

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА САЙТУ АВТОЗАПРАВКИ ІЗ
ЗАСТОСУВАННЯМ NODE.JS ТА MYSQL»

Виконав: студент 3 курсу, групи 6.1211-пі-с
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми програмна інженерія (зі скороченим
терміном навчання)
(назва освітньої програми)

В.О. Чернишов

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
доцент, к.т.н. Лимаренко Ю.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри електроніки, інформаційних систем та
програмного забезпечення ІННІ ЗНУ,
доцент, к.т.н. Заяц В.І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія (зі скороченим терміном навчання)

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Чернишову Владиславу Олександровичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка сайту автозаправки із застосуванням Node.js та MySQL

керівник роботи Лимаренко Юлія Олексіївна, к.т.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Дослідження та аналіз предметної області.

2. Проектування вебсайту.

3. Розробка сайту.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

презентація до захисту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	07.02.2024	
3.	Обробка методичних та теоретичних джерел.	28.02.2024	
4.	Розробка першого та другого розділу.	16.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	21.06.2024	

Студент _____
(підпис)

В.О. Чернишов _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

Ю.О. Лимаренко _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка сайту автозаправки із застосуванням Node.js та MySQL»: 44 с., 17 рис., 13 джерел.

АВТЕНТИФІКАЦІЯ, ВЕБДИЗАЙН, ВЕБПРОГРАМУВАННЯ, ІНТЕРНЕТ-МАГАЗИН, ЛОГІКА БАЗ ДАНИХ, МОВА РОЗМІТКИ ГІПЕРТЕКСТУ, РОЗРОБКА ВЕБДОДАТКІВ, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, ТОРГІВЛЯ ОНЛАЙН, УПРАВЛІННЯ КОНТЕНТОМ, ХОСТИНГ ВЕБСАЙТІВ.

Об'єкт дослідження – процес розробки вебсайт АЗС Barrel.

Мета роботи: розробити функціональний та зручний вебсайт АЗС Barrel, який забезпечить ефективне управління контентом, зручний інтерфейс для користувачів та можливість здійснення онлайн-покупок.

Метод дослідження – розробка і вебсайт АЗС Barrel: HTML, CSS, JavaScript, MySQL.

Для розробки сайту та написання кваліфікаційної роботи були вивчені та розглянуті засоби для створення вебсайтів.

Під час виконання завдання були розглянуті основи створення сайтів, вивчена теорія, що пізніше була реалізована на практиці у вигляді створеного програмного продукту.

Результатом даної роботи є розробка вебсайт АЗС Barrel.

SUMMARY

Bachelor's qualifying paper "Development of a Filling Station Site Using Node.js and MySQL": 44 p., 17 figures, 13 references.

AUTHENTICATION, WEB DESIGN, WEB PROGRAMMING, ONLINE STORE, DATABASE LOGIC, HYPERTEXT MARKUP LANGUAGE, WEB APPLICATION DEVELOPMENT, DATABASE MANAGEMENT SYSTEM, ONLINE TRADING, CONTENT MANAGEMENT, WEBSITE HOSTING WORDPRESS ABLONS.

The object of the study is the development process of the website of the Barrel gas station.

The aim of the study is develop a functional and convenient Web site of the Barrel gas station, which will provide effective content management, a convenient interface for users and the ability to make online purchases.

The methods of research are the is using modern software development methods such as HTML, CSS, JavaScript, MySQL.

For the development of the website and the writing of the qualification paper, the tools for creating websites were studied and considered.

During the execution of the task, the basics of creating websites were considered, studied theory, which was later implemented in practice and in the form of a created software product.

The result of this work is the development of the Barrel gas station website.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	8
1 Аналіз предметної області та об'єкту дослідження	9
1.1 Характеристика предметної області	9
1.2 Огляд існуючих сайтів АЗС	10
1.3 Засоби для створення вебсайта.....	13
1.3.1 Огляд бази даних MySQL	14
1.4 Формування вимог до сайту	16
2 Розробка проєкту.....	18
2.1 Технічне завдання для вебсайту АЗС “Barrel”	18
2.2 Функціональні вимоги.....	19
2.2.1 Найменування і область застосування.....	19
2.2.2 Призначення розробки	21
2.2.3 Технічні вимоги до програмного продукту	22
2.3 Етапи створення вебсайту.....	23
2.3.1 Планування	23
2.3.2 Дизайн	24
2.3.3 Розробка	24
2.3.4 Тестування	25
2.3.5 Запуск.....	26
2.3.6 Підтримка та оновлення.....	26
2.4 Діаграма прецедентів.....	27
2.5 Схема бази даних	29
2.6 Проєктування інтерфейсу.....	30
2.6.1 Визначення вимог до інтерфейсу.....	30

2.6.2 Прототипування	31
2.6.3 Дизайн UI/UX	31
2.6.4 Тестування інтерфейсу	32
2.6.5 Впровадження та оновлення	33
3 Програмна реалізація	34
3.1 Вибір технологічного стеку	34
3.1.1 Розробка функціональності	35
3.1.2 Опис інструментарію	37
Висновки	43
Перелік посилань	44

ВСТУП

У сучасному світі вебтехнології відіграють ключову роль у веденні бізнесу. Вони не тільки спрощують управління бізнес-процесами, але й значно розширюють можливості для залучення клієнтів та їх обслуговування. Однією з галузей, де ці технології можуть принести особливу користь, є сфера автозаправних станцій (АЗС). Розробка функціонального та зручного вебсайту для АЗС може значно підвищити рівень сервісу та забезпечити компанії переваги у конкурентній боротьбі.

Темою цієї дипломної роботи є «Розробка вебсайту для АЗС Barrel». Обрана тема є важливою та актуальною, адже вебсайт для АЗС не лише слугуватиме електронним каталогом послуг та товарів, але й стане важливим інструментом для взаємодії з клієнтами та управління лояльністю. Ціль проєкту – створити інтерактивний, зручний для користувачів вебсайт, який допоможе підвищити впізнаваність бренду, оптимізувати клієнтський досвід та максимізувати продажі.

Для досягнення поставленої мети будуть використані сучасні технології веброзробки, включаючи HTML, CSS, JavaScript, а також системи управління базами даних. Важливою частиною проєкту стане дослідження потреб цільової аудиторії, що дозволить створити користувацький інтерфейс, який найкраще відповідає вимогам та побажанням клієнтів.

Очікується, що результатом цієї роботи стане повнофункціональний вебсайт, який не тільки задовольнить поточні потреби компанії, але й матиме можливість адаптації до майбутніх вимог ринку. Розробка такого сайту дозволить “Barrel” підняти якість обслуговування, підвищити зручність використання послуг та зміцнити свої позиції на ринку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБ'ЄКТУ ДОСЛІДЖЕННЯ

1.1 Характеристика предметної області

В сучасних умовах динамічного розвитку ринкових відносин та технологій, індустрія автозаправних станцій (АЗС) активно впроваджує цифрові рішення для забезпечення конкурентоспроможності, оптимізації бізнес-процесів та покращення взаємодії з клієнтами. Вебсайт для АЗС виступає не тільки як вітрина основної інформації про послуги і товари, але й як стратегічний інструмент у створенні цілісної, інтегрованої системи обслуговування клієнтів.

Розробка вебсайту для АЗС охоплює кілька ключових аспектів, включаючи технічну реалізацію, дизайн інтерфейсу, інтеграцію з внутрішніми системами, маркетинг та забезпечення безпеки. Для технічної реалізації використовуються сучасні технології, такі як HTML, CSS і JavaScript для фронтенду, а також серверні рішення для бекенду, що гарантує високу швидкість роботи та безпеку сайту. Дизайн інтерфейсу та UX/UI зосереджений на створенні інтуїтивно зрозумілого та легкого в навігації інтерфейсу, що забезпечує позитивний досвід користувачів. Важливою частиною проєкту є інтеграція сайту з внутрішніми ERP та CRM-системами, що дозволяє ефективно управляти запасами, клієнтською базою та логістикою. Маркетингова стратегія включає оптимізацію сайту для пошукових систем, розробку контент-стратегії та використання інструментів цифрового маркетингу для залучення нових клієнтів. Для забезпечення безпеки впроваджуються заходи, такі як шифрування даних, аутентифікація користувачів та захист від кібератак, що захищає дані користувачів і забезпечує безпечну роботу сайту. Також, важливо враховувати вплив зовнішніх і внутрішніх факторів на розвиток ринку АЗС, включаючи зміни в законодавстві, коливання цін на нафту, зростання екологічних стандартів та збільшення популярності електромобілів. Все це вимагає гнучкого підходу до

розробки та адаптації сайту, щоб він відповідав актуальним трендам та викликам.

Розвиток вебресурсів для АЗС також передбачає глибоке розуміння потреб кінцевого споживача, оскільки основною метою є не просто інформування про послуги, а створення доданої вартості через зручність, доступність та персоналізацію обслуговування. Тому, дослідження цільової аудиторії та її поведінкових факторів стає невід’ємною частиною процесу розробки вебсайтів.

1.2 Огляд існуючих сайтів АЗС

У рамках дослідження вебсайтів автозаправних станцій було ретельно проаналізовано ведучі ресурси в цій галузі, включаючи як національні, так і міжнародні платформи. Цей аналіз дозволив виявити низку ключових функцій, необхідних для ефективної взаємодії з клієнтами та підвищення лояльності.

Багато вебсайтів АЗС мають інтерактивні карти (див. рис. 1.1), що допомагають користувачам легко знаходити найближчі станції, переглядати доступні послуги, години роботи та актуальні ціни на паливе.

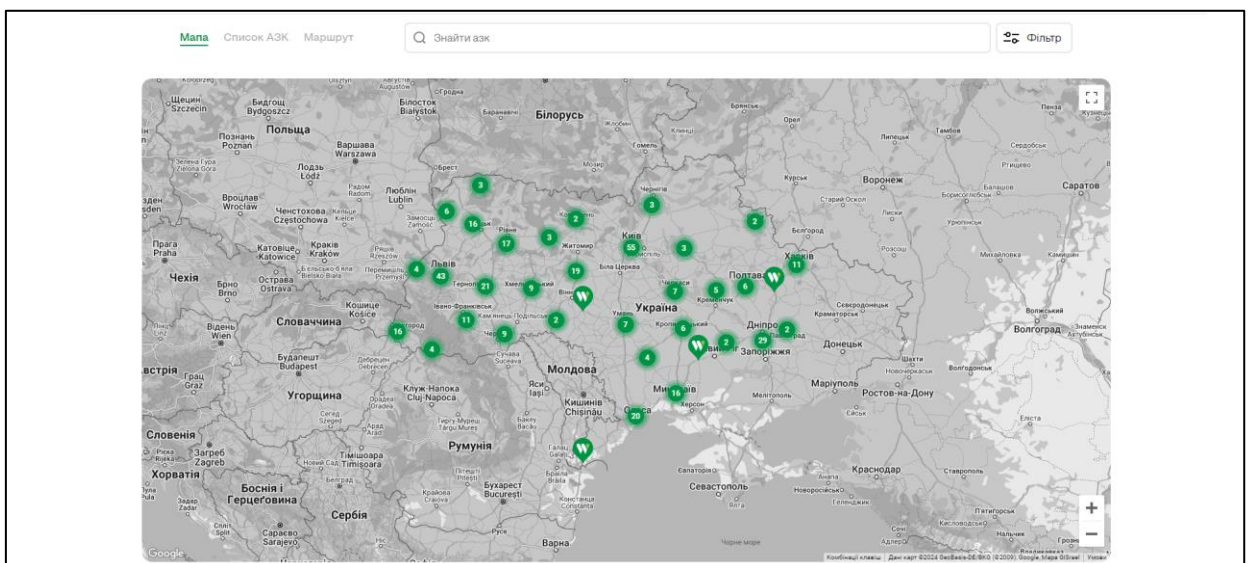


Рисунок 1.1 – Карта Заправок WOG

Також учасні сайти АЗС надають детальну інформацію про наявні види пального, масла, автомийки, кафе та інші зручності при станціях (див. рис. 1.2).

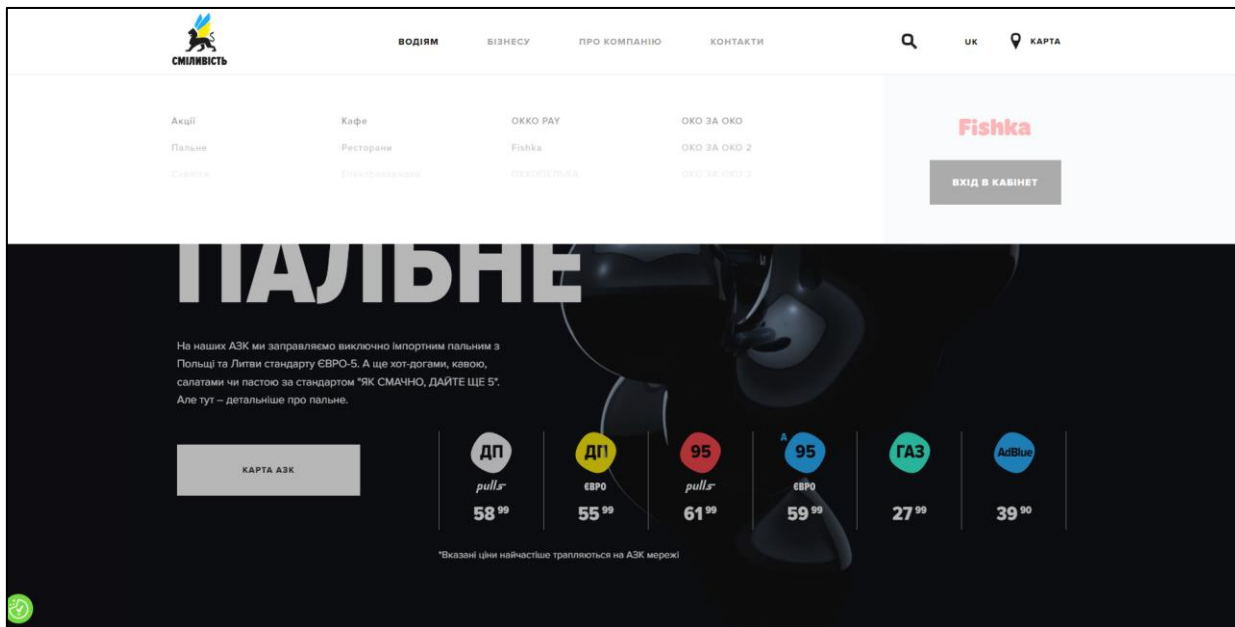


Рисунок 1.2 – Інформація про продукти та послуги ОККО

А функціонал особистого кабінету користувача дозволяє клієнтам керувати своїми лояльними картками, переглядати історію покупок, накопичувати та використовувати бонусні бали (див. рис. 1.3).

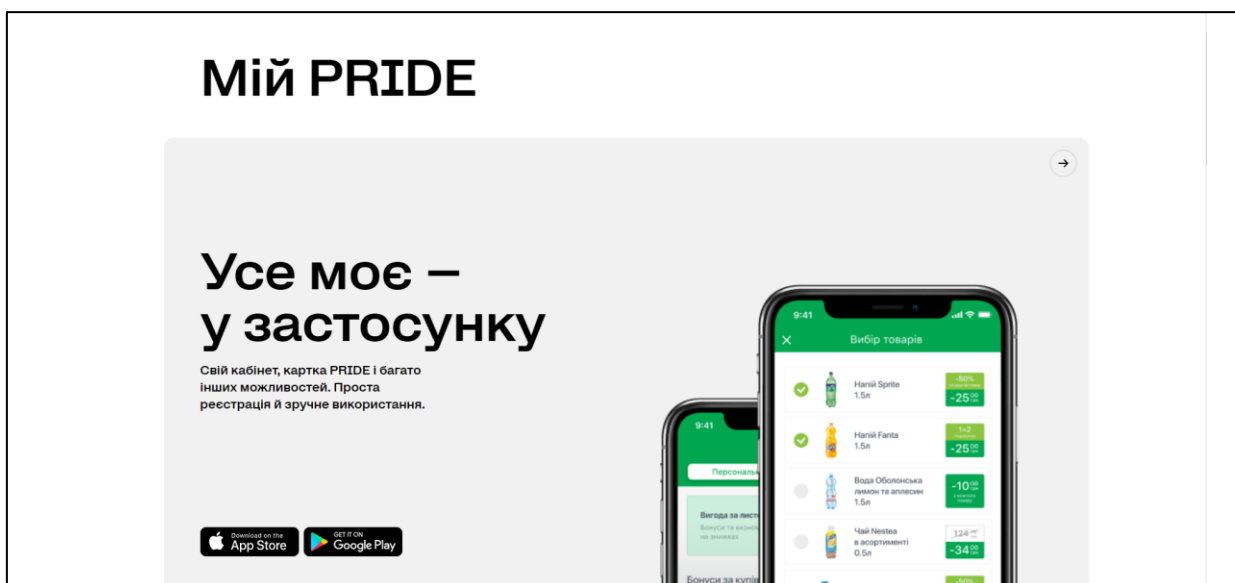


Рисунок 1.3 – Особистий кабінет користувача WOG

Також є різні програми лояльності (див. рис. 1.4) та більшість сайтів включають інформацію про ці програми, які пропонують систему нагород та знижок для регулярних клієнтів.



Рисунок 1.4 – Програми лояльності BRCM

Та невід’ємною частиною сучасних сайтів є чітка контактна інформація та наявність служби підтримки для надання допомоги клієнтам (див. рис. 1.5).

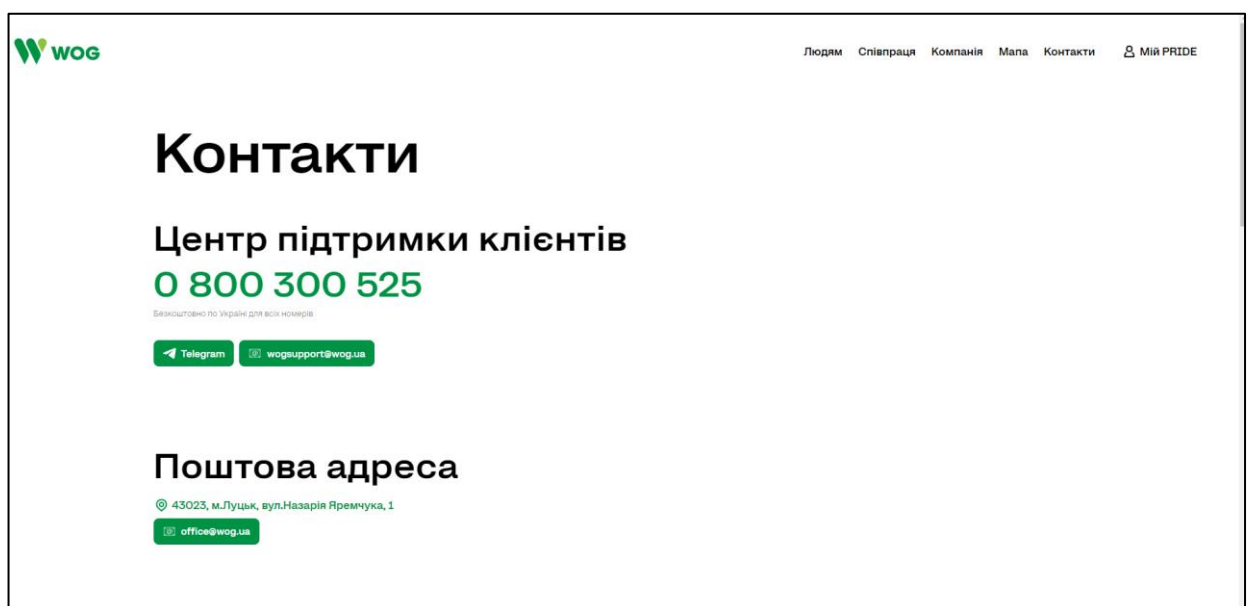


Рисунок 1.5 – Контактна інформація та служба підтримки WOG

Все вищесказане ілюструє основні функції та переваги, які мають бути присутні на сучасному вебсайті АЗС для забезпечення високої задоволеності та залученості користувачів.

1.3 Засоби для створення вебсайта

Для розробки сучасного та ефективного вебсайту АЗС, існує багато технологій та інструментів. Основні з них – HTML, CSS і JavaScript, які є стовпами фронтенд-розробки. Кожен з цих інструментів відіграє свою унікальну роль у створенні вебсайтів, що відрізняються зручністю, функціональністю та привабливим дизайном.

HTML становить основу будь-якого вебсайту. Він використовується для структурування контенту на вебсторінках, включаючи текст, зображення, відео та інші медіа-файли. З допомогою HTML розробники створюють каркас сайту, визначаючи заголовки, параграфи, списки, посилання та інші елементи сторінки [1].

CSS використовується для задання візуального стилю вебсторінок, наприклад, шрифтів, кольорів, маржинів, відступів, фонів тощо. За допомогою CSS, веброзробники можуть контролювати презентацію вебсайту на різних пристроях, що дозволяє робити дизайн чуйний та адаптивним. Також CSS допомагає уникнути повторення HTML-коду, оскільки стилі можуть бути застосовані одразу до кількох елементів або цілих сторінок [1].

JavaScript є мовою програмування, яка використовується для додавання інтерактивності на вебсайти. З його допомогою можна створювати складні ефекти та анімації, обробляти події (наприклад, кліки мишею або натискання клавіш), валідувати форми, динамічно змінювати вміст на сторінці без перезавантаження, та багато іншого. JavaScript також відіграє ключову роль у розробці SPA (Single Page Applications), що дозволяє користувачам взаємодіяти з вебсайтом більш плавно і без затримок [3].

Разом, HTML, CSS і JavaScript формують потужний інструментарій для створення вебсайтів. Використання цих технологій дозволяє розробникам створювати високоадаптивні, швидкі та привабливі вебресурси, які можуть задовольнити вимоги сучасних користувачів і стандарти інтернету. Під час розробки сайту для АЗС, важливо зосередити увагу на таких аспектах, як швидкість завантаження сторінок, SEO-оптимізація, безпека користувачів та зручність використання [6].

1.3.1 Огляд бази даних MySQL

MySQL є однією з найпопулярніших систем управління реляційними базами даних, яка широко використовується в розробці вебсайтів. Це відкритий продукт, підтримуваний Oracle, що забезпечує надійну, стабільну та гнучку платформу для зберігання даних [2]. Завдяки високій продуктивності, сумісності з багатьма мовами програмування та легкій інтеграції з різними вебдодатками, MySQL часто обирають для розробки. Веброзробники цінують цю систему за її здатність забезпечувати швидкий доступ до даних, що є критично важливим для сучасних вебдодатків.

Основні переваги MySQL включають її надійність та здатність обробляти великі об'єми даних з високою швидкістю, що робить її ідеальною для бізнес-додатків з інтенсивною обробкою транзакцій. Наприклад, популярні вебплатформи, як WordPress і Joomla, використовують MySQL як свій основний механізм для управління даними, що свідчить про її ефективність і надійність у реальних проєктах [2]. Масштабованість системи дозволяє легко розширювати базу даних у міру зростання потреб користувача, а високий рівень безпеки забезпечує захист даних від несанкціонованого доступу через шифрування, автентифікацію користувачів та контроль доступу. Дотримання стандартів SQL забезпечує сумісність з більшістю існуючих рішень для управління базами даних, а підтримка різних

операційних систем, таких як Linux, Windows, та MacOS, надає гнучкість у виборі платформи для розгортання.

В контексті розробки вебсайту для АЗС, MySQL може використовуватись для зберігання даних про клієнтів, включаючи особисті дані, історію транзакцій та інформацію про програми лояльності. Також, база даних може зберігати деталі про товари та послуги, транзакційні дані, маркетингову інформацію, та інші важливі дані. Наприклад, інформація про наявність пального та його ціни може бути миттєво доступна користувачам, що значно покращує їхній досвід взаємодії з сайтом. MySQL дозволяє легко інтегрувати вебсайт з зовнішніми програмами для управління відносинами з клієнтами, електронною комерцією, аналітикою та іншими системами, що забезпечують повноцінне функціонування сайту АЗС.

Node.js є потужною платформою для розробки серверної частини вебдодатків, що забезпечує високу ефективність та масштабованість. Використання JavaScript на серверній стороні дозволяє уніфікувати розробку вебдодатків, знижуючи кількість помилок та спрощуючи підтримку коду [8]. Асинхронна та подієорієнтована архітектура Node.js робить її ідеальною для обробки численних запитів, забезпечуючи високу продуктивність в середовищах з великим навантаженням, таких як сайт АЗС “Barrel”. Ця технологія дозволяє розробникам створювати інтерактивні та динамічні вебдодатки, які можуть обробляти великі об’єми даних у реальному часі.

Масштабованість Node.js дозволяє ефективно обробляти велику кількість одночасних з’єднань, що робить її чудовим вибором для бізнесу, який зростає [8]. Наприклад, великі компанії, такі як LinkedIn та Netflix, використовують Node.js для забезпечення високої продуктивності та швидкого часу відгуку своїх сервісів. Активна спільнота розробників та велика кількість доступних пакетів у менеджері npm скорочують час розробки та сприяють інноваціям. Використання Node.js для сайту “Barrel” підвищує швидкість обробки запитів, дозволяє створювати RESTful API для інтеграції з базами даних та зовнішніми сервісами, і забезпечує кращий користувацький досвід завдяки швидкій і надійній обробці даних.

1.4 Формування вимог до сайту

Для створення ефективного вебсайту автозаправної станції необхідно врахувати не лише основні технічні, функціональні та дизайнерські аспекти, але й забезпечити гнучкість, масштабованість та високий рівень користувацької взаємодії. Це вимагає всебічного підходу, де кожна деталь відіграє важливу роль у створенні привабливого та ефективного ресурсу, що відповідатиме сучасним вимогам та потребам користувачів [4].

На початковому етапі слід обрати надійну платформу розробки, яка підтримує інтеграцію зі сторонніми сервісами та забезпечує стабільну роботу сайту цілодобово. Це дозволяє уникнути можливих технічних проблем та забезпечує високу доступність ресурсу для користувачів у будь-який час. Надійна платформа також сприяє легкій масштабованості вебсайту, що важливо для зростання бізнесу та збільшення кількості відвідувачів.

Інтеграція з ERP/CRM системами є ключовим аспектом для автоматизації процесів та покращення сервісу. Це дозволяє забезпечити безперервний зв'язок з внутрішніми системами управління ресурсами та відносинами з клієнтами. Така інтеграція сприяє ефективнішому обслуговуванню клієнтів, швидшому реагуванню на їхні запити та покращенню загального користувацького досвіду.

Важливим компонентом сучасного вебсайту є системи аналітики. Вбудовані аналітичні інструменти дозволяють моніторити поведінку користувачів на сайті та оцінювати ефективність маркетингових кампаній. Це дає змогу приймати обґрунтовані рішення на основі реальних даних, що допомагає оптимізувати роботу вебсайту та підвищувати рівень залученості користувачів [6].

У функціональному аспекті вебсайт автозаправної станції має надавати можливість для онлайн-платежів, що значно спрощує процес оплати для клієнтів. Персоналізований досвід на основі попередньої взаємодії користувача з сайтом є ще одним важливим аспектом, який допомагає

забезпечити індивідуальний підхід до кожного клієнта. Це може включати рекомендації на основі попередніх покупок, спеціальні пропозиції та інші персоналізовані елементи.

Підтримка багатомовності є необхідною умовою для залучення інтернаціональних користувачів. Вебсайт має бути доступним на кількох мовах, що дозволяє розширити аудиторію та зробити сервіс зручнішим для людей з різних країн. Наявність API для інтеграції з іншими сервісами та додатками забезпечує гнучкість та розширюваність функціоналу вебсайту, що дозволяє швидко адаптуватися до нових вимог та потреб бізнесу.

Дизайнерські рішення повинні орієнтуватися на створення інтуїтивно зрозумілого інтерфейсу, який забезпечує просту та зручну навігацію. Важливо створити унікальну візуальну ідентичність, яка відображає бренд та цінності компанії. Це допомагає зробити вебсайт впізнаваним та привабливим для користувачів.

Доступність є ще одним важливим аспектом, який слід враховувати при розробці вебсайту. Реалізація принципів доступності дозволяє забезпечити комфортне використання сайту людьми з обмеженими можливостями, що підвищує його інклюзивність та загальну зручність.

Використання анімацій та інтерактивних елементів сприяє підвищенню залученості користувачів, роблячи взаємодію з вебсайтом більш динамічною та цікавою. Це дозволяє утримувати увагу відвідувачів та стимулює їх до подальшої взаємодії з ресурсом.

Дотримання цих вимог дозволить створити не тільки функціональний, а й технологічно просунутий вебсайт, який відповідатиме сучасним тенденціям веброзробки та очікуванням користувачів. Інтеграція новітніх технологій та інноваційних рішень сприятиме підвищенню ефективності бізнесу та задоволенню потреб клієнтів на найвищому рівні.

2 РОЗРОБКА ПРОЄКТУ

2.1 Технічне завдання для вебсайту АЗС “Barrel”

Технічне завдання є ключовим документом у процесі створення вебсайту, який визначає всі технічні та функціональні вимоги. Воно служить основою для проєктування і розробки, а також встановлює договірні зобов’язання між замовником та виконавцем. Для вебсайту АЗС “Barrel” це завдання охоплює кілька важливих аспектів.

Основною метою проєкту є створення вебсайту для АЗС “Barrel”, який покращить зручність використання послуг станцій для клієнтів, підвищить імідж компанії на ринку, та забезпечить інтеграцію з існуючими системами управління та аналітики.

Вебсайт стане віртуальною вітриною компанії, що дозволить ефективніше комунікувати з клієнтами, забезпечуючи їм доступ до необхідної інформації в будь-який час.

Серед основних завдань вебсайту – надання вичерпної інформації про послуги і товари АЗС “Barrel”, забезпечення можливості онлайн-замовлення продуктів та послуг, а також інтеграція з системою лояльності для управління особистими бонусами та знижками користувачів. Крім того, сайт має забезпечувати збір та аналіз даних для подальшого вдосконалення послуг.

Важливим елементом є також оптимізація сайту для мобільних пристроїв, щоб користувачі могли зручно взаємодіяти з ресурсом незалежно від типу гаджета.

Таким чином, технічне завдання не лише визначає функціональність майбутнього вебсайту, але й спрямоване на створення інструменту, що допоможе АЗС “Barrel” досягти нових висот у сфері обслуговування клієнтів та ефективності бізнес-процесів.

2.2 Функціональні вимоги

Сайт має мати сучасний і інтуїтивно зрозумілий інтерфейс, який автоматично адаптується під різні типи пристроїв. Каталог продуктів повинен містити повну інформацію про доступні види пального, масла та супутні товари, що є в АЗС. Система повинна забезпечувати онлайн-замовлення з зручними варіантами оплати. Модуль для управління акаунтами користувачів повинен дозволяти реєстрацію, авторизацію, відновлення доступу та перегляд історії покупок. Сайт має бути синхронізований з внутрішніми системами компанії через CRM для ефективного управління даними. Безпека користувачів повинна бути забезпечена за допомогою SSL-шифрування та захисту від XSS і CSRF атак.

Технічні вимоги включають використання HTML5, CSS3 та JavaScript для клієнтської частини, Node.js і Express для серверної, а також MySQL або MongoDB для бази даних. Необхідно вибрати надійний хостинг з високою пропускнуою спроможністю і цілодобовою підтримкою. Сайт має бути сумісний з останніми версіями основних браузерів (Chrome, Firefox, Safari, Edge) і спроектований для легкого масштабування та оновлення.

Дизайн сайту повинен відповідати корпоративному стилю компанії “Barrel” і забезпечувати високий рівень зручності користування. Розробка технічної документації повинна включати інструкції для користувачів та системних адміністраторів, надаючи їх українською та англійською мовами.

Це технічне завдання стане основою для подальшої розробки та впровадження вебсайту АЗС “Barrel”, сприяючи зростанню компанії та підвищенню задоволеності клієнтів.

2.2.1 Найменування і область застосування

Назва проєкту: розробка вебсайту для АЗС “Barrel”. Область застосування проєкту включає всі етапи створення вебсайту, починаючи від

концептуального дизайну і програмування, до ретельного тестування і впровадження у промислову експлуатацію. Метою цього проєкту є створення інноваційного та зручного у використанні вебсайту, що надаватиме інформаційну підтримку та онлайн-сервіси клієнтам АЗС “Barrel”. Вебсайт призначений для широкого кола користувачів: від кінцевих споживачів, які шукають інформацію про продукти і послуги, до внутрішніх корпоративних користувачів, що потребують інструментів для управління даними, аналітики та проведення маркетингових кампаній.

Основними завданнями сайту є надання користувачам доступу до актуальної інформації про продукти та послуги, включаючи ціни на пальне та спеціальні пропозиції. Крім цього, вебсайт запропонує широкий спектр онлайн-сервісів, таких як можливість онлайн-замовлення товарів і послуг, включаючи пальне, мастильні матеріали та автомобільні аксесуари. Важливою функцією вебсайту стане інтеграція з існуючими CRM-системами, що дозволить ефективно керувати клієнтською базою та оптимізувати маркетингові стратегії

Для зручності користувачів на сайті буде впроваджено інструменти зворотного зв'язку, включаючи форми для запитань, чати з підтримкою та номери гарячих ліній. Це забезпечить швидкий та ефективний доступ до підтримки клієнтів, що є критично важливим для підтримання високого рівня задоволеності клієнтів.

Крім цього, вебсайт стане важливим елементом у стратегії цифрового маркетингу компанії “Barrel”. Використання сучасних технологій дозволить не лише покращити взаємодію з клієнтами, але й забезпечити збір та аналіз даних для прийняття більш обґрунтованих бізнес-рішень. Вебсайт також сприятиме підвищенню впізнаваності бренду та зміцненню його позицій на ринку.

Метою створення вебсайту є підвищення задоволеності та залученості клієнтів, покращення іміджу компанії “Barrel” на ринку, забезпечення ефективного управління клієнтськими даними, а також поліпшення доступності послуг АЗС через цифрові канали зв'язку. Ця область

застосування визначає основні цілі та функціональні можливості вебсайту, а також встановлює рамки його використання у контексті бізнес-цілей компанії “Barrel”.

Враховуючи швидкий розвиток технологій і зростаючі потреби користувачів, розробка сучасного вебсайту для АЗС “Barrel” є важливим кроком на шляху до забезпечення високоякісного обслуговування та створення унікального користувацького досвіду.

2.2.2 Призначення розробки

Метою проєкту є створення інноваційного та багатофункціонального вебсайту для АЗС “Barrel”. Це рішення сприятиме збільшенню обсягів продажів за рахунок покращеної взаємодії з клієнтами та оптимізації внутрішніх процесів компанії.

Вебсайт буде спрямований на підвищення зручності для користувачів, пропонуючи інтуїтивно зрозумілий інтерфейс. Це дозволить легко навігувати по сайту, швидко знаходити необхідну інформацію про продукти та послуги, а також здійснювати покупки або замовлення онлайн.

Інтеграція з існуючими CRM та ERP системами забезпечить єдиний інформаційний простір, що дозволить автоматизувати процеси управління клієнтами, запасами та продажами. Це підвищить ефективність роботи компанії та зменшить витрати на ручну обробку даних.

Розширення функціональності вебсайту включатиме нові можливості, такі як програма лояльності, персоналізовані рекомендації та онлайн-бронювання послуг (наприклад, мийка автомобіля). Це додає додаткову цінність для клієнтів, сприяючи їх залученню та утриманню.

Для покращення комунікації з клієнтами будуть використані інструменти зворотного зв'язку, включаючи чати, форми для запитань та опитування задоволеності клієнтів. Це допоможе збирати важливу інформацію для аналізу та удосконалення сервісу.

Забезпечення безпеки даних клієнтів стане пріоритетом, включаючи впровадження сучасних методів захисту, таких як шифрування та аутентифікація. Це гарантуватиме конфіденційність інформації клієнтів і відповідність нормативним вимогам.

Завершення проєкту дозволить компанії “Barrel” значно покращити свої позиції на ринку, підвищити обізнаність бренду серед потенційних клієнтів та забезпечити більшу лояльність існуючих клієнтів. Впровадження ефективних інструментів онлайн-сервісу та управління відносинами сприятиме стійкому розвитку та зростанню компанії.

2.2.3 Технічні вимоги до програмного продукту

Для забезпечення ефективної роботи та надійності вебсайту АЗС “Barrel”, було визначено низку технічних вимог. По-перше, система має бути побудована на багаторівневій архітектурі, що включає клієнтський, серверний рівні та рівень бази даних, з можливістю горизонтального та вертикального масштабування для підтримки високої продуктивності під час пікових навантажень.

У розробці інтерфейсу використовуються сучасні технології, зокрема HTML5, CSS3, JavaScript і такі фреймворки, як Vue.js або Angular для створення інтерактивних та адаптивних інтерфейсів. На серверному боці застосовуються Node.js і Express.js для розробки RESTful API, а база даних може бути як реляційною (MySQL, PostgreSQL), так і NoSQL (MongoDB) в залежності від потреб зберігання та обробки даних.

Інтерфейс сайту має автоматично адаптуватися до різних розмірів екранів і відповідати стандартам доступності WCAG 2.1 для забезпечення зручності користування для людей з обмеженими можливостями. Задля безпеки даних використовуються HTTPS для шифрування переданих даних, а також реалізуються механізми аутентифікації та авторизації, захист від атак

SQL Injection, Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF).

Сайт повинен підтримувати інтеграцію з зовнішніми сервісами через розробку RESTful API, включаючи платіжні системи та зовнішні додатки, а також взаємодію з CRM та ERP системами для ефективного управління даними та процесами [9].

Для завершення проєкту необхідна розробка повної технічної документації, що охоплює архітектуру системи, базу даних, кодову базу та API, а також інструкцій для кінцевих користувачів, які описують функціонал та основні процедури роботи з сайтом. Ці вимоги встановлюють стандарти якості та безпеки для розробки вебсайту АЗС “Barrel” і слугують основою для наступних етапів проєкту, включаючи дизайн, розробку, тестування та впровадження.

2.3 Етапи створення вебсайту

Процес створення вебсайту для АЗС “Barrel” включає декілька ключових етапів. Кожен етап важливий для забезпечення якості та ефективності кінцевого продукту.

2.3.1 Планування

Цей етап є фундаментом для всього проєкту, де команда визначає цілі сайту, його аудиторію, і функціональність, яку потрібно реалізувати.

- визначення потреб бізнесу, функціональних та нефункціональних вимог;
- планування архітектури сайту, вибір технологій та інструментів;
- формулювання документу, який буде направляти розробку та дизайн.

2.3.2 Дизайн

На цьому етапі здійснюється розробка дизайну інтерфейсу, який поєднує привабливість і зручність для користувача. Спочатку створюються прототипи сторінок, які включають макети та переходи між ними, забезпечуючи чітке уявлення про структуру сайту. Потім розробляється візуальний стиль, що охоплює вибір кольорової гами, шрифтів та розміщення елементів, аби створити гармонійний і функціональний інтерфейс. Завершальним кроком є представлення готового дизайну замовнику для отримання зворотного зв'язку та внесення необхідних коректив.

2.3.3 Розробка

На цьому етапі впровадженні функціональності сайту згідно з затвердженим дизайном та технічним завданням. Цей процес включає кілька ключових аспектів.

Розробка фронтенду:

- HTML/CSS: використання HTML для створення структури сторінок вебсайту та CSS для стилізації елементів, що забезпечує візуальне представлення контенту і логічну візуальну ієрархію [1];
- JavaScript: впровадження інтерактивних елементів на клієнтській стороні, таких як меню навігації, слайдери, модальні вікна та відгуківі форми, що покращують користувацький досвід [3];
- фреймворки та бібліотеки: інтеграція фреймворків як Bootstrap для швидкої розробки адаптивних макетів [10] та Vue.js для динамічного управління станами інтерфейсу, що підвищує швидкість і зручність в роботі [7].

Розробка бекенду:

- серверна логіка: програмування логіки обробки даних, управління

сесіями користувачів та авторизації;

- бази даних: впровадження і керування базами даних для зберігання даних користувачів, транзакцій, продуктів тощо;
- API та інтеграції: розробка API для взаємодії з іншими сервісами та інтеграція з зовнішніми системами, наприклад, системами оплати, соціальними мережами або зовнішніми базами даних;
- безпека: забезпечення безпечної передачі та зберігання даних за допомогою технологій шифрування та аутентифікації.

Тестування:

- Unit tests: написання та виконання модульних тестів для кожного компоненту системи, що допомагає швидко виявляти та виправляти помилки на ранніх етапах розробки;
- Integration tests: перевірка взаємодії між різними модулями та сервісами сайту для забезпечення їх коректної роботи як єдиного цілого;
- Performance testing: тестування продуктивності та швидкості реакції сайту під високим навантаженням, що критично для високо відвідуваних ресурсів;
- Security audits: виконання оглядів безпеки для виявлення потенційних вразливостей та забезпечення захисту користувацьких даних і бізнес-інформації.

2.3.4 Тестування

Комплексне тестування охоплює всі аспекти вебсайту, щоб забезпечити його надійність та ефективність. На цьому етапі здійснюється перевірка окремих компонентів програми, що дозволяє виявити та виправити помилки на рівні модулів. Інтеграційне тестування гарантує коректну взаємодію між різними модулями та сервісами, а тестування взаємодії фокусується на тому,

щоб користувачі могли ефективно користуватися сайтом, надаючи їм безперебійний та інтуїтивно зрозумілий досвід [11].

2.3.5 Запуск

Завершальний етап проєкту включає розгортання вебсайту на продуктивному сервері та його публічний запуск. Це включає перенесення сайту на продуктивне середовище, а також постійний моніторинг його роботи для виявлення можливих проблем.

При необхідності, вносяться зміни для покращення продуктивності та безпеки сайту, забезпечуючи його стабільну роботу та відповідність найвищим стандартам якості.

2.3.6 Підтримка та оновлення

Після запуску, вебсайт потребує постійного оновлення та підтримки, щоб залишатися актуальним та безпечним для користувачів. Це включає оперативне вирішення технічних проблем користувачів, регулярне оновлення контенту та функціоналу сайту, а також постійне оновлення системи безпеки для захисту від нових загроз. Наприклад, впровадження нових функцій або зміна дизайну може покращити взаємодію користувачів із сайтом, роблячи його більш зручним та привабливим.

Ці етапи забезпечують систематичний підхід до розробки вебсайту, що дозволяє ефективно управляти проєктом і досягти високої якості кінцевого продукту. Тільки завдяки уважному тестуванню, надійному запуску та постійній підтримці можна гарантувати, що вебсайт задовольнить потреби як бізнесу, так і користувачів, забезпечуючи надійну платформу для взаємодії та розвитку.

2.4 Діаграма прецедентів

Діаграма прецедентів (Use Case Diagram) є важливим інструментом в аналізі та проектуванні систем, який допомагає визначити взаємодії між користувачами (акторами) та системою. Для вебсайту АЗС “Barrel” ця діаграма візуалізує основні функції сайту та взаємодію користувачів з цими функціями, що сприяє чіткому розумінню ролей і завдань кожного актора [12].

Основні актори включають клієнтів, адміністраторів та CRM-систему. Клієнти, як відвідувачі сайту, можуть шукати інформацію, робити замовлення та користуватися іншими послугами. Вони мають можливість переглядати детальну інформацію про продукти та послуги, які пропонує АЗС, обирати товари для покупки або бронювання, реєструватися в програмі лояльності для отримання знижок та бонусів, а також залишати відгуки про сервіс. Це створює інтерактивне середовище, де клієнти можуть отримувати всю необхідну інформацію та користуватися послугами з максимальною зручністю.

Адміністратори керують контентом сайту, налаштовують системні параметри та обслуговують користувачів. Їхня роль полягає у підтримці актуальності контенту, управлінні замовленнями, обробці звернень клієнтів та аналізі поведінки користувачів для підвищення ефективності маркетингових кампаній. Вони також відповідають за безпеку сайту, забезпечуючи захист даних від можливих загроз.

CRM-система взаємодіє з вебсайтом для обробки даних клієнтів, замовлень та програм лояльності. Вона забезпечує автоматичну синхронізацію даних між різними платформами, керує рахунками лояльності, нарахуванням балів та адмініструванням знижок і спеціальних пропозицій. Це дозволяє ефективно управляти взаємовідносинами з клієнтами, забезпечуючи персоналізований підхід та підвищуючи рівень задоволеності клієнтів [13].

На діаграмі прецедентів кожен актор з'єднаний з прецедентами, в яких він бере участь або які ініціює. Це наочно показує, які дії доступні для кожного

типу користувачів і як ці дії пов'язані з функціональністю системи. Наприклад, клієнт може ініціювати прецедент перегляду інформації про продукти, тоді як адміністратор ініціює управління контентом. Це дозволяє проєктній команді і зацікавленим сторонам краще зрозуміти структуру взаємодії та виявити можливості для покращення або додавання нових функцій.

Така діаграма (див. рис. 2.1) допомагає всім зацікавленим сторонам проєкту краще зрозуміти взаємодію між користувачами та системою, а також визначити потенційні області для розширення функціональності або вдосконалення. Це сприяє більш ефективному плануванню та реалізації проєкту, забезпечуючи високу якість кінцевого продукту та задоволення потреб користувачів.

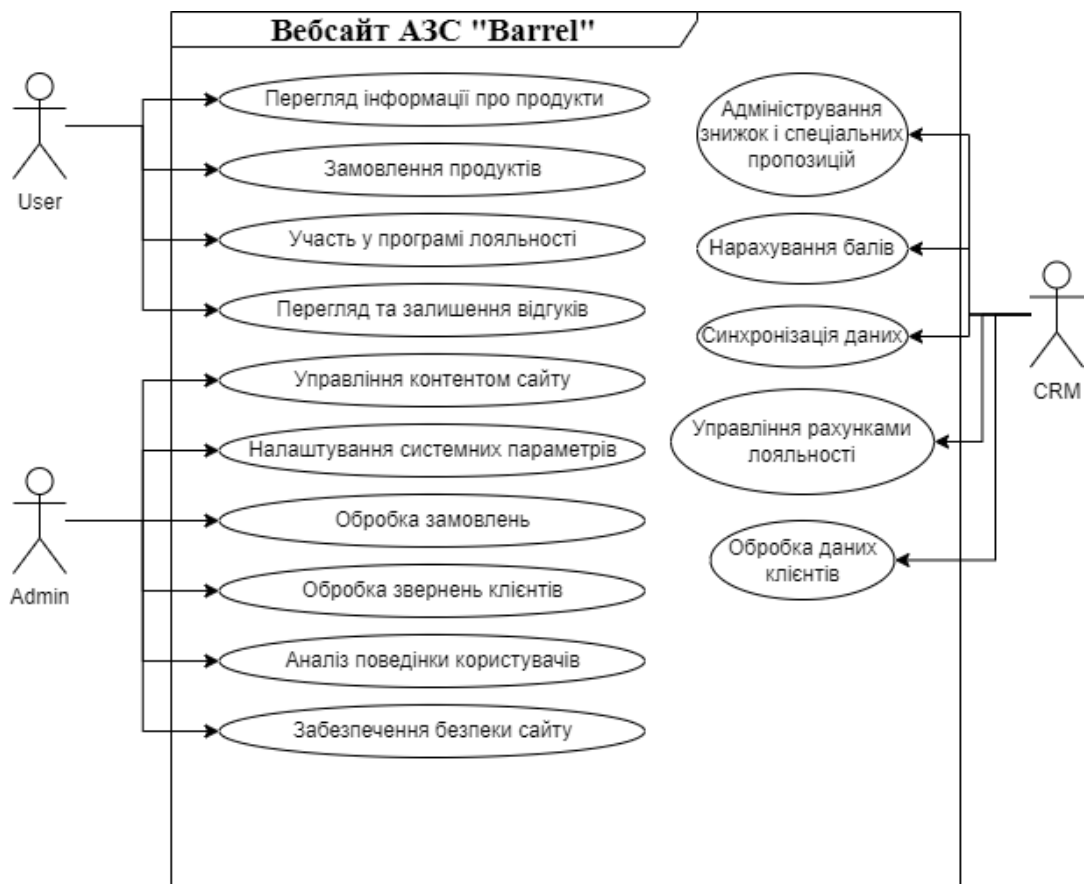


Рисунок 2.1 – Діаграма варіантів використання

Основними актами для розроблюваного програмного продукту є покупець та адміністратор сайту.

2.5 Схема бази даних

Розробка схеми бази даних є ключовим етапом у створенні вебсайту АЗС “Barrel”, оскільки вона визначає структуру, в якій будуть зберігатися та оброблятися всі дані. Схема бази даних повинна ефективно підтримувати операції, які будуть виконуватися через вебсайт, забезпечуючи швидкий доступ та високу продуктивність.

Основні сутності та їх атрибути (див. рис. 2.2).

	A	B
1	Клієнти (Customers)	
2	CustomerID	унікальний ідентифікатор клієнта
3	FirstName	ім'я клієнта
4	LastName	прізвище клієнта
5	Email	електронна пошта клієнта
6	PhoneNumber	контактний номер телефону
7	Address	адреса клієнта
8		
9	Продукти (Products)	
10	ProductID	унікальний ідентифікатор продукту
11	Name	назва продукту
12	Description	опис продукту
13	Price	ціна продукту
14	QuantityInStock	кількість на складі
15		
16	Замовлення (Orders)	
17	OrderID	унікальний ідентифікатор замовлення
18	CustomerID	ідентифікатор клієнта, який робить замовлення
19	OrderDate: дата замовлення.	
20	TotalAmount: загальна сума замовлення.	
21		
22	Деталі замовлення (OrderDetails)	
23	OrderDetailID: унікальний ідентифікатор деталі замовлення.	
24	OrderID: ідентифікатор замовлення.	
25	ProductID: ідентифікатор продукту.	
26	Quantity: кількість замовленого продукту.	
27	Price: ціна за одиницю.	
28		
29	Працівники (Employees)	
30	EmployeeID: унікальний ідентифікатор працівника.	
31	FirstName: ім'я працівника.	
32	LastName: прізвище працівника.	
33	Position: посада в компанії.	
34	Email: електронна пошта працівника.	
35		
36	Логін (Login)	
37	LoginID: унікальний ідентифікатор логіну.	
38	EmployeeID: ідентифікатор працівника, якому належить логін.	
39	Username: ім'я користувача.	
40	Password: пароль.	

Рисунок 2.2 – Сутності та їх атрибути

Відносини між сутностями такі: Клієнти роблять Замовлення, які містять Деталі замовлення, що включають Продукти. Працівники керують Замовленнями та обслуговують Клієнтів, використовуючи Логін для доступу до адміністративної панелі.

Для кращого розуміння структури бази даних створюється ER-діаграма (Entity-Relationship Diagram), яка візуально представляє всі сутності, їх атрибути та зв'язки між ними. Це допомагає переконатися, що база даних буде здатна підтримувати всі необхідні операції та забезпечити легкий доступ до інформації для всіх користувачів системи.

2.6 Проектування інтерфейсу

Проектування інтерфейсу вебсайту для АЗС “Barrel” є важливим кроком у створенні зручного та ефективного засобу спілкування з користувачами. Цей етап включає в себе розробку дизайну, який відповідає бренду компанії та забезпечує інтуїтивно зрозумілу навігацію для користувачів.

2.6.1 Визначення вимог до інтерфейсу

Перед початком проектування важливо чітко визначити вимоги до інтерфейсу.

Цільова аудиторія: «Хто є основними користувачами сайту? Що вони очікують знайти?»

Основні задачі користувача: «Які дії користувачі будуть виконувати на сайті?»

Адаптивність: сайт повинен бути зручним для перегляду як на настільних комп'ютерах, так і на мобільних пристроях.

2.6.2 Прототипування

На цьому етапі створюються макети сторінок, що демонструють базову структуру інтерфейсу, що є ключовим для подальшого розвитку дизайну вебсайту. Спочатку розробляються каркасні макети (wireframes), які відображають розміщення основних елементів інтерфейсу. Ці каркасні макети є основою для розуміння розташування контенту та навігації на сайті без відволікання на візуальні деталі.

Після затвердження каркасних макетів, створюються більш детальні макети (mockups) з використанням фірмового стилю, кольорів і шрифтів, що надає їм завершеного вигляду. Цей етап дозволяє побачити, як виглядатиме кінцевий продукт, забезпечуючи візуальну узгодженість та естетичну привабливість.

Завершальним кроком є створення інтерактивних прототипів (clickable prototypes), які можна тестувати на користувачах. Інтерактивні прототипи дозволяють симулювати реальний досвід взаємодії з сайтом, що є надзвичайно важливим для виявлення потенційних проблем та удосконалення користувацького досвіду. Тестування таких прототипів на ранніх стадіях розробки допомагає отримати цінний зворотний зв'язок від користувачів, що дозволяє ввести необхідні корективи до остаточного дизайну.

Таким чином, поетапне створення макетів, від каркасних до інтерактивних прототипів, забезпечує систематичний підхід до розробки інтерфейсу, дозволяючи врахувати всі аспекти користувацького досвіду та досягти високої якості кінцевого продукту.

2.6.3 Дизайн UI/UX

Остаточне оформлення інтерфейсу повинно охоплювати кілька ключових аспектів для забезпечення його ефективності та привабливості.

Передусім, важливо зосередитися на юзабіліті, адже інтерфейс має бути зрозумілим та інтуїтивно легким у використанні. Це забезпечить користувачам зручний доступ до всіх необхідних функцій без зайвих ускладнень.

Візуальний дизайн також відіграє важливу роль. Необхідно створити привабливий зовнішній вигляд, який буде відображати ідентичність бренду та стимулюватиме інтерес користувачів. Гармонійне поєднання кольорів, шрифтів та графічних елементів сприятиме позитивному сприйняттю інтерфейсу.

Крім того, дизайн має сприяти легкій взаємодії з сайтом, що мінімізує можливість виникнення помилок під час використання. Зручне та логічне розташування елементів інтерфейсу допоможе користувачам швидко орієнтуватися та виконувати необхідні дії без зайвих зусиль.

Таким чином, увага до юзабіліті, візуального дизайну та зручності взаємодії є критично важливою для створення ефективного інтерфейсу, який не лише приваблює, але й задовольняє потреби користувачів.

2.6.4 Тестування інтерфейсу

Перевірка інтерфейсу на зручність та ефективність є важливим етапом розробки, що включає кілька ключових методів.

Перш за все, тестування з користувачами дозволяє зібрати зворотний зв'язок від реальних користувачів, що допомагає виявити проблеми в юзабіліті. Це надає розробникам цінну інформацію для вдосконалення інтерфейсу.

Також важливо використовувати метод А/Б тестування, який передбачає проведення експериментів з різними версіями інтерфейсу. Це дозволяє визначити найефективніші рішення, які найкраще задовольняють потреби користувачів.

Аналіз поведінки користувачів за допомогою інструментів аналітики

допомагає відстежувати, як користувачі взаємодіють з сайтом. Це надає можливість виявляти патерни поведінки та оптимізувати інтерфейс для покращення загального користувацького досвіду.

Таким чином, перевірка інтерфейсу на зручність та ефективність включає тестування з користувачами, А/Б тестування та аналіз поведінки, що разом допомагає створити більш досконалий і зручний інтерфейс.

2.6.5 Впровадження та оновлення

Після завершення тестування та збору зворотного зв'язку, інтерфейс підлягає оновленню та оптимізації для кращого задоволення потреб користувачів.

Важливо вносити зміни відповідно до отриманих даних та відгуків. Це дозволяє адаптувати інтерфейс, покращуючи його зручність і функціональність. Регулярне оновлення інтерфейсу також має велике значення. Враховуючи новітні технології та актуальні тренди у дизайні, інтерфейс повинен постійно вдосконалюватися, щоб залишатися сучасним та привабливим.

Таким чином, систематичний підхід до проєктування інтерфейсу, включаючи внесення змін та регулярні оновлення, забезпечує створення кінцевого продукту, який є функціональним, візуально привабливим і зручним для користувачів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Програмна реалізація вебсайту для АЗС “Barrel” охоплює всі аспекти створення та налаштування функціональності сайту, включаючи серверну логіку, клієнтську частину, інтеграцію з базами даних та іншими сервісами. Важливо забезпечити, що сайт буде надійним, ефективним, і зручним для користувачів.

3.1 Вибір технологічного стеку

Для розробки сучасного вебсайту необхідно вибрати технології, які забезпечать ефективність, безпеку та користувацьку привабливість продукту. Нижче детально описано обраний технологічний стек для клієнтської (frontend) та серверної (backend) частин, а також систему управління базами даних і інтеграційні інструменти.

Frontend (Клієнтська частина):

- HTML/CSS/JavaScript: ці технології формують основу веброзробки. HTML відповідає за структуру сторінки, CSS визначає стильове оформлення, а JavaScript забезпечує інтерактивність та динамічність вебсайту [1];
- Vue.js: це ефективний фреймворк JavaScript, який дозволяє створювати інтерактивні користувацькі інтерфейси; Vue використовує компонентний підхід, що спрощує управління станом та повторне використання коду в масштабованих проєктах [7];
- Bootstrap: фреймворк, який допомагає швидко створювати адаптивний дизайн; Bootstrap містить готові стильові шаблони та компоненти, які можуть бути легко інтегровані з HTML та JavaScript, забезпечуючи консистентність і чуйність інтерфейсу на різних пристроях [10].

Backend (Серверна частина):

- Node.js: це серверна платформа, яка дозволяє виконувати JavaScript на сервері. Node.js популярна своєю високою продуктивністю та великою екосистемою модулів, що робить її ідеальною для створення швидких та ефективних вебдодатків [8];
- Express.js: фреймворк для Node.js, який спрощує розробку серверних додатків, забезпечуючи потужні інструменти для управління маршрутами, запитами і відповідями сервера [5];
- JWT (JSON Web Tokens): технологія, яка використовується для забезпечення безпеки сесій користувачів. JWT дозволяє безпечно передавати інформацію між клієнтом та сервером у формі компактного, самодостатнього токена.

Database (База даних):

- MySQL: популярна реляційна база даних, відома своєю надійністю та ефективністю у зберіганні структурованих даних. MySQL підтримує складні SQL-запити, транзакції та забезпечує високий рівень безпеки, що робить її ідеальною для широкого спектру вебдодатків [2].

Інтеграційні інструменти:

- REST API: інтерфейс, який дозволяє вебсайтам взаємодіяти з зовнішніми сервісами. REST API є стандартом для створення мережових запитів між клієнтами та серверами, дозволяючи інтегрувати такі сервіси, як системи оплати, соціальні мережі та інші зовнішні бази даних [9].

3.1.1 Розробка функціональності

Під час виконання завдань з верстки, був використаний файл для скидання базових стилів, знайдений на просторах інтернету.

Далі будуть зразки коду з виконаних завдань. У випадку з завданнями по верстці будуть додані тільки код HTML, тому що стилів дуже багато Почнемо з верстки вікна з планами (див. рис. 3.1).

```

<header class="header" v-if="!isMobile">
  <div class="header_logo">
    
  </div>
  <div class="header_menu">
    <a v-for="item in header_data" class="header_menu_item" v-bind:href="'#'+item.to">
      {{item.name}}
    </a>
    <div class="header_menu_button red_button" v-on:click="modal_show = true">
      Заповіти талони
      <div class="button_triangle"></div>
    </div>
  </div>
</header>
<header v-else class="header_burger">
  <Slide>
    <a v-for="item in header_data" class="header_menu_item header_burger_item" v-bind:href="'#'+item.to">
      {{item.name}}
    </a>
    <div class="header_menu_button red_button header_burger_button" v-on:click="modal_show = true">
      Заповіти талони
      <div class="button_triangle"></div>
    </div>

    <div class="header_burger_logo">
      
    </div>
  </Slide>
</header>

```

Рисунок 3.1 – Код верстки

Було впроваджено систему авторизації з використанням електронної пошти та пароллю, забезпечено безпеку паролів через хешування. Також розроблено CRUD (Create, Read, Update, Delete) операції для управління продуктами через адміністративну панель. Реалізовано корзину покупок для замовлення продуктів та інтегровано з платіжними системами для обробки онлайн-оплати. Крім того, були розроблені механізми для нарахування та використання бонусних балів у програмі лояльності.

Завдання на виробничу практику складається з двох пунктів.

Завдання на верстку. Створити першу сторінку на сайті заправок барель (див. рис. 3.2), меню (див. рис. 3.4), також талони (див. рис. 3.3), що рухаються, реалізувати інтерфейс по макету.

Завдання для JavaScript. Завдання передбачає повний розвиток функціональної частини сайту АЗС, включаючи інтеграцію обробників подій, реалізацію реакцій на натискання кнопок, а також моніторинг змін даних. Потрібно забезпечити безперебійну роботу всіх елементів інтерфейсу, щоб користувачі могли зручно та швидко отримувати необхідну інформацію та взаємодіяти із сервісом.

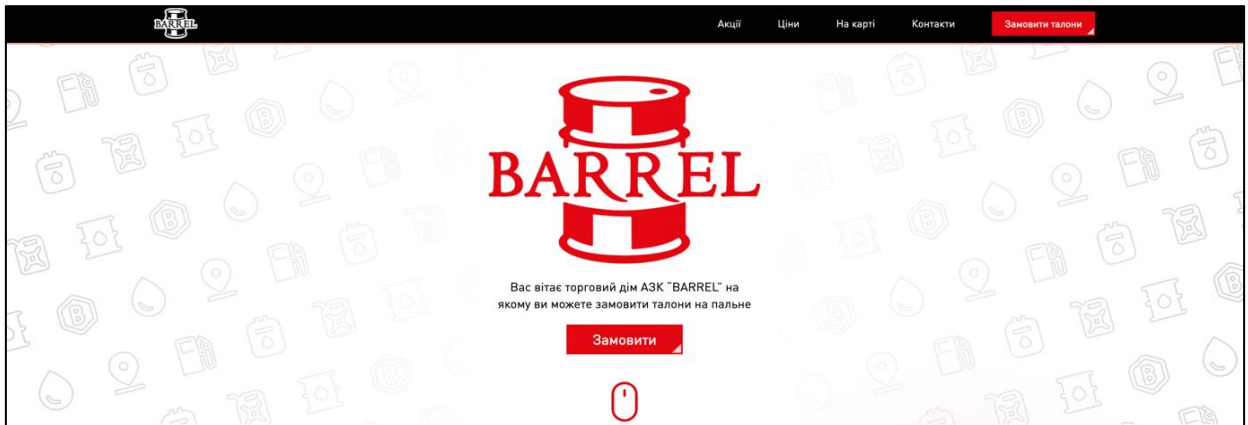


Рисунок 3.2 – Головна сторінка

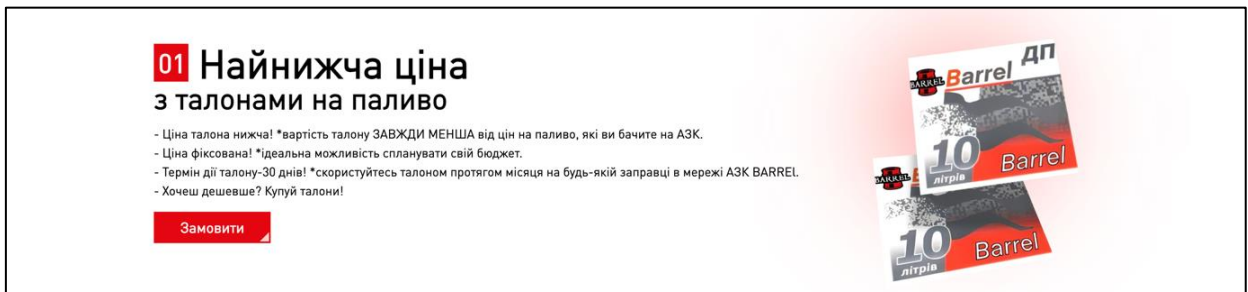


Рисунок 3.3 – Плюси купівлі талонів

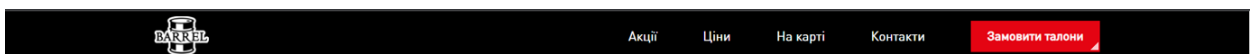


Рисунок 3.4 – Результат

3.1.2 Опис інструментарію

Я використовував широкий спектр інструментів, які можна використовувати для розробки вебсайтів.

Ось деякі з них: HTML/CSS використовується для створення структури та візуального оформлення вебсторінок. HTML відповідає за структуру контенту, а CSS за його стилізацію.

JavaScript: це мова програмування, яка дозволяє додавати інтерактивність до вебсторінок. Використовується для створення динамічних елементів, взаємодії з користувачем, обробки подій тощо.

Наведений код представляє два різних вигляди заголовка (`<header>`) залежно від умови `v-if="!isMobile"`. Якщо умова виконується (не мобільний пристрій), то відображається перший блок коду, якщо умова не виконується (мобільний пристрій), то відображається другий блок коду.

У першому блоці коду (не мобільний пристрій) розташовано логотип (``) у вигляді картинки з використанням `getImageUrl('logo_header.png')`. Після логотипу розміщено меню (`<div class="header__menu">`), що складається з набору посилань (`<a>`) з назвами елементів меню, які отримуються з даних `header_data`. Також у меню є кнопка «Замовити талони» (`<div class="header__menu_button red_button">`), при кліку на яку встановлюється значення `modal_show` на `true`.

У другому блоці коду (мобільний пристрій) використовується компонент `Slide`, в якому розташовано аналогічне меню, але з відповідними класами для адаптивного вигляду. Тут також є кнопка «Замовити талони», логотип та анімація.

Цей код використовує фреймворк або бібліотеку `Vue.js`, оскільки використовується директива `v-if` та `v-else`, а також директиви `v-for` та `v-bind:href`. Він працює з даними, що передаються в компонент та змінними, такими як `isMobile` та `header_data`. Також з'являється можливість взаємодії зі змінною `modal_show` при кліку на кнопку.

Для відображення місцезнаходження заправок було додано мапу з позначками станцій `Barrel` (див. рис. 3.5), а нижче наведено список із їхніми адресами (див. рис. 3.6).

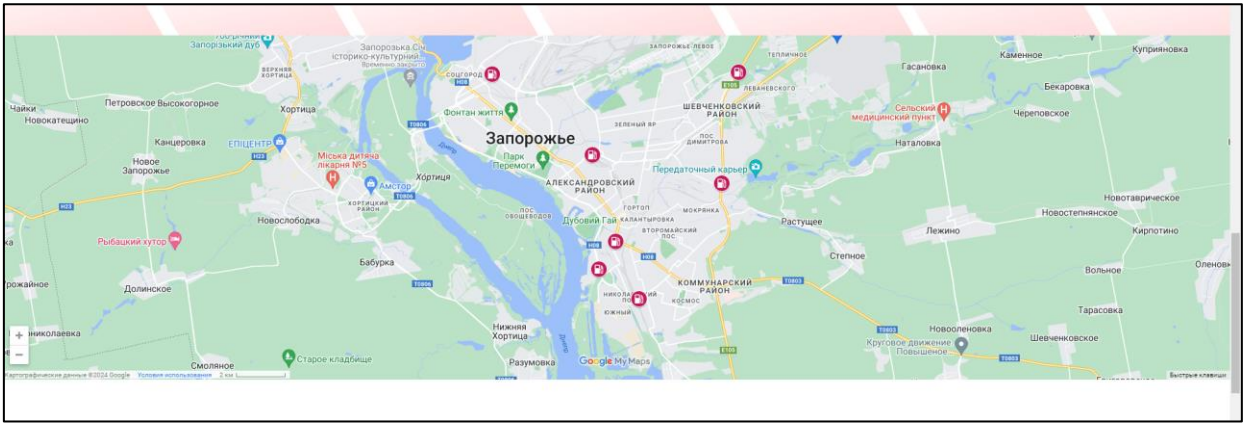


Рисунок 3.5 – Карта заправок Barrel

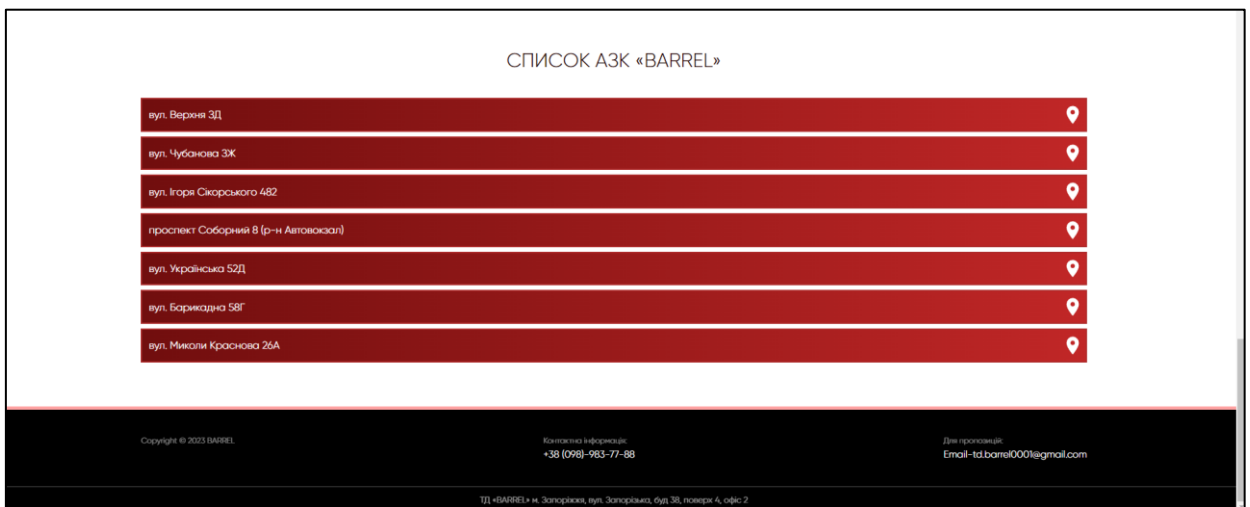


Рисунок 3.6 – Перелік заправок Barrel

За наступним завданням потрібно було зверстати головний екран. Код головного екрану (див. рис. 3.7). представляє блок з заголовком (`<div class="title">`), що складається з кількох елементів.

```

<div class="title">
  <div class="title_logo">
    
  </div>
  <div class="title_text">
    <span>Вас вітає торговий дом АЗС "BARREL" на якому ви можете замовити талони на пальне</span>
  </div>
  <div class="title_button red_button" v-on:click="modal_show = true">
    Замовити
    <div class="button_triangle"></div>
  </div>
  <div class="title_mouse">
    
  </div>
</div>

```

Рисунок 3.7 – Код головного екрану

У блоці заголовка розміщено логотип () у вигляді картинки з використанням `getImageUrl('title_logo.png')`. Після логотипу є елемент з текстом (<div class="title__text">), в якому розташовано рядок з вітанням та описом. Цей рядок містить текст «Вас вітає торговий дім АЗК “BARREL” на якому ви можете замовити талони на пальне» українською мовою (див. рис. 3.8).

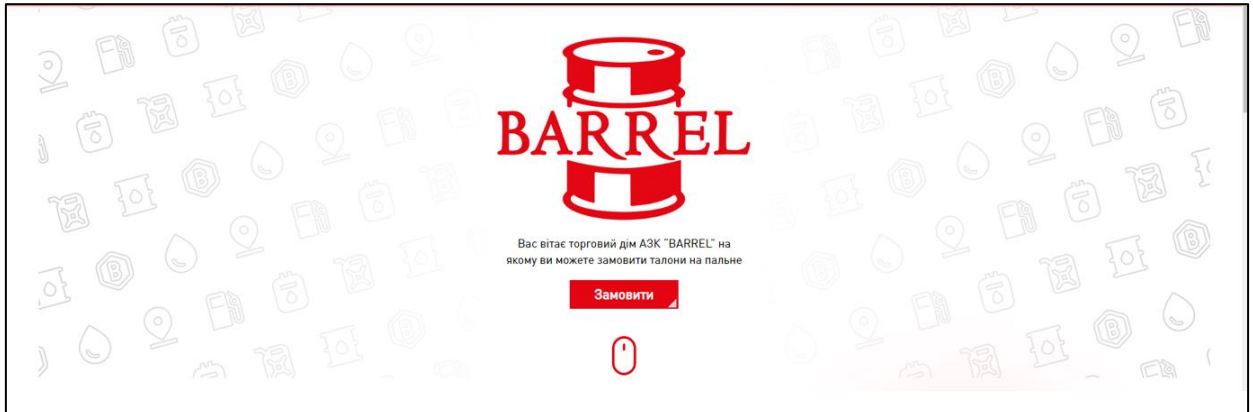


Рисунок 3.8 – Результат

Також у блоці заголовка є кнопка «Замовити» (<div class="title__button red_button">), при кліку на яку встановлюється значення `modal_show` на `true`. Ця кнопка має також елемент з трикутником (<div class="button_triangle"></div>), який, ймовірно, відображається як декоративний елемент.

Нарешті, є ще один елемент – зображення миші (<div class="title__mouse">) з використанням `getImageUrl('mouse.svg')`. Це може бути іконка миші, що вказує на прокрутку або взаємодію користувача зі сторінкою.

Цей код також використовує фреймворк або бібліотеку `Vue.js`, оскільки використовується директива `v-on:click` для обробки події кліку та директива `v-bind` для динамічного встановлення значень атрибутів `src` для зображень. Код також залежить від даних, які передаються в компонент, та взаємодії зі змінною `modal_show` при кліку на кнопку.

За наступним завданням потрібно було зверстати екран переваги (див. рис. 3.9).

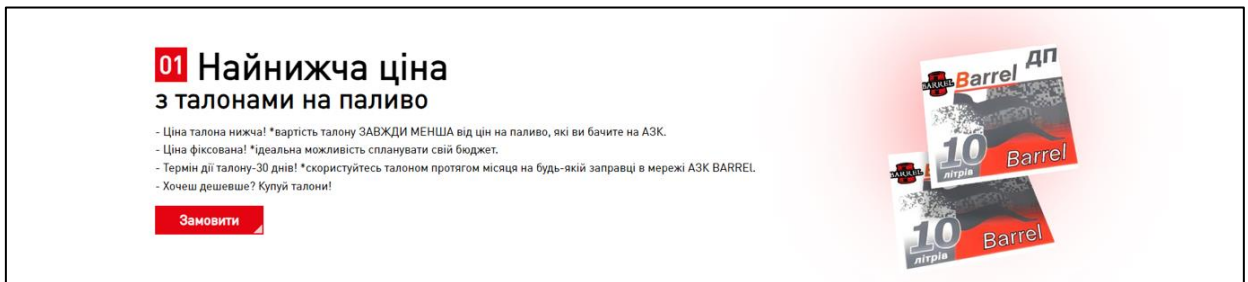


Рисунок 3.9 – Результат

Наведений код (див. рис. 3.10) представляє блок з перевагами (`<div class="advantages">`), який складається з кількох елементів.

```

<div class="advantages">
  <div class="advantages__content">
    <div class="advantages__content_title">
      <div class="advantages__content_title_first_row">
        <div class="point pl-7">01</div>
        <span>Найнижча ціна</span>
      </div>
      <div class="advantages__content_title_second_row">
        <span>з талонами на паливо</span>
      </div>
    </div>
    <div class="advantages__content_list">
      <div v-for="advantage in advantages" class="advantages__content_list_item">
        - {{advantage}}
      </div>
    </div>
    <div class="title_button advantages__content_button red_button" v-on:click="modal_show = true">
      Замовити
      <div class="button_triangle"></div>
    </div>
  </div>
  <div class="advantages__img">
    
    
  </div>
</div>

```

Рисунок 3.10 – Код екрану переваги

У блоці з перевагами є контейнер для вмісту (`<div class="advantages__content">`), в якому розташовані заголовок, список переваг та кнопка «Замовити».

Заголовок (`<div class="advantages__content_title">`) містить два рядки

тексту. Перший рядок містить номер переваги (`<div class="point pl-7">01</div>`) та назву переваги (`Найнижча ціна`). Другий рядок містить додатковий текст (`з талонами на паливо`).

У блоку переваг є список переваг (`<div class="advantages__content_list">`), який генерується з даних `advantages` за допомогою директиви `v-for`. Кожна перевага відображається як окремий елемент списку (`<div class="advantages__content_list__item">`) з префіксом «-» та текстом переваги.

Також у блоку переваг є кнопка «Замовити» (`<div class="title__button advantages__content_button red_button">`), при кліку на яку встановлюється значення `modal_show` на `true`. Ця кнопка також має елемент з трикутником (`<div class="button_triangle"></div>`), який, ймовірно, відображається як декоративний елемент.

У блоку переваг також є контейнер для зображень (`<div class="advantages__img">`), в якому розміщені два зображення з використанням `getImageUrl()`. Зображення можуть мати ідентифікатори `advantage_img_front` та `advantage_img_back`.

Цей код також залежить від фреймворка або бібліотеки Vue.js, оскільки використовує директиви `v-for`, `v-on:click` та `v-bind` для взаємодії з даними та подіями. Він використовує дані, які передаються в компонент, і змінну `modal_show` для відображення модального вікна при кліку на кнопку «Замовити».

ВИСНОВКИ

У ході написання кваліфікаційної роботи було створено функціональний і зручний вебсайт для автозаправної станції «Баррель», використовуючи CMS WordPress та технології HTML, CSS, JavaScript, а також систему управління базами даних MySQL. Ретельний аналіз вебсайтів-аналогів, розробка макетів сторінок та UML діаграм сприяли створенню ефективного та привабливого онлайн-ресурсу.

Проєкт зосереджувався на ключових аспектах вебдизайну і розробки: від аналізу предметної області до проєктування, реалізації функціональності та її тестування. Інтеграція сучасних технологій та дотримання стандартів користувацького досвіду значно підвищили якість вебдодатку, що є критично важливим для задоволення потреб сучасних користувачів.

Вебсайт «Баррель» вирізняється унікальним дизайном, який забезпечує легку навігацію і високу доступність його функцій, роблячи користувацький досвід простим і приємним. Виконання всіх поставлених завдань у рамках кваліфікаційного проєкту демонструє успішну реалізацію вебсайту автозаправної станції, що задовольняє всі сучасні вимоги до функціональності та дизайну.

ПЕРЕЛІК ПОСИЛАНЬ

1. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com> (дата звернення: 23.01.2024).
2. MySQL Official Website. URL: <https://dev.mysql.com/doc/> (дата звернення: 01.03.2024).
3. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/> (дата звернення: 04.03.2024).
4. Interaction Design Foundation. URL: <https://www.interaction-design.org> (дата звернення: 17.04.2024).
5. Express Documentation. URL: <https://expressjs.com/en/guide/routing.html> (дата звернення: 23.04.2024).
6. Google Developers. URL: <https://developers.google.com/learn?hl=en> (дата звернення: 03.05.2024).
7. Vue.js Documentation. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 07.05.2024).
8. Node.js v22.3.0 documentation. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 10.05.2024).
9. REST API Tutorial. URL: <https://dev.mysql.com/doc/> (дата звернення: 13.05.2024).
10. Bootstrap Documentation. URL: <https://getbootstrap.com/docs/4.1/getting-started/introduction> (дата звернення: 13.05.2024).
11. What Is Software Testing? All the Basics You Need to Know. URL: <https://www.testim.io/blog/software-testing-basics/> (дата звернення: 13.05.2024).
12. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 05.05.2024).
13. Що таке CRM-система: повний гід по вибору CRM для початківців. URL: <https://nethunt.ua/blog/shcho-takie-crm-sistiema-povnii-ghid-po-viboru-crm-dlia-pochatktivsiv/> (дата звернення: 23.04.2024).