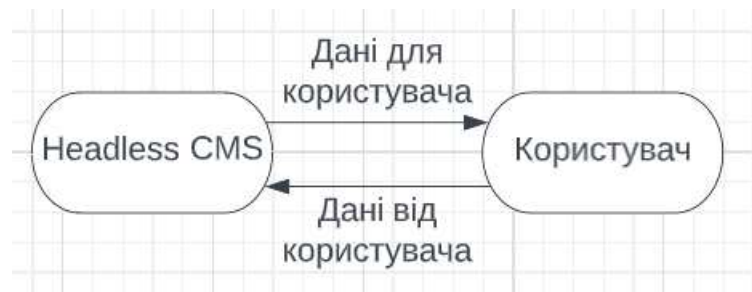


Рисунок 2.2 Діаграма потоків даних

На поданій діаграмі можна побачити, як буде відбуватися процес відображення даних користувачеві. Тобто для отримання даних на стороні користувача потрібно передати налаштування доступу для отримання даних.

2.2 Рішення в ПЗ

2.2.1 Вибір технологій для виконання задач

Для реалізації поставленої задачі потрібно порівняти та вибрати найкраще рішення для підприємства з досить великого переліку сучасних веб-технологій.

Перш за все, розробка веб-сайтів – це процес створення веб-сайтів та додатків. Існують поняття розробка фронт-енд частини та розробка бекенд частини – це два різні типи веб-розробки.

Фронтенд-розробка – це розділ комп'ютерного програмування, який зосереджується на кодуванні та створенні елементів і функцій веб-сайту, які потім побачить користувач. Мова йде про те, щоб візуальні аспекти веб-сайту були функціональними. Ви також можете розглядати інтерфейс як «клієнтську сторону» програми. Це означає, що робота розробника полягає в кодуванні та втіленні в життя візуальних елементів веб-сайту. Більше зосереджені на тому, що бачить користувач, коли відвідує веб-сайт або додаток. Потрібно зробити так, щоб з сайтом було легко взаємодіяти, а також безперебійно працювати.

Ці розробники беруть візуальний дизайн від UX і UI дизайнерів і оживляють веб-сайт, переконавшись, що він добре функціонує для користувача. Один із багатьох способів використання навичок фронт-енду — це створення статичного веб-сайту, який є веб-сайтом із фіксованим вмістом, який доставляється у браузер користувача точно так, як він зберігається. Можна зіткнутися зі статичним веб-сайтом, якщо натрапите на просту цільову сторінку або веб-сайт малого бізнесу, який не дозволяє користувачам виконувати будь-які інтерактивні завдання.

Розробники інтерфейсу створюють такі елементи, як:

- Верстка;
- Кнопки;
- Макети;
- Анімації;

- Організація контенту;
- Навігація;
- Зображення;
- Графіка.

Фронтенд-розробка складається з різних частин, які можна розділити на компоненти за допомогою яких реалізується створення веб-сайту чи веб-додаток:

- HTML (HyperText Markup Language) - мова розмітки документів для створення структури сторінки: заголовки, абзаци, списки і так далі;
- CSS (Cascading Style Sheets) - мова для опису і стилізації зовнішнього вигляду документа. Завдяки CSS-коду браузер розуміє, як саме відображати елементи. CSS задає кольори і параметри шрифтів, визначає, як будуть розташовуватися різні блоки сайту, і так далі. Ще він дозволяє виводити один і той же документ в різних стилях, наприклад, для друку (звичайним або шрифтом Брайля), виведення передачі на екран або читання голосом;
- JavaScript - це мова, що створювалася, щоб оживити веб-сторінки. Її завдання - реагувати на дії користувача, обробляти кліки мишкою, переміщення курсора, натискання клавіш. Ще вона посилає запити на сервер і завантажує дані без перезавантаження сторінки, дозволяє вводити повідомлення і багато іншого.

Що стосується розробки бекенд частини, або ще називають серверної частини, зосереджена на стороні веб-сайту, яку користувачі не можуть побачити. Розробка серверної частини займається зберіганням і впорядкуванням даних, а також забезпеченням належної роботи передньої частини.

Бекенд частина - це процес об'єднання сервера з користувачем. Сервер зберігає дані і відповідає на запити користувачів.

Бекенд-розробник застосовує ті інструменти, що доступні на його сервері. Він має право вибрати будь-яку з універсальних мов програмування,

наприклад, Ruby, PHP, Python, Java. Все залежить від конкретного проекту і завдання замовника.

Також для серверної розробки використовуються системи управління базами даних:

- MySQL;
- PostgreSQL;
- SQLite;
- MongoDB.

Залежно від продукту обов'язки бекенд-розробника сильно змінюються. Такий фахівець може створювати та інтегрувати бази даних, забезпечувати безпеку або налаштовувати технології резервного копіювання та відновлення.

Взаємодія клієнтської та бекенд частин відбувається по колу:

- frontend відправляє призначену для користувача інформацію в backend;
- інформація обробляється;
- і повертається назад, прийнявши зрозумілу форму.

Над цим займаються різні фахівці, але кожному з них бажано розуміти принципи, за якими працюють колеги. Навіть дизайнеру інтерфейсів (UI-дизайнер) важливо хоча б в загальних рисах знати, що являє собою backend проекту, яким він займається. Це допоможе адекватно оцінити, які технічні можливості є у сайту або програми.

Існує кілька варіантів взаємодії фронтенд і бекенд:

- HTTP-запит безпосередньо відправляється на сервер, сервер шукає інформацію, вбудовує її в шаблон і повертає у вигляді HTML-сторінки.
- Варіант з використанням інструментарію AJAX (Asynchronous JavaScript and XML). У цьому випадку запит відправляє JavaScript, завантажений в браузер, а відповідь приходить в форматі XML або JSON.

Враховуючи те, що посталевна задача реалізувати сайт-візитку, для отримання максимального результату є можливість залучити кваліфікованого фронтенд-розробника, який зможе створити сучасне та

гнучке рішення за допомогою сучасних веб-технологій без залучення додаткових профільних програмістів. При цьому веб-сайт повинен мати можливість додавати та редагувати контент динамічно за допомогою зручної адмін-панелі, щоб це завдання могла виконувати людина без технічних знань кодування або складних систем.

Також важливий пункт реалізації SEO-оптимізації, тому що в разі створення простого веб-додатку, односторінковий JavaScript програма, яка запускається і працює в браузері, JavaScript не обробляється більшістю пошукових систем. Причина в тому, що рендеринг сторінки повністю відбувається на стороні клієнта, а не на сервері, саме тому пошукові боти не зможуть проаналізувати сайт-візитку та просунути його за запитами користувачів пошукових-сервісів.

2.2.2 Порівняння технологій

Для створення сучасного, масштабуючого та конкурентноспроможного рішення, використання технологій, які мають велику спільноту підтримки та дискусій на різних інтренет-ресурсах, тому чудовим вибором є технологія React.js.

React.js — найпопулярніша бібліотека JavaScript для створення веб-додатків. React.js або Reactjs або просто React — це різні способи представлення React.js.

React.js — це бібліотека JavaScript з відкритим кодом, яка використовується для створення інтерфейсів користувача спеціально для односторінкових програм. Він використовується для обробки шару перегляду для веб- та мобільних додатків. React також дозволяє нам створювати багаторазові компоненти інтерфейсу користувача. React дозволяє розробникам створювати великі веб-додатки, які можуть змінювати дані, не перезавантажуючи сторінку. Основна мета React — бути швидким, масштабованим і простим. Це відповідає подання в шаблоні MVC.

React був розгорнутий Facebook у 2011 та 2012 роках. Instagram був повністю написаний на React. Згідно з аналітичним сервісом JavaScript Libscore, React наразі використовується на домашніх та інших веб-сторінках Netflix, Airbnb, Imgur, Feedly та інших.

Один із важливих показників розвитку технологічного стека чи засобів розробки – це кількість завантажувальних пакетів. NPM trends [12] показує найбільш точну статистику для інструментів, включаючи не тільки кількість завантажень, а також дані Github. React.js у цьому плані значно випереджає своїх конкурентів. Angular.js та Vue.js.

Підбиваючи підсумки стосовно вибору кращої технології для побудови інтерфейсу користувача, React.js є кращим варіантом, маючи такі переваги:

- Маркетингова історія. Дуже важливий аргумент те, що одні з найвідоміших продуктів, як Facebook та Instagram цю технологію, і саме команда Facebook створила її. Тому для підприємства важливо те, що використовується настільки сучасні та найкращі технології, щоб бути впевненими в неї;
- Популярність. Технологія є найпопулярнішою серед своїх конкурентів, тому знайти програміста з потрібними навичками, для розширення чи змінення функціоналу після реалізації не буде викликом, що зменшить затрати на пошук та виконання самої роботи;
- Велика спільнота. Дуже багато програмістів, які активно підтримають та використовують технологію. Це збільшує впевненість в тому, що в разі виникнення труднощів, можна буде знайти допомогу на різних форумах, по типу Stackoverflow [13];
- Підтримка відтворення на стороні сервера (Server-Side Rendering, або SSR). Відтворення на стороні сервера за допомогою бібліотек JavaScript, таких як React, — це те, коли сервер повертає готову до відтворення HTML-сторінку та сценарії JS, необхідні для того, щоб зробити сторінку інтерактивною. HTML відображається відразу з усіма статичними елементами. Значить, що вміст веб-сторінки індексується швидше, що

призводить до кращого рейтингу SEO. Цього можна досягнути за використанням технології Next.js, яка побудована на React.js.

Саме тому використання React.js є найкращим рішенням для реалізації певної частини всього проекту.

З цього випливає досить зрозуміле використання Next.js.

Next.js - відкритий JavaScript фреймворк, створений поверх React.js для створення веб-застосунків. Фреймворк призначений для вирішення проблеми React.js, пов'язаної з відтворенням програми на стороні сервера - SSR. Працює на сервері та в браузері.

Відтворення на стороні сервера за допомогою бібліотек JavaScript, таких як React, - це те, коли сервер повертає готову до відтворення HTML-сторінку та сценарії JS, необхідні для того, щоб зробити сторінку інтерактивною. HTML відображається відразу з усіма статичними елементами. Тим часом браузер завантажує та виконує JS-код, після чого сторінка стає інтерактивною. Взаємодія на цій сторінці тепер обробляється браузером на стороні клієнта. Для будь-якого нового вмісту або нових даних браузер надсилає запит на сервер через API, і отримує лише нову необхідну інформацію.

2.2.3 Запровадження SEO-оптимізації сайту-візитки

Опираючись на інформацію описану раніше, можна зрозуміти, що пошукова оптимізація має вирішальне значення для залучення трафіку на веб-сторінки. Наразі більшість пошукових систем, крім Google, недостатні для відтворення сторінок перед індексуванням. Навіть у Google є проблеми з навігацією сайтів із відображенням на стороні клієнта.

При відображенні на стороні сервера всі елементи, необхідні для SEO, доступні в початковій відповіді. Крім того, веб-сторінки, відтворені на стороні сервера, індексуються точніше, оскільки браузери віддають перевагу сторінкам, які завантажуються швидше.

Як приклад, на платформах соціальних медіа посилання на веб-сайти з відображенням на стороні сервера краще для правильного представлення назви та мініатюри сайту.

Говорячи про продуктивність, повинні враховувати три основні параметри:

1. TFB (Time to First Byte) – проміжок часу між натисканням посилання та отриманням першого біта вмісту.

2. FCP (First Contentful Paint) – момент, коли відтворюється деякий запитаний вміст.

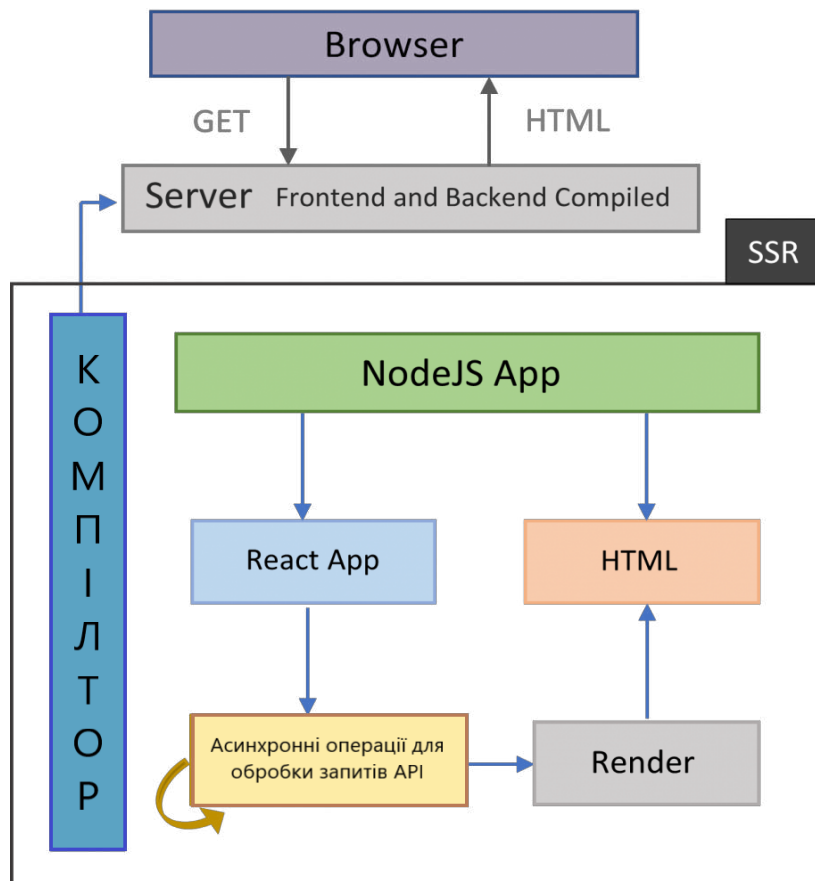
3. TTI (Time To Interactive) – час, коли сторінка стає інтерактивною.

TFB може бути більш ефективним із рендерингом на стороні сервера, оскільки повністю відтворена HTML-сторінка надсилається до браузера, але FCP набагато швидше, що сприяє підвищенню продуктивності та зручності для користувачів. TTI може залежати від складності веб-сторінки та кількості сценаріїв, які потрібно запустити для її відтворення. Більшість сторінок мають помірний рівень інтерактивності, що призводить до низького TTI при відтворенні на стороні сервера.

Необхідні вимоги в середовищі для запуску SSR

Перша і головна вимога для відтворення на стороні сервера — це середовище виконання, яке може реалізувати веб-сервер і обробляти події.

Node.js — одна з найпопулярніших платформ для налаштування SSR для програми React, а Express — чудовий варіант для створення HTTP-сервера. Далі нам потрібен компілятор JavaScript, наприклад Babel, і пакет модулів JS, як-от Webpack, Rollup або подібний інструмент.



Схематичне зображення застосування SSR [14]

Розглядаючи дане зображення стає зрозумілим гостра потреба оптимізувати процеси, щоб отримати перевагу в легкості реалізації та зменшенні витрат часу на створення рішень. Тому що така система може швидко ускладнитися і створити проблеми зі стилями, зображеннями, маршрутизацією, історією браузера тощо.

Загалом, існує досить багато бібліотек і пакетів, які потрібно встановити та налаштувати під час налаштування відтворення на стороні сервера на етапі розробки на комп'ютері, і може бути ресурсозатратним реалізовувати все самостійно.

Саме для оптимізації багатьох процесів є наявна потреба у використанні технології Next.js.

Окрім головної переваги Next.js в можливості оптимізувати перелічені вище процеси, також є покращення в роботі самого веб-додатку:

- Швидке початкове завантаження веб-сторінки, оскільки готовий до відображення HTML надається браузеру.

- Чудовий досвід роботи з користувачем, навіть якщо у користувача погане з'єднання, застарілий пристрій або вимкнений JavaScript у браузері, оскільки весь основний вміст готовий до відтворення.
- Вміст веб-сторінки індексується швидше, що призводить до кращого рейтингу SEO.
- Чудовий варіант для статичних сторінок, оскільки рендеринг на стороні сервера завантажує вміст швидко й ефективно.



Схематичне зображення переваг використання Next.js для бізнесу [13]

2.2.4 Можливість редагувати контент

Однією з важливих задач є створення можливості зручної роботи з редагуванням контенту.

Щоб залучити й утримати якомога більше відвідувачів за допомогою веб-сайту, необхідний інформативний контент. Тексти, зображення, відео та графіка не тільки забезпечують додаткову цінність ресурсу для користувачів, але і позитивно відзначаються пошуковими системами.

Після створення, будь-який контент повинен бути опублікований, а потім їм треба управляти, оновлювати і поширювати в Інтернеті. Незалежно від розміру веб-сайту, це трудомістке завдання, що реалізується за допомогою системи управління контентом (англ. Content Management System, або CMS).

В нашому випадку, при використанні технологій React.js та Next.js система управління контентом повинна бути сучасна, інтерактивна та найкраще підходити під наш проект. Таким чином з переліку існуючих CMS найкращим рішенням буде використання Contentful.

Contentful - сучасна безголовна (headless) CMS, що просуває підходи, що ґрунтуються на API та хмарі

Headless CMS

Серед різноманітних рішень бекенд як сервісне рішення (Backend as a Service solution), headless CMS займає особливу ринкову нішу і завоювала велику популярність серед розробників [12]. У порівнянні з «традиційними» CMS, такими як Wordpress або Joomla, вони зосереджені виключно на розміщенні контенту, інструментах для зручного редагування вмісту та наданні API для доступу до вмісту з програми. Це дає розробникам і дизайнерам повну свободу реалізації бізнес-логіки та зовнішнього вигляду програми. Це відповідний інструмент для роботи з контентом, який є легким для людей без досвіду розробки. Також важлива можливість розділити процеси розробки додатків та оновлення контенту, оскільки за це відповідають різні відділи.

Наприклад, один з найпопулярніших CMS Wordpress, який дозволяє доволі легко створити сайт з використанням шаблону та з наявністю можливості редагувати контент зі зручної адмін-панелі, має ряд недоліків, які в кінцевому результаті будуть впливати на загальний результат сайту-візитки, де для досягнення потрібного результату, потрібно буде використовувати додаткові плагіни, в тому числі для SEO-оптимізації, та бути прив'язаним конкретно до цієї CMS Wordpress.

Основні мінуси WordPress [12]:

- Відчутне на сервер навантаження;
- Іноді низька швидкість завантаження веб-ресурсу;
- Невеликий набір базових можливостей;
- Дублювання зображень та сторінок;
- Величезна кількість неякісно зроблених тем та розширень з помилками;

- Упереджене ставлення пошукових систем до сайту на безкоштовному движку.

CMS WordPress, а саме безліч доповнень, що встановлюються до нього, суттєво навантажують сервер, де розміщений сайт. Щоб уникнути такого навантаження скоротите кількість додаткових плагінів до можливого мінімуму, а головне, розміщуйте сайт на якісному хостингу. Ці невеликі тонкощі сприятимуть збільшенню швидкості завантаження сайту.

Ще одна вада WordPress - автоматична генерація дублів сторінок, що негативно впливає на просування сайту. Також цей двигун при завантаженні зображень робить їх копії, причому в різних розмірах. Щоб запобігти цьому, треба встановлювати додатковий плагін, наприклад, плагін DNUI.

Загальне використання WordPress, його гідність і в той же час - недолік, тому пошукові системи насторожено ставляться до сайтів на таких CMS.

Навіть при використанні шаблону, для зекономлення витрат на розробку кастомного рішення, то шаблон має обмежений набір функцій. Хоча їх можна розширити плагінами, але надмірна кількість може вплинути на швидкість роботи веб-ресурсу.

Для запобігання такій ситуації додатково можна редагувати сам код. Однак знадобляться певні навички або послуги програміста, що суперечить поставленій задачі в легкості редагування контенту.

Стосовно переваг використання Contentful:

- Має широке кешування, а також численні резерви, які забезпечують швидку доставку вмісту. Таким чином, майже 98% вмісту буде доставлено з кешу, коли в разі всі резерви інфраструктури доставки вийдуть з ладу.
- Платформа має можливість редагування вмісту в реальному часі, що покращує командну співпрацю. Ви можете легко імпортувати свій вміст з будь-якого джерела за допомогою надійного API та спеціальних скриптів. Це означає, що можна створювати й оновлювати моделі вмісту та вмісту за допомогою API керування.

- Платформа не блокує користувачів у певній архітектурі або попередньо визначеній моделі; натомість це залишається за професійним розробником. Contentful дає свободу, яку ви можете використовувати для створення додатків для iOS, Android або будь-якої іншої платформи.

Беручи до уваги всі описані вище особливості та зробовши дослідження серед сучасності та гнучкості рішень дійшли висновку використовувати React.js, Next.js Contentful.

2.2.5 Структура проєкту

Беручи до уваги те, що основні технології в розробці – це React.js та Next.js, які мають власний процес розгортання проєкту та його структури, будемо слідувати базовому налаштуванню.

Для початку створимо додаток Next, схожу на створення додатку React, використовуючи команду `prx`, яка постачається в комплекті з Node.js. Команда ініціалізує програму Next і створить нову папку з назвою програми. Усі необхідні пакети будуть автоматично завантажені та посилатися в `package.json`.

Ми можемо негайно запустити зразок програми, перейшовши до кореневого каталогу програми та запустивши сервер `prn`.

Структу корневої директорії складеться з:

- Файлу `package.json`, який зберігає команди запуску та необхідні пакети, для роботи;
- Директорії `pages`, яка побудована на основі концепції сторінок Next.js. Сторінка — це компонент React, експортований з файлів `.js`, `.jsx`, `.ts` або `.tsx` у каталозі сторінок;
- Директорія `public`, зберігає статичні ресурси, такі як зображення, шрифти тощо;

Дана структура є необхідною для роботи веб-додатка, а інші додаткові файли та директорії слугують розширенням технічного рішення.

2.3 Інформаційне забезпечення

Для використання можливості взаємодії з даними, потрібно слідувати правилами валідного введення даних в системі управління вмістом (CMS).

В даному випадку використання технології Contentful дозволяє автоматизувати цей процес тим, що перевірка на валідність даних вже реалізована в адмін-панелі.

Наприклад, для введення тексту, для необхідної секції на веб-сайті, потрібно створити необхідну модель контенту. Ввівши назву та пройшовши далі, з'являється можливість додати необхідне поле, де є чітко виржені валідаційні правила та обмеження.

Перелік деяких полів з конкретними визначеннями:

- Текстове поле (коротке) – це строка, призначена для невеликого тексту та назв. Має обмеження в 256 символів;
- Текстове поле (довге) – це строка, призначена для довгого тексту з параграфами. Має обмеження в 50 тисяч символів;
- Медіа-файл (один файл) – це об'єкт, який в собі збергіє посилання до завантаженого медіа-файлу.

2.4 Організаційне забезпечення

Маючи дане технічне рішення, яке є дуже гнучким та простим в взаємодії, для підприємства постає потреба в наявності всього в одному новому співробітнику – адміністраторі.

В обов'язки адміністратора буде входити:

- Підтримка актуального контенту на сайті;
- Створення нового контенту в разі потреби, це більше відноситься до створення нових коичних статей для залучення потенціальних відвідувачів;
- Приймання звінків від користувачів сайту.

Враховуючи даний перелік обов'язків, достатньо буде найняти належного співробітника з зайнятістю на пів-ставки. Що дозволяє значно зменшити витрати на заробітну платню, тим самим зробивши ефективнішою бізнес-модель по залученню нових клієнтів.

3. Реалізація ІС підприємства

3.1 Реалізація модулів рішення

Для рішення даної задачі також важливою задачею було реалізувати гнучке технічне рішення. Це дозволяє мати можливість досить легкого розширення функціональних можливостей веб-додатка, тому що технічне рішення є масштабуючих та розділеним на компоненти, та окремі ізольовані сервіси для виконання бізнес-логіки. В даному випадку основна бізнес логіка показана в реалізації отримання даних з системи управління вмістом.

Підхід реалізації технічних рішень за допомогою розділення повторюваного коду на компоненти, дозволяє гнучко настроювати компонування сторінок, створюючи базовий шаблон, який приймає реалізовує в собі роботу з даними, а саме їх отримання та виведення.

Поданий нижче приклад коду продемонструє реалізацію запиту на отримання вмісту для головного розділу та його виведення на веб-сайті:

Функція отримання вмісту для розділу

```
async function getHero () {  
  const client = createClient()  
  
  const params = {  
    'content_type': 'hero',  
  }  
  
  const data = await client.getEntries(params)  
  
  const { items } = client.parseEntries(data)  
  
  return {  
    data: items[0].fields  
  }  
}
```



```
}  
}
```

Виведення вмісту для головного розділу додатку:

```
<div className="left__item">  
  <h1>{hero.data.title}</h1>  
</div>  
  
<picture className="right__img-wrapper">  
  <img  
    src={hero.data.image.fields.file.url}  
    alt={hero.data.image.fields.title}  
    title={hero.data.image.fields.title}  
  />  
</picture>
```

В даному прикладі можна побачити реалізацію функції отримання даних для головної секції та подальшим виведенням даних для користувача.

Таке гнучке рішення дозволяє виводити потрібну інформацію не змінюючи логіку інших технічних рішень, тому що саме запит є ізольованим та імпортується в потрібний файл, де потім він і викликається для отримання даних.

Однією з переваг програмного рішення є зручність редагування контенту всього веб-додатку за допомогою використання адмін-панелі технології Contentful. Адміністратору не потрібно мати технічні навички чи змінювати щось в базі даних, тому що адмін-панелі дозволяє взаємодіяти з контентом в інтуїтивно зрозумілому інтерфейсі.

Name	Description	Fields	Updated	By	Status
Article	Here is the content of a helpful article.	4	Jun 19, 2022	Me	ACTIVE
ArticleInfo	The info in the article section	1	Tue, 12:56 AM	Me	ACTIVE
Diagnose	Diagnose section	4	Jun 19, 2022	Me	ACTIVE
Hero	Hero section	2	Jun 19, 2022	Me	ACTIVE
mainContactData	Data contains the main info about company.	4	Tue, 12:29 AM	Me	ACTIVE
OfferInfo	What we offer info	2	Jun 19, 2022	Me	ACTIVE
OfferItems	Offer items	2	Jun 20, 2022	Me	ACTIVE
ServicesInfo	Info for the Services section	1	Jun 19, 2022	Me	ACTIVE
ServicesItems	Services items	3	Jun 20, 2022	Me	ACTIVE

Рисунок 3.1 Список створених моделей даних

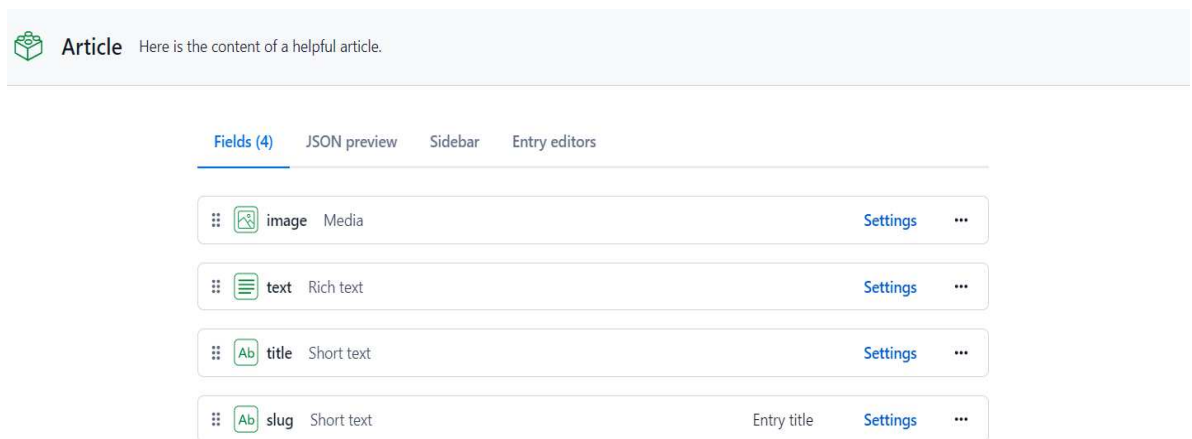


Рисунок 3.2 Список створених полів для моделі даних «Article»

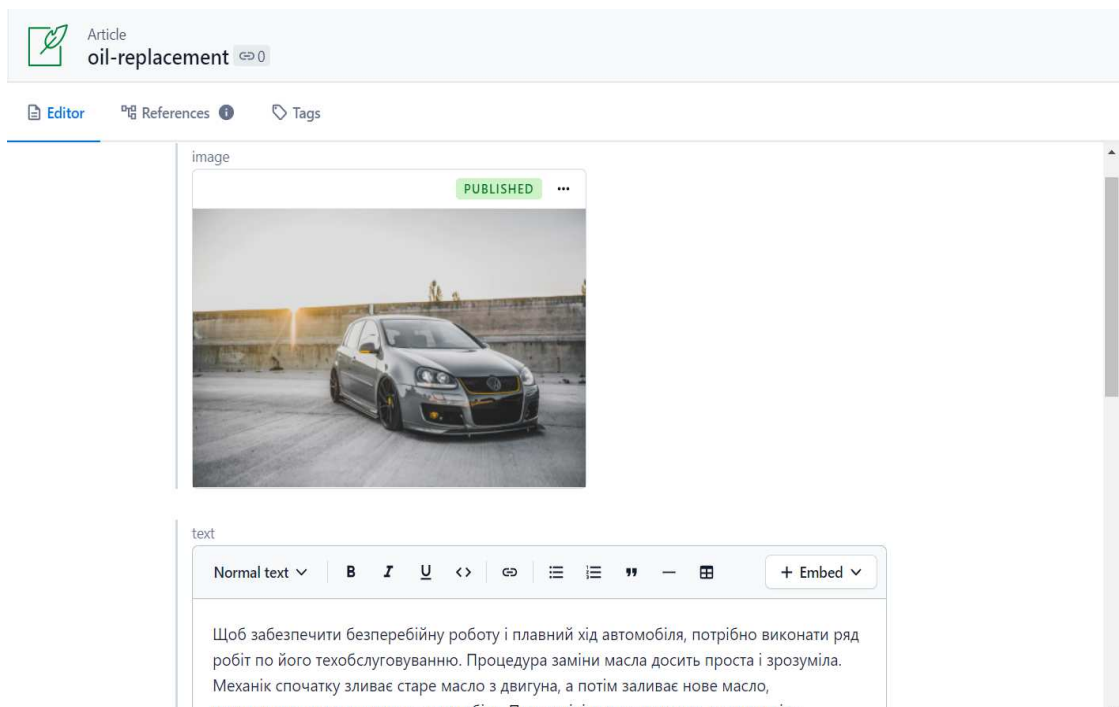


Рисунок 3.3 Створення даних згідно моделі даних «Article» (частина 1)

Article
oil-replacement

Editor References Tags

Normal text **B** *I* U <> + Embed

масла в ньому.

Поліпшення робочих характеристик двигуна. Регулярна заміна моторного масла набагато важливіша для двигуна, ніж може здатися. Що може бути приємніше для водія, ніж ефективно працюючий двигун його автомобіля? Своєчасна заміна масла гарантує належне змащування внутрішніх рухомих компонентів двигуна, завдяки чому можна насолоджуватися плавним, легким і комфортним ходом автомобіля і абсолютно спокійно відправлятися в будь-яку поїздку.

title
ЗАМІНА МАСЛА
12 characters Maximum 256 characters

slug
 oil-replacement
name the correct article url

Рисунок 3.4 Створення даних згідно моделі даних «Article» (частина 2)

4. Охорона праці

4.1 Вступ

Згідно ст. 1 Закону України «Про охорону праці» охорона праці має велике соціально-економічне значення. Соціальне значення охорони праці полягає в сприянні росту ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань.

Зростання продуктивності праці відбувається в результаті збільшення фонду робочого часу завдяки скороченню внутрішньо-змінних простоїв шляхом ліквідації мікротравм або зниження їх кількості, а також завдяки запобіганню передчасного стомлення шляхом раціоналізації і покращення умов праці та введенню оптимальних режимів праці і відпочинку та інших заходів, які сприяють підвищенню ефективності використання робочого часу.

Економічне значення охорони праці визначається ефективністю заходів з покращення умов і підвищення безпеки праці та є економічним виразом соціальної значущості охорони праці. Економічне значення охорони праці оцінюється за результатами, отриманими при зміні соціальних показників шляхом впровадження заходів з покращення умов праці.

Даний розділ дипломного проєкту присвячений аналізу і розгляду питань охорони праці при роботі за комп'ютером.

4.2 Основні законодавчі і нормативні документи, які регламентують Охорону праці України

Основними законодавчими актами, якими керується підприємство при організації охорони праці та забезпеченні безпеки життєдіяльності своїх працівників є Конституція України, Кодекс законів про працю та Закон України «Про охорону праці» [15]. Саме ці документи визначають основні положення щодо реалізації конституційного права працівників на охорону їх життя й здоров'я в процесі трудової діяльності, регулюють за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки і гігієни праці та виробничого середовища, встановлюють єдиний порядок організації охорони праці в Україні.

Згідно з законом України «Про охорону праці» служба охорони праці на досліджуваному підприємстві створюється для організації виконання правових, організаційно-технічних, санітарно-гігієнічних, соціально-економічних і лікувально-профілактичних заходів, спрямованих на запобігання нещасних випадків, професійних захворювань і аварій у процесі праці.

4.3 Санітарно-гігієнічні вимоги до організації та обладнання робочих місць з ПК

Під час проведення будь-яких робіт, де обробка отриманих даних здійснюється за допомогою комп'ютерів, потрібно дотримуватися гігієнічних

норм, правил і вимог техніки безпеки при роботі з персональним комп'ютером (ПК).

Користувачі персональних комп'ютерів мають бути забезпечені відповідними робочими місцями, які відповідатимуть гігієнічним нормам. Конструкції всіх елементів робочого місця та їх взаємного розташування повинні відповідати ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності (ДСТ 12.2.032-78, ДСТ 22.269-76, ДСТ 21.889-76). Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози.

Робочі місця з персональними комп'ютерами слід так розташовувати відносно світлових прорізів, щоб природне світло падало збоку, переважно зліва.

При розміщенні робочих столів з персональними комп'ютерами слід дотримуватись таких відстаней між їх бічними поверхнями – 1,2м, відстань від тильної поверхні однієї машини до екрана іншої – 2,5м.

Висота поверхні робочого столу з комп'ютером має регулюватися в межах 680-800мм, а ширині і глибина - забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Робочий стілець має бути підйомно-поворотним, регульованим за висотою, за кутом нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння, поверхня сидіння має бути плоскою, передній край -заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися. Крок регулювання елементів стільця має становити: для лінійних розмірів - 15-20мм, для кутових 2-5°. Зусилля регулювання має не перевищувати 20Н. Висота поверхні сидіння має регулюватися в межах 400...500мм, а ширина і глибина становити не менше ніж 400мм. Кут нахилу сидіння – до 15° вперед і до 5° назад. Висота спинки стільця має становити 300±20мм, ширина – не менше ніж 380мм, радіус кривизни горизонтальної площини – 400мм. Кут нахилу спинки має регулюватися в меж 1-30° від вертикального положення.

Відстань від спинки до переднього краю сидіння має регулюватися в межах 260-400мм. Для зниження статичного напруження м'язів верхніх кінцівок слід використовувати стаціонарні або змінні підлокітники завдовжки не менше ніж 250мм, завширшки 50-70мм, що регулюються за висотою над сидінням у межах 230-260мм і відстанню між підлокітниками в межах 350-500мм. Поверхня сидіння і спинки стільця має бути напівм'якою з нековзним, повітронепроникним покриттям, що легко чиститься і не електризується.

4.4 Перелік шкідливих та небезпечних факторів, які супроводжують роботу програміста

Мікроклімат робочого приміщення - це клімат внутрішнього середовища цього приміщення, який визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря.

Принцип нормування мікроклімату:

- Створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем;
- Створення оптимального освітлення робочого місця;
- Зменшення впливу електромагнітних випромінювань і електричного поля;
- Зменшення сукупності факторів появи шуму і вібрації.

4.5 Висновки

Таким чином, дотримання описаних вище умов допоможе зберегти здоров'є, покращити працездатність протягом робочого дня, що в свою чергу сприятиме створенню програмних рішень.

Висновки

Ефективним кроком у підвищенні ефективності підприємства за допомогою структуризації інформації, послуг та контактних даних, для відображення їх на Інтернет-ресурсі, запобігає зменшенню витрат на неефективні способи представлення діяльності підприємства в інформаційному просторі, тим самим підвищує рівень залучання та опрацювання клієнтів, продуктивності та рентабельності підприємства, як кінцевий результат створення сайту-візитки.

Для досягнення цієї мети були вирішені наступні задачі:

- виконаний аналіз предметної області звернувши увагу на проблеми представлення діяльності підприємств в інформаційному просторі;
- розроблене технічне завдання;
- розроблені проектні рішення;
- розроблена ІС підприємства;
- проведені необхідні дослідження з охорони праці.

При виконанні аналізу предметної області та зверненні уваги на проблеми представлення діяльності підприємств в інформаційному просторі, дозволили створити конкурентноспроможний програмний продукт, який виявився потрібним рішення підприємству.

У результаті дослідження були опрацьовані необхідні технології для розробки інформаційної системи, використані сучасні веб-технології React.js, Next.js та Contentful, які дозволили створити інтерактивний веб-додаток з реалізацією SSR. В свою чергу це дозволило користуватися веб-додаток, який швидко завантажується, плавно працює та має значні показники в пошуковій видачі.

Розроблений сайт-візитка відповідає технічному завданню і проходить дослідну експлуатацію у користуванні.

Перелік використаних джерел

1. MSG equipment: види автосервісів. // Режим доступу:
<https://servicems.ru/news/post/252-vidy-avtoservisov.html>
2. Документація та опис React бібліотеки для побудови інтерфейсів користувача. // Режим доступу: <https://reactjs.org/>
3. Документація Next.js. // Режим доступу: <https://nextjs.org/docs>
4. Чи є Contentful альтернативою WordPress. // Режим доступу:
<https://www.contentful.com/blog/2020/03/26/contentful-wordpress-alternative/>
5. W3C валідатор HTML коду. // Режим доступу: <https://validator.w3.org/>
6. Валідатор CSS коду. // Режим доступу: <https://jigsaw.w3.org/css-validator/>
7. Сервіс PageSpeed Insights API. // Режим доступу: <https://pagespeed.web.dev/>
8. Веб-сайт атосервісу конкурента «Автоклініка». // Режим доступу:
<https://avtoklinika.com.ua/>
9. Веб-сайт атосервісу конкурента «У Сергія». // Режим доступу: <https://car-service-nikolaev.business.site/>
10. Сервіси перевірки швидкості завантаження сайту. // Режим доступу:
<https://www.plerdy.com/ua/blog/10-servisov-skorosti-sayta/>
11. Опис хто такі frontend- і backend-розробники. // Режим доступу:
<https://te.itstep.org/blog/frontend-and-backend-developers>
12. Порівняння в кількості завантажень найпопулярніших бібліотек на створення користувацького інтерфейсу. // Режим доступу:
<https://www.npmtrends.com/angular-vs-react-vs-vue>
13. Stackoverflow. // Режим доступу: <https://stackoverflow.com/>
14. Опис переваг та потреби використання SSR з React.js. // Режим доступу:
<https://asperbrothers.com/blog/server-side-rendering-in-react>
15. Закон України «Про охорону праці».
// Режим доступу: <http://zakon2.rada.gov.ua/laws/show/2694-12>

Додаток А – Розробка технічного завдання

Ціллю роботи є створення сучасного та конкурентоспроможного методу просування підприємства, залучення нових клієнтів, розміщення інформації про підприємство в Інтернет-просторі. Необхідно створити сайт-візитку та оптимізувати процес подальшої підтримки веб-сайту і взаємодію з користувачами цього програмного продукту, розробити оптимальну програмну реалізацію ІС підприємства.

До функціональних вимог відносяться:

- Відображення інформації про підприємство;
- Адаптивність веб-сайту для комфортного перегляду на різних пристроях;
- Висока швидкість завантаження сайту в межах не більше ніж 2-3 секунди;
- Управління всім контентом за допомогою адмін-панелі.

Для більш детального визначення функцій ІС сервісного підприємства визначаємо базовий набір надходжень та відправлень інформації.

Основними вхідними даними підприємства є:

- Інформація про послуги;
- Інформація про графік роботи;
- Контактні дані;
- Корисні статті з теми послуг автосервісу.

Висновки

Фактори знаходження нових клієнтів та витрати ресурсів на це є одними з найважливіших показників в системі незалежного автосервісу. Дана реалізація дозволяє покращити репутаційний чинник, суттєво забезпечити конкуренту спроможність підприємства на рівні населеного пункту України. Врешті решт призведе до збільшення потоку клієнтів та дозволить більше автомобілів обробляти за робочий час, максимально заповнюючи робочий графік клієнтами.

Впровадження сучасних систем просування підприємства дозволяє збільшувати потік клієнтів витрачаючи менше часу на знаходження та обробку кожного клієнта. Зменшувати кількість робочої сили на виконання певних ролей підприємства дозволяє суттєво зменшити зарплатні видатки, витрати на розхідники та збільшити дохід підприємства.

Додаток Б – Код програмних засобів

Головний файл налаштувань проекту

```
{
  "name": "car-service",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "bootstrap": "^5.1.3",
    "contentful": "^9.1.32",
    "next": "12.1.6",
    "react": "18.2.0",
    "react-dom": "18.2.0"
  },
  "devDependencies": {
    "eslint": "8.18.0",
    "eslint-config-next": "12.1.6",
    "sass": "^1.52.3"
  }
}
```

Файл конфігурацій Next.js

```
const nextConfig = {
  reactStrictMode: true,
}

module.exports = nextConfig
```

Головний файл вхідної точки додатку

```
import Head from 'next/head'

import 'bootstrap/dist/css/bootstrap.css'

import Layout from '../components/layout'

import '../styles/globals.css'
import '../styles/styles.scss'

function MyApp({ Component, pageProps }) {
  return (
    <>
      <Head>
        <meta name="viewport" content="width=device-width, initial-scale=1"
      />
```

```

    <meta property="og:title" content="Автосервіс Finsweet" key="title"
  />

  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin
/>

  <link
href="https://fonts.googleapis.com/css2?family=Manrope:wght@700&display=swap"
    rel="stylesheet"
  />
</Head>

  <Layout>
    <Component {...pageProps} />
  </Layout>
</>
)
}

export default MyApp

```

Файл головної сторінки

```

import Head from 'next/head'
import Link from 'next/link'
import Image from 'next/image'

import {
  MainContactData,
  Hero,
  Articles,
  Services,
  Diagnose,
  Offers,
} from '../services'
function Home({ mainContactData, hero, articlesInfo, articles,
servicesItems, servicesInfo, diagnose, offerInfo, offerItems }) {
  return (
    <>
      <Head>
        <title>Автосервіс Finsweet</title>
        <meta name="og:title" content="Автосервіс Finsweet" />
        <meta name="description" content="Автосервіс Finsweet в
Києві 1800 265 24 52. Обслуговуйте свій автомобіль в одному місці." />
        <meta name="og:site_name" content="Finsweet" />
        <link rel="icon" href="/favicon.ico" />
      </Head>

      <main>
        <section className="hero">
          <div className="hero__top-bar">
            <div className="container">
              <div className="row wrapper">
                <div className="col-12 col-md-5 inner">
                  <div className="left">
                    <div className="left__item">
                      <h1>{hero.data.title}</h1>
                    </div>
                    <div className="left__image">
                      <Image width={24} height={24}
objectFit='contain' src="/img/header-top/1.png" alt="" />
                    <div className="left__image-text">

```

[illegible]


```

        return (
            <div className="col-md-3 item"
key={item.fields.title}>
                <div className="top">
                    <picture className="offer__image-
wrapper">
                        <img
src={item.fields.icon.fields.file.url}
alt={item.fields.icon.fields.title}
title={item.fields.icon.fields.title}
                        />
                    </picture>
                    <p>{item.fields.title}</p>
                </div>
            </div>
        )
    }
}
</div>
</div>
</section>

<section className="section article">
    <div className="container">
        <div className="title">
            <h1>{articlesInfo.data.title}</h1>
        </div>
        <div className="row">
            {articles.map((item) => {
                return (
                    <div className="col-md-4 item"
key={item.fields.title}>
                        <Link
href={` /article/${item.fields.slug}`}>
                            <a className="top">
                                <picture className="article__img-
wrapper">
                                    <img
src={item.fields.image.fields.file.url}
alt={item.fields.image.fields.file.title}
title={item.fields.image.fields.file.title}
                                    />
                                </picture>
                                <div className="top__text">
                                    <h2>{item.fields.title}</h2>
                                <p>{item.fields.text.content[0].content[0].value.substr(0, 60)}...</p>
                                </div>
                            </a>
                        </Link>
                    </div>
                )
            })}
        </div>
    </div>
</section>
</main>

```

```

    </>
  )
}

Home.getInitialProps = async () => {
  const mainContactData = await MainContactData.getMainContactData()

  const hero = await Hero.getHero()

  const articlesInfo = await Articles.getArticleInfo()
  const articles = await Articles.getArticles()

  const servicesInfo = await Services.getServicesInfo()
  const servicesItems = await Services.getServicesItems()

  const diagnose = await Diagnose.getDiagnose()

  const offerInfo = await Offers.getOfferInfo()
  const offerItems = await Offers.getOfferItems()

  return {
    mainContactData,
    hero,
    articlesInfo,
    articles,
    servicesItems,
    servicesInfo,
    diagnose,
    offerInfo,
    offerItems,
  }
}

export default Home

```

Файл сторінки статті

```

import Head from 'next/head'
import { useRouter } from 'next/router'

import { Articles } from '../../services'

const Article = ({ content }) => {
  const router = useRouter()

  if (!content) {
    router.push('/')
  }

  return (
    <>
      <Head>
        <title>Finsweet - {content.title}</title>
      </Head>

      <section className="articles">
        <div className="articles__nav">
          <div className="container">
            <div className="title">
              <h1>{content.title}</h1>
            </div>

            <picture className="image">

```



```

        <img
          src={content.image.fields.file.url}
          alt={content.image.fields.file.title}
          title={content.image.fields.file.title}
        />
      </picture>

      <div className="text">
        {content.text.content.map((item, indexItem) => {
          return (
            item.content.map((el, indexEl) => {
              return (
                <p key={`_${indexItem} - ${indexEl}`}>{el.value}</p>
              )
            })
          )
        })}
      </div>
    </div>
  </div>
</section>
</>
)
}

Article.getInitialProps = async ({ query: { id } }) => {
  const { data } = await Articles.getArticle(id)

  return {
    content: data,
  }
}

export default Article

```

Файл головного опису будови сторінок

```

import Header from './header'
import Footer from './footer'

export default function Layout({ children }) {
  return (
    <>
      <Header />
      <main>{children}</main>
      <Footer />
    </>
  )
}

```

Компонент «шапки» додатку

```

import Link from 'next/link'
import Image from 'next/image'

import React, { useState, useEffect } from 'react'

import {
  MainContactData,
} from '../services'

```

```

function Header() {
  const [mainContactData, setMainContactData] = useState({})

  async function getMainContactData() {
    const mainContactData = await MainContactData.getMainContactData()

    setMainContactData(mainContactData)
  }

  useEffect(() => {
    getMainContactData()
  }, [])

  return (
    <>
      <header className="header">
        <div className="header__nav">
          <div className="container">
            <div className="header__nav_bar">
              <div className="header__nav_bar-left">
                <Link href="/">
                  <a><Image src="/img/info/Logo1.png"
alt="" width={120} height={30} /></a>
                  </Link>
                </div>

                <div className="header__nav_bar-right">
                  <Image src="/img/header/Icon.png" alt=""
width={36} height={36} />

                  <div className="header__nav_bar-
right__text">
                    <p>Допомога на дорозі</p>
                    <a href={`tel:${mainContactData.data &&
mainContactData.data.phone}`}>{mainContactData.data &&
mainContactData.data.phone}</a>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </header>
    </>
  )
}

export default Header

```

Компонент «підвалу» додатку

```

import Link from 'next/link'
import Image from 'next/image'

import React, { useState, useEffect } from 'react'

import {
  MainContactData,
} from '../services'

function Footer() {
  const [mainContactData, setMainContactData] = useState({})

  async function getMainContactData() {

```

```

const mainContactData = await MainContactData.getMainContactData()

setMainContactData(mainContactData)
}

useEffect(() => {
  getMainContactData()
}, [])

return (
  <>
    <footer className="section footer">
      <div className="container">
        <div className="row">
          <div className="col-md-6">
            <div className="left">
              <h1>{mainContactData.data &&
mainContactData.data.contactUsText}</h1>
            </div>
          </div>

          <div className="col-md-6">
            <div className="right">
              <div className="right_top">
                <p>Номер довідкової лінії:</p>
                <a href={`tel:${mainContactData.data &&
mainContactData.data.phone}`}>{mainContactData.data &&
mainContactData.data.phone}</a>
              </div>

              <div className="right_center">
                <p>Адреса:</p>
                <span>{mainContactData.data &&
mainContactData.data.adress}</span>
              </div>

              <div className="right_bottom">
                <p>Ми відкриті:</p>
                <span>{mainContactData.data &&
mainContactData.data.openingScheduleText}</span>
              </div>
            </div>
          </div>
        </div>
      </div>

      <div className="maps">
        <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d325515.6816598763!2d30
.25250249915429!3d50.40213674931967!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1
m2!1s0x40d4cf4ee15a4505%3A0x764931d2170146fe!2z0JrQuNC10LIsIDAYMDAw!5e0!3m2!1sru!
2sua!4v1654026663724!5m2!1sru!2sua" width={'100%'} height={450} style={{ border:
0}} loading="lazy" referrerPolicy="no-referrer-when-downgrade"></iframe>
        </div>

      <div className="container">
        <div className="info">
          <div className="info_image">
            <Link href="/">
              <a> <Image src="/img/info/Logo.png"
alt="fintsweet" width={120} height={30} /></a>
            </Link>
          </div>
        </div>
      </div>
    </div>
  </div>
)

```

```

        <div className="info_text">
            <p>© Copyright Finsweet {new
Date().getFullYear()}</p>
        </div>
    </div>
</div>
</footer>
</>
)
}

export default Footer

```

Головний файл експортування всіх запитів (сервісів)

```

import MainContactData from './mainContactData'
import Hero from './hero'
import Articles from './articles'
import Services from './services'
import Diagnose from './diagnose'
import Offers from './offers'

export {
    MainContactData,
    Hero,
    Articles,
    Services,
    Diagnose,
    Offers,
}

```

Файл запиту для отримання даних для головної секції

```

import * as contentful from 'contentful'

const createClient = () => {
    return contentful.createClient({
        space: 't0lyrtzf1pfw',
        accessToken: 'VjqeSP5uPF9Ftk-3cXmB7EdzrwcaWWPlDf0aoDD7bOU'
    })
}

async function getHero () {
    const client = createClient()

    const params = {
        'content_type': 'hero',
    }

    const data = await client.getEntries(params)

    const { items } = client.parseEntries(data)

    return {
        data: items[0].fields
    }
}

export default {

```

```
    getHero,  
  }  
}
```

Файл запиту для отримання даних для секції «Діагностика»

```
import * as contentful from 'contentful'  
  
const createClient = () => {  
  return contentful.createClient({  
    space: 't0lyrtzf1pfw',  
    accessToken: 'VjqeSP5uPF9Ftk-3cXmB7EdzrwcaWWPlDf0aoDD7bOU'  
  })  
}  
  
async function getDiagnose () {  
  const client = createClient()  
  
  const params = {  
    'content_type': 'diagnose',  
  }  
  
  const data = await client.getEntries(params)  
  
  const { items } = client.parseEntries(data)  
  
  return {  
    data: items[0].fields  
  }  
}  
  
export default {  
  getDiagnose,  
}
```

Файл запитів для отримання даних для статей

```
import * as contentful from 'contentful'  
  
const createClient = () => {  
  return contentful.createClient({  
    space: 't0lyrtzf1pfw',  
    accessToken: 'VjqeSP5uPF9Ftk-3cXmB7EdzrwcaWWPlDf0aoDD7bOU'  
  })  
}  
  
async function getArticleInfo () {  
  const client = createClient()  
  
  const params = {  
    'content_type': 'articleInfo',  
  }  
  
  const data = await client.getEntries(params)  
  
  const { items } = client.parseEntries(data)  
  
  return {  
    data: items[0].fields,  
  }  
}
```

```

async function getArticles () {
  const client = createClient()

  const params = {
    'content_type': 'article',
  }

  const data = await client.getEntries(params)

  const { items } = client.parseEntries(data)

  return items
}

async function getArticle (slug) {
  const client = createClient()

  const params = {
    'content_type': 'article',
    'fields.slug': slug
  }

  const data = await client.getEntries(params)

  const { items } = client.parseEntries(data)

  return {
    data: items[0].fields,
  }
}

export default {
  getArticleInfo,
  getArticles,
  getArticle,
}

```

Файл запиту для отримання даних про інформацію підприємства

```

import * as contentful from 'contentful'

const createClient = () => {
  return contentful.createClient({
    space: 't0lyrtzf1pfw',
    accessToken: 'VjqeSP5uPF9Ftk-3cXmB7EdzrwcaWWPlDf0aoDD7bOU'
  })
}

async function getMainContactData () {
  const client = createClient()

  const params = {
    'content_type': 'mainContactData',
  }

  const data = await client.getEntries(params)

  const { items } = client.parseEntries(data)

  return {

```

```

      data: items[0].fields
    }
  }

export default {
  getMainContactData,
}

```

Файл запитів для отримання даних для секції «Ми пропонуємо»

```

import * as contentful from 'contentful'

const createClient = () => {
  return contentful.createClient({
    space: 't0lyrtzf1pfw',
    accessToken: 'VjqeSP5uPF9Ftk-3cXmB7EdzrwcaWWPlDf0aoDD7bOU'
  })
}

async function getOfferInfo () {
  const client = createClient()

  const params = {
    'content_type': 'offer',
  }

  const data = await client.getEntries(params)

  const { items } = client.parseEntries(data)

  return {
    data: items[0].fields,
  }
}

async function getOfferItems () {
  const client = createClient()

  const params = {
    'content_type': 'offerItems',
  }

  const data = await client.getEntries(params)

  const { items } = client.parseEntries(data)

  return items
}

export default {
  getOfferInfo,
  getOfferItems,
}

```

Файл запитів для отримання даних для секції «Послуги»

```

import * as contentful from 'contentful'

const createClient = () => {

```

```

    return contentful.createClient({
      space: 't0lyrtzf1pfw',
      accessToken: 'VjqeSP5uPF9Ftk-3cXmB7EdzrwcaWWPlDf0aoDD7bOU'
    })
  }

  async function getServicesItems () {
    const client = createClient()

    const params = {
      'content_type': 'servicesItems',
    }

    const data = await client.getEntries(params)

    const { items } = client.parseEntries(data)

    return items
  }

  async function getServicesInfo () {
    const client = createClient()

    const params = {
      'content_type': 'servicesInfo',
    }

    const data = await client.getEntries(params)

    const { items } = client.parseEntries(data)

    return {
      data: items[0].fields,
    }
  }

  export default {
    getServicesItems,
    getServicesInfo,
  }

```

Глобальні стилі додатку

```

html,
body {
  padding: 0;
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, Segoe UI, Roboto, Oxygen,
    Ubuntu, Cantarell, Fira Sans, Droid Sans, Helvetica Neue, sans-serif;
}

a {
  color: inherit;
  text-decoration: none;
}

* {
  box-sizing: border-box;
}

```


Стили для сторінок та компонентів додатку

```
body {
  font-family: 'Manrope', sans-serif;
  background: #ffffff;;
  color: #1E1B1B;
}

picture {
  display: block;

  img {
    max-width: 100%;
    width: 100%;
    max-height: 100%;
    object-fit: cover;
  }
}

.section {
  padding-top: 80px;
  padding-bottom: 80px;
}

// 1 Section
.header {
  &__nav {
    &__bar {
      display: flex;
      justify-content: space-between;
      padding-top: 50px;
      padding-bottom: 50px;
      &-right {
        display: flex;
        &__text {
          line-height: 10px;
          padding-left: 10px;
          p {
            font-size: 14px;
            opacity: 0.7;
          }
          a {
            font-size: 14px;
            text-decoration: none;
            color: #1E1B1B;
          }
        }
      }
    }
  }
}

.hero {
  &__top {
    &-bar {
      padding-bottom: 80px;
      .left {
        padding-top: 55px;
        h1 {
          font-size: 40px;
        }
        &__image {
          display: flex;
          padding-top: 90px;
        }
      }
    }
  }
}
```

```

        &-text {
            padding-left: 10px;
            line-height: 5px;
            p {
                opacity: 0.7;
            }
            span {
                font-size: 11px;
            }
        }
    }
}

.right {
    display: flex;
    align-items: center;
    position: relative;
    height: 475px;

    &__img-wrapper {
        position: relative;
        width: 100%;
        height: 100%;

        img {
            width: 100%;
            height: 100%;
        }
    }
}

&__top {
    top: 50px;
    left: -42px;
    background-color: #ffffff;
    border-radius: 100px;
    justify-content: center;
    width: 300px;
    padding: 12px;
    display: flex;
    position: absolute;
    box-shadow: 1px 1px 15px 1px gray;
    h2 {
        font-size: 16px;
        padding-top: 27px;
        padding-left: 10px;
    }
}

&__lower {
    display: flex;
    position: absolute;
    background-color: #ffffff;
    border-radius: 100px;
    justify-content: center;
    width: 300px;
    padding: 12px;
    top: 334px;
    left: 196px;
    box-shadow: 1px 1px 15px 1px gray;
    &-text {
        line-height: 10px;
        h2 {
            font-size: x-large;
        }
        p {
            opacity: 0.7;

```

```

        }
    }
    &-image {
        padding-top: 10px;
        padding-left: 10px;
    }
}

}

}

}
}
// 2 Section
.services {
    background-color: #E2E6E9;
    // .item {
    //     display: contents;
    // }
    .title {
        padding-bottom: 80px;
        h1 {
            font-size: 55px;
            text-align: center;
        }
    }
    .expert {
        display: flex;
        &__text {
            padding-left: 10px;
            h2 {
                font-size: 24px;
            }
            p {
                font-size: 14px;
                opacity: 0.7;
            }
        }

        &__image {
            position: relative;
            width: 77px;
            height: 38px;
        }
    }
}
// 3 Section
.diagnose {
    .image {
        display: flex;
        justify-content: space-between;
        height: 360px;

        &-wrapper {
            position: relative;
            width: calc(50% - 5px);
            height: 100%;

            img {
                height: 100%;
            }
        }

        &__right {
            padding-left: 10px;

```

```

    }
  }
  .text {
    h1 {
      font-size: 35px;
    }
    p {
      font-size: 14px;
      opacity: 0.7;
      padding-top: 60px;
    }
  }
}
// 4 Section
.offer {
  background-color: #F2F2F2;
  .title {
    text-align: center;
    padding-bottom: 60px;
    h2 {
      font-size: 24px;
    }
    h1 {
      font-size: 35px;
    }
  }
  .item {
    margin-top: 20px;
  }
  .top {
    text-align: center;
    background: #ffffff;
    padding: 40px;
    height: 100%;

    &:hover {
      background: #EAE9E9;
    }
    p {
      font-size: 15px;
      padding-top: 5%;
      margin-bottom: 0;
    }
  }
}

&__image-wrapper {
  margin: 0 auto;
  width: 50px;
  height: 50px;

  img {
    max-width: 100%;
  }
}

.bottom {
  margin-top: 20px;
  text-align: center;
  background: #ffffff;
  padding: 40px;
  height: 100%;
  &:hover {
    background: #EAE9E9;
  }
}

```

```

    }
    p {
        font-size: 15px;
        padding-top: 5%;
    }
}

// 5 Section
.article {
    &__img-wrapper {
        position: relative;
        width: 100%;
        height: 140px;
    }

    .item {
        margin-top: 50px;
    }

    .title {
        text-align: center;
        padding-bottom: 60px;
        h1 {
            font-size: 55px;
        }
    }

    .top {
        text-decoration: none;
        &:hover h2 {
            text-decoration: underline;
        }
        img {
            max-width: 100%;
        }
        &__text {
            padding-top: 20px;
            h2 {
                font-size: 17px;
                color: #000;
                text-decoration: none;
            }
            p {
                color: #000;
                text-decoration: none;
                padding-top: 10px;
                font-size: 15px;
                opacity: 0.7;
            }
        }
    }
}

// 8 Section
.footer {
    background-color: #1E1B1B;
    color: #ffffff;
    padding-bottom: 40px;
    .left {
        h1 {
            padding-bottom: 50px;
        }
        &__image {
            a {
                img {
                    &:hover {

```

```
background:#1E1B1B;
opacity: 50%;

    }
padding: 10px;
}
}
}
}
.right {
padding-left: 50px;
p {
opacity: 0.7;
}
a {
font-size: 22px;
text-decoration: none;
color: #ffffff;
}
span {
font-size: 15px;
}
&__top {
padding-top: 20px;
line-height: 10px;
}
&__center {
padding-top: 50px;
line-height: 10px;
}
&__bottom {
padding-top: 50px;
line-height: 10px;
}
}
.maps {
padding-top: 60px;
padding-bottom: 40px;
}
.info {
display: flex;
justify-content: space-between;
&__text {
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
p {
margin-top: 0;
margin-bottom: 0;
align-items: center;
opacity: 0.7;
}
}
}
}
// ARTICLES
.articles {
padding-bottom: 60px;
&__nav {
.title {
max-width: 600px;
```

```

        margin-left: auto;
        margin-right: auto;
        padding-top: 80px;
        h1 {
            text-align: center;
            font-size: 32px;
        }
    }
    .image {
        position: relative;
        margin-top: 50px;
        margin-bottom: 50px;
        width: 100%;

        img {
            max-width: 100%;
            max-height: 400px;
        }
    }
    .text {
        max-width: 600px;
        margin-left: auto;
        margin-right: auto;
        h2 {
            font-size: 30px;
        }
        p {
            font-size: 15px;
            opacity: 0.7;
        }
    }
}

@media (max-width: 1196px) {
    .hero {
        &__top {
            &-bar {
                .right {
                    &__top {
                        left: 10px;
                    }
                    &__lower {
                        left: 140px;
                    }
                }
            }
        }
    }
    .diagnose {
        .item {
            .image {
                height: 100%;

                &__left {
                    img {
                        max-width: 100%;
                    }
                }
                &__right {
                    margin-left: auto;

                    img {
                        max-width: 100%;
                    }
                }
            }
        }
    }
}

```

```

    }
  }
}

@media (max-width: 992px) {
  .section {
    padding-top: 40px;
    padding-bottom: 40px;
  }
  .hero {
    &__top {
      &-bar {
        padding-left: 0px;
        .left {
          left: 136px;
          margin-left: 0px;
          h1 {
            font-size: 30px;
          }
          &__image {
            padding-top: 20px;
          }
        }
        .right {
          img {
            max-width: 100%;
          }
          &__top {
            width: 250px;
            padding: 6px;

            left: 8px;
            h2 {
              padding-top: 15px;
            }
          }
          &__lower {
            left: 142px;
            width: 250px;
            padding: 6px;
          }
        }
      }
    }
  }
}

.header {
  &__nav {
    &__bar {
      &-right {
        &__text {
          line-height: 5px;
          padding-left: 10px;
          padding-top: 5px;
        }
      }
    }
  }
  &__top {
    &-bar {
      padding-left: 0px;
      .left {
        margin-left: 0px;
        h1 {

```



```

        font-size: 30px;
    }
    &__image {
        padding-top: 20px;
    }
}
.right {
    img {
        max-width: 100%;
    }
    &__top {
        width: 250px;
        padding: 6px;
        h2 {
            padding-top: 15px;
        }
    }
    &__lower {
        left: 142px;
        width: 250px;
        padding: 6px;
    }
}
}
}
}
.services {
    padding-top: 20px;
    .title {
        padding-bottom: 10px;
        h1 {
            font-size: 20px;
        }
    }
    .expert {
        &__text {
            h2 {
                font-size: 12px;
            }
            p {
                font-size: 10px;
            }
        }
        &__image {
            width: 50px;
            height: 32px;
        }
    }
}
.diagnose {
    .item {
        .text {
            padding-top: 20px;
            h1 {
                text-align: center;
                font-size: 20px;
            }
            p {
                font-size: 10px;
                padding-top: 0px;
            }
        }
    }
}

```

```

    }
}
.offer {
  .title {
    padding-bottom: 0px;
    h2 {
      font-size: 15px;
    }
    h1 {
      font-size: 20px;
    }
  }
  .item {
    .top {
      p {
        font-size: 12px;
      }
    }
  }
}
}
.article {
  .title {
    padding-bottom: 20px;
    h1 {
      font-size: 20px;
    }
  }
  .item {
    text-align: center;
    margin-top: 0px;
    .top {
      &__text {
        padding-top: 10px;
        padding-bottom: 40px;
        h2 {
          font-size: 15px;
        }
        p {
          padding-top: 0px;
          font-size: 10px;
        }
      }
    }
  }
}
}
}
.reviews {
  .title {
    padding-bottom: 20px;
    h1 {
      font-size: 20px;
    }
  }
}
&-slider {
  margin-top: 10px;
  &__block {
    width: 250px;
    padding: 0px;
    p {
      font-size: 10px;
      opacity: 0.7;
      padding-left: 10px;
      padding-right: 10px;
    }
  }
  .image {

```

```
padding: 10px;
&__text {
    padding-left: 10px;
    h2 {
        font-size: 15px;
    }
    p {
        margin-left: -10px;
        font-size: 10px;
        opacity: 0.7;
    }
}
}

}

.footer {
    .left {
        h1 {
            font-size: 20px;
            padding-bottom: 0px;
            text-align: center;
        }
        &__image {
            text-align: center;
        }
    }
    .right {
        padding-left: 0px;
        text-align: center;
        &__top {
            padding-top: 20px;
            line-height: 5px;
            p {
                font-size: 15px;
            }
            a {
                font-size: 14px;
            }
        }
        &__center {
            padding-top: 20px;
            line-height: 5px;
            p {
                font-size: 15px;
            }
            span {
                font-size: 14px;
            }
        }
        &__bottom {
            padding-top: 20px;
            line-height: 5px;
            p {
                font-size: 15px;
            }
            span {
                font-size: 14px;
                line-height: 12px;
            }
        }
    }
}
```

```

.info {
    font-size: 10px;
}
.articles {
    &__nav {
        .title {
            max-width: 600px;
            margin-left: auto;
            margin-right: auto;
            padding-top: 10px;
            padding-bottom: 50px;
            text-align: center;
            h1 {
                text-align: center;
                font-size: 32px;
            }
        }
        .image {
            img {
                max-width: 100%;
            }
        }
        .text {
            padding-top: 10px;
            max-width: 600px;
            margin-left: auto;
            margin-right: auto;
            p {
                padding-top: 5px;
                font-size: 15px;
                opacity: 0.7;
            }
        }
    }
}
}
@media (max-width: 768px) {
    .section {
        padding-top: 40px;
        padding-bottom: 40px;
    }
    .hero {
        &__top {
            &-bar {
                padding-bottom: 0;
            }
            .wrapper {
                position: relative;
                .inner {
                    display: flex;
                    flex-direction: column;
                    align-items: center;
                    justify-content: center;
                    position: absolute;
                    z-index: 1;
                    color: #ffffff;
                    background: rgba(0, 0, 0, 0.6);
                    height: 100%;
                    padding: 20px;
                    left: 0;
                    top: 0;
                    .left {
                        display: flex;
                        flex-direction: column;
                        align-items: center;

```

[illegible]

```

        }
        p {
            font-size: 15px;
        }
    }
}

@media (max-width: 560px) {
    .section {
        padding-top: 40px;
        padding-bottom: 40px;
    }
    .header {
        &__nav {
            &__bar {
                &-right {
                    &__text {
                        line-height: 5px;
                        padding-left: 10px;
                        padding-top: 5px;
                    }
                }
            }
        }
    }
    .services {
        padding-top: 20px;
        .title {
            padding-bottom: 10px;
            h1 {
                font-size: 20px;
            }
        }
        .expert {
            padding-top: 20px;
            padding-bottom: 20px;
            &__text {
                h2 {
                    font-size: 15px;
                }
                p {
                    font-size: 10px;
                }
            }
        }
    }
    .diagnose {
        .item {
            .text {
                padding-top: 20px;
                h1 {
                    text-align: center;
                    font-size: 20px;
                }
                p {
                    font-size: 10px;
                    padding-top: 0px;
                }
            }
        }
    }
}

```

```

.offer {
  .title {
    padding-bottom: 0px;
    h2 {
      font-size: 15px;
    }
    h1 {
      font-size: 20px;
    }
  }
  .item {
    .top {
      p {
        font-size: 12px;
      }
    }
  }
}

.article {
  .title {
    padding-bottom: 20px;
    h1 {
      font-size: 20px;
    }
  }
  .item {
    text-align: center;
    margin-top: 0px;
    .top {
      &__text {
        padding-top: 10px;
        padding-bottom: 40px;
        h2 {
          font-size: 15px;
        }
        p {
          padding-top: 0px;
          font-size: 10px;
        }
      }
    }
  }
}

}

.reviews {
  background-color: #F2F2F2;
  .title {
    h1 {
      font-size: 45px;
      text-align: center;
    }
  }
}

&-slider {
  margin-top: 100px;
  &__block {
    padding: 20px 50px;
    background-color: #ffffff;
    width: 420px;
    text-align: left;
    opacity: 0.4;
    transform: scale(0.85);
    transition: all 300ms ease;
    &.slick-center {
      opacity: 1;
      transform: scale(1);
    }
  }
}

```

```

    }

    .image {
      display: flex;
      padding-bottom: 20px;
      &__text {
        padding-left: 12px;
        h2 {
          font-size: 18px;
        }
        p {
          opacity: 0.7;
        }
      }
    }
    p {
      font-size: 15px;
    }
  }
}

.footer {
  .left {
    h1 {
      font-size: 20px;
      padding-bottom: 0px;
      text-align: center;
    }
    &__image {
      text-align: center;
    }
  }
  .right {
    padding-left: 0px;
    text-align: center;
    &__top {
      padding-top: 20px;
      line-height: 5px;
      p {
        font-size: 15px;
      }
      a {
        font-size: 14px;
      }
    }
    &__center {
      padding-top: 20px;
      line-height: 5px;
      p {
        font-size: 15px;
      }
      span {
        font-size: 14px;
      }
    }
  }
  &__bottom {
    padding-top: 20px;
    line-height: 5px;
    p {
      font-size: 15px;
    }
    span {
      font-size: 14px;
      line-height: 12px;
    }
  }
}

```



```

    }
  }
}
.info {
  font-size: 10px;
}
.articles {
  &__nav {
    .title {
      max-width: 600px;
      margin-left: auto;
      margin-right: auto;
      padding-top: 10px;
      h1 {
        text-align: center;
        font-size: 20px;
      }
    }
    .image {
      img {
        max-width: 100%;
      }
    }
    .text {
      max-width: 600px;
      margin-left: auto;
      margin-right: auto;
      p {
        padding-top: 5px;
        font-size: 12px;
        opacity: 0.7;
      }
    }
  }
}
}
.hero {
  &__top {
    &-bar {
      .wrapper {
        position: relative;
        .inner {
          display: flex;
          flex-direction: column;
          align-items: center;
          justify-content: center;
          position: absolute;
          z-index: 1;
          color: #ffffff;
          background: rgba(0, 0, 0, 0.6);
          height: 100%;
          padding: 20px;
          left: 0;
          top: 0;
          .left {
            display: flex;
            flex-direction: column;
            align-items: center;
            text-align: center;
            h1 {
              text-align: center;
              font-size: 30px;
            }
          }
          &__image {

```

[illegible]

```

        text-align:left;
    }
}
}
.winner {
padding: 0;
.right {
img {
width: 100%;
}
justify-content: center;
&__top {
display: none;
}
&__lower {
display: none;
}
}
}
}
}
}
}
}
.services {
padding-top: 40px;
.title {
padding-bottom: 10px;
h1 {
font-size: 15px;
}
}
.expert {
padding-top: 20px;
padding-bottom: 20px;
&__text {
h2 {
font-size: 12px;
}
p {
font-size: 10px;
}
}
}
}
}
.diagnose {
.item {
.image {
&__left {
img {
width: 400px;
}
}
&__right {
display: none;
}
}
.text {
padding-top: 15px;
h1 {

text-align: center;
font-size: 15px;
}
p {
```



```

    &__top {
        padding-top: 20px;
        line-height: 5px;
        p {
            font-size: 10px;
        }
        a {
            font-size: 10px;
        }
    }
    &__center {
        padding-top: 20px;
        line-height: 5px;
        p {
            font-size: 10px;
        }
        span {
            font-size: 10px;
        }
    }
    &__bottom {
        padding-top: 20px;
        line-height: 5px;
        p {
            font-size: 10px;
        }
        span {
            font-size: 10px;
            line-height: 12px;
        }
    }
}
.info {
    font-size: 10px;
}
}
.articles {
    padding-bottom: 50px;
    &__nav {
        .title {
            max-width: 600px;
            margin-left: auto;
            margin-right: auto;
            h1 {
                text-align: center;
                font-size: 15px;
            }
        }
        .image {
            img {
                max-width: 100%;
            }
        }
        .text {
            max-width: 600px;
            margin-left: auto;
            margin-right: auto;
            p {
                font-size: 10px;
                opacity: 0.7;
            }
        }
    }
}
}

```

```

.reviews {
  background-color: #F2F2F2;
  .title {
    h1 {
      font-size: 45px;
      text-align: center;
    }
  }
  &-slider {
    margin-top: 100px;
    &__block {
      padding: 20px 50px;
      background-color: #ffffff;
      width: 420px;
      text-align: left;
      opacity: 0.4;
      transform: scale(0.85);
      transition: all 300ms ease;
      &.slick-center {
        opacity: 1;
        transform: scale(1);
      }
    }
    .image {
      display: flex;
      padding-bottom: 20px;
      &__text {
        padding-left: 12px;
        h2 {
          font-size: 18px;
        }
        p {
          opacity: 0.7;
        }
      }
    }
  }
  p {
    font-size: 15px;
  }
}

```

Файл налаштування для роботи з кодом

```

{
  "extends": "next/core-web-vitals"
}

```