

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ

Навчально-науковий інститут денної освіти

Форма навчання денна

Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА

(підпис)

«___» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

**«РОЗРОБКА ВЕБ-САЙТУ АВТОСАЛОНУ З ВИКОРИСТАННЯМ
HTML, CSS ТА JAVASCRIPT»**

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи Лукаш Денис Ігорович

_____ «___» _____ 202_ р.
(підпис)

Науковий керівник к. ф.-м. н., доц., Ольховська Олена Володимирівна

_____ «___» _____ 202_ р.
(підпис)

Рецензент _____

ПОЛТАВА 2024

РЕФЕРАТ

Записка: 74 с., 15 рис., 2 додатки, 18 джерел.

ВЕБ-САЙТ АВТОСАЛОНУ З ВИКОРИСТАННЯМ HTML, CSS ТА JAVASCRIPT.

Об'єкт розробки – веб-сайт для автосалону, який забезпечує користувачам зручний доступ до повної інформації про доступні автомобілі, включаючи їхні характеристики, фотографії та ціни.

Мета роботи – розробка веб-сайту для автосалону, який надаватиме повну інформацію про доступні автомобілі, включаючи їхні характеристики, фотографії та ціни.

Методи дослідження – аналіз літератури, порівняльний аналіз веб-сайтів автосалонів, проектування макету, розробка за допомогою HTML, CSS і JavaScript, тестування та оцінка ефективності.

Структура роботи – робота складається з наступних розділів: вступ – обґрунтування вибору теми, мета та завдання дослідження; аналіз існуючих рішень – огляд та аналіз веб-сайтів автосалонів; проектування веб-сайту – розробка макету, структури та дизайну; реалізація веб-сайту – використані технології та реалізація функціоналу; Тестування та оптимізація – методи тестування, виявлення помилок та оптимізація; висновки – підсумки, оцінка результатів та рекомендації.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	5
ВСТУП.....	7
1. ПОСТАНОВКА ЗАДАЧІ	10
1.1 Постановка задачі проект веб-сайт автосалону.....	10
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	11
2.1 Огляд історії розвитку веб-технологій.....	11
2.2 Важливість веб-сайтів для бізнесу	13
2.3 Основні засоби розробки веб-сайтів	15
3. ТЕОРЕТИЧНИЙ ЧАСТИНА	18
3.1 Аналіз сучасних тенденцій в автомобільній індустрії	18
3.2 Порівняльний аналіз існуючих сайтів автосалонів	20
3.3 Вибір технології розробки сайту автосалону	25
4. ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	29
4.1 Проектування веб-сайту	29
4.1.1 Визначення функціональних вимог до сайту	29
4.1.2 Розробка дизайну та інтерфейсу користувача (UI/UX).....	33
4.2 Програмна реалізація веб-сайту.....	36
4.2.1 Розробка фронтенду з використанням HTML та CSS.....	36
4.3 Використання JavaScript для створення інтерактивних елементів ..	40
4.3.1 Реалізація форм для замовлення автомобілів або запису на тест-драйв	48
4.3.2 Оптимізація сайту для роботи на різних пристроях (респонсивний дизайн).....	48
4.4 Тестування та валідація	50
4.4.1 Проведення тестування сайту на різних браузерах.....	50
4.4.2 Валідація HTML та CSS коду	52
4.5 Запуск сайту на сервері MongoDB.....	55
4.6 Аналіз можливих покращень	58
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63

ДОДАТОК А.....	65
ДАДАТОК Б	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
HTML	HyperText Markup Language - стандартна мова розмітки для створення веб-сторінок.
CSS	Cascading Style Sheets - мова стилів, яка використовується для опису вигляду документа, написаного мовою розмітки HTML.
JavaScript	Мова програмування для створення динамічних веб-сторінок і веб-додатків.
React	JavaScript бібліотека для створення компонентів користувацького інтерфейсу.
jQuery	JavaScript бібліотека, яка спрощує роботу з HTML-документами, обробку подій, анімацію та Ajax.
Vue.js	JavaScript фреймворк для створення односторінкових додатків.
Python	Мова програмування, яка часто використовується для веб-розробки, аналізу даних, автоматизації тощо.
MySQL	Реляційна система керування базами даних з відкритим кодом.
MongoDB	Нереляційна база даних, що використовує документоорієнтовану модель даних.

WordPress	Система керування контентом з відкритим кодом, яка часто використовується для створення веб-сайтів та блогів.
Joomla	Система керування контентом з відкритим кодом, яка дозволяє створювати потужні онлайн-додатки.
Visual Studio Code	Редактор коду з підтримкою багатьох мов програмування і розширень.
UI/UX	Інтерфейс користувача/досвід користувача - дизайн інтерфейсу і взаємодія користувача з продуктом.
Adobe Photoshop/Illustrator	Графічні редактори для створення і обробки зображень, ілюстрацій та графіки.
Frontend	Клієнтська частина веб-розробки, що включає все, що користувач бачить і з чим взаємодіє на веб-сайті.
Backend	Серверна частина веб-розробки, що включає логіку, бази даних і сервери.
URL	Уніфікований вказівник ресурсів - адреса ресурсу в Інтернеті.
WebP	Формат зображень для веб, що забезпечує високу якість при меншому розмірі файлу.
CDN	Мережа доставки контенту - система серверів, що розподіляє доставку веб-ресурсів користувачам.
Webpack	Інструмент для зборки модулів JavaScript і управління залежностями.
SEO	Оптимізація для пошукових систем - набір заходів для покращення позицій веб-сайту в результатах пошуку.

ВСТУП

З розвитком інформаційних технологій та Інтернету все більше компаній переходять до цифрових методів ведення бізнесу. Веб-сайти стають основним засобом комунікації з клієнтами, надання інформації про продукти та послуги, а також проведення онлайн-транзакцій. Автосалони не є винятком і потребують сучасних веб-ресурсів для ефективної взаємодії з потенційними покупцями. Розробка веб-сайту автосалону з використанням HTML, CSS та JavaScript дозволить створити інтерактивну та зручну платформу для представлення асортименту автомобілів, їхніх характеристик та цін, що підвищить інтерес клієнтів та сприятиме збільшенню продажів.

Метою даної роботи є розробка веб-сайту для автосалону, який надаватиме повну інформацію про доступні автомобілі, включаючи їхні характеристики, фотографії та ціни. Для досягнення цієї мети передбачається виконання наступних завдань:

- аналіз існуючих веб-сайтів автосалонів для визначення кращих практик та необхідних функціональних можливостей;
- Розробка макету та дизайну веб-сайту з урахуванням вимог зручності користування.
- Реалізація веб-сайту за допомогою HTML для структурування контенту, CSS для стилізації та JavaScript для забезпечення інтерактивності.
- Тестування веб-сайту для виявлення та усунення можливих помилок.
- Оцінка ефективності та зручності веб-сайту для користувачів.

Предметом дослідження є розробка веб-сайту для автосалону, що дозволяє представити автомобілі, їхні характеристики, фотографії та ціни. Сайт має бути зручним для користувачів, інтуїтивно зрозумілим та функціональним, забезпечуючи повну інформацію про пропоновані автомобілі.

Об'єктом дослідження є процес створення веб-сайту для автосалону з використанням HTML, CSS та JavaScript. Це включає аналіз існуючих веб-

сайтів, розробку макету та дизайну, реалізацію структури та функціоналу, а також тестування та оптимізацію сайту.

Методи дослідження:

- Аналіз літератури: вивчення існуючих рішень та підходів до розробки веб-сайтів автосалонів.
- Порівняльний аналіз: аналіз конкурентних веб-сайтів для визначення кращих практик та необхідних функцій.
- Проектування: розробка макету та дизайну веб-сайту з урахуванням вимог зручності користування.
- Реалізація: створення веб-сайту за допомогою HTML для структурування контенту, CSS для стилізації та JavaScript для забезпечення інтерактивності.
- Тестування: виявлення та усунення можливих помилок, оптимізація продуктивності веб-сайту.
- Оцінка ефективності: оцінка зручності та функціональності веб-сайту для користувачів.

На сьогоднішній день важливість наявності веб-сайту для будь-якого бізнесу, включаючи автосалони, важко переоцінити. Веб-сайт є не лише засобом представлення товарів та послуг, але й інструментом для залучення нових клієнтів та підтримки зв'язку з постійними покупцями. Враховуючи зростаючу конкуренцію в сфері продажу автомобілів, наявність сучасного та функціонального веб-сайту стає необхідністю для успішного ведення бізнесу.

Робота складається з наступних розділів: вступ – обґрунтування вибору теми, мета та завдання дослідження, актуальність проблеми, структура роботи; аналіз існуючих рішень – огляд веб-сайтів автосалонів, аналіз їхніх переваг та недоліків; проектування веб-сайту – розробка макету, структури та дизайну веб-сайту; реалізація веб-сайту – опис технологій та інструментів, використаних для створення веб-сайту, реалізація функціоналу за допомогою HTML, CSS та JavaScript; тестування та оптимізація – методи тестування веб-сайту, виявлення та виправлення помилок, оптимізація продуктивності. У

висновках – підведення підсумків роботи, оцінка досягнутих результатів, рекомендації щодо подальшого розвитку веб-сайту.

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка задачі проект веб-сайт автосалону

Розробка веб-сайту автосалону передбачає виконання наступних задач:

- Збір вимог – визначення основних функцій та інформаційних блоків, які повинні бути присутніми на веб-сайті автосалону.
- Створення структури веб-сайту – розробка ієрархії сторінок та навігаційного меню.
- Дизайн інтерфейсу користувача – створення макетів сторінок, вибір колірної гами, шрифтів та графічних елементів.
- Розробка HTML-коду – створення HTML-сторінок для структурування контенту.
- Стилiзація за допомогою CSS – застосування стилів для покращення зовнішнього вигляду веб-сайту.
- Інтерактивність за допомогою JavaScript – реалізація динамічних елементів, таких як слайдери, форми зворотного зв'язку, пошук по сайту тощо.
- Тестування веб-сайту – перевірка функціональності та зручності користування, виявлення та усунення помилок.
- Оптимізація продуктивності – зменшення часу завантаження сторінок, оптимізація зображень та коду.

Розробка веб-сайту автосалону з використанням HTML, CSS та JavaScript дозволить створити сучасний, функціональний та привабливий веб-ресурс, який буде сприяти залученню нових клієнтів та підвищенню рівня продажів автомобілів.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд історії розвитку веб-технологій

Розвиток веб-технологій пройшов через кілька етапів [2-3], кожен з яких приніс нові можливості та інструменти для створення веб-сторінок. Нижче наведено короткий огляд ключових етапів розвитку веб-технологій.

Історія веб-технологій розпочалася з моменту, коли Тім Бернерс-Лі, науковець Європейської організації з ядерних досліджень (CERN), запропонував створення гіпертекстової системи для обміну інформацією між вченими. У 1989 році він розробив першу версію World Wide Web (WWW), а в 1990 році створив перший веб-браузер і веб-сервер. Основними технологіями того часу були:

- HTML (HyperText Markup Language) – мова розмітки для створення веб-сторінок.

- HTTP (HyperText Transfer Protocol) – протокол передачі даних у веб.

З середини 1990-х років веб почав швидко зростати. У 1993 році з'явився веб-браузер Mosaic, який значно спростив доступ до веб-ресурсів і сприяв популяризації Інтернету. У цей період з'явилися перші комерційні веб-сайти, а також нові веб-технології:

- CSS (Cascading Style Sheets) – мова стилів для покращення вигляду веб-сторінок.

- JavaScript – мова програмування для створення інтерактивних веб-сторінок, розроблена компанією Netscape у 1995 році.

З початку 2000-х років розвиток веб-технологій зосередився на створенні динамічних веб-сайтів, які дозволяли взаємодіяти з користувачами у реальному часі. Це стало можливим завдяки появі серверних мов програмування та баз даних:

- PHP (Hypertext Preprocessor) – серверна мова програмування для створення динамічних веб-сторінок.

- MySQL – система керування базами даних для зберігання інформації.

- AJAX (Asynchronous JavaScript and XML) – технологія для створення інтерактивних веб-додатків, що дозволяє оновлювати частини веб-сторінки без перезавантаження.

Поняття Веб 2.0 позначає етап розвитку Інтернету, який характеризується появою соціальних медіа, блогів та інших платформ для взаємодії користувачів. Ключовими технологіями цього періоду стали:

- AJAX – широке використання для створення інтерактивних веб-додатків.

- RSS (Really Simple Syndication) – формат для публікації оновлень на веб-сайтах.

- Content Management Systems (CMS) – системи управління контентом, такі як WordPress та Joomla, які спростили створення та адміністрування веб-сайтів.

Сучасний веб характеризується використанням адаптивного дизайну, що дозволяє веб-сайтам коректно відображатися на різних пристроях, таких як комп'ютери, планшети та смартфони. Ключовими технологіями цього періоду є:

- HTML5 – оновлена версія HTML з новими елементами та API для покращення функціональності веб-сторінок.

- CSS3 – оновлена версія CSS з новими можливостями для стилізації та анімації.

- JavaScript фреймворки та бібліотеки – такі як React, Angular та Vue.js, що дозволяють створювати складні та інтерактивні веб-додатки.

- Node.js – серверна платформа для виконання JavaScript на сервері, що дозволяє створювати масштабовані мережеві додатки.

Розвиток веб-технологій триває, постійно з'являються нові інструменти та підходи, що дозволяють створювати ще більш функціональні та зручні веб-додатки.

2.2 Важливість веб-сайтів для бізнесу

У сучасному світі наявність веб-сайту є однією з основних умов успішного ведення бізнесу. Веб-сайт служить не лише інструментом комунікації з клієнтами, але й важливим засобом маркетингу, продажу та підтримки іміджу компанії. Нижче наведено основні причини, чому веб-сайти є важливими для бізнесу [4-6].

- Підвищення видимості та доступності

Веб-сайт дозволяє компанії бути доступною для широкої аудиторії 24/7. Це особливо важливо в умовах глобалізації, коли потенційні клієнти можуть знаходитися в різних часових поясах. Завдяки веб-сайту, бізнес стає видимим для клієнтів у будь-який час і з будь-якого місця, де є доступ до Інтернету.

- Інструмент маркетингу та реклами

Веб-сайт є ефективним інструментом для проведення маркетингових кампаній. За допомогою різних технік, таких як SEO (оптимізація для пошукових систем), контент-маркетинг, PPC (плата за клік) реклама та соціальні медіа, компанії можуть залучати нових клієнтів і підвищувати свою популярність. Веб-сайт дозволяє проводити аналітику та відстежувати ефективність маркетингових заходів [3].

- Підтримка іміджу та довіри

Веб-сайт є відображенням бренду компанії та її цінностей. Відвідувачі часто оцінюють професіоналізм та надійність бізнесу на основі його веб-сайту. Добре спроектований, інформативний та зручний у користуванні веб-сайт сприяє формуванню позитивного іміджу компанії та підвищує довіру клієнтів.

- Підвищення обслуговування клієнтів

Веб-сайт дозволяє автоматизувати багато процесів обслуговування клієнтів, таких як надання інформації про продукти та послуги, відповіді на часті питання, оформлення замовлень та підтримка клієнтів. Це зменшує навантаження на персонал компанії та підвищує задоволеність клієнтів.

- Збільшення продажів

Веб-сайт є ефективним каналом продажів, що дозволяє компаніям пропонувати свої товари та послуги в Інтернеті. Електронна комерція (e-commerce) відкриває можливості для збільшення продажів без обмежень географічного розташування. Веб-сайт може інтегруватися з платіжними системами, забезпечуючи зручний процес покупки для клієнтів.

- Аналітика та зворотний зв'язок

Веб-сайти надають можливість збирання та аналізу даних про поведінку користувачів. Це дозволяє компаніям краще розуміти потреби своїх клієнтів, аналізувати ефективність маркетингових заходів та приймати обґрунтовані рішення. Крім того, за допомогою форм зворотного зв'язку та опитувань, компанії можуть отримувати безпосередні відгуки від своїх клієнтів.

- Переваги над конкурентами

У багатьох галузях наявність веб-сайту є стандартом. Відсутність веб-сайту може спричинити втрату конкурентних переваг, оскільки клієнти часто порівнюють компанії в Інтернеті перед прийняттям рішення про покупку. Веб-сайт дозволяє виділитися серед конкурентів, пропонуючи унікальні функції, зручний інтерфейс та високий рівень обслуговування.

- Розширення ринку

Веб-сайт дає можливість бізнесу вийти на нові ринки, не обмежуючись локальними кордонами. Це особливо актуально для малого та середнього бізнесу, який може пропонувати свої товари та послуги міжнародній аудиторії, розширюючи таким чином свою клієнтську базу.

- Економія ресурсів

Веб-сайт може допомогти зменшити витрати на рекламу, підтримку клієнтів та інші операційні витрати. Наприклад, інформація, яка раніше вимагала друку та поширення у фізичному вигляді, тепер може бути розміщена на веб-сайті. Це сприяє економії часу та грошей.

Узагальнюючи, можна сказати, що веб-сайт є важливим інструментом для розвитку бізнесу в сучасному цифровому світі. Він підвищує видимість та

доступність компанії, сприяє залученню нових клієнтів, збільшує продажі та підтримує позитивний імідж бренду.

2.3 Основні засоби розробки веб-сайтів

Розробка веб-сайтів включає використання різноманітних технологій, інструментів та засобів, які забезпечують створення структурованого, стильного та функціонального веб-ресурсу. Нижче наведено огляд основних засобів розробки веб-сайтів [7].

- **HTML (HyperText Markup Language)**

HTML є основною мовою розмітки для створення веб-сторінок. Він визначає структуру веб-документів за допомогою елементів і тегів. HTML забезпечує базову структуру веб-сайту, яка включає заголовки, абзаци, списки, посилання, зображення та інші елементи. Найновіша версія, HTML5, додала багато нових елементів та API для покращення функціональності та семантики веб-сторінок.

- **CSS (Cascading Style Sheets)**

CSS використовується для стилізації та оформлення веб-сторінок. Він дозволяє розділити візуальне представлення від структурного коду, забезпечуючи гнучкість у дизайні. CSS3, остання версія CSS, додає нові можливості для анімації, градієнтів, трансформацій та адаптивного дизайну, що дозволяє створювати сучасні та привабливі веб-сайти.

- **JavaScript**

JavaScript є мовою програмування, яка використовується для додавання інтерактивності та динаміки на веб-сторінки. За допомогою JavaScript можна створювати такі елементи, як слайдери зображень, інтерактивні форми, анімації та багато інших функцій. Сучасні фреймворки та бібліотеки JavaScript, такі як React, Angular та Vue.js, дозволяють розробляти складні односторінкові додатки (SPA).

- Фреймворки та бібліотеки

- Bootstrap: популярний CSS-фреймворк для створення адаптивних та мобільних веб-сайтів. Він включає готові компоненти, такі як навігаційні панелі, кнопки, форми та інші елементи інтерфейсу.

- jQuery: JavaScript-бібліотека, яка спрощує маніпуляцію DOM (Document Object Model), обробку подій та створення анімацій. Хоча сучасні фреймворки частково замінили jQuery, він все ще залишається популярним.

- React: бібліотека для створення користувацьких інтерфейсів, розроблена Facebook. Вона дозволяє створювати компоненти, які можна повторно використовувати, що полегшує розробку великих додатків.

- Angular: фреймворк для створення динамічних веб-додатків, розроблений Google. Angular надає інструменти для побудови односторінкових додатків з використанням MVC (Model-View-Controller) архітектури.

- Vue.js: прогресивний фреймворк для створення користувацьких інтерфейсів. Він простий у вивченні та використанні, що робить його популярним серед новачків та досвідчених розробників.

- Системи керування контентом (CMS)

CMS дозволяють створювати та керувати контентом веб-сайтів без необхідності глибоких знань у програмуванні. Найпопулярніші CMS включають:

- WordPress: найпопулярніша CMS у світі, яка використовується для створення блогів, інтернет-магазинів та інших типів веб-сайтів.

- Joomla: потужна CMS з широкими можливостями для створення складних веб-сайтів.

- Drupal: гнучка та масштабована CMS, яка підходить для великих та складних проектів.

- Серверні мови програмування та бази даних

- PHP: серверна мова програмування, яка широко використовується для створення динамічних веб-сайтів. PHP легко інтегрується з базами даних, такими як MySQL.

- Node.js: серверна платформа, яка дозволяє виконувати JavaScript на сервері. Node.js забезпечує високу продуктивність і масштабованість для мережеских додатків.

- Python: популярна мова програмування, яка використовується для веб-розробки завдяки своїм фреймворкам, таким як Django та Flask.

- Ruby: мова програмування, яка використовується разом з фреймворком Ruby on Rails для швидкої розробки веб-додатків.

- MySQL та PostgreSQL: реляційні системи керування базами даних, які забезпечують зберігання та обробку даних для веб-додатків.

- MongoDB: нереляційна база даних (NoSQL), яка використовується для зберігання великих обсягів неструктурованих даних.

- Інструменти для розробки

- Редактори коду: такі як Visual Studio Code, Sublime Text та Atom, які забезпечують зручне середовище для написання коду.

- Системи контролю версій: такі як Git, які дозволяють відслідковувати зміни в коді та працювати в команді.

- Інструменти для автоматизації: такі як Webpack та Gulp, які спрощують процес розробки шляхом автоматизації завдань, таких як компіляція CSS та JavaScript, оптимізація зображень тощо.

Узагальнюючи, сучасна розробка веб-сайтів вимагає використання різноманітних технологій та інструментів, які забезпечують створення високоякісних, функціональних та привабливих веб-ресурсів. Знання основних засобів розробки веб-сайтів дозволяє розробникам ефективно реалізовувати свої ідеї та створювати проекти, які відповідають вимогам сучасних користувачів та бізнесу.

3. ТЕОРЕТИЧНИЙ ЧАСТИНА

3.1 Аналіз сучасних тенденцій в автомобільній індустрії

Автомобільна індустрія постійно розвивається та адаптується до змін у технологіях, споживчих уподобаннях та регуляторних вимогах. Нижче наведено аналіз основних сучасних тенденцій, які впливають на автомобільну галузь [8].

- Електрифікація транспорту

Електромобілі (EV) стають все більш популярними завдяки екологічним перевагам та зниженню вартості експлуатації. Основні тренди включають:

- Зростання виробництва електромобілів: Все більше автовиробників інвестують у розробку та виробництво електромобілів. Ведучі бренди, такі як Tesla, Nissan, BMW, і Volkswagen, розширюють свої модельні ряди електромобілів.

- Розвиток інфраструктури зарядки: Зростає кількість зарядних станцій для електромобілів, що сприяє їх масовому прийняттю. Багато країн інвестують в інфраструктуру для зарядки електромобілів.

- Покращення технологій акумуляторів: Технології акумуляторів постійно вдосконалюються, забезпечуючи більший запас ходу та швидкість зарядки.

- Автономні транспортні засоби

Розробка автономних транспортних засобів (AV) прогресує швидкими темпами завдяки досягненням в галузі штучного інтелекту та машинного навчання. Основні аспекти включають:

- Рівні автономності: Існує п'ять рівнів автономності, де рівень 5 представляє повністю автономні транспортні засоби без потреби в людському водію. Більшість сучасних технологій знаходяться на рівнях 2-3.

- Тестування та впровадження: Багато компаній, включаючи Google (Waymo), Uber, та Tesla, активно тестують автономні транспортні засоби на дорогах загального користування.

- Регуляторні виклики: Безпека та правові аспекти є основними викликами для масового впровадження автономних транспортних засобів. Багато країн розробляють нові регуляції для адаптації до цих технологій.

- Конективність та Інтернет речей (IoT)

Сучасні автомобілі стають все більш підключеними до Інтернету, що забезпечує нові можливості для водіїв та пасажирів. Основні напрямки включають:

- Підключені автомобілі: Автомобілі з вбудованими сенсорами та з'єднанням з Інтернетом можуть отримувати та передавати дані в режимі реального часу, що покращує безпеку та комфорт водіння.

- Інтелектуальні системи допомоги водієві (ADAS): Такі системи, як адаптивний круїз-контроль, автоматичне гальмування та асистенти паркування, значно підвищують безпеку та зручність керування автомобілем.

- Телематика та діагностика: Підключені автомобілі можуть здійснювати діагностику своїх систем у реальному часі та передавати ці дані до сервісних центрів, що дозволяє прогнозувати та запобігати можливим поломкам.

- Зміни у споживчих уподобаннях

Сучасні споживачі все більше цінують екологічність, безпеку та новітні технології в автомобілях. Основні тенденції включають:

- Попит на екологічні автомобілі: Зростає попит на електромобілі та гібриди через їхню екологічність та зниження вартості експлуатації.

- Інтерес до підключених та автономних технологій: Споживачі все більше зацікавлені в автомобілях, які пропонують новітні технології для покращення безпеки та комфорту.

- Зміни у власності на автомобілі: Зростає популярність каршерінгу та оренди автомобілів, особливо серед молодих людей, які віддають перевагу використанню транспортних засобів без необхідності їхнього придбання.

- Регуляторні зміни

Регуляторні органи в усьому світі впроваджують нові норми та стандарти для зниження викидів та підвищення безпеки на дорогах. Основні напрямки включають:

- Стандарти викидів: Багато країн встановлюють жорсткіші стандарти викидів для транспортних засобів, що стимулює виробників переходити на електромобілі та гібриди.

- Безпека на дорогах: Нові регуляції спрямовані на підвищення безпеки дорожнього руху, включаючи вимоги щодо впровадження систем допомоги водієві та автономних технологій.

- Інфраструктурні інвестиції: Уряди багатьох країн інвестують у розвиток інфраструктури для підтримки нових технологій, таких як зарядні станції для електромобілів та мережі для автономних транспортних засобів.

Таким чином, сучасні тенденції в автомобільній індустрії спрямовані на підвищення екологічності, безпеки та інтеграції новітніх технологій, що забезпечують зручність та комфорт для користувачів. Ці тенденції впливають на стратегії автовиробників та формують майбутнє автомобільного ринку.

3.2 Порівняльний аналіз існуючих сайтів автосалонів

Аналіз веб-сайтів автосалонів є важливим кроком для розуміння найкращих практик та визначення можливостей для покращення власного проекту. Нижче наведено порівняльний аналіз кількох популярних сайтів автосалонів, враховуючи такі аспекти, як дизайн, функціональність, зручність користування та інтерактивність.

- Tesla (www.tesla.com)

На малюнку нижче зображено головну сторінку сайту «Tesla»:

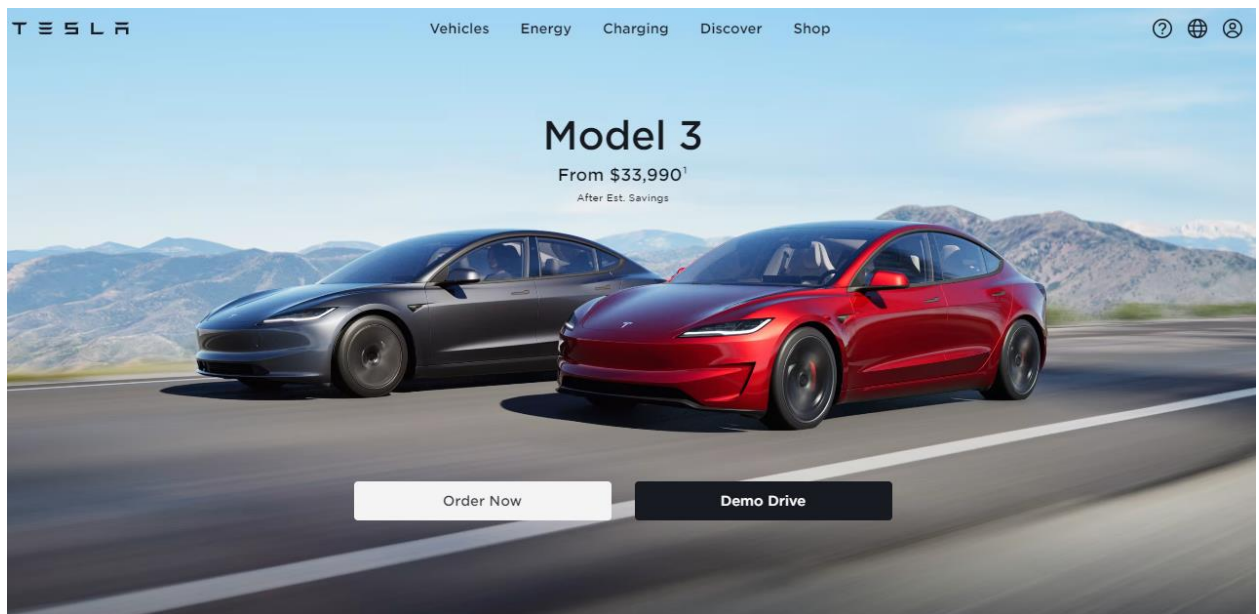


Рисунок 3.1 Головна сторінка «Tesla»

Дизайн:

- Сучасний і мінімалістичний дизайн з акцентом на високоякісні зображення автомобілів.
- Використання білих простірв для створення відчуття простоти та елегантності.

Функціональність:

- Інтерактивні конфігуратори автомобілів, що дозволяють користувачам персоналізувати автомобіль.
- Інформаційні розділи про кожну модель, включаючи технічні характеристики, ціни та опції.
- Інтеграція з системою онлайн-замовлення та бронювання тест-драйвів.

Зручність користування:

- Інтуїтивно зрозуміла навігація з чіткою структурою та легко доступними меню.
- Адаптивний дизайн, що забезпечує коректне відображення на різних пристроях.

Інтерактивність:

- Відео та анімації для демонстрації функцій автомобілів.
- Живий чат для консультацій з клієнтами.
- BMW (www.bmw.com)

На малюнку нижче зображено головну сторінку сайту «Bmw»:

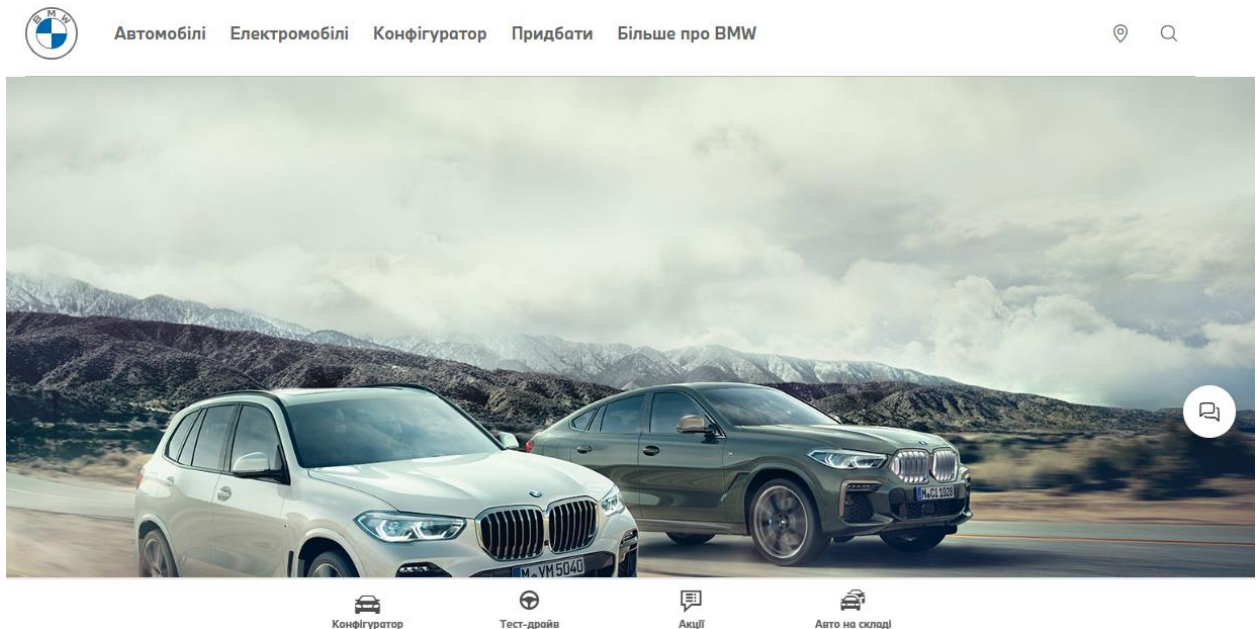


Рисунок 3.2 Головна сторінка «Bmw»

Дизайн:

- Преміум-дизайн з використанням високоякісних зображень та відео.
- Темний фон з контрастними елементами, що підкреслюють розкіш та елегантність бренду.

Функціональність:

- Детальні конфігуратори автомобілів з багатьма опціями персоналізації.
- Розділи новин та блогів з актуальною інформацією про нові моделі та технології.
- Можливість онлайн-замовлення та бронювання сервісного обслуговування.

Зручність користування:

- Просте та логічне меню навігації.
- Адаптивний дизайн для мобільних пристроїв.

Інтерактивність:

- Відеоролики з оглядами нових моделей.

- Toyota (www.toyota.com)

На малюнку нижче зображено головну сторінку сайту «Toyota»:

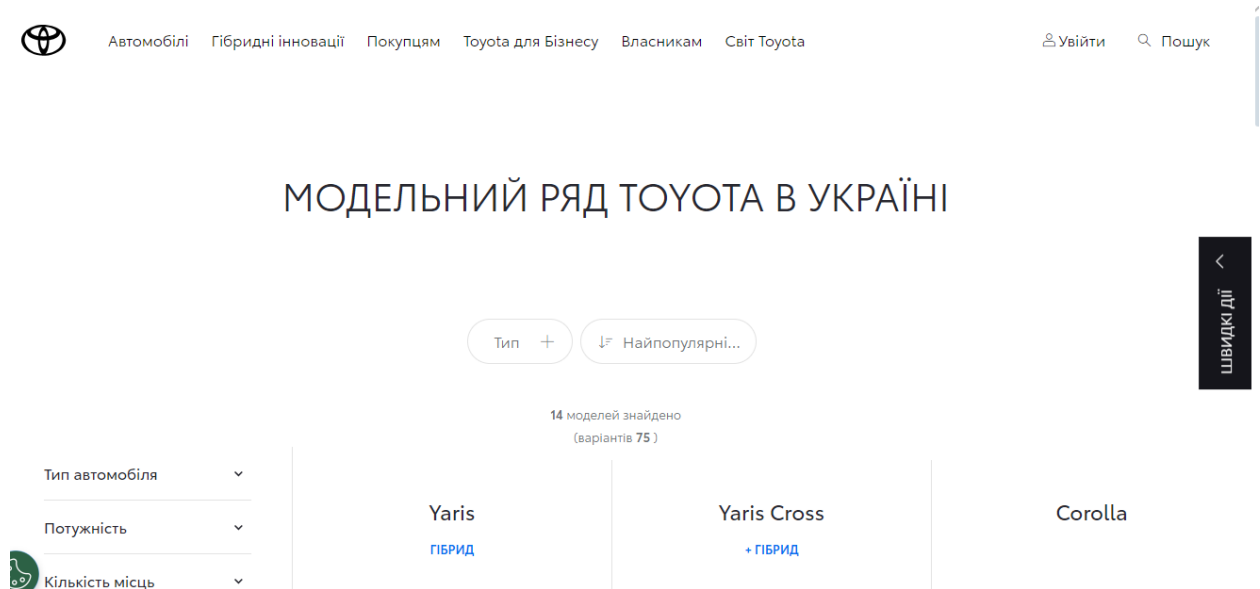


Рисунок 3.3 Головна сторінка «Toyota»

Дизайн:

- Чистий та функціональний дизайн з акцентом на зручність користування.

- Використання яскравих кольорів для виділення ключових елементів.

Функціональність:

- Інтерактивні конфігуратори з можливістю порівняння різних моделей.
- Розділи для фінансування та страхування автомобілів.
- Інтеграція з системою онлайн-замовлення та бронювання тест-драйвів.

Зручність користування:

- Інтуїтивна навігація з чітко структурованими розділами.
- Адаптивний дизайн для зручного перегляду на мобільних пристроях.

Інтерактивність:

- Віртуальні тури та інтерактивні демонстрації функцій автомобілів.

- Форми зворотного зв'язку та онлайн-консультації.

- Mercedes-Benz (www.mercedes-benz.com)

На малюнку нижче зображено головну сторінку сайту «Mercedes-Benz»:

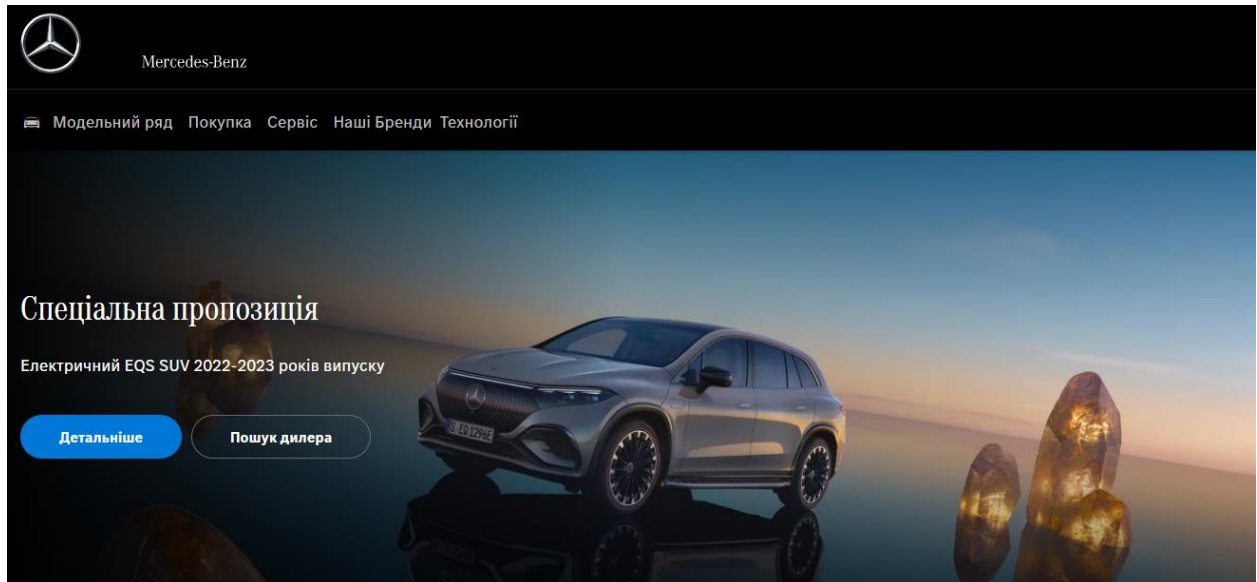


Рисунок 3.4 Головна сторінка «Mercedes-Benz»

Дизайн:

- Розкішний та стильний дизайн з використанням темних кольорів і високоякісних зображень.
- Анімації та відео, що підкреслюють елегантність та технологічність автомобілів.

Функціональність:

- Розширені конфігуратори автомобілів з багатьма опціями персоналізації.
- Інформаційні розділи про технології, інновації та історію бренду.
- Онлайн-замовлення та бронювання тест-драйвів.

Зручність користування:

- Чітка структура меню та логічна навігація.
- Адаптивний дизайн для мобільних пристроїв.

Інтерактивність:

- Відео та анімації для демонстрації технологій та функцій.

Висновок

Аналіз веб-сайтів автосалонів показує, що успішні сайти мають кілька спільних характеристик:

- Сучасний дизайн: Використання високоякісних зображень, відео та мінімалістичного дизайну.
- Функціональність: Інтерактивні конфігуратори, можливість онлайн-замовлення та бронювання послуг.
- Зручність користування: Інтуїтивна навігація та адаптивний дизайн для різних пристроїв.
- Інтерактивність: Віртуальні тури, відеоролики, живі чати та форми зворотного зв'язку.

Враховуючи ці аспекти, можна розробити власний веб-сайт автосалону, який буде відповідати сучасним вимогам та залучати більше клієнтів.

3.3 Вибір технології розробки сайту автосалону

Вибір правильних технологій для розробки веб-сайту автосалону є критичним для забезпечення його функціональності, продуктивності та зручності користування. Нижче наведено детальний аналіз технологій, які можуть бути використані для створення сучасного та ефективного веб-сайту автосалону [9-10].

- Мова розмітки HTML5

HTML5 є основою для створення структурованого контенту на веб-сторінках. Він дозволяє створювати семантичну розмітку, яка полегшує роботу з контентом для браузерів і пошукових систем. HTML5 також підтримує мультимедійні елементи (відео, аудіо), що є важливим для демонстрації автомобілів.

Переваги:

- Широка підтримка браузерами.
- Семантичні теги для покращення SEO.
- Підтримка мультимедіа та інтерактивних елементів.
- Мова стилів CSS3

CSS3 використовується для створення стильного та привабливого дизайну веб-сайту. Він дозволяє задавати макет, кольорові схеми, шрифти та анімації, що робить сайт візуально привабливим і зручним для користувачів.

Переваги:

- Підтримка адаптивного дизайну для мобільних пристроїв.
- Можливість створення анімацій та трансформацій.
- Відокремлення стилю від змісту, що полегшує підтримку сайту.
- JavaScript та бібліотеки

JavaScript є мовою програмування, яка дозволяє додати інтерактивність до веб-сайту. За допомогою JavaScript можна реалізувати динамічні елементи, такі як слайдери зображень, інтерактивні форми та пошук по сайту.

Бібліотеки:

- jQuery: Спрощує маніпуляції з DOM, обробку подій та анімації.
- React: Використовується для створення компонентів користувацького інтерфейсу, що полегшує розробку складних веб-додатків.
- Vue.js: Прогресивний фреймворк для створення інтерфейсів, що поєднує простоту та продуктивність.

Переваги:

- Широкий спектр можливостей для створення інтерактивних елементів.
- Підтримка компонентного підходу, що покращує масштабованість та повторне використання коду.
- Серверні технології

Для обробки запитів користувачів та взаємодії з базами даних необхідно використовувати серверні технології.

PHP: Популярна серверна мова програмування, яка легко інтегрується з базами даних та підтримує динамічний контент.

- Node.js: Серверна платформа, яка дозволяє виконувати JavaScript на сервері. Забезпечує високу продуктивність та масштабованість.

- Python (Django): Потужний фреймворк для швидкої розробки веб-додатків з великою кількістю готових компонентів.

- Ruby on Rails: Фреймворк для розробки веб-додатків на мові Ruby, відомий своєю простотою та швидкістю розробки.

- Бази даних

Для зберігання інформації про автомобілі, клієнтів та інші дані необхідно використовувати бази даних.

MySQL: Популярна реляційна система керування базами даних, відома своєю надійністю та продуктивністю.

- PostgreSQL: Потужна реляційна база даних з підтримкою складних запитів та розширень.

- MongoDB: Нереляційна база даних, яка підходить для зберігання великих обсягів неструктурованих даних.

- Системи керування контентом (CMS)

Для спрощення процесу управління контентом веб-сайту можна використовувати CMS.

WordPress: Найпопулярніша CMS, яка забезпечує просте управління контентом та має велику кількість плагінів та тем.

- Joomla: Гнучка CMS, яка підходить для створення складних веб-сайтів з великою кількістю функцій.

- Drupal: Потужна CMS, яка підходить для великих та складних проєктів, забезпечуючи високу масштабованість та безпеку.

- Інструменти для розробки

- Visual Studio Code: Популярний редактор коду з підтримкою численних розширень для різних мов програмування.

- Git: Система контролю версій, яка дозволяє відстежувати зміни в коді та працювати в команді.

- Webpack: Інструмент для зборки модулів JavaScript, що спрощує процес розробки.

Рекомендована технологія для розробки сайту автосалону

Враховуючи сучасні тенденції та вимоги до веб-сайтів автосалонів, рекомендую наступний технологічний стек:

- HTML5 та CSS3 для структурування та стилізації контенту.
- JavaScript з використанням React для створення інтерактивних та компонентних інтерфейсів.
- Node.js для серверної частини, що забезпечить високу продуктивність та масштабованість.
- MySQL або PostgreSQL для зберігання даних про автомобілі та клієнтів.
- WordPress як CMS для спрощення управління контентом, з інтеграцією кастомних тем та плагінів для специфічних потреб автосалону.
- Git для контролю версій та спільної роботи над проектом.
- Visual Studio Code як основний редактор коду з підтримкою необхідних розширень для розробки.

Цей технологічний стек дозволить створити сучасний, функціональний та зручний у користуванні веб-сайт автосалону, який відповідатиме всім сучасним вимогам та забезпечить позитивний користувацький досвід.

4. ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1 Проектування веб-сайту

4.1.1 Визначення функціональних вимог до сайту

Функціональні вимоги визначають, які саме функції та можливості повинен мати веб-сайт автосалону для задоволення потреб користувачів і досягнення цілей бізнесу. Нижче наведено основні функціональні вимоги для веб-сайту автосалону.

- **Головна сторінка**

- Привітання та короткий опис автосалону: Інформація про компанію, її місію та ключові переваги.

- Слайдер з акціями та новинами: Динамічний слайдер з актуальними пропозиціями, новинками та акціями.

- Основні розділи сайту: Швидкі посилання на основні розділи сайту, такі як каталог автомобілів, послуги, контакти тощо.

- **Каталог автомобілів**

- Список автомобілів: Відображення всіх доступних автомобілів з можливістю сортування та фільтрації за різними параметрами (марка, модель, рік випуску, ціна, тип кузова, тощо).

- Детальна інформація про автомобіль: Окрема сторінка для кожного автомобіля з детальною інформацією, включаючи технічні характеристики, фотографії, відеоогляди та ціни.

- Конфігуратор автомобіля: Інтерактивний інструмент для персоналізації автомобіля, вибору комплектації, кольору, додаткових опцій тощо.

- **Пошук та фільтрація**

- Пошук по сайту: Інструмент для швидкого пошуку автомобілів за ключовими словами.

- Фільтрація результатів: Можливість фільтрації автомобілів за різними параметрами для швидкого знаходження потрібного варіанту.

- Інтерактивні елементи

- Відеоогляди: Вбудовані відео з оглядами автомобілів для надання більш детальної інформації про них.

- Послуги

- Опис послуг: Розділ з інформацією про послуги автосалону, такі як технічне обслуговування, фінансування, страхування, тест-драйви тощо.

- Бронювання послуг: Можливість онлайн-бронювання послуг, включаючи запис на тест-драйв та сервісне обслуговування.

- Онлайн-консультації та зворотний зв'язок

- Форми зворотного зв'язку: Інтерактивні форми для зв'язку з менеджерами автосалону, замовлення консультацій, зворотний дзвінок тощо.

- Живий чат: Інтеграція живого чату для надання миттєвих консультацій користувачам.

- Новини та акції

- Блог та новини: Розділ з новинами, статтями та блогами про автосалон, нові моделі автомобілів, акції та інші події.

- Підписка на новини: Форма для підписки на новини та акції автосалону.

- Користувацький кабінет

- Реєстрація та авторизація: Можливість створення облікового запису користувача для доступу до персоналізованих послуг.

- Особистий кабінет: Інтерфейс для керування особистою інформацією, замовленнями, бронюваннями та налаштуваннями користувача.

- Контакти та місцезнаходження

- Контактна інформація: Адреси, телефони, електронна пошта та інші контактні дані автосалону.

- Карта проїзду: Інтерактивна карта з вказівками щодо місцезнаходження автосалону та його філій.

- Години роботи: Інформація про години роботи автосалону та сервісних центрів.

- Інтеграція з соціальними мережами

- Посилання на соціальні мережі: Інтеграція з профілями автосалону в соціальних мережах (Facebook, Instagram, Twitter тощо).

- Шерінг контенту: Можливість поділитися контентом з веб-сайту в соціальних мережах.

Створення структури сайту (мапа сайту).

Структура сайту, або мапа сайту, є важливим елементом процесу розробки, оскільки вона визначає, як інформація буде організована та як користувачі будуть переміщатися між різними розділами сайту. Нижче (рис. 4.1) наведено структуру сайту автосалону з основними розділами та підрозділами.

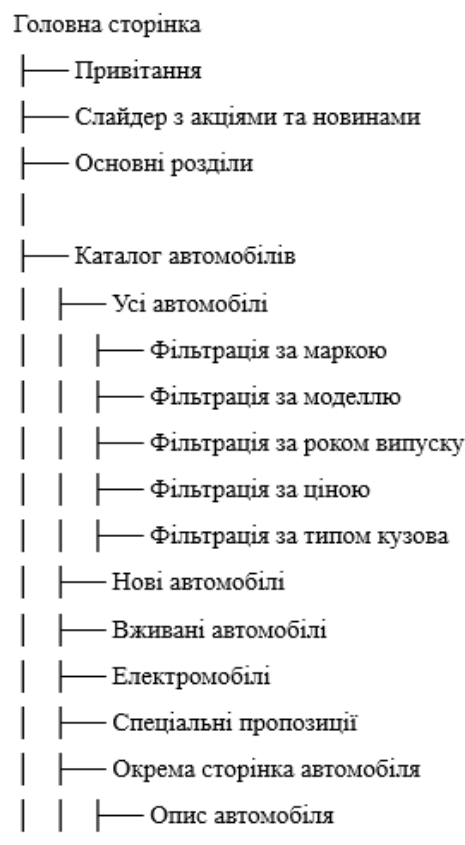


Рисунок 4.1 - Структуру сайту автосалону

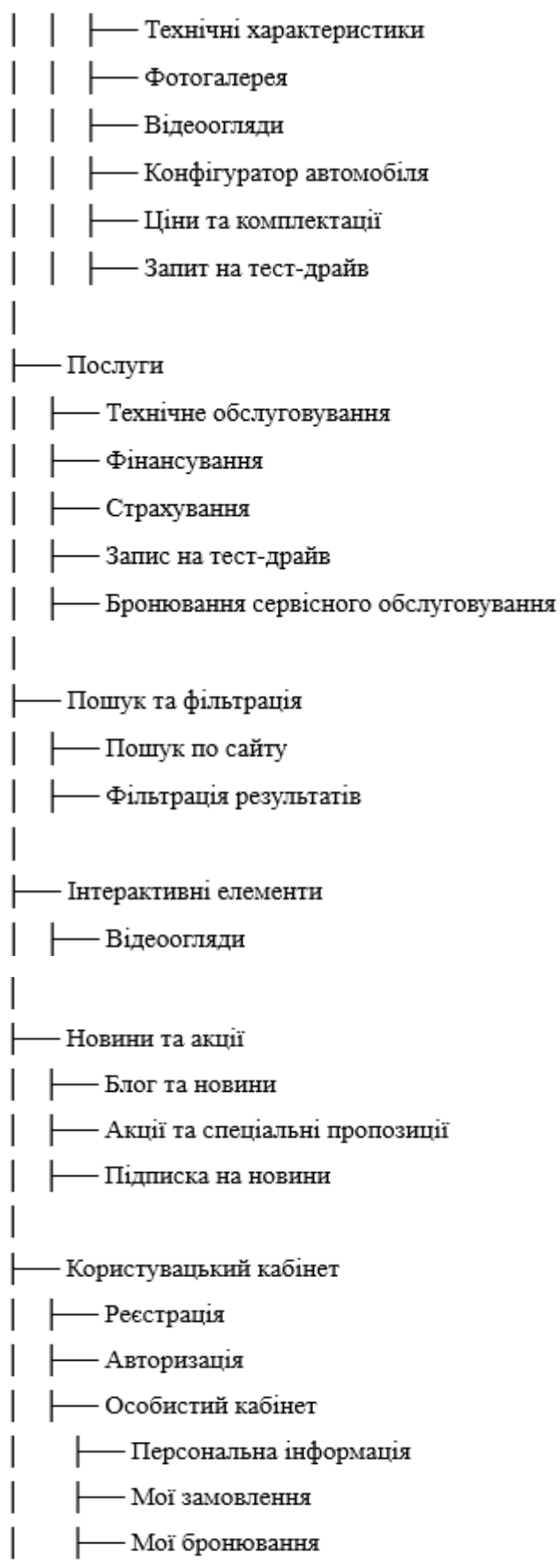


Рисунок 4.2 – Продовження структуру сайту автосалону

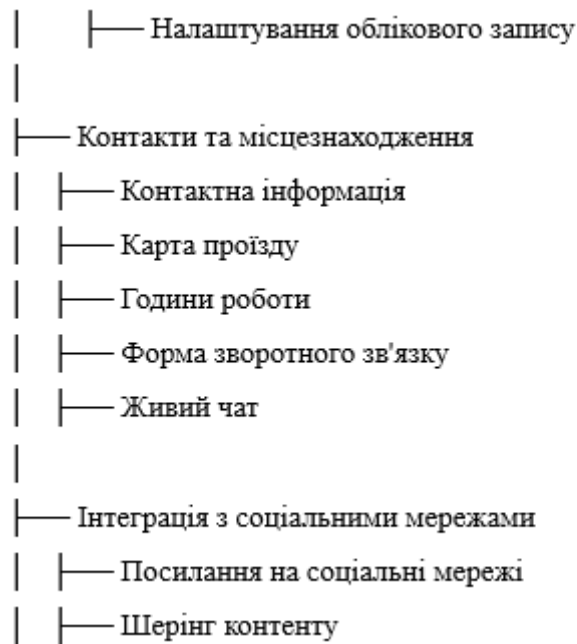


Рисунок 4.3 – Продовження структуру сайту автосалону

4.1.2 Розробка дизайну та інтерфейсу користувача (UI/UX)

Розробка дизайну та інтерфейсу користувача (UI/UX) є важливим етапом створення веб-сайту автосалону, оскільки він визначає візуальне представлення та зручність користування сайтом. Цей етап включає створення естетичного, інтуїтивного та функціонального дизайну, який задовольняє потреби користувачів та сприяє досягненню бізнес-цілей [11-13].

- Аналіз цільової аудиторії

Перший крок у розробці UI/UX – це розуміння цільової аудиторії, яка буде користуватися веб-сайтом автосалону. Це допоможе визначити, які елементи дизайну та функціональності будуть найбільш важливими. Основні групи цільової аудиторії можуть включати:

- Потенційні покупці автомобілів (нових та вживаних).
- Клієнти, які шукають фінансові та страхові послуги.
- Користувачі, які бажають замовити сервісне обслуговування або тест-драйв.

- Основні принципи дизайну

Основні принципи, які слід враховувати при розробці дизайну веб-сайту автосалону:

- Простота: Дизайн має бути простим та зрозумілим, щоб користувачі легко знаходили потрібну інформацію.
- Когерентність: Усі елементи дизайну повинні бути узгодженими та відповідати загальному стилю бренду.
- Зручність користування (Usability): Інтерфейс повинен бути інтуїтивним та легким у використанні.
- Адаптивність: Дизайн повинен коректно відображатися на різних пристроях (десктопах, планшетах, смартфонах).

- Створення прототипів

Прототипи допомагають візуалізувати структуру та функціональність сайту перед початком розробки. Основні етапи створення прототипів:

- Каркасні схеми (Wireframes): Скелетні структури сторінок, які визначають розташування основних елементів (меню, блоків контенту, кнопок тощо).
- Інтерактивні прототипи: Прототипи з базовою інтерактивністю, що дозволяють протестувати навігацію та взаємодію користувачів з інтерфейсом.

- Дизайн інтерфейсу користувача (UI)

Створення візуального дизайну сайту включає:

- Колірна гамма: Вибір кольорів, які відповідають бренду автосалону та створюють привабливий візуальний образ.
- Шрифти: Використання читабельних та привабливих шрифтів для текстових елементів.
- Графічні елементи: Додавання зображень, іконок, відео та анімацій, які покращують візуальне сприйняття сайту.
- Макет сторінок: Розміщення основних елементів на сторінках (заголовків, текстів, зображень, кнопок тощо).
- Інтерфейс користувача (UX)

Розробка UX-дизайну включає забезпечення зручності та ефективності взаємодії користувачів з сайтом:

- Навігація: Простий та інтуїтивний навігаційний меню з чіткою структурою та швидким доступом до основних розділів.

- Функціональність: Забезпечення безперебійної роботи інтерактивних елементів (конфігураторів, форм зворотного зв'язку, пошуку тощо).

- Зворотний зв'язок: Інтерактивні елементи, які надають користувачам зворотний зв'язок (анімації при натисканні на кнопки, спливаючі повідомлення про успішне виконання дій тощо).

- Тестування з користувачами: Проведення тестування прототипів з реальними користувачами для виявлення недоліків та їх усунення.

- Приклади дизайну сторінок

Головна сторінка:

- Заголовок та привітання.
- Слайдер з акціями та новинками.
- Основні розділи сайту (каталог, послуги, новини тощо).
- Кнопки швидкого доступу до важливих розділів (запис на тест-драйв, контактна інформація).

Каталог автомобілів:

- Фільтри та сортування.
- Карточки автомобілів з короткою інформацією.
- Кнопка для перегляду детальної інформації про автомобіль.

Сторінка автомобіля:

- Детальна інформація про автомобіль.
- Конфігуратор автомобіля.
- Запит на тест-драйв або покупку.

Послуги:

- Опис кожної послуги.
- Кнопка для бронювання послуги.

Контакти:

- Карта проїзду.
- Контактна інформація (телефон, електронна пошта).
- Форма зворотного зв'язку.
- Живий чат для миттєвих консультацій.
- Інструменти для розробки дизайну та прототипування
 - Sketch/Figma/Adobe XD: Інструменти для створення каркасних схем та прототипів, а також для дизайну інтерфейсу користувача.
 - InVision: Платформа для створення інтерактивних прототипів та спільної роботи над дизайном.
 - Adobe Photoshop/Illustrator: Інструменти для створення графічних елементів та обробки зображень.

4.2 Програмна реалізація веб-сайту

4.2.1 Розробка фронтенду з використанням HTML та CSS

Фронтенд-розробка включає створення видимої частини веб-сайту, з якою взаємодіють користувачі. HTML забезпечує структуру контенту, тоді як CSS відповідає за стилізацію та візуальне оформлення [7, 9].

• Створення структури HTML

HTML (HyperText Markup Language) використовується для створення базової структури веб-сторінок. Основні елементи HTML включають заголовки, абзаци, списки, посилання, зображення та інші компоненти. Нижче наведено приклад базової структури HTML для головної сторінки автосалону.

Приклад HTML-коду для головної сторінки - програмний код реалізацій в Додатку А.

Нижче показується реалізація коду: головна сторінка та мапа сайту (рис. 4.4); каталог автомобілів (рис. 4.5); послуги сайту (рис. 4.6); відгуки клієнтів (рис. 4.7).

Про-Нас Головна Навички Автопарк Рекомендовані



Рисунок 4.4 – Головна сторінка та мапа сайту

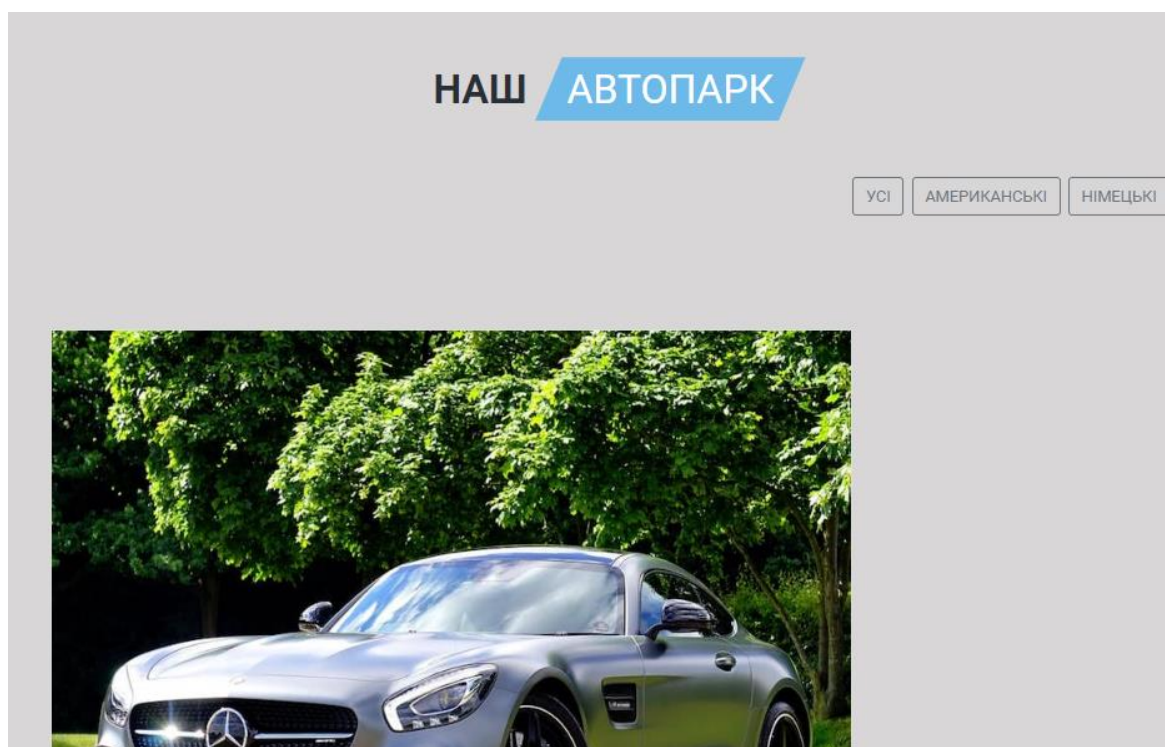


Рисунок 4.5 – Каталог автомобілів

НАШІ ПОСЛУГИ

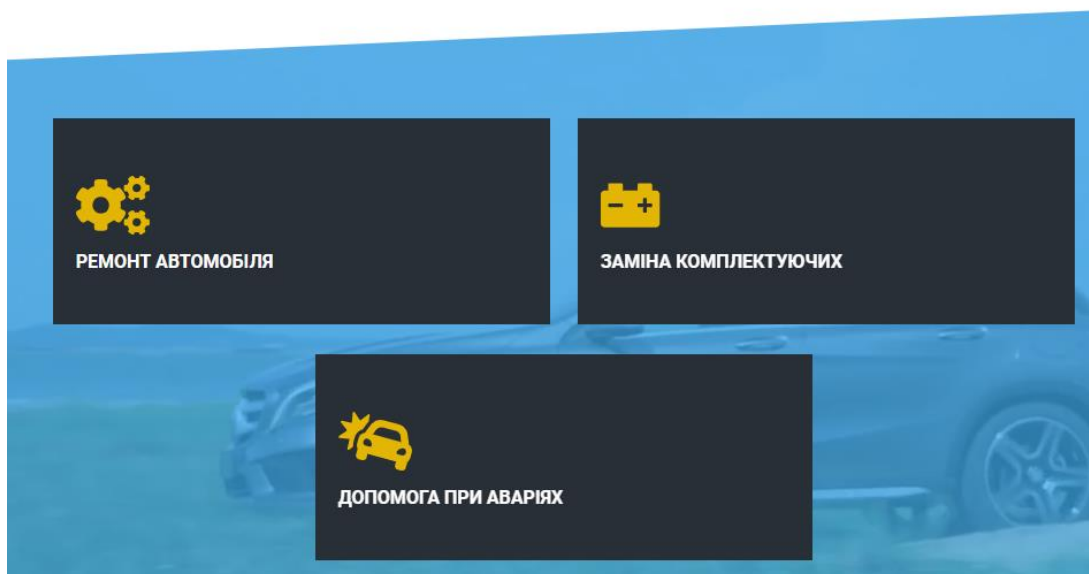


Рисунок 4.6 – Послуги

ВІДГУКИ КЛІЄНТІВ

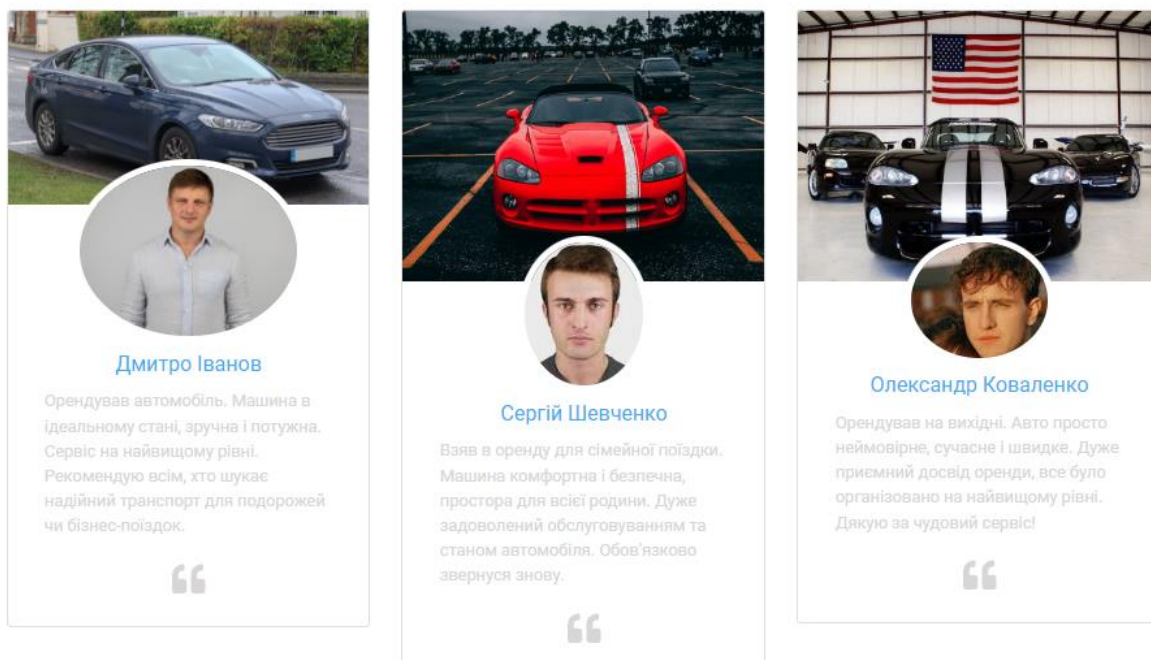


Рисунок 4.7 – Відгуки клієнтів

- Стилiзацiя з використанням CSS

CSS (Cascading Style Sheets) використовується для створення стилів та покращення візуального вигляду веб-сторінок. За допомогою CSS можна визначати кольори, шрифти, розміри, відступи, макети та багато інших властивостей елементів HTML.

Приклад CSS-коду для головної сторінки:

Основні стилі

```
@import url('https://fonts.googleapis.com/css?family=Roboto:400,700');
:root {
  --mainBlue: #40ACF1;
  --transparentBlue: rgba(64, 172, 241, 0.7);
  --mainWhite: #FFFFFF;
  --mainBlack: #292F36;
  --mainGrey: rgb(216, 214, 214);
  --mainYellow: #e3b505;
}
body {
  font-family: 'Roboto', sans-serif;
  color: var(--mainBlack);
  background: var(--mainWhite);
}
/* Стилi для кнопок */
.stylish-btn {
  font-weight: bold;
  border-radius: 20px;
  transition: background-color 0.3s ease, color 0.3s ease;
}
.stylish-btn:hover {
  background-color: #333;
  color: #fff;
}
/ Шапка сайту /
header {
  background-color: 333;
  color: fff;
  padding: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

4.3 Використання JavaScript для створення інтерактивних елементів

JavaScript є мовою програмування, яка дозволяє додати інтерактивність до веб-сайту. Завдяки JavaScript можна створювати динамічні елементи, що покращують користувацький досвід. Нижче наведено приклади використання JavaScript для реалізації різних інтерактивних функцій на веб-сайті автосалону[14-15].

- Форма зворотного зв'язку
- Форма зворотного зв'язку дозволяє користувачам залишати свої запитання та коментарі. JavaScript може бути використаний для валідації введених даних перед відправкою форми.
- HTML для форми зворотного зв'язку:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/main.css">
  <link href="https://fonts.googleapis.com/css?family=Courgette"
rel="stylesheet">
  <script src="js/all.js"></script>
  <title>Запис в автосалон</title>
  <style>
    /* Стили для вікна чату */
  <!-- Вікно чату -->
  <div class="chat-window">
    <div class="chat-window-header">Менеджер Оксана</div>
    <div class="chat-window-body" id="chat-window-body"></div>
    <div class="chat-window-footer">
      <textarea class="chat-input" id="chat-input"
placeholder="Напишіть повідомлення..."></textarea>
      <button class="send-button"
onclick="sendMessage()">Надіслати</button>
    </div>
  </div>
```

```

<script>
    function sendMessage() {
        var message = document.getElementById('chat-input').value;
        var chatBody = document.getElementById('chat-window-body');
        var messageElement = document.createElement('div');
        messageElement.textContent = message;
        chatBody.appendChild(messageElement);
        document.getElementById('chat-input').value = '';
        setTimeout(function () {
            var response = "Дякую за ваше повідомлення! Зв'яжусь з вами
найближчим часом.";
            var responseElement = document.createElement('div');
            responseElement.textContent = "Олена: " + response;
            chatBody.appendChild(responseElement);
        }, 5000);
    }
</script>
<script src="js/jquery-3.3.1.min.js"></script>
<script src="js/bootstrap.bundle.min.js"></script>
<script src="js/script.js"></script>
<script>
    $(document).ready(function () {
        var cars = $('.car-card').detach();
        $('.btn-filter').click(function () {
            var category = $(this).data('category');
            var filteredCars = (category === 'all') ? cars :
cars.filter(function () {
                return $('img', this).attr('src').includes(category);
            });
            $('#car-container').empty().append(filteredCars);
        });
        $('#car-container').append(cars);
    });
</script>
<script>
    $(document).ready(function () {
        $('.featured-item').click(function () {
            var imgPath = $(this).data('src');
            $('.featured-photo').attr('src', imgPath);
        });
    });
</script>
</body>
</html>

```

CSS для форми зворотного зв'язку:

```
.chat-window {
```

```

    position: fixed;
    bottom: 20px;
    right: 20px;
    width: 300px;
    background-image: url('your-image.jpg'); /* Замініть 'your-
image.jpg' на посилання на ваше зображення */
    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    z-index: 9999;
}
.chat-window-header {
    padding: 10px;
    background-color: rgba(255, 255, 255, 0.7); /* Прозорий фон для
заголовка */
    border-bottom: 1px solid #ccc;
}
.chat-window-body {
    padding: 10px;
    height: 200px;
    overflow-y: auto;
    font-size: 16px; /* Збільшуємо розмір шрифту */
}
.chat-window-footer {
    padding: 10px;
    border-top: 1px solid #ccc;
    background-color: rgba(255, 255, 255, 0.7); /* Прозорий фон для
підвалу */
}
.chat-input {
    width: calc(100% - 20px);
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 3px;
    resize: none;
}
.send-button {
    width: 100px; /* Збільшуємо ширину кнопки */
    padding: 12px; /* Збільшуємо відступи всередині кнопки */
    margin-left: 10px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}
</style>
</head>

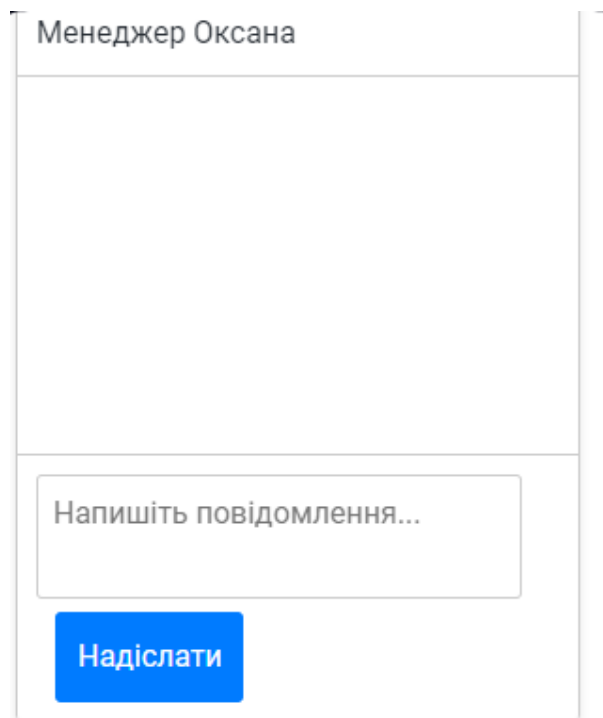
```

<body>

JavaScript для форми зворотного зв'язку:

```
// Функція для відображення чату
function showChat() {
    var chatContainer = document.getElementById('chat-container');
    chatContainer.innerHTML = `
        <div id="chat">
            <div id="chat-messages"></div>
            <input type="text" id="chat-input" placeholder="Напишіть
повідомлення...">
            <button onclick="sendMessage()">Надіслати</button>
        </div>
    `;
}
// Функція для надсилання повідомлення
function sendMessage() {
    var messageInput = document.getElementById('chat-input');
    var message = messageInput.value;
}
```

На (рис. 4.8) нижче зображено чат зворотнього зв'язку:



Менеджер Оксана

Напишіть повідомлення...

Надіслати

Рисунок 4.8 Зворотній зв'язок

Конфігуратор автомобіля дозволяє користувачам персоналізувати автомобіль за різними параметрами.

Приклад коду HTML для конфігуратора автомобіля:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Замовити автомобіль</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/main.css">
  <style>
    /* Ваші стилі */
  </style>
</head>
<body>
  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="index.html">Головна</a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="order-a-
car.html">Замовити автомобіль</a>
        </li>
      </ul>
    </div>
  </nav>

  <!-- Main Content -->
  <main class="container my-5">
    <div class="row">
      <div class="col-md-6 mx-auto">
        <h2 class="text-center mb-4">Замовити автомобіль</h2>
        <!-- Форма замовлення -->
        <form id="orderForm">
          <!-- Вибір автомобіля -->
          <div class="form-group">
            <label for="carModel">Оберіть модель
автомобіля:</label>
```

```

name="carModel">
xDrive</option>
RAV4</option>
Avant</option>
E213</option>
Taycan</option>
Mondeo</option>
srt</option>
Challenger</option>
Octavia RS</option>
Benz AMG</option>
Series GT</option>
GT</option>
Pajero">Mitsubishi Pajero</option>
Mazda3</option>
Monza</option>
Fusion</option>
Ranger</option>
S60</option>
XC90</option>
XC60</option>
XE</option>
<select class="form-control" id="carModel"
<option value="BMW 750 xDrive">BMW 750
<option value="Toyota RAV4">Toyota
<option value="Audi A6 Avant">Audi A6
<option value="Mercedes E213">Mercedes
<option value="Porsche Taycan">Porsche
<option value="Ford Mondeo">Ford
<option value="Dodge viper srt">Dodge viper
<option value="Dodge Challenger">Dodge
<option value="Škoda Octavia RS">Škoda
<option value="Mercedes-Benz AMG">Mercedes-
<option value="BMW 3 Series GT">BMW 3
<option value="BMW-320i">BMW-320i</option>
<option value="BMW F-30">BMW F-30</option>
<option value="Audi e-tron GT">Audi e-tron
<option value="Mitsubishi
Pajero">Mitsubishi Pajero</option>
<option value="BMW X8">BMW X8</option>
<option value="2017 Mazda3">2017
<option value="BMW M4">BMW M4</option>
<option value="Opel Monza">Opel
<option value="Ford Fusion">Ford
<option value="Ford Ranger">Ford
<option value="Volvo S60">Volvo
<option value="Volvo XC90">Volvo
<option value="Volvo XC60">Volvo
<option value="Jaguar XE">Jaguar

```

```

        <option value="Jaguar XJ">Jaguar
XJ</option>
        <option value="Lamborghini
Revuelto">Lamborghini Revuelto</option>
        <!-- Додайте інші варіанти автомобілів тут
-->
    </select>
</div>
<!-- Вибір комплектації -->
<div class="form-group">
    <label for="configuration">Оберіть
комплектацію:</label>
    <select class="form-control" id="configuration"
name="configuration">
        <option value="Standard">Стандарт</option>
        <option value="Premium">Преміум</option>
        <!-- Додайте інші варіанти комплектації тут
-->
    </select>
</div>
<!-- Ваше ім'я -->
<div class="form-group">
    <label for="fullName">Ваше повне ім'я:</label>
    <input type="text" class="form-control"
id="fullName" name="fullName" required>
</div>
<!-- Ваш email -->
<div class="form-group">
    <label for="email">Ваш Email:</label>
    <input type="email" class="form-control"
id="email" name="email" required>
</div>
<!-- Ваш номер телефону -->
<div class="form-group">
    <label for="phone">Ваш номер телефону:</label>
    <input type="tel" class="form-control"
id="phone" name="phone" required>
</div>
<!-- Кнопка відправки -->
<button type="button" class="btn btn-primary btn-
block" onclick="submitOrder()">Відправити замовлення</button>
</form>
</div>
</div>
</main>
<!-- Підвал -->
<footer class="footer mt-auto py-3">
    <div class="container">

```

```

        <span class="text-muted">© 2024 Автосалон</span>
    </div>
</footer>
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script>
    function submitOrder() {
        // Спочатку можна провести перевірку введених даних та їх
        // валідацію, якщо необхідно
        // Сповіщення про прийняття замовлення
        alert("Вашу заявку прийнято. Найближчим часом з вами
        зв'яжеться наш менеджер. Дякуємо, що вибрали нас!");
        // Очищення форми після відправки
        document.getElementById("orderForm").reset();
    }
</script>
</body>
</html>

```

На (рис. 4.9) зображено конфігуратор автомобіля, відфільтровування за брендом, моделлю, комплектацією та ціновим діапазоном:

Головна

Замовити автомобіль

Оберіть бренд:

Всі моделі
▼

Оберіть комплектацію:

Стандарт
▼

Оберіть модель:

Введіть бренд
▼

Ціновий діапазон:

Мінімальна ціна
▼

Максимальна ціна
▼

Фільтрувати

Рисунок 4.9 Конфігуратор автомобіля

4.3.1 Реалізація форм для замовлення автомобілів або запису на тест-драйв

Форми для замовлення автомобілів або запису на тест-драйв є важливими елементами веб-сайту автосалону, оскільки вони дозволяють користувачам легко взаємодіяти з компанією та оформлювати свої запити. Нижче наведено приклади реалізації таких форм з використанням HTML, CSS та JavaScript [7, 9, 14].

Форма для замовлення автомобіля – програмний код реалізацій в Додатку Б. На (рис. 4.10) зображено форму запису на тест-драйв:

The screenshot displays a web application titled "Оренда авто" (Car Rental). The navigation bar includes links for "ГОЛОВНА" (Home), "АВТОПАРК" (Fleet), "ПРО НАС" (About Us), and "КОНТАКТИ" (Contacts). The main content area is divided into two sections. On the left, there is a list of cars for rent, each with a small image, the car model, the daily price, a brief description, and a green "Орендувати" (Rent) button. The cars listed are Toyota RAV4 (1000 грн/день), Volvo XC90 (1200 грн/день), and Ford Fusion. On the right, there is a form titled "Форма оренди авто" (Car Rental Form). The form contains input fields for "Ім'я" (Name), "Електронна пошта" (Email), and "Телефон" (Phone). It also has date pickers for "Дата початку оренди" (Start date) and "Дата завершення оренди" (End date), both set to "дд.мм.rrrr". A green "Оформити" (Submit) button is located at the bottom of the form.

Рисунок 4.10 Запис на тест-драйв

4.3.2 Оптимізація сайту для роботи на різних пристроях (респонсивний дизайн)

Респонсивний дизайн (адаптивний дизайн) забезпечує коректне відображення та функціонування веб-сайту на різних пристроях, таких як десктопи, планшети та смартфони. Це дозволяє користувачам отримати

позитивний досвід незалежно від того, який пристрій вони використовують. Нижче наведено основні принципи та приклади реалізації респонсивного дизайну для веб-сайту автосалону [16].

- Медіа-запити (Media Queries)

Медіа-запити дозволяють застосовувати різні стилі CSS в залежності від характеристик пристрою, таких як ширина екрану, висота екрану, орієнтація та інші.

Приклад медіа-запитів для адаптивного дизайну:

```
/* Основні стилі */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}
header, nav, main, footer {
    padding: 1em;
}
header {
    background: #333;
    color: white;
    text-align: center;
}
nav ul {
    list-style: none;
    padding: 0;
    display: flex;
    justify-content: center;
    background: #444;
}
nav ul li {
    margin: 0 1em;
}
nav ul li a {
    color: white;
    text-decoration: none;
}
main {
    padding: 2em;
    text-align: center;
}
footer {
    background: #333;
    color: white;
    text-align: center;
    padding: 1em 0;
}
```

```
/* Адаптивні стилі */
@media (max-width: 768px) {
    nav ul {
        flex-direction: column;
        align-items: center;
    }
}
@media (max-width: 480px) {
    main {
        padding: 1em;
    }
}
```

- Гнучкі зображення та медіа

Використання гнучких зображень та медіа гарантує, що вони автоматично підлаштовуються під розмір контейнера.

Приклад гнучких зображень:

```
css
img {
    max-width: 100%;
    height: auto;
}
```

- Адаптивні типографічні рішення

Типографіка повинна бути читабельною на всіх пристроях, що досягається за допомогою відносних одиниць вимірювання, таких як em та rem.

Приклад адаптивної типографіки:

```
css
body {
    font-size: 16px; / Базовий розмір шрифту /
}
```

4.4 Тестування та валідація

4.4.1 Проведення тестування сайту на різних браузерях

Автоматизоване тестування дозволяє автоматично перевіряти сумісність сайту на різних браузерах за допомогою спеціальних інструментів та скриптів [17-18].

Приклад використання BrowserStack з Selenium:

1. Встановлення Selenium WebDriver.

```
bash
pip install selenium
```

2. Скрипт для тестування веб-сайту.

```
python
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
Введіть ваші дані BrowserStack
USERNAME = 'your_browserstack_username'
ACCESS_KEY = 'your_browserstack_access_key'
URL = f"https://{USERNAME}:{ACCESS_KEY}@hub-
cloud.browserstack.com/wd/hub"
desired_cap = {
    'os': 'Windows',
    'os_version': '10',
    'browser': 'Chrome',
    'browser_version': 'latest',
    'name': 'BrowserStack Test'    Назва тесту
}
driver = webdriver.Remote(
    command_executor=URL,
    desired_capabilities=desired_cap
)
driver.get("https://www.yourwebsite.com")
Перевірка заголовка сторінки
assert "Your Website Title" in driver.title
Закриття браузерa
driver.quit()
```

3. Запустити скрипт для виконання автоматизованого тестування.

4. Тестування на мобільних пристроях

Сайт правильно відображається і функціонує на мобільних пристроях.

- Chrome DevTools: Інструмент для емуляції мобільних пристроїв у браузері Chrome.

- BrowserStack: Платформа для тестування на реальних мобільних пристроях.

4.4.2 Валідація HTML та CSS коду

Валідація HTML та CSS коду є важливим кроком у забезпеченні якості та коректності веб-сайту. Валідація допомагає виявити помилки, які можуть вплинути на відображення та функціональність сайту. Нижче наведено основні кроки для валідації HTML та CSS коду.

1. Валідація HTML коду

Для валідації HTML коду використовується W3C Markup Validation Service. Цей інструмент перевіряє HTML код на відповідність стандартам W3C і вказує на помилки та попередження.

Приклад HTML коду для валідації:

```
html
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Автосалон "АвтоСвіт"</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <div class="logo">
      
    </div>
    <nav>
      <ul>
        <li><a href="home">Головна</a></li>
        <li><a href="catalog">Каталог автомобілів</a></li>
        <li><a href="services">Послуги</a></li>
        <li><a href="news">Новини</a></li>
        <li><a href="contacts">Контакти</a></li>
      </ul>
    </nav>
  </header>
  <!-- Інші розділи сайту -->
</body>
</html>
```

2. Валідація CSS коду

Для валідації CSS коду використовується W3C CSS Validation Service. Цей інструмент перевіряє CSS код на відповідність стандартам W3C та вказує на помилки та попередження.

Кроки для валідації CSS:

- Відкрити [W3C CSS Validation Service](https://jigsaw.w3.org/css-validator/).
- Ввести URL вашого веб-сайту або завантажити файл CSS для перевірки.
- Натиснути кнопку "Check" для початку валідації.
- Переглянути результати та виправити помилки, якщо вони є.

Приклад CSS коду для валідації:

```
css
/ Загальні стилі /
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: f5f5f5;
}
header {
    background-color: 333;
    color: fff;

    padding: 20px;

    display: flex;

    justify-content: space-between;
    align-items: center;
}
header .logo img {
    width: 150px;
}
header nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
    display: flex;
}
header nav ul li {
    margin-left: 20px;
}
header nav ul li a {
    color: fff;
    text-decoration: none;
    font-weight: bold;
```

```

}
header nav ul li a:hover {
    text-decoration: underline;
}

```

3. Автоматизована валідація

Автоматизована валідація дозволяє інтегрувати перевірку коду в процес розробки, забезпечуючи регулярний контроль якості.

Використання linting інструментів:

- HTMLHint: Інструмент для перевірки HTML коду на відповідність стандартам.

- stylelint: Інструмент для перевірки CSS коду на відповідність стандартам.

Приклад використання HTMLHint:

1. Встановити HTMLHint.

```
bash
```

```
npm install htmlhint --save-dev
```

2. Створити файл конфігурації `.htmlhintc``.

```
json
```

```

{
  "tagname-lowercase": true,
  "attr-lowercase": true,
  "attr-value-double-quotes": true,
  "doctype-first": true,
  "tag-pair": true,
  "spec-char-escape": true,
  "id-unique": true,
  "src-not-empty": true,
  "attr-no-duplication": true
}

```

3. Запустіть HTMLHint для перевірки HTML файлів.

```
bash
```

```
npx htmlhint yourfile.html
```

Приклад використання stylelint:

1. Встановити stylelint.

```
bash
```

```
npm install stylelint stylelint-config-standard --save-dev
```

2. Створити файл конфігурації `.stylelintrc.json`.

```
json
{
  "extends": "stylelint-config-standard",
  "rules": {
    "block-no-empty": true,
    "color-no-invalid-hex": true,
    "declaration-colon-space-after": "always",
    "max-empty-lines": 2,
    "unit-whitelist": ["em", "rem", "%", "s"]
  }
}
```

3. Запустити stylelint для перевірки CSS файлів.

```
bash
npx stylelint "/*.css"
```

4.5 Запуск сайту на сервері MongoDB

1. Встановлення Node.js та npm.

2. Перейти на [офіційний сайт Node.js](https://nodejs.org/) і завантажити останню LTS версію [10].

3. Встановлення Node.js, слідуючи інструкціям інстальатора. Це також встановить npm (Node Package Manager).

4. Підготовка проекту

- Створення нової папки для проекту, наприклад, `myproject` та перейти до неї:

```
sh
mkdir myproject
cd myproject
```

- Ініціалізувати новий проект Node.js, створивши файл `package.json`:

```
sh
npm init -y
```

- Встановлення необхідних залежностей:

```
sh
npm install express mongoose
```

5. Створіть наступну структуру директорій і файлів у вашій папці “myproject”:

```
myproject/
├── public/
│   └── index.html
├── server.js
└── package.json
```

2. Створення файлу `index.html` у папці `public` з наступним вмістом:

```
html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>My Project</title>
</head>
<body>
  <h1>Welcome to My Project</h1>
</body>
</html>
```

3. Створення файлу `server.js` з наступним вмістом:

```
javascript
const express = require('express');
const mongoose = require('mongoose');
const path = require('path');
const app = express();
const port = 3000;
// Підключення до MongoDB
mongoose.connect('mongodb://localhost:27017/mydatabase', {
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(() => {
  console.log('Підключено до MongoDB');
}).catch((err) => {
  console.error('Помилка підключення до MongoDB:', err);
});
// Обслуговування статичних файлів з папки public
app.use(express.static(path.join(__dirname, 'public')));
// Маршрут для головної сторінки
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'index.html'));
});
// Запуск сервера
app.listen(port, () => {
  console.log(`Сервер працює на http://localhost:${port}`);
});
```

Запуск сервера

1. Зайти у директорію `myproject`.

2. Запустити сервер:

```
sh
```

```
npm start
```

Перевірка роботи сервера

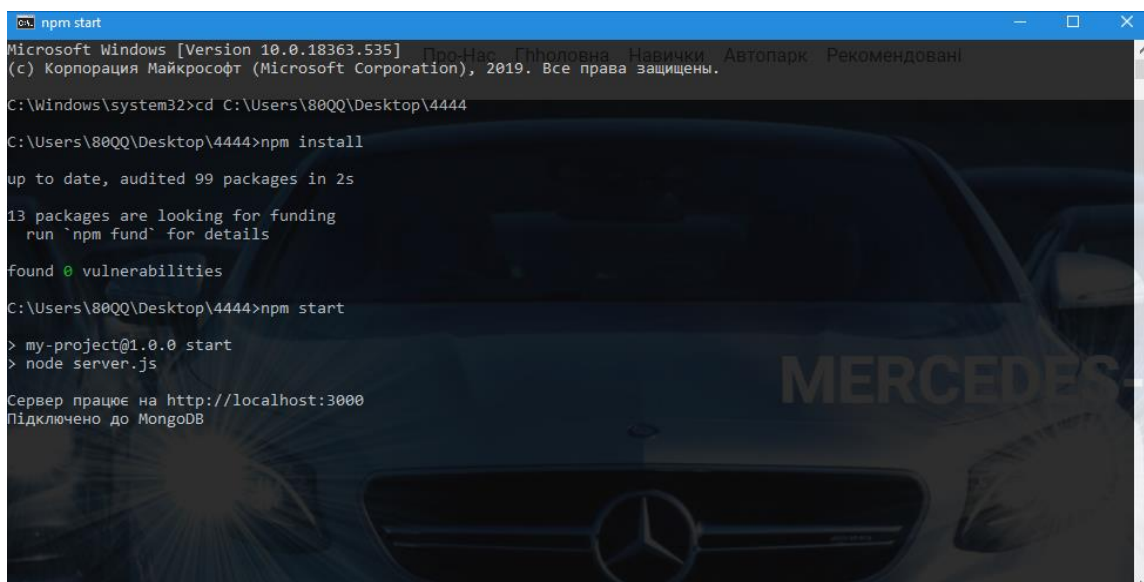
- Відкрити веб-браузер і перейти за адресою “http://localhost:3000”.
- Ви повинні побачити вміст вашого файлу “index.html”.

Підсумкова структура проекту

Після виконання всіх кроків ваша структура проекту повинна виглядати так:

```
myproject/
├─ public/
|   └─ index.html
├─ server.js
└─ package.json
```

На (рис. 4.11) зображено командний рядок, в якому введені команди для запуску сервера MongoDB:



```

C:\Windows\system32>cd C:\Users\80QQ\Desktop\4444
C:\Users\80QQ\Desktop\4444>npm install
up to date, audited 99 packages in 2s
13 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
C:\Users\80QQ\Desktop\4444>npm start
> my-project@1.0.0 start
> node server.js
Сервер працює на http://localhost:3000
Підключено до MongoDB
  
```

Рисунок 4.11 Запуск сервера MongoDB

4.6 Аналіз можливих покращень

Аналіз можливих покращень веб-сайту є важливим етапом для забезпечення його ефективності, зручності та конкурентоспроможності. Нижче наведено кілька ключових аспектів, які можуть бути покращені для підвищення загальної якості веб-сайту автосалону.

1. Покращення дизайну та інтерфейсу користувача (UI/UX)

- Зручність навігації:
 - Спрощення меню навігації та додавання випадаючих списків для швидкого доступу до підрозділів.
 - Використання breadcrumb навігації для покращення користувацького досвіду при переході між сторінками.
- Анімації та мікровзаємодії:
 - Додавання анімацій та мікровзаємодій для покращення візуального сприйняття та взаємодії з елементами сайту.
 - Використання плавних переходів та ефектів наведенні на кнопки та посилання.
- Адаптивний дизайн:
 - Оптимізація адаптивного дизайну для кращого відображення на різних пристроях, включаючи мобільні телефони та планшети.
 - Додавання спеціальних стилів для забезпечення зручного користування на невеликих екранах.

• Покращення функціональності

2. Фільтрація та пошук:

- Розширення можливостей фільтрації автомобілів за додатковими параметрами (наприклад, пробіг, тип палива, рік випуску).
- Додавання покращеного пошуку з підказками та автозаповненням.
- Конфігуратор автомобіля:

- Розширення функціоналу конфігуратора автомобіля, додавання більше опцій для персоналізації та інтерактивних елементів (наприклад, 3D огляд автомобіля).

- Інтеграція з системою бронювання для миттєвого резервування конфігурованого автомобіля.

- Інтерактивні елементи:

- Додавання інтерактивних турів та відеооглядів для кращого представлення автомобілів.

- Використання віртуальної реальності (VR) або доповненої реальності (AR) для демонстрації автомобілів.

3. Оптимізація продуктивності

- Швидкість завантаження сторінок:

- Оптимізація зображень та використання сучасних форматів (наприклад, WebP) для зменшення розміру файлів.

- Використання технік lazy loading для відкладеного завантаження зображень та інших медіа-контенту.

- Кешування:

- Впровадження кешування на стороні клієнта та сервера для прискорення завантаження часто використовуваних ресурсів.

- Використання CDN (Content Delivery Network) для розповсюдження контенту та зниження часу завантаження.

- Мінімізація коду:

- Мінімізація та об'єднання CSS та JavaScript файлів для зменшення кількості HTTP-запитів.

- Використання сучасних інструментів для зборки та оптимізації коду (наприклад, Webpack).

4. Покращення SEO

- Оптимізація контенту:

- Використання ключових слів у заголовках, мета-тегах та текстовому контенті для покращення пошукової видимості.

- Створення унікального та інформативного контенту, який відповідає запитам користувачів.

- Структуровані дані:

- Використання структурованих даних (Schema.org) для покращення відображення інформації у пошукових системах.

- Додавання мікророзмітки для покращення видимості у пошукових системах та підвищення CTR (click-through rate).

- Мобільна оптимізація:

- Переконавання, що сайт повністю оптимізований для мобільних пристроїв, оскільки мобільна версія впливає на рейтинги в пошукових системах.

- Використання Google Mobile-Friendly Test для перевірки мобільної оптимізації.

5. Підвищення безпеки

- SSL сертифікат:

- Забезпечення наявності та правильного налаштування SSL сертифікату для шифрування даних та захисту користувацької інформації.

- Захист від атак:

- Використання інструментів для захисту від атак типу SQL injection, XSS (cross-site scripting) та інших вразливостей.

- Впровадження системи моніторингу та регулярного оновлення програмного забезпечення для захисту від нових загроз.

ВИСНОВКИ

Створення веб-сайту для автосалону є комплексним процесом, який включає декілька етапів, від початкового аналізу до фінального розгортання та демонстрації робочого продукту. У цьому проекті успішно розроблений веб-сайт, який відповідає вимогам сучасних стандартів та забезпечує зручний і функціональний інтерфейс для користувачів.

Розглянемо основні результати, які були зроблені під час виконання проекту:

- Проведено вибір технологій реалізації проєкту: HTML5 та CSS3 забезпечили створення структурованого та стильного контенту, що відповідає сучасним веб-стандартам. Використання JavaScript дозволило додати інтерактивні елементи, такі як слайдери, форми та конфігуратори автомобілів. Медіа-запити та Flexbox/Grid технології сприяли створенню адаптивного дизайну, який коректно відображається на різних пристроях.
- Розроблено функціонал сайту: реалізовані форми для замовлення автомобілів та запису на тест-драйв з валідацією даних та інтерактивними повідомленнями. Інтерактивні елементи, такі як слайдер зображень та випадаючі меню, забезпечили зручну та привабливу взаємодію з користувачами. Функціональність сайту була протестована на різних браузерах та пристроях для забезпечення сумісності та коректної роботи.
- Проведено оптимізація та продуктивність програмного продукту: впроваджені техніки оптимізації, такі як кешування, lazy loading та мінімізація коду, сприяли підвищенню швидкості завантаження сторінок. Було перевірено коректність HTML та CSS коду за допомогою валідаційних інструментів W3C, що дозволило виявити та виправити можливі помилки.
- Проведення покращення та аналіз сайту: аналіз можливих покращень показав, що існує потенціал для подальшого розвитку сайту, включаючи покращення UI/UX, розширення функціональності, оптимізацію

SEO та підвищення безпеки. Регулярне оновлення та впровадження нових технологій допоможуть зберігати сайт актуальним та конкурентоспроможним.

- Демонстрація результатів: робочий веб-сайт був успішно продемонстрований, що дозволило наочно показати його функціональність та дизайн. Підготовка та проведення демонстрації забезпечили позитивне враження та підтвердили відповідність сайту встановленим вимогам.

Проект з розробки веб-сайту автосалону був успішно реалізований завдяки чітко визначеним етапам роботи, використанню сучасних технологій та постійному тестуванню і оптимізації. Результатом став функціональний, зручний та привабливий веб-сайт, який відповідає потребам бізнесу та користувачів.

Регулярний аналіз та впровадження покращень дозволять зберігати високий рівень якості та забезпечувати успішне функціонування веб-сайту в майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ольховська О. В. Методичні рекомендації до виконання кваліфікаційної роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» ступеня бакалавра / О. В. Ольховська, О. О. Черненко – Полтава: ПУЕТ, 2022. – 71с.
2. Tim Berners-Lee. "Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor". HarperSanFrancisco. 2000. 246 p.
3. Thomas A. Powell. "The Complete Reference HTML & CSS". McGraw-Hill Education. 2017. 880 p.
4. Dave Chaffey, Fiona Ellis-Chadwick. "Digital Marketing: Strategy, Implementation and Practice". Pearson Education. 2019. 712 p.
5. Philip Kotler, Kevin Lane Keller. "Marketing Management". Pearson Education. 2016. 816 p.
6. Ryan Deiss, Russ Henneberry. "Digital Marketing for Dummies". Wiley. 2020. 368 p.
7. Jon Duckett. "HTML and CSS: Design and Build Websites". Wiley. 2011. 512 p.
8. Martin Murray. "The Automotive Industry and the Fourth Industrial Revolution". Springer. 2021. 250 p.
9. Jon Duckett. "HTML and CSS: Design and Build Websites". Wiley. 2011. 512 p.
10. Brad Dayley, Brendan Dayley, Caleb Dayley. "Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications". Addison-Wesley Professional. 2017. 640 p.
11. Steve Krug. "Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability". New Riders. 2014. 216 p.
12. Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel. "About Face: The Essentials of Interaction Design". Wiley. 2014. 720 p.

13. Joel Marsh. "UX for Beginners: A Crash Course in 100 Short Lessons". O'Reilly Media. 2015. 256 p.
14. Ethan Brown. "Learning JavaScript: Add Sparkle and Life to Your Web Pages". O'Reilly Media. 2016. 352 p.
15. David Flanagan. "JavaScript: The Definitive Guide". O'Reilly Media. 2021. 706 p.
16. Ethan Marcotte. "Responsive Web Design". A Book Apart. 2011. 144 p.
17. Dustin Mitchell. "Selenium Testing Tools Cookbook". Packt Publishing. 2012. 328 p.
18. Paul Grizzaffi. "Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing". Wiley. 2009. 480 p.

ДОДАТОК А

Приклад HTML-коду для головної сторінки:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/main.css">
  <link href="https://fonts.googleapis.com/css?family=Courgette"
rel="stylesheet">
  <script src="js/all.js"></script>
  <title>Запис в автосалон</title>
</head>
<body>
  <nav class="navbar navbar-expand-lg px-3" id="navBar">
    <a href="#" class="navbar-brand">
      
    </a>
    <button class="navbar-toggler" type="button" data-toggle='collapse'
data-target='#myNav'>
      <span class="navbar-icon">
        <i class="fas fa-bars"></i>
      </span>
    </button>
    <div class="collapse navbar-collapse" id="myNav">
      <ul class="navbar-nav mx-auto">
        <li class="nav-item active">
          <a href="#navBar" class="nav-link">Головна</a>
        </li>
        <li class="nav-item">
          <a href="#skills" class="nav-link">Навички</a>
        </li>
        <li class="nav-item">
          <a href="#inventory" class="nav-link">Автопарк</a>
        </li>
        <li class="nav-item">
          <a href="#featured" class="nav-link">Рекомендовані</a>
        </li>
      </ul>
      <div class="nav-icons d-none d-lg-block">
        <a href="#" class="nav-icon">
          <i class="fab fa-facebook"></i>
        </a>
        <a href="#" class="nav-icon">
          <i class="fab fa-twitter"></i>
        </a>
      </div>
    </div>
  </nav>
```

```

        </a>
        <a href="#" class="nav-icon">
            <i class="fab fa-instagram"></i>
        </a>
    </div>
</div>
</nav>
<header class="header" id="header">
    <div class="container-fluid">
        <div class="row height-max align-items-center">
            <div class="col col-md-9 ml-auto text-right pr-5">
                <h5 class="title-heading d-inline-block p-2 text-
uppercase">Ton-1</h5>
                <h2 class="text-uppercase my-2 title">Mercedes-Benz 2018</h2>
                <h3 class="text-uppercase">E213</h3>
                <h6 class="text-capitalize">ціна від</h6>
                <h2>$60,000 <a href="" class="title-icon d-inline-block mx2">
                    <i class="fas fa-play"></i>
                </a></h2>
            </div>
        </div>
    </div>
</div>
</header>
<section class="skills py-5" id="skills">
    <div class="container">
        <div class="row mb-5">
            <div class="col d-flex flex-wrap text-uppercase justify-
content-center">
                <h1 class="font-weight-bold align-self-center mx-1">Чому
обирають нас</h1>
                <h1 class="section-title--special mx-1">Автодилер</h1>
            </div>
        </div>
        <div class="row">
            <div class="col-sm-6 col-lg-3 text-center my-3">
                <a href="all-cars.html" class="skills-icon p-2 rounded-
circle">
                    <i class="fas fa-car fa-fw"></i>
                </a>
                <h6 class="text-uppercase font-weight-bold my-3">Уci
марки</h6>
                <div class="skills-underline"></div>
            </div>
            <div class="col-sm-6 col-lg-3 text-center my-3">
                <a href="#" onclick="openSupportWindow()" class="skills-icon
p-2 rounded-circle">
                    <i class="fas fa-comments fa-fw"></i>
                </a>
                <h6 class="text-uppercase font-weight-bold my-3">Безкоштовна
підтримка</h6>
                <div class="skills-underline"></div>
            </div>
        </div>
    </div>
</section>

```

```

    </div>
    <div class="col-sm-6 col-lg-3 text-center my-3">
      <a href="rental-cars.html" class="skills-icon p-2 rounded-
circle">
        <i class="fas fa-taxi fa-fw"></i>
      </a>
      <h6 class="text-uppercase font-weight-bold my-3">Аренда
авто</h6>
      <div class="skills-underline"></div>
    </div>
    <div class="col-sm-6 col-lg-3 text-center my-3">
      <a href="rental-cars.html" class="skills-icon p-2 rounded-
circle">
        <i class="fas fa-wallet fa-fw"></i>
      </a>
      <h6 class="text-uppercase font-weight-bold my-
3">Доступність</h6>
      <div class="skills-underline"></div>
    </div>
  </div>
</div>
</section>
<section class="inventory py-5" id="inventory">
  <div class="container">
    <div class="row mb-5">
      <div class="col d-flex flex-wrap text-uppercase justify-
content-center">
        <h1 class="font-weight-bold align-self-center mx-1">Наш</h1>
        <h1 class="section-title--special mx-1">Автопарк</h1>
      </div>
    </div>
    <div class="row mb-5">
      <div class="col-10 mx-auto col-md-12 d-flex justify-content-
end">
        <button class="btn btn-outline-secondary text-uppercase mx-1
btn-filter" data-category="all">Усі</button>
        <button class="btn btn-outline-secondary text-uppercase mx-1
btn-filter" data-category="american">Американські</button>
        <button class="btn btn-outline-secondary text-uppercase mx-1
btn-filter" data-category="german">Німецькі</button>
      </div>
    </div>
    <div class="row" id="car-container">
      <!-- Приклади карток автомобілів -->
      <div class="col-10 mx-auto my-3 col-md-6 col-lg-4">
        <div class="card car-card">
          
          <div class="card-body">
            <div class="car-info d-flex justify-content-between">
              <div class="car-text text-uppercase">

```

```

        <h6 class="font-weight-bold">Mercedes-Benz</h6>
        <h6>AMG GT 2015</h6>
    </div>
    <h5 class="car-value align-self-center py-2 px-3">$
        <span class="car-price">54,000</span>
    </h5>
</div>
</div>
<div class="card-footer text-capitalize d-flex justify-
content-between">
    <p><span><i class="fas fa-car"></i></span> Седан</p>
    <p><span><i class="fas fa-cogs"></i></span> Автомат</p>
    <p><span><i class="fas fa-gas-pump"></i></span> 4.0</p>
</div>
</div>
</div>
<!-- Інші картки автомобілів можна додати тут -->
</div>
</div>
</section>
<footer class="footer py-5">
    <div class="footer-icons d-flex justify-content-around">
        <a href="#" class="footer-icon">
            <i class="fab fa-facebook"></i>
        </a>
        <a href="#" class="footer-icon">
            <i class="fab fa-twitter"></i>
        </a>
        <a href="#" class="footer-icon">
            <i class="fab fa-instagram"></i>
        </a>
        <a href="#" class="footer-icon">
            <i class="fab fa-linkedin"></i>
        </a>
        <a href="#" class="footer-icon">
            <i class="fab fa-google-plus"></i>
        </a>
    </div>
</footer>
<script src="js/jquery-3.3.1.min.js"></script>
<script src="js/bootstrap.bundle.min.js"></script>
<script src="js/script.js"></script>
<script>
    $(document).ready(function() {
        var cars = $('<div>.car-card</div>').detach();
        $('<div>.btn-filter</div>').click(function() {
            var category = $(this).data('category');
            var filteredCars = (category === 'all') ? cars :
cars.filter(function() {
                return $('<div>img</div>', this).attr('src').includes(category);
            });
        });
    });
</script>

```

```

        $('#car-container').empty().append(filteredCars);
    });
    $('#car-container').append(cars);
});
function openSupportWindow() {
    var windowWidth = 350;
    var windowHeight = 500;
    var screenHeight = window.screen.height;
    var supportWindow = window.open('technical-support.html',
'_blank', 'width=' + windowWidth + ',height=' + windowHeight);
    var leftPosition = 1000;
    var topPosition = screenHeight - windowHeight;
    supportWindow.moveTo(leftPosition, topPosition);
}
</script>
<script>
    $(document).ready(function() {
        $('.featured-item').click(function() {
            var imgPath = $(this).data('src');
            $('.featured-photo').attr('src', imgPath);
        });
    });
</script>
</body>
</html>

```

ДАДАТОК Б

HTML для форми замовлення автомобіля:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Оренда авто</title>
  <style>
</style>
  <script>
    function showForm() {
      document.getElementById('rentForm').style.display =
'block';
    }
  </script>
</head>
<body>
  <header class="header">
    <h1>Оренда авто</h1>
  </header>
  <nav class="nav">
    <div class="container">
      <a href="index.html">Головна</a>
      <a href="all-cars.html">Автопарк</a>
      <a href="about.html">Про нас</a>
      <a href="technical-support.html">Контакти</a>
    </div>
  </nav>
  <div class="container">
    <div class="car-list-container">
      <ul class="car-list">
        <li>
          <div class="car">
            
            <div class="car-details">
              <h3>Toyota RAV4</h3>
              <p>Ціна: <span>1000 грн/день</span></p>
              <p>Опис: Toyota RAV4 - це компактний
кросовер, який виробляється компанією Toyota.</p>
              <button class="rent-button"
onclick="showForm()">Орендувати</button>
            </div>
          </div>
        </li>
        <li>
          <div class="car">
```

```

        
        <div class="car-details">
            <h3>Volvo XC90</h3>
            <p>Ціна: <span>1200 грн/день</span></p>
            <p>Опис: Volvo XC90 - це преміум
позашляховик середнього розміру, що виробляється шведською
автовиробницею Volvo Cars.</p>
            <button class="rent-button"
onclick="showForm()">Орендувати</button>
        </div>
    </div>
</li>
<li>
    <div class="car">
        
        <div class="car-details">
            <h3>Ford Fusion</h3>
            <p>Ціна: <span>1100 грн/день</span></p>
            <p>Опис: Ford Fusion - це сімейство
автомобілів, вироблених американською компанією Ford Motor Company.</p>
            <button class="rent-button"
onclick="showForm()">Орендувати</button>
        </div>
    </div>
</li>
</ul>
</div>
<div class="form-container">
    <!-- Форма оренди авто -->
    <div id="rentForm" class="rent-form">
        <h3>Форма оренди авто</h3>
        <form action="/submit_rent" method="post">
            <label for="name">Ім'я</label>
            <input type="text" id="name" name="name" required>

            <label for="email">Електронна пошта</label>
            <input type="email" id="email" name="email"
required>

            <label for="phone">Телефон</label>
            <input type="tel" id="phone" name="phone" required>

            <label for="start_date">Дата початку оренди</label>
            <input type="date" id="start_date"
name="start_date" required>

            <label for="end_date">Дата завершення
оренди</label>

```

```

        <input type="date" id="end_date" name="end_date"
required>

        <input type="submit" value="Відправити">
    </form>
</div>
</div>
</div>
</div>
</body>
</html>

```

CSS для форми замовлення автомобіля на тест-драйв:

```

    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
    display: flex;
    flex-direction: column;
    height: 100vh;
}
.header {
    background-color: #333;
    color: #fff;
    padding: 20px 0;
    text-align: center;
}
.header h1 {
    margin: 0;
}
.container {
    flex: 1;
    display: flex;
    overflow: hidden;
}
.nav {
    background: #333;
    color: #fff;
    padding: 15px;
    text-align: center;
}

.nav a {
    color: #fff;
    padding: 14px 20px;
    text-decoration: none;
    text-transform: uppercase;
}
.nav a:hover {
    background: #575757;
}
.car-list-container {

```

```

        flex: 1;
        overflow-y: auto;
        background: #f4f4f4;
        padding: 20px;
        box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);
    }
    .car-list {
        list-style: none;
        padding: 0;
    }
    .car-list li {
        background: #fff;
        margin-bottom: 10px;
        padding: 10px;
        border: 1px solid #ddd;
        display: flex;
        align-items: center;
    }
    .car-list li:hover {
        background: #e9e9e9;
    }
    .car img {
        width: 100px;
        margin-right: 20px;
    }
    .car-details {
        flex-grow: 1;
    }
    .car-details h3 {
        margin: 0 0 10px 0;
    }
    .car-details p {
        margin: 0 0 5px 0;
    }
    .car-details span {
        font-weight: bold;
    }
    .rent-button {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        margin-top: 10px;
        cursor: pointer;
        border: none;
    }
    .rent-button:hover {
        background-color: #45a049;
    }

```

```

.form-container {
  flex: 1;
  padding: 20px;
  background: #fff;
}
.rent-form {
  display: none;
  background: #f4f4f4;
  padding: 20px;
  border: 1px solid #ddd;
}
.rent-form label {
  display: block;
  margin-bottom: 5px;
}
.rent-form input[type="text"],
.rent-form input[type="email"],
.rent-form input[type="date"],
.rent-form input[type="tel"] {
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
}
.rent-form input[type="submit"] {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}
.rent-form input[type="submit"]:hover {
  background-color: #45a049;
}
.footer {
  background: #333;
  color: #fff;
  text-align: center;
  padding: 10px 0;
}

```