

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут післядипломної освіти
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка системи інтернет-магазину "Автосервіс"

Виконав студент групи КН-5
спеціальності 122 Комп'ютерні науки
Хасратова Хаяла Самад кизи

Керівник доктор філософії, доцент
Бучинська Ірина Вікторівна

Рецензент д.ф.-м.н. професор
Ковальчук Володимир Володимирович

ЗМІСТ

СКРОЄННЯ ТА УМОВНІ ПОЗНАЧКИ	6
ВСТУП	7
1 СУЧАСНІ ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБ -САЙТІВ.....	9
2 ОГЛЯД ІСНУЮЧИХ СИСТЕМ	18
2.1 Огляд систем для автосервісу.....	18
2.1.1 Опис системи Carbook.....	18
2.1.2 Характеристика системи Бітрікс 24	19
2.1.3 Характеристка системи РемОнлайн.....	20
2.1.4 Характеристики системи Моє STO	20
2.1.5 Характеристика системи Альфа-Авто	21
2.2 Переваги автоматизованого автосервісу	23
2.3 Важливість планування сайту.....	23
2.3.1 Аналітика	24
2.3.2 Стратегія	24
2.3.3 Створення технічного завдання.....	25
2.3.4 Прототипування	25
2.3.5 Створення макетів дизайну.....	26
2.3.6 Верстка	26
2.3.7 Програмування	27
2.3.8 Базове наповнення	27
2.3.9 Тестування	28
2.3.10 Технічна підтримка.....	28
3 ПЕРЕВАГИ СТВОРЕННЯ САЙТУ АВТОСЕРВІСА	29
3.1 Структура розділів сайту автосервісу.....	30
3.2 Розкритка сайтів сто	31
3.3 Дизайн сайту автосервісу.....	32
3.4 Готовий сайт чи індивідуальна розробка	32

4 АНАЛІЗ СИСТЕМ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІОННОЇ СИСТЕМИ АВТОСЕРВІС	34
4.1 Аналіз вибору СУБД.....	34
4.2 Вибір технології програмування автоматизованої інформаційної системи	39
5 ОПИС ПРОГРАМНОГО ДОДАТКУ	41
ВИСНОВКИ.....	49
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	51

СКРОЄННЯ ТА УМОВНІ ПОЗНАЧКИ

CSM – Content Management System

HTML – HyperText Markup Language – мова маркування гіпертексту

W3C – World Wide Web

СУБД – Системи управління базами даних

ВСТУП

Інформаційна система реалізує інформаційну систему, що призначається для організації обліку, розподілу робіт за відділами та працівниками, а також веде облік вартості виконаних робіт. Планується, що система буде проводити введення, редагування та перегляд баз даних, відповідати на запити користувача та створювати різноманітні звіти.

Дана інформаційна система може бути використана в автосервісах. Вона дуже легка в обігу, дозволяє зберігати об'ємну кількість значень бази даних, за рахунок автоматизації деяких процесів (облік вартості робіт, облік розподілу робіт по відділах) заощаджує робочий час. Тому запропонована інформаційна система має суттєво спростити роботу автосервісу.

Створення сайту автосервісу – досить просте завдання під час використання відповідного інструменту.

Її реалізація під силу людині, яка не має досвіду розробки, програмування, верстки дизайну, роботи з хостингом та іншими технічними навичками.

Ця інформаційна система може використовуватися в службах автомобілів. Це легко розповсюджувати, дозволяє зберігати велику кількість інформації в одній базі даних, економить робочий час через автоматизацію деяких процесів, таких як, враховуючи вартість роботи, а також враховувати розподіл роботи на відділах. Тому запропонована програма повинна значно спростити роботу автомобільних послуг.

Для підвищення конкурентоспроможності необхідно впровадити інформаційну систему, розроблену з урахуванням сучасних Web-технологій, яка повинна містити елементи Customer Relationship Management (CRM)

CRM – це така стратегія організації бізнесу, в якій відносини з клієнтом одна з пріоритетних напрямів діяльності компанії [9].

Роль CRM-систем можна коротко охарактеризувати як ефективне використання всіх каналів комунікацій із клієнтом задля забезпечення його лояльності до компанії.

Наявність солідної бази клієнтів сьогодні є основним і чи не єдиним фактором стійкості та процвітання підприємств, що працюють у сфері надання послуг населенню.

Метою роботи є розробка автоматизованої інформаційної системи управління замовленнями клієнтів з елементами системи CRM для Автосервісу.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести аналіз предметної галузі;
- вибрати методології та технології проектування інформаційної системи;
- сформулювати вимоги до системи, що впроваджується;
- розробити логічну модель системи;
- виконати програмну реалізацію системи.

Робота складається з вступу, 5 розділів, таблиць, рисунків, переліку посилань.

1 СУЧАСНІ ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБ -САЙТІВ

Інформація, яка зберігається в Інтернеті, розташована на комп'ютерах або на так званих веб-серверах, на яких, у свою чергу, встановлено спеціальне програмне забезпечення, яке дозволяє користувачеві знайти необхідну інформацію. Значна частина такої інформації впорядкована у вигляді веб -сайтів, кожна з яких має власне ім'я або адресу в Інтернеті.

Веб-сайт (з англ. website: Web-Web, Network and Site-Place, буквальный переклад, сегмент, частина в мережі) – набір електронних документів (файлів) приватної особи або організації в комп'ютерній мережі, поєднана під однією адресою (доменне ім'я або IP-адреса) [1]. За замовчуванням розуміється, що сайт розташований в Інтернеті.

Таким чином, веб-сайт – це поєднання логічно пов'язаної з гіпертекстовою інформацією, розробленою у вигляді окремих сторінок та доступних в Інтернеті.

Для перегляду веб-сайтів на комп'ютері користувача необхідно встановити спеціальний браузер програмного забезпечення.

Браузер або програмне забезпечення, що застосовується веб-браузер для перегляду сторінок, обслуговування веб-документів, комп'ютерні файли та їх каталоги; управління веб-додатками; А також вирішувати інші проблеми. У глобальній мережі браузери використовуються для запиту, обробки, маніпулювання та відображення веб-сайтів [2].

На основі того, яка адреса (ім'я) веб-сайту встановлюється в рядку "Адреса", браузер відображає відповідну інформацію у своєму робочому вікні. Будь-який веб -сайт включає взаємопов'язані веб -сторінки. Кожна сторінка сайту містить текстовий файл з розширенням *.html, який містить текстову інформацію та спеціальні команди HTML, який, у свою чергу, визначає тип інформації для відображення в браузері.

HTML (з англ. HyperText Markup Language – Мова маркування гіпертексту) – це стандартизована мова мови у всесвітній павутині [3].

Таким чином, код HTML – це код, написаний мовою HTML.

Код HTML не включає в себе графічну, аудіо- та відео-інформацію. Це окремі файли, які посилання в коді HTML вказані [4].

В даний час при створенні веб-сайтів найчастіше та успішно використовуються два основні методи:

– перший – за допомогою готових двигунів (тобто використання готового шаблону);

– друге – це незалежне написання коду з використанням мови опису структури веб-сторінок HTML та каскадних стилів CSS.

Особливістю створення веб-сайтів з використанням готових двигунів є CSM (Content Management System) – система управління вмістом, яка складається з розробки сайту, без використання навичок програмування.

Цей метод створення сайту може бути реалізований у двох формах:

– платна опція – для прикладу, 1С-Бітрікс тощо;

– безкоштовний варіант – наприклад, WordPress, Drupal, Joomla тощо.

Незалежне створення сайту складається з структурування сайту, використовуючи мову мови мови веб-сторінок HTML та каскадних стилів CSS. У той же час Java Script використовується для забезпечення необхідної функціональності та динамізму.

Ці два методи можна порівняти за кількома аспектами.

Перший аспект – це виступ. Швидкість є важливою характеристикою роботи будь-якого Інтернет-ресурсу, що суттєво впливає на якість його роботи. Повільні сайти незручні для користувачів – мало хто чекатиме завантаження, якщо сайт буде чекати буде завантажуватися набагато довше, ніж інші. З таких ресурсів, як правило, вони йдуть швидше. Зайнятість також впливає на індексацію сайту.

Індексація в пошукових системах (веб-індексація) – процес додавання інформації (на сайті) роботом пошукової системи до бази даних, яка згодом використовується для (повного тексту) пошуку інформації про індексовані сайти [5].

Інформація про сайт найчастіше включає ключові слова (алгоритм визначення ключових слів залежить від пошукової системи), статті, посилання, документи, зображення, аудіо та тощо.

Ключові слова – це окремі слова (або їх комбінації), які користувачі Інтернету вступають у пошукову лінію, щоб знайти товари, послугу чи інформацію. Правильно вибрані ключові слова

Вони здатні забезпечити сайт великою притоком цільової аудиторії [6].

У зв'язку з цим індексація сайту в пошукових системах безпосередньо впливає на його відвідування.

На сайтах, створених на готових двигунах, буде представлена велика кількість файлів, які абсолютно непотрібні для його функціональності – тому сервер буде виконувати багато зайвих операцій. Додаткові файли дозволяють готові двигуни, щоб забезпечити універсальність використання, можливість задоволення різноманітні запити. Як правило, немає ідентичних сайтів, і кожен веб-ресурс має власні функції з різними завданнями та можливостями.

Навпаки, версія, записана самотійно, вказує на ті коди, необхідні для виконання функціональності інтернет-ресурс і більше нічого. У зв'язку з цим їхній код значно менший, що сприяє їх високій швидкості порівняно з готовими двигунами.

Наступний аспект – безпека. Один з ключових недоліків популярного CSM може бути визнаний відкритістю та доступністю їх коду. Відкритий код дозволяє всім, хто хоче побачити, як працює система, а потім виявити слабкі сторони коду, вразливість та помилку системи.

Третій аспект – функціональність. Якщо сайт зроблений на готовому двигуні CMS, то для зміни його функціональності необхідно

використовувати різні додаткові розширення. На сайтах, створених за допомогою написання власного коду, навпаки, усі необхідні доповнення та зміни спочатку створюються спеціально для їх завдань.

Більше того, якщо сайт зроблений незалежно, тоді подальший розвиток його функціональності не залежатиме ні від кого, а лише від себе в особі адміністратора веб-сайту.

Останній аспект, який ми розглянемо, – це простота створення. Сайт, розроблений на готовому двигуні, вимагає значного часу в процесі регулювання шаблону для конкретних завдань клієнта, а також для оптимізації роботи цього веб-ресурсу. У свою чергу, у другому варіанті створення сайту-велика кількість часу потребує незалежно запису коду, використовуючи мову опису структури веб-сторінок HTML та каскадних таблиць стилів CSS.

Таким чином, розробка сайту на готовому двигуні займе набагато менше часу, ніж незалежний код письма.

Враховуючи вище, здається доречним створити сайт щодо використання технологій HTML, CSS, JavaScript.

Позначення мови – HTML-коду – це рідна мова для будь-якого браузера та веб-сайту.

HTML був винайдений Тімом Бернерс-Лі, фізика з науково-дослідного інституту CERN у Швейцарії, він придумав ідею Інтернет-гіпертекстної системи [7].

Гіпертекст означає текст, що містить посилання на інші тексти, які глядачі можуть отримати негайно. Він опублікував першу версію HTML у 1991 році, що складається з 18 тегів HTML [8]. Відтоді кожна нова версія мови HTML з'явилася з маркуванням нових тегів та атрибутів (модифікатори тегів).

Тег (іноді тег, англійський тег - названа позначка; більш правильним іменем є дескриптор). У HTML є елемент мови маркування гіпертексту.

Текст, що міститься між початковим та кінцевим тегом, відображається та розміщується відповідно до властивостей, зазначених у початковому тегу [9].

Згідно з посиланням на елемент HTML від Mozilla Developer Network, в даний час існує 140 тегів HTML, хоча деякі з них вже застаріли (не підтримуються сучасними браузерами) [10].

Через швидке зростання популярності-HTML зараз вважається офіційним веб-стандартом. Технічні характеристики HTML підтримуються та розробляються консорціумом World Wide Web (W3C). Ви можете перевірити останній стан мови в будь-який час на веб-сайті W3C (<https://www.w3.org/>).

Найбільшим оновленням мови було введення HTML5 у 2014 році. До позначок було додано кілька нових семантичних тегів, які показують значення власного вмісту, наприклад, <statia>, <doader> та <pooter> [39].

Отже, мова сторінок HTML HyperText (HTML – Мова розмітки гіпертексту) – це мова для створення веб-документів. Він визначає синтаксис та розміщення спеціальних інструкцій (тегів), які не відображаються на екрані, але вказують браузеру, як відобразити вміст документа [11].

Він також використовується для створення посилань на інші документи, локальні чи мережу, наприклад, розташовані в Інтернеті.

Документи HTML – це файли, які закінчуються розширенням *.html або *.htm. Будь-який веб-браузер дозволяє переглядати документи HTML (наприклад, Google Chrome, Safari або Mozilla Firefox). Браузер зчитує файл HTML та відображає його вміст, щоб користувачі Інтернету могли його переглянути.

Зазвичай середній веб-сайт включає кілька різних сторінок HTML (головна сторінка, тематичні сторінки, контакти контактів та інше), які мають окремі документи HTML.

Використовуючи єдину мову сторінок HTML HyperText у нашій роботі, ми не отримуємо повністю чуйного веб-сайту. Оскільки ця мова дозволяє лише створювати структуру вмісту та додавати текстові елементи.

Однак HTML чудово працює з каскадними стилями CSS та JavaScript. Разом вони можуть забезпечити багатий інтерфейс користувача та реалізувати розширені функції:

- CSS наразі повністю відповідає за знуцання (позиціонування) та дизайн сторінок;
- JavaScript дозволяє додавати різні анімаційні ефекти, реагувати на події (обробить рухи вказівника миші, натискаючи клавіші з клавіатури), перегляньте запис даних у поля перед надсиланням на сервер, що, в свою чергу сервер тощо.

Каскадні таблиці CSC CSS (каскадні аркуші стилю) надають додаткові можливості для позначення документа HTML та свободи для створення унікального дизайну для веб -сторінок [12].

CSS – це технологія опису появи документа, розробленого мовою маркування.

Якщо HTML надає інформацію про структуру документа, то таблиці стилів повідомляються, як це повинно виглядати.

Коли браузер обробляє код HTML, він використовує стиль представлення за замовчуванням елементів HTML на веб-сторінці.

Стиль – це набір правил, застосованих до елемента гіпертексту та визначення способу його відображення. Стиль включає всі типи елементів дизайну: шрифт, фон, текст, посилання, поля та розташування об'єктів на сторінці.

Таблиця стилів – це набір стилів, що застосовуються до документа з гіпертексту.

Каскад – це процедура застосування різних стилів на веб-сторінку [13]. Браузер, який підтримує таблиці стилів, буде послідовно використовувати їх

відповідно до пріоритету: спочатку підключений, потім реалізований і, нарешті, побудований в стилі. Інший аспект каскадування – успадкування (успадкування) - означає, що якщо в іншому випадку конкретний стиль буде застосований до всіх дочірніх підприємств документа з гіпертексту.

За допомогою CSS ви можете зменшити встановлений стиль за замовчуванням для елементів для власних, тим самим створивши унікальний стиль дизайну для елементів веб-сторінки Червона рама тощо.

Стиль – це правило, що описує окремий елемент на сторінці. Стили були додані для вирішення проблем проектування веб-сторінок. Стили можна зберігати в окремих документах, що зменшує розмір HTML -коду.

Ви також можете зібрати всі стилі, які використовуються на сайті в одному зовнішньому файлі з розширенням *.css та з'єднати його з усіма сторінками сайту. Після цього, редагуючи стиль, змінюється миттєво впливає на всі елементи на сторінках сайту, де є посилання на цей зовнішній файл із стилями. Таким чином, ви можете повністю змінити зовнішній вигляд, редагуючи єдиний файл таблиці стилів, який значно спрощує роботу, ніж стилі редагування на кожній сторінці окремо.

Можливості CSS та HTML спрямовані на досягнення різних цілей: використання CSS призначене безпосередньо для дизайну

Веб-сторінки та HTML використовуються лише для своєї передбачуваної мети-тобто саме для позначення веб-сторінки до логічних фрагментів, не піклуючись про форматування та появу сторінки.

Мова програмування сценарію – це повна динамічна мова програмування, яка застосовується до документа HTML і може забезпечити динамічну інтерактивність на веб-сайтах [14]. Використовуючи JavaScript, ви можете відображати додаткову інформацію про посилання, створювати інтерактивні ефекти під час роботи з мишею, змінити вміст сторінок за певних умов, випадковим чином відображати вміст сторінки та (за

допомогою CSS) Перемістіть елементи разом сторінка. Додаткові посібники JavaScript можна знайти на сторінках WebCoder.com.

Сценарії JavaScript, якщо їх є невеликою кількістю, зазвичай розміщуються безпосередньо в документі HTML (сценарії можуть бути розташовані або в заголовку, або в тілі документа). В іншому випадку найбільш необхідні сценарії розміщуються в HTML, решта в окремому файлі.

У браузерах за замовчуванням спеціальне програмне забезпечення будується інтерпретатором JavaScript. Це необхідно, щоб браузер міг виконати код, написаний мовою JavaScript.

Як правило, JavaScript називається мовою клієнта, тим самим підкреслюючи, що сценарій виконується на клієнтському комп'ютері в браузері, а не на веб-сервері.

Ви можете розрізнити, принаймні дві сильні сторони JavaScript:

- повна інтеграція з HTML та CSS;
- він підтримується всіма основними браузерами і включається за замовчуванням.

Щоб спростити роботу, ми також будемо використовувати Bootstrap та jQuery. Ці інструменти дають готові рішення, і їх використання дозволяє скоротити час, витрачений на створення макета сайту:

- Bootstrap (також відомий як Twitter Bootstrap) – безкоштовний набір інструментів для створення сайтів та веб-додатків. Включає форми дизайну HTML та CSS для типографії, веб-форми, кнопки, мітки, блоки навігації та інші компоненти веб-інтерфейсу, включаючи розширення JavaScript [15];
- JQuery – це бібліотека JavaScript, яка зосереджується на взаємодії JavaScript, HTML та CSS [16].

Таким чином, у нашій роботі ми будемо використовувати:

- мова маркування HTML-A, яка використовується для візуального та семантичного структурування веб-вмісту (визначає розділи документа, заголовки, таблиці даних, зображення, інформація про аудіо/відео тощо);
- таблиці стилів CSS, з якими встановлено стиль дисплея вмісту HTML (фоновий колір та колір шрифту, розміщення елемента на сторінці, розміри елементів тощо);
- JavaScript – мова програмування, яка дозволяє створювати динамічно оновлений вміст, керує мультимедією, оживляє зображення тощо;
- Freimvorki jQuery та Bootstrap для забезпечення інтерактивності веб-сторінок та сучасного дизайну.

2 ОГЛЯД ІСНУЮЧИХ СИСТЕМ

2.1 Огляд систем для автосервісу

Проведене одним з дистриб'юторів опитування, показало, що в Україні найбільш популярним інструментом керування СТО залишається традиційна 1С. Втім автосервіси, які прагнуть розвиватися й бути в тренді сучасних тенденцій, використовують спеціалізований софт. Адже європейська практика доводить, що майбутнє – за програмним забезпеченням, розробленим під потреби СТО. В розділі описано п'ятірку кращих функціональних програм на вітчизняному ринку.

Головна перевага CRM-системи, заточеної під автосервіс – вона враховує особливості роботи цього бізнесу, сприяє комунікації спеціалістів, задіяних у різних напрямках роботи СТО, допомагає налагодити ефективну роботу з клієнтами, партнерами та працівниками. Сьогодні практично всі системи працюють на різних гаджетах, що дозволяє контролювати свій бізнес з різної точки планети де є інтернет. [17]

2.1.1 Опис системи Carbook

Багатофункціональний софт, який налаштований на автоматизацію складського обліку та електронного документообігу. Програмне забезпечення дозволяє контролювати перебіг робіт, організувати оповіщення клієнтів, запис на зустріч, повідомлення про закінчення ремонту, розсилка інформації про акції та особливі пропозиції СТО. Керівник може контролювати простої, бачити перебіг ремонтних робіт з будь-якого девайсу. Розробник заявляє про багаторівневий захист від хакерських атак.

Це програмне забезпечення для СТО може інтегрувати через API різні бази даних. Наприклад, базу запчастин TecDoc. Залити дані з 1С, що дозволяє синхронізувати інформацію про клієнтів, оформлення наряд-

замовлень. Є змога під'єднати IP-телефонію для інтегрування дзвінків у власну CRM та багато інших можливостей.

2.1.2 Характеристика системи Бітрікс 24

Головна мета цієї CRM-системи для автосервіса (рис. 1) – автоматизація прийому звернень клієнтів й підвищення швидкості обробки заявок. А також налагодження роботи менеджерів та автоматизація продажів запчастин. Програма дозволяє контролювати розклад роботи майстрів та їх робочого часу. Є можливість реалізувати лендінг або власний сайт компанії з онлайн-чатом.

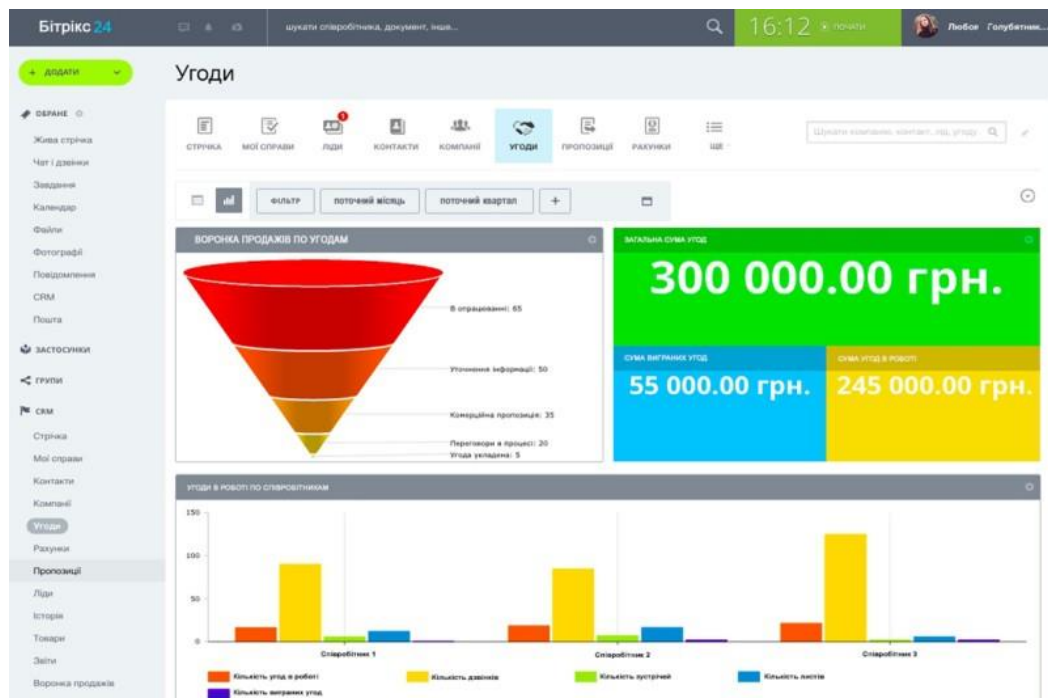


Рисунок 1 – Система Бітрікс 24

Загалом система передбачає як хмарну версію, так і коробкову. Хмарна безкоштовна версія містить 5 Гб вільного простору на віртуальному диску, чат, календар та можливість здійснювати відеодзвінки в межах компанії. Коробкова версія за замовчуванням містить інтернет-магазин та безпосередньо CRM.

2.1.3 Характеристика системи РемОнлайн

Програмне забезпечення не може надати широкого функціоналу, втім, чудово дозволяє обробляти потік клієнтів. Автоматизація спілкування з клієнтом, інформування його про перебіг робіт, також має позитивний вплив на ефективність роботи постів. Добре організований планувальник замовлень, дозволяє раціонально витратити робочий час та уникнути черг та простоїв. Софт розділяє клієнтів на фізичних та юридичних осіб, дозволяє отримати інформацію про рівень задоволення автосервісом. Клієнт має змогу коментувати виконані роботи, залишати фото тощо (рис. 2).

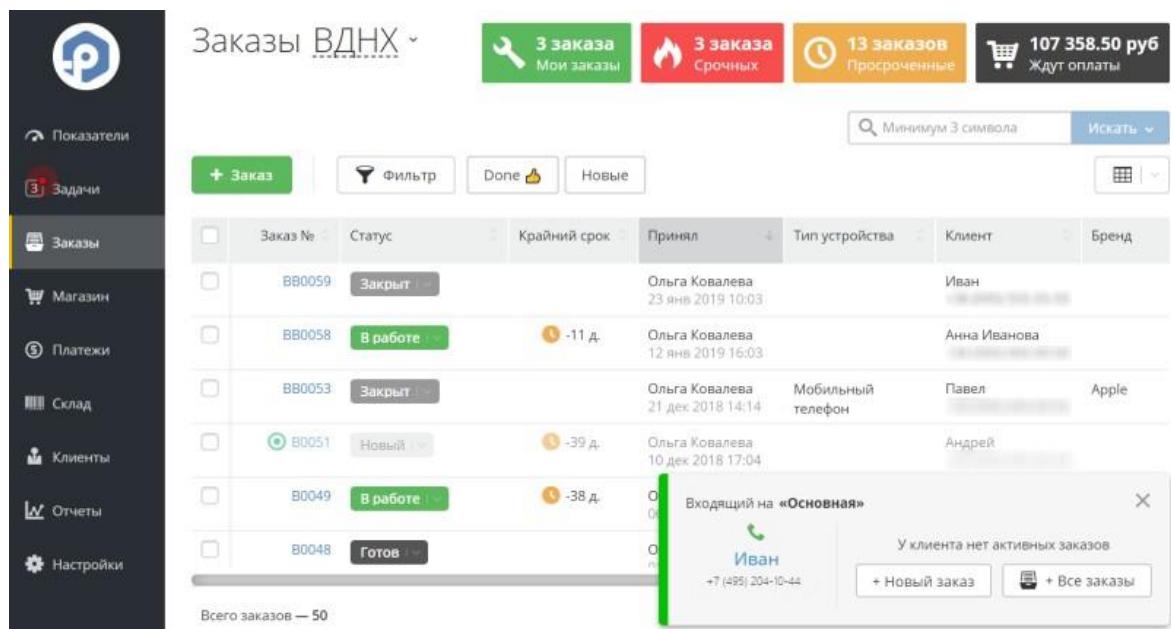


Рисунок 2 – Система РемОнлайн

2.1.4 Характеристики системи Моє STO

Нещодавно з'явилася нова CRM-система МОЕ-STO, яку українські програмісти розробили спільно німецькими фахівцями з компанії ТесAlliance. Головна фішка софту – інтегрований офіційний каталог ТесDoc. Система пропонує найвищий рівень автоматизації – підбір запчастин, пошук та облік нормо-годин, автоматизований розрахунок вартості послуг. Є змога

керувати списками постачальників запчастин, вартістю окремих послуг, створювати базу нестандартних послуг для спрощення обліку.

Програма має також налагоджену систему комунікації з клієнтами: онлайн-запис на СТО (рис. 3), інформування про перебіг робіт. Це дозволить уникнути черг або простоїв. Загалом CRM-система МОЕ-СТО має на меті допомогти керівнику ефективніше використовувати робочий час працівників та стандартизувати робочі процеси автосервісу. Відеоогляд основних функцій МОЕ-СТО ви можете подивитися на YouTube каналі.



Рисунок 3 – Система РемОнлайн

2.1.5 Характеристика системи Альфа-Авто

Досить поширений серед СТО програмний пакет Альфа-Авто побудовано цілком на базі софту 1С:Підприємство. Його розроблено задля охоплення максимальної кількості підприємств, що працюють у технічній галузі.

Система може однаково ефективно працювати як з автосалоном з продажу нових автомобілів, складом або магазином запчастин, так і з автосервісом. Щодо СТО, Альфа-Авто пропонує облік запчастин за

призначенням, оформлення індивідуальних карток автомобілів та їх власників й потужні інструменти для ведення документації.



Рисунок 4 – Система РемОнлайн

Оскільки платформа 1С це першою чергою робота з документами, то і для автосервісів тут відкриваються певні можливості. Наприклад, керівництво сервісу може оперативно формувати наряд-замовлення, контролювати витрати на закупівлю запчастин, синхронізувати дані з бухгалтерськими системами, відслідковувати дані щодо каталогів виробників.

Наразі понад 3000 підприємств в Україні та країнах СНД використовують саме версію Альфа-Авто. Єдине, що спиняє багатьох – система ліцензування. Тому що підприємству необхідно купити ліцензію не тільки на програмне забезпечення Альфа-Авто, але й мати повну ліцензію на пакет 1С. До того ж окремі необхідні модулі, наприклад, система контролю робочих місць, каталоги даних Silver-Dat, Audatex, AutoData потрібно купувати додатково.[17]

2.2 Переваги автоматизованого автосервісу

Безумовно, CRM-система здатна істотно покращити ефективність роботи автосервісу. Незалежно від типу системи, загальні переваги будуть відчутні відразу:

- підвищення ефективності працівників сервісу (більшість систем дають змогу рівномірно розподілити навантаження між постами та майстрами, аби уникати черг та простоїв);
- аналітика (система дає змогу аналізувати витрати та доходи підприємства, простежити ефективність акцій та реклами, контролювати рух запчастин тощо, все це допоможе приймати правильні стратегічні управлінські рішення);
- автоматизація документообігу (програмне забезпечення допомагає уникнути помилок та позбавитись рутинної паперової роботи, сконцентруватись на основних виробничих процесах);
- робота з клієнтом (у різних CRM різні можливості, але всі вони намагаються якомога більше уваги приділяти саме клієнту – сповіщення про перебіг робіт, дату та час візиту, про акції та знижки, проводити опитування, що збільшує кількість лояльних клієнтів);
- облік запчастин та доступ до технічної документації, автоматизований підбір запчастин з офіційних каталогів, доступ до технологічних карт ремонту та обслуговування за стандартами виробника автомобіля виводить станцію на якісно новий рівень роботи обслуговування клієнтів.[17]

2.3 Важливість планування сайту

Занадто багато людей відразу починають «забивати цвяхи», не витрачаючи часу створення належного плану. Але суть у тому, що яку б роботу не виконували (створювали сайт, будували будинок або починали маркетингову компанію), необхідно дві третини часу витратити на

планування і лише одну третину – на здійснення задуманого. Економія часу на етапі планування проекту призводить до непередбачуваних помилок, через які в результаті доводиться переробляти частину роботи.

2.3.1 Аналітика

Почати варто з визначення цільової аудиторії та аналізу конкурентів. Власнику сайту важливо розуміти свою цільову аудиторію, її потреби та біль. Тоді він зможе адаптувати свій сайт під їхні інтереси та отримати з цього власну вигоду. Варто знайти відповіді на запитання:

- для кого призначено веб-ресурс?
- які переваги ваших потенційних клієнтів?
- яка інформація буде для них корисною та цікавою?
- які комунікаційні канали вони користуються?

Перегляньте своїх прямих конкурентів, вивчіть їх рішення, знайдіть переваги та недоліки. А потім використовуйте це у своїх цілях. Визначте свої переваги, свою родзинку, яка вас відрізнятиме і показуватиме у вигідному світлі.

2.3.2 Стратегія

Визначте мету вашого сайту, чого ви хочете досягти за допомогою нього та які інструменти вам у цьому допоможуть. Цілі у бізнес-ресурсу можуть бути абсолютно різні:

- продавати (для e-commerce);
- залучати клієнтів;
- презентувати продукт чи лінійку товарів;
- розповісти про послугу;
- збирати ліди (контактні дані);
- розповісти про вашу компанію;
- вибудувати довгострокові відносини із клієнтами;
- інформувати ЦА;

- підвищити охоплення та впізнаваність бренду, використовуючи digital-канали.

Один сайт може підходити одразу для кількох цілей, існують також складні індивідуальні проекти, які виконують кожне з перерахованих завдань. Дуже важливо правильно визначити, який результат потрібний і на основі цієї інформації формувати інші етапи роботи над створенням сайту.

2.3.3 Створення технічного завдання

Вкрай важливий крок. Похибки тут неприпустимі, оскільки можуть проявитися будь-якому етапі й у результаті відкинути проект далеко тому. У ТЗ прописуються основні вимоги проекту, його можливості та спосіб роботи, враховуючи аналіз ЦА та конкурентів, цілі та стратегію. По суті, два попередні етапи проводять заради того, щоб скласти якісне ТЗ та передати у роботу фахівцям.

Основні вимоги до техзавдання:

- Подробиці: все, що має зробити фахівець, необхідно вказати у ТЗ, навіть якщо це обов'язкова частина роботи, навіть якщо розробник робив це сотні разів і не потребує нагадування, та кожну вимогу, кожен аспект робіт слід прописувати;
- відсутність неточностей: будь-яким неоднозначним формулюванням чи суб'єктивізму не місце у техзаданні, це інструкція, в якій йдеться про те, що і як зробити, а поняття: «добре», «красиво», «чіпляє» у всіх різні;
- зрозуміла мова: фахівець не повинен розшифровувати та перепитувати, що саме від нього хочуть.

2.3.4 Прототипування

Для наочності створюється прототип із докладним розташуванням елементів та навігацією. Остання потребує особливої уваги. Занадто багато сайтів створюються без урахування зручності навігації. Чому так багато

людей люблять продукти Apple? Тому що вони інтуїтивно зрозумілі. Користувач при першому контакті з продуктом розуміє, як ним користуватися.

Дизайнер насамперед забезпечує зручність інтерфейсу. Всі елементи на сторінці повинні бути розташовані таким чином, щоб користувач легко міг знайти і скористатися ними. За допомогою розташування інформаційних блоків дизайнер керує увагою відвідувача, демонструє йому те, що він повинен або хоче побачити зараз.

2.3.5 Створення макетів дизайну

Етап розробки дизайну сайту відповідає за візуальну складову ресурсу:

- колір;
- форма;
- шрифт;
- анімація.

Дизайнер дотримується фірмового стилю замовника, вписує логотип, робить сайт повністю пізнаваним та відповідним компанії.

Наслідувати тренди добре, але не робіть дизайн занадто "модним", він повинен бути сучасним і зберігати свою актуальність хоча б два роки. Пізніше, коли ресурс заслуговує на довіру пошукових роботів, відвідувачів і клієнтів, ви зможете змінити його зовнішній вигляд.

2.3.6 Верстка

Після завершення роботи дизайнер відправляє макет на затвердження замовнику і потім передає на адаптивну верстку. Вона має на увазі переклад ескізів в HTML, тобто адаптацію зображення сайту під будь-який браузер та пристрій (ПК, планшет, смартфон та інші гаджети).

Це якраз той випадок, коли «сім разів відміряй, один відріж». Краще 7 разів промальовувати дизайн сайту і один - його зверстати в HTML, ніж відправити на верстку перший варіант, а потім нескінченно вносити правки і

переверстувати. І тут має бути уважним насамперед замовник. Саме він утверджує концепцію сайту. Чим активніше замовник бере участь в обговоренні дизайну, тим швидше він отримає ресурс, що відповідає його очікуванням.

2.3.7 Програмування

Навіщо потрібне програмування? Начебто дизайнери все промалювали, верстальники переклали ескізи в HTML - адаптували макет сайту під Всесвітню мережу. Що далі? А далі на сцену виходить програміст, який пожвавлює картинки. Тут краще пояснити з прикладу кнопки призыва. Зробив її дизайнер прямокутною і зеленого кольору, помістив у неї гасло (припустимо, «Забронювати квитки зараз»). Все це добре. Але ж потрібно зробити кнопку активною! Щоб відвідувач одразу попадав на заповітну сторінку бронювання квитка, а не бився в істериці, бачачи, що від клікання мишки дива не походить.

Загалом, програміст – це той, хто всі зроблені дизайнером та зібрані верстальником атрибути сайту прописує певною мовою програмування (PHP, Java, JavaScript, Python або технології ASP.NET).

Іноді етап верстки та програмування виконують одночасно. Якщо, звісно, дозволяють масштаби проекту.

2.3.8 Базове наповнення

Сайт без інформаційного наповнення, як шпаківня без птахів. Етап заповнення передбачає переведення контенту у необхідний формат та розміщення на сторінках ресурсу. Матеріали готуються заздалегідь. Їх можуть писати як копірайтери за заздалегідь складеним технічним завданням (ТЗ), так і власник сайту або будь-яка людина, яка знає специфіку текстів і володіє мистецтвом слова. Але на цьому робота не припиняється. Контент потребує стабільного оновлення. Інакше пошукові боти під час

ранжирування обходитимуть сайт. Важливим фактором є зацікавленість клієнтів. Їх важливо постійно підігрівати актуальною інформацією.

2.3.9 Тестування

Хоча після кожного етапу розробки ресурсу перевіряють на предмет «працює – не працює», без фінального тестування не обійтись. А раптом десь загубилося бите посилання, некоректно відображається інформація чи спливаюче вікно не вискакує на екран? Недоопрацювання під час тестування виявляються різні і команда їх відразу усуває. Як правило, для тестування та демонстрації сайт розміщують на спеціальному майданчику. Пізніше, коли всі похибки усунуті, проект з усіма файлами переносять на його постійне місце прописки в Інтернеті – хостинг. Але тут ще не все. Після перенесення ресурс знову тестують.

2.3.10 Технічна підтримка

Під час передачі готового проекту фахівці навчають роботі з адмінпанеллю, а також із сервісами збору статистики. Після цього власник може повністю самостійно публікувати новий контент, створювати сторінки, розміщувати товари, збирати дані для аналітики поведінкових показників та користуватись іншим функціоналом ресурсу. Якщо виникають питання, компанія-розробник завжди готова допомогти розібратися. Це стосується і багів, якщо раптом вони з'явилися після релізу сайту - команда техпідтримки їх тут же усуне.

3 ПЕРЕВАГИ СТВОРЕННЯ САЙТУ АВТОСЕРВІСА

Сайт автосервісу, як і будь-який інший бізнес-проект, призначений для залучення клієнтів, інформування їх про послуги та ціни. Роблять його задля душі, а справи. Багато людей переїхали в Інтернет, у всіх є смартфони і авто, що раптово зламалося, здатне змусити власника пошукати в Мережі найближчу швидку допомогу травмованому вірному залізному другу. Або навпаки: лежачи не дивані, людина моніторить пропозиції СТО у своєму районі, щоб нарешті полагодити машину, що дісталася від тестя. Ситуацій маса і всі вони здатні привести вам нових клієнтів.

Головне – мати хороший сайт, здатний вселити довіру потенційному клієнту. Що цьому сприяє? Підходящий дизайн, повна інформація щодо послуг та їх вартості, наявність карти проїзду, контактних даних, всіляких акцій, відгуків клієнтів та всього так. Чим більше корисних для справи подробиць зрозуміє людина, зайшовши на сайт СТО і чим менше він часу на це витратить, тим більша ймовірність звернення в сервіс. Загалом, як і будь-якому іншому бізнесі. Вам повинні повірити перед тим, як нести гроші.

Сайт буде наводити клієнтів на постійній основі без перерв на обіди, лікарняні та декрети. Звичайно, є великий сенс у тому, щоби робити його якісним. Навряд чи ви часто будете вносити зміни в дизайн або наповнення, тому потурбуватися про написання якісних текстів, зйомку класних фото і придумування соковитих акцій варто відразу. Після публікації бажано підключити рекламу в соціальних мережах та Яндекс.Дірект. Чим більший кількості народу ваш проект траплятиметься на очі, тим більший прибуток він принесе. Серед відвідувачів сайтів автосервісів мало випадкових перехожих – 90% є представниками цільової, платоспроможної аудиторії.

Те, за чийі послуги люди понесуть гроші – ваші чи конкурентів, багато в чому залежить від якості сайту та пропозицій, розміщених на його сторінках.

Загалом створення сайту СТО – справа відповідальна. Підходити до розв'язання задачі необхідно у всеозброєнні, а саме правильно вибравши платформу для його створення.

3.1 Структура розділів сайту автосервісу

Розділ із описом послуг сервісу:

- марки авто (спеціалізуєтесь на сервісі та ремонті певних марок та моделей авто);
- каталог продукції (можна додати на сайт каталог автотоварів, запчастин та будь-яких товарів; ступінь деталізації інформації може бути різним – сторінки по товарних групах або окремі сторінки для опису кожної моделі товару, і це може бути як вітрина товару, так і повноцінний інтернет магазин з можливістю замовлення прямо на сайті);
- про компанію (розповідь про компанію потенційним покупцям, розміщення історії компанії, місії, ліцензії та дозвільну документацію, список клієнтів та партнерів);
- новини (розділ для розміщення новин компанії, галузевих новин або новин інтернету);
- акції та спецпропозиції (привернення більшої уваги до знижок та найвигідніших пропозицій);
- відгуки, поради (скромність не прикрасить сайт, дозволити всім побачити, як інші оцінили переваги роботи з Вами);
- приклади робіт (розміщення на сайті прикладів робіт, наприклад автотюнінгу, відновлення авто "до і після");
- питання відповідь (у цьому розділі можна розміщувати відповіді на запитання клієнтів);

- статті, документація, інша інформація (розміщення більшої корисної інформації для користувача: тематичні статті, ГОСТи та техдокументація,
- фотогалерея (створення на сайті альбомів з фото – фото сервісу, подій компанії, та показати найкращі сторони).
- контакти (детальна інформація із контактними даними: адреси сервісів, офісів, магазинів, філій, телефони, інші контакти. Схема проїзду чи Goggle Картб допоможуть легко знайти автосервіс).

Для розробки структури та функціоналу сайту можна включати будь-які з вищезгаданих розділів, виходячи з особливостей бізнесу.

3.2 Розкрутка сайтів сто

Додатково необхідно розробляти розділи, що підвищують інформативність та привабливість сайту. Наприклад: "Майстер місяця", "Новинки автосалонів", "Автопрофі". І, звичайно, не забудемо про Зворотній зв'язок, Карту сайту.

Ще на етапі розробки починається формування довіри до автосервісу. Мета – збільшення кількості лояльних Клієнтів, привернення уваги до спеціальних пропозицій, підвищення рівня прийняття рішення під час відвідування сайту, збільшення кількості переглянутих сторінок та кількості замовлень.

Вміло підкреслити переваги: професійно, вигідно, швидко.

Сфокусувати увагу на сильних сторонах:

- професіоналізм;
- зручний час роботи;
- швидка діагностика;
- сучасне обладнання;
- досвідчені фахівці;

3.3 Дизайн сайту автосервісу

Сайт повинен мати красивий та стильний дизайн. Сайт має бути з унікальним дизайном, розробленим спеціально для компанії з урахуванням усіх його нюансів та особливостей.

Дизайн забезпечить правильне емоційне сприйняття сайту, підкреслить солідність, надійність, успішність.

3.4 Готовий сайт чи індивідуальна розробка

Для розробки сайту автосервісу під ключ, запропоновано різні варіанти розробки. Заощадити час дозволить використання готового рішення для сайту автосервісної компанії. Готові рішення дозволяють створити сайт автосервісу та запустити проект із мінімальними витратами ресурсів. У такому варіанті створення веб-сайту можна зробити сайт і запустити його за 5-10 днів.

Що створити: сайт-візитку, повнофункціональний сайт компанії чи інтернет-магазин вибирати заказчику. Можна розвивати сайт, додавати розділи, розширювати функціональність.

До розробки сайту автосервісу входить:

- розробка структури сайту з урахуванням перспективи подальшого просування;
- реєстрація домену;
- розробка дизайну сайту;
- верстка дизайну;
- інтеграція системи керування сайтом;
- інтеграція додаткових модулів та сервісів, що розширюють функціональність сайту;
- написання контенту, SEO-оптимізація матеріалів;

– заповнення сайту контентом.

4 АНАЛІЗ СИСТЕМ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІОННОЇ СИСТЕМИ АВТОСЕРВІС

Розробка веб-сайтів включатиме кілька ключових етапів. Першим з них є створення концепції майбутнього Інтернет-ресурсу. На основі цілей та цілей, сформульованих на цьому етапі, пропонується структура та створюється дизайн веб-сайтів. Ідея та веб-дизайн розроблені в тісній взаємодії з замовником, тому ми можемо врахувати всі побажання клієнта щодо зовнішності та дизайну сторінок.

4.1 Аналіз вибору СУБД

Системи управління базами даних (СУБД), яка використовуються в архітектурі клієнт-сервер належать до категорії серверів баз даних.

Сервер баз даних забезпечує наступні завдання:

- 1) реалізація програмного коду (логіка бізнесу) на сервері;
- 2) передача даних клієнтам, які на даний час підключені до сервера.

В розділі розглянуто та проведено порівняльний аналіз некомерційних СУБД, а саме: MySQL, PostgreSQL, MongoDB.

MySQL – це вільно розповсюджені СУБД, підтримка яких здійснюється корпорацією Oracle, яка отримала права на СУБД з поглинанням Sun Microsystems.

Postgresql – вільно розподілений об'єкт – революційна система управління базами даних. До недоліку цієї СУБД можна віднести відносно низьку продуктивність

MongoDB – це СУБД, орієнтована на документ, з вихідним кодом, що не потребує опису схеми таблиці бази даних.

В табл. 1 показані результати порівняння перелічених СУБД відповідно до основних особливостей:

- швидкість отримання даних («+» – швидке отримання даних з їх бази даних, «-» – низька швидкість зчитування даних із бази даних);
- швидкість запису даних («+» – дуже швидка швидкість запису даних, «-» – повільна швидкість додавання даних);
- транзакції («+» – наявність підтримки транзакцій, «-» – транзакції не підтримуються);
- підтримка реляційних баз даних («+» – база даних використовує реляційну модель зберігання даних, «-» – база даних використовує інші моделі для зберігання даних).

Таблиця 1 – Порівняльний аналіз СУБД

Характеристики	MySQL	PostgreSQL	MongoDB
Швидкість отримання даних	+	-	-
Швидкість запису даних	+	-	+
Транзакції	+	+	-
Підтримка реляційних баз даних	+	+	-

З порівняльної таблиці робимо висновок, що СУБД MySQL забезпечує найкращу продуктивність, тому в роботі буде використано ця СУБД для динамічного веб-сайту розробленої інформаційної системи.

На рис. 5 представлена фізична модель бази даних Автосервіс



Рисунок 5 – Фізична модель БД

У таблиці List_Of_Clients міститься інформація про всіх клієнтів, які вже відвідали послуг автосервісу. Поля, що містяться у таблиці: Id_Client (код клієнта), nvarchar (50), Surname (прізвище), nvarchar (50) Name (ім'я), Nvarchar (50), Patronimic (по-батькові), DateTime DateOfBirth (дата народження), Nvarchar (50) Phonenumber (номер телефону).

Для операцій з даними було створено три збережені процедури AddClient (додайте клієнта), updateclient (зміна інформації про клієнта), deleteclient (видалити інформацію про клієнта).

У таблиці List_Of_Clients є з'єднання "один на безліч" з таблицею списків через поля IDCLIENT_IDCLIENT. Ця таблиця містить інформацію

про автомобілі, що належать клієнту. Один клієнт може мати кілька автомобілів. Це пояснює з'єднання "один на багато".

Поля таблиці List_Of_Cars: int idCar (код автомобіля), int idClient (код клієнта), nvarchar(50) VIN, nvarchar(50) engineNumber (номер двигуна), nvarchar(50) bodyNumber (номер кузова), nvarchar(50) chassisNumber (номер шасі), , nvarchar(50), carModel (модель автомобіля), int idEngine (код типу двигуна), nvarchar(50), enginePower (потужність двигуна), int idTransmission (код типу коробки передач), int carMileage (пробіг автомобіля), int yearOfIssue (рік випуску автомобіля), nvarchar(MAX) comment (коментар).

Для операцій з даними створено три збережені процедури AddCar (додати автомобіль клієнту), UpdateCar (змінити інформацію про автомобіль клієнта), DeleteCar (видалити інформацію про автомобіль клієнта).

Таблиця List_Of_Cars має зв'язок «один-багатьом» з таблицею AllRenderedServices по полях idCar_idCar. Ця таблиця містить у собі інформацію про всі виконані роботи раніше у даного автомобіля. Має поля: int idRecord (код запису), int idCar (код автомобіля), int idService (код послуги), DateTime dateService (дата надання послуги).

Для операцій з даними були створені три процедури AddRenderedService (додати надану послугу автомобілю), UpdateRenderedService (змінити надану послугу автомобілю), DeleteRenderedService (видалити інформацію про надану послугу автомобілю).

Таблиця All_Rendered_Services має зв'язок із таблицею List_Of_Services за полями idService_idService. У таблиці описані всі види послуг, що надаються автосервісом. Поля таблиці ListOfServices: int idService (код послуги), int idGroupOfServices (код групи послуг), nvarchar(50) nameOfService (найменування послуги), real standardClock (нормо-годинник), real price (ціна).

Для операцій з даними створено три збережені процедури AddService (Додати вид послуги), UpdateService (змінити вид послуги), DeleteService (видалити вид послуги).

Таблиця List_Of_Services має зв'язок із таблицею Groups_Of_Services за полями idGroupOfServices_idGroupOfServices. Ця таблиця містить назви груп послуг, щоб надалі при перегляді видів послуг було зручно фільтрувати послуги з груп. Поля таблиці GroupsOfServices: int idGroupOfServices (код групи послуг), nvarchar(50) nameOfGroup (найменування групи послуг).

Для операцій з даними створено три збережені процедури AddGroupOfServices (додати групу послуг), UpdateGroupOfServices (змінити групу послуг), DeleteGroupOfServices (видалити групу послуг).

Таблиця List_Of_Cars має зв'язку з таблицями List_Of_Engines, що зберігає типи двигунів і List_Of_Transmissions зберігають типи коробок передач. Зв'язки є по полях idEngine_idEngine та idTransmission_idTransmission відповідно. Додаткові процедури додавання (AddEngine, AddTransmission), зміни (UpdateEngine, UpdateTransmission) та видалення (DeleteEngine, DeleteTransmission).

Таблиця List_Of_Clients має зв'язок «один-багатьом» з таблицею All_Solded_Parts по полях idClient_idClient. Таблиця містить інформацію про всіх проданих клієнту запчастини. Поля таблиці AllSoldedParts: int idRecord (код запису), int idClient (код клієнта), int idPart (код запчастини), int countParts (кількість у запчастин), DateTime dateSelling (дата продажу).

Для операцій з даними створено три процедури, що зберігаються AddSoldedPart (додати інформацію про продаж запчастини), UpdateSoldedPart (змінити інформацію про продаж запчастини) DeleteSoldedPart (видалити інформацію про продаж запчастини).

Таблиця All_Solded_Parts має зв'язок з таблицею List_Of_Parts по полях idPart_idPart. Таблиця містить найменування всіх запчастин, які використовує автосервіс. Поля таблиці ListOfParts: int idPart (код

запчастини), int idGroupOfParts (код групи запчастин), nvarchar(50) nameOfPart (найменування запчастин).

Для операцій з даними створено три збережені процедури AddPart (додати вид запчастини), UpdatePart (змінити вигляд запчастини) DeletePart (видалити вигляд) запчастини).

Таблиця List_Of_Parts має зв'язок із таблицею Groups_Of_Parts по полях idGroupOfParts_idGroupOfPart. Ця таблиця містить у собі інформацію про групи запчастин та має такі поля: int idGroupOfParts (код групи запчастин), nvarchar(50) nameOfGroup (найменування групи запчастин).

Таблиця List_Of_Parts має зв'язок із таблицею Coming_Of_Parts по полях idPart_idPart, яка містить інформацію про прихід запчастин до складу. Поля таблиці ComingOfParts: int idRecord (код запису), int idPart (код запчастини), int countOfPart (кількість запчастин у приході), DateTime dateOfArrival (дата приходу запчастин), real purchasePrice (закупівельна ціна).

Таблиця List_Of_Parts має зв'язок із таблицею List_Of_Orders за полями idPart_idPart. Ця таблиця містить інформацію про замовлення запчастин, дату оформлення, дату приходу та кількість. Поля таблиці List_Of_Orders: int idRecord (код запису), int idPart (код запчастини), int countOfPart (кількість виду запчастини), DateTime dateOfIssue (дата оформлення замовлення), DateTime dateOfCompletion (дата приходу).

4.2 Вибір технології програмування автоматизованої інформаційної системи

При виборі засобів розробки програмного забезпечення ІС необхідно взяти до уваги наступні характеристики (табл. 2):

- підтримка технології швидкої розробки програм RAD (Rapid Application Development);
- підтримка сучасних Web-технологій;
- знання розробником даної технології програмування;

– низькі витрати на розробку якісних Web-додатків.

На даний момент для розробки Web-додатку широко застосовуються такі технології програмування, як Java, ASP.NET і PHP

- об'єктно-орієнтований підхід до розробки ІС;
- вільно розповсюджені середовища розробки;
- підтримка СУБД MySQL;
- підтримка архітектури «клієнт-сервер».

Таблиця 2 – Порівняльний аналіз технологій Web-програмування

Характеристики	ASP.NET	Java	PHP
Об'єктно-орієнтований підхід до розробки ІС	+	+	+
Вільно розповсюджені середовища розробки	-	-	+
Підтримка СУБД MySQL	-	+	+
Підтримка архітектури «клієнт-сервер»	+	+	+

Таким чином, на підставі результатів аналізу для реалізації ІС було обрано мову PHP та СУБД MySQL

5 ОПИС ПРОГРАМНОГО ДОДАТКУ

Відразу після входу на сайт компанії клієнт має можливість ввести заявку на ремонт автомобіля без попередньої реєстрації (рис. 6).

Можлива також реєстрація клієнта зі створенням особистого кабінету.

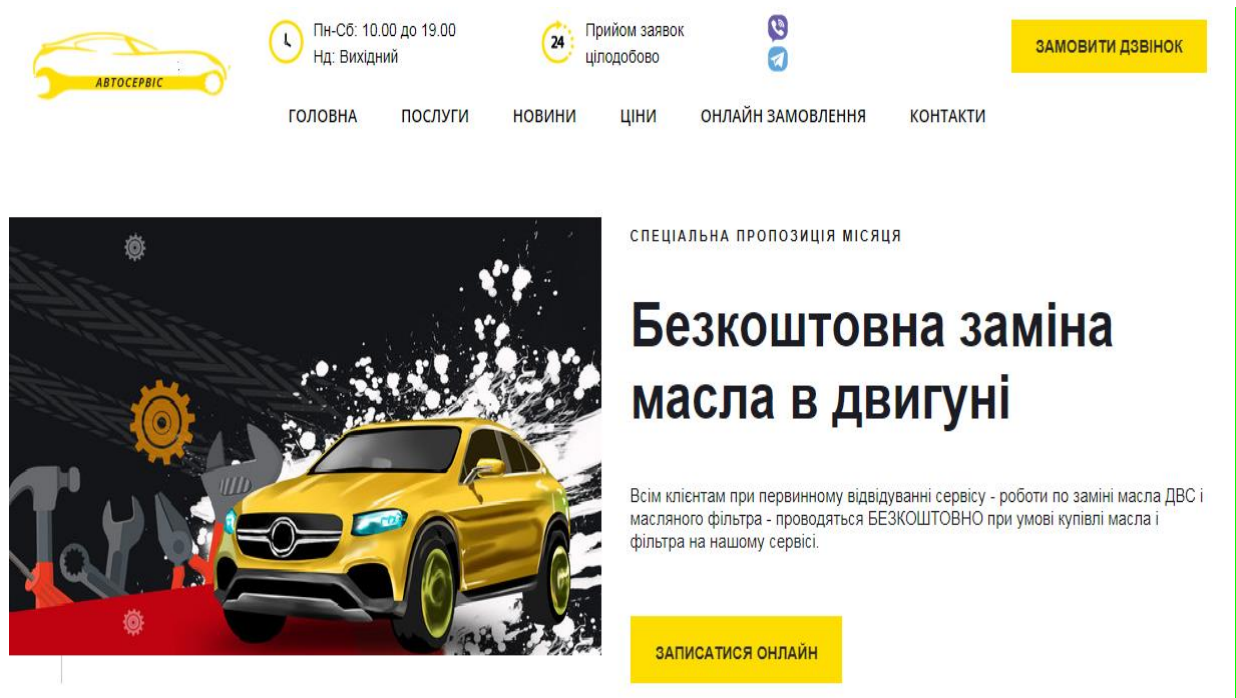


Рисунок 6 – Головна сторінка сайту

Структура сайту Автосервісу налічує стандартний набір розділів:

- "Головна" – ця сторінка є головною точкою введення сайту, вона повинна відображати основну інформацію про профіль підприємства;
- "Новини" – розділ, що містить інформацію про діяльність автомобіля;
- "Послуги" – розділ, що містить інформацію про специфіку послуг, їх вартість та зміни в них;
- "Зворотній зв'язок" – розділ, що містить можливість консультування з будь-якого питання з фахівцем;

- "Контакти" – розділ, що містить всю контактну інформацію про автомобільну службу.

Основними функціональними завданнями сайту є доступ до інформації:

- Про сфери активності СТО;
- Про режим роботи;
- про вартість послуг;
- Про акції.

Враховуючи типові елементи та вимоги замовника, була визначена така структура макета сайту (рис. 1.6):

Веб -сторінки повинні містити типові елементи навігації та інформації:

- шапка сайту;
- головне меню;
- бічне меню;
- основне поле вмісту;
- Субстрат.

Шапка сайту включає вміст логотипу та назву компанії, зображення (колаж), а також контактну інформацію.

Логотип - це посилання на головну сторінку сайту.

Головне меню повинно розташовуватися у верхній частині вікна (під ковпачком) і містити посилання на всі розділи.

Навігація на сайті говорить про те, що з кожного розділу Сайту ви можете потрапити в будь -який інший розділ (що допоможе користувачам не плутатись у навігації).

Патеральне меню має бути розташоване з лівої або правої частини вікна (є необов'язковою частиною сайту).

Основне поле вмісту повинно бути розташоване в центрі сторінки. У цьому полі він повинен відображати основний вміст вибраного розділу.

Дизайн стилю матеріалів та їх елементи (посилання, заголовки, основний текст, зображення, форми, таблиці тощо)

Будьте на всіх сторінках веб-сайту.

Замовник надає всі текстові та графічні матеріали, необхідні для підрядника у зручній формі, для подальшої обробки, а також коментарів щодо їх вмісту, обсягу, дизайну та розміщення.

Підкладка повинна мати інформацію про власників авторських прав веб-ресурсу.

Розробка макета та впровадження програми Сайту шляхом створення сайту, враховуючи його призначення, зручність наступного супровід та розвиток. Сайт створений для відвідувачів, і тому повинен бути зручним для них художнім способом.

Склад та дизайн елементів - це те, що в основному визначає зовнішній вигляд ділянки. Його можна намалювати ще до розробки технічної частини. Це робиться за допомогою графічного редактора, я використовував Adobe Photoshop. Сторінки сайту створюються за допомогою HTML, CSS та частково сценаріїв (сценаріїв), написаних на JavaScript.

Щоб створити головну сторінку, я використовував у The Text Text Text Editor Notepad ++ (безкоштовний текстовий – Source – Source Editor для Windows з підсвічуванням

Синтаксис великої кількості мов програмування та маркування) та веб-браузера.

Код головної сторінки зберігається у файлі home.html.

Маркування та дизайн основних елементів (стилів) описані у файлі Style.css (рис. 7).

```

1  * {margin:0;padding:0}
2  html, body {height:100%}
3  body {background:#0184d4;font-family:Georgia; text-align="justify"; sans-serif;font-size:24px; line-height:1.125em;color:#565656}
4  img {border:0;vertical-align:top;text-align:left}
5  object {vertical-align:top;outline:none}
6  ul, ol {list-style:none}
7  .left {float:left}
8  .right {float:right}
9  .clear {clear:both}
10 .col-1, .col-2, .col-3 {float:left}
11 .alignright {text-align:right}
12 .aligncenter {text-align:center}
13 .wrapper {width:100%;overflow:hidden}
14 .container {width:100%}
15 /* GLOBAL */
16 #main {width:1300px;margin:0 auto;background:url(images/main-bg2.png) no-repeat left top;font-size:.85em;position:relative}
17 #header {height:360px}
18 #content {padding:0 20px 38px 30px}
19 #content .aside {float:left;width:100px;margin-right:40px}
20 #content .mainContent {float:left;width:635px;padding-top:110px}
21 #footer {height:100px}

```

Рисунок 7 – Фрагмент листинга файла style.css

Меню, згідно з розробленою макетами структури сайту, містить такі розділи:

- Дім;
- Новини;
- Послуги;
- Замовити дзвінок;
- Контакти.

Переходячи до будь -якого розділу, користувач вводить сторінку з відповідною інформацією про ім'я. Клацнувши на точку меню "Домашня" або кнопку, розташованій з лівої сторони "ковпачка" сайту, користувач повертається на головну сторінку сайту (рис. 8).

Натиснувши кнопку "Про нас" меню, користувач вводить сторінку, на якій розміщується інформація про послугу автомобіля номер один.

Переходячи до розділу «Послуги», користувач вводить сторінку з відповідною інформацією (рис. 9).

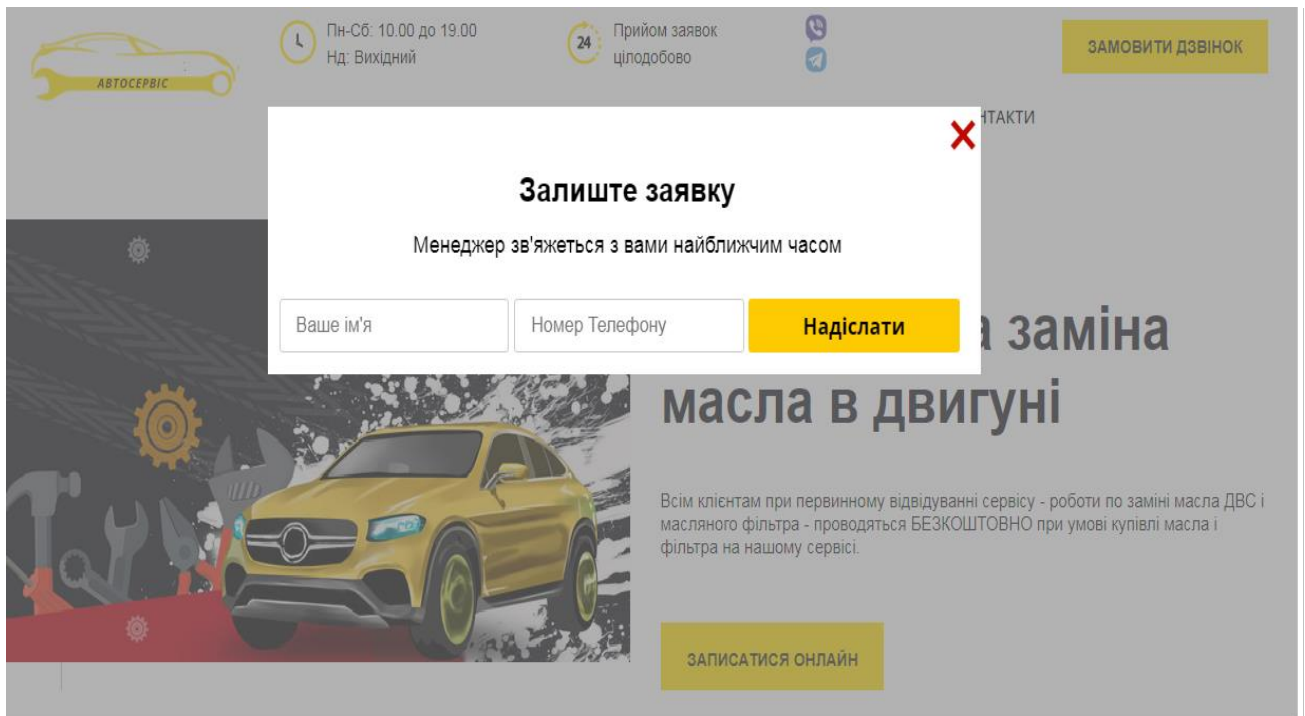


Рисунок 8 – Заявка

Кожен тип послуги містить власну сторінку інформації. Наприклад, врахуйте вкладку Діагностика автомобіля. Коли ви натискаєте назву послуги, користувач переходить на відповідну сторінку.

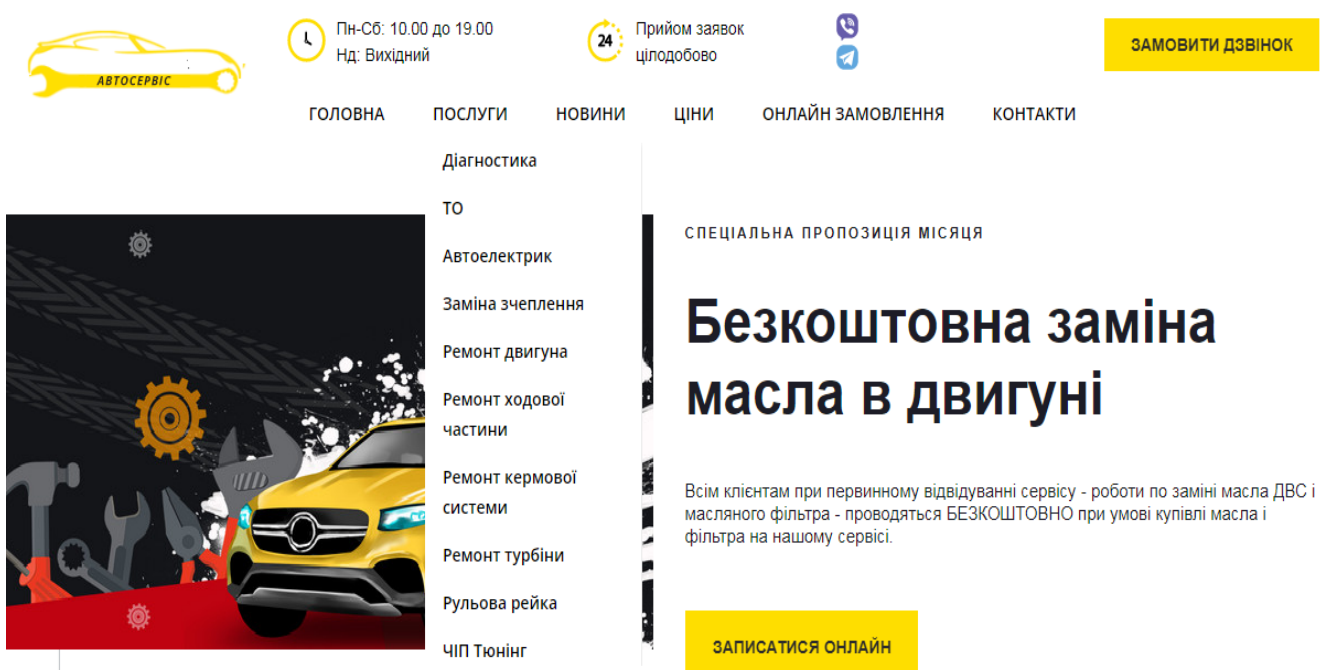


Рисунок 9– Послуги

На цій сторінці використовується елемент JavaScript –"Акардальне меню". При натисканні на потрібний елемент меню відображається прихована частина інформації (рис. 10).

Сторінки "шини та диски", "технічне обслуговування", "комплекс RРMON" та "Авто-lome" мають власні сторінки HTML, але не наповнені суттєво (основою для подальшої роботи).

Розділ "Замовити дзвінок" містить форму для надсилання повідомлення. Користувач вказує свої дані та свою привабливість та надсилає повідомлення при натисканні кнопки "Надіслати повідомлення".

Розділ "Новини" містить форми детальною інформацією з ремонту різних моделей авто. Користувач може ознайомитись детально з даною інформацією по різним моделям.

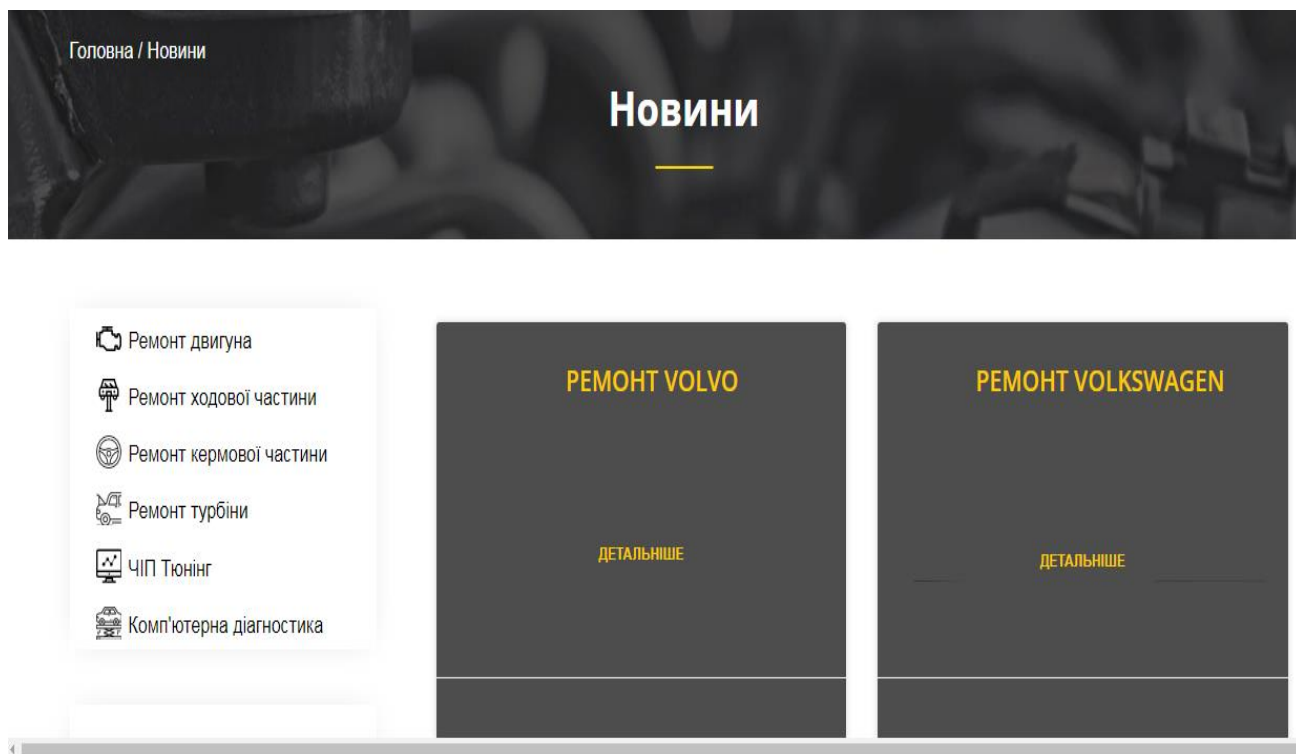


Рисунок 10 – Новини

У розділі "Ціни" містить форму з описом різних діагностичних послуг та цінової політики. Користувач може переглядати інформацію з послуг які надає автосервіс та ціни (рис. 11).

Найменування		Ціна, грн. *
1. Діагностика автомобіля		
1.1.	Комп'ютерна діагностика	200
1.2.	Діагностика ходової частини	160
1.3.	Діагностика гальмівної системи	160
1.4.	Діагностика стану робочих і технічних рідин	60
1.5.	Перевірка компресії в циліндрах двигуна	Від 400
2. Технічне обслуговування		
2.1.	Заміна мастила в двигуні	160
2.2.	Заміна мастила з промивкою двигуна	400
2.3.	Заміна мастила гідролідкювача керма	200
2.4.	Заміна мастила в мосту	200
2.5.	Заміна мастила в механічній КПП	300
2.7.	Заміна ременя ГРМ	1800-2700

Рисунок 11 – Ціни

Ця сторінка також передбачає прохідну карту до служби автомобіля.

Розділ "Контакти" містить контактну інформацію, а також посилання на соціальні мережі автомобільної служби.

Посилання на соціальні мережі організуються за допомогою використання гіперпосилання на відповідні піктограми, які також містять «поради» (рис. 12).

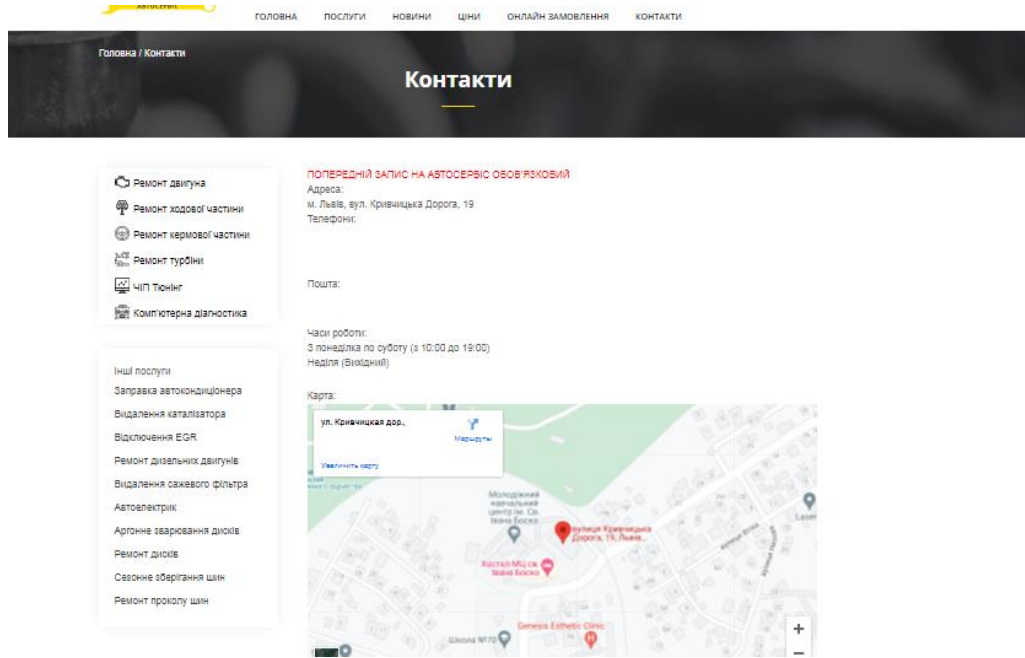


Рисунок – Контакти

Поле для введення тексту повідомлення має необмежену кількість знаків, тобто. Звернення може бути довгим та істотним (рис. 13)

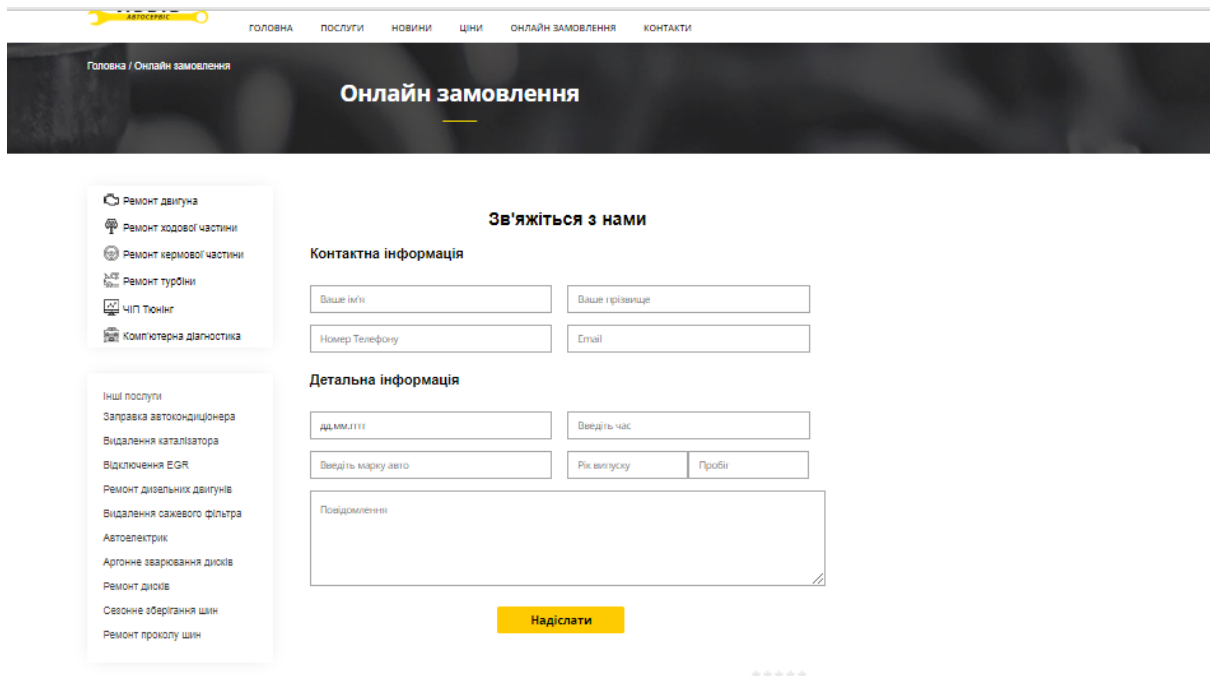


Рисунок 13 – Онлайн замовлення

ВИСНОВКИ

У роботі була спроектована та реалізована інформаційна система "Автосервіс", яка забезпечує:

- введення, редагування та перегляд даних;
- відповіді на запити користувача;
- пошук необхідної інформації;
- формування звітів.

Освоєно навички проектування баз даних методом ER-діаграм.

Реалізація системи проводилася за допомогою JavaScript, елементів системи CRM, СУБД.

Основна увага в цій роботі була приділена зручності роботи користувача з програмою та побудові дружнього інтерфейсу.

Результатом тестування продемонстровано, що дана програма працює чітко у всіх передбачуваних ситуаціях.

Додаток клієнт-сервер бере участь у зборі інформації про клієнта, веде свою історію візитів, виконану технічну роботу та використані деталі.

Орієнтовані завдання для підприємства для створення сучасного веб - сайту.

У процесі виконання роботи було вирішено наступні завдання:

- розглянуто сучасні технології для розробки веб - сайтів;
- проаналізовані веб-ресурси в сучасному авто-бізнесі;
- розробка макета сайту для послуги автомобілів;

На початковому етапі дослідження було встановлено, що автосервіс з різних послуг автомобілів є конкурентним підприємством, але його слабкою стороною є відсутність інформаційної системи, а саме Веб-сайт. Оскільки сайт дозволяє транслювати діяльність автомобільної послуги для більш широкої громадськості, повідомити про всі типи послуг та поточні акції,

своєчасно оновлюючи відповідну інформацію, було вирішено створити веб – ресурс.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 Гаевский А.Ю. Создание Web-страниц и Web-сайтов: HTML и JavaScript: учеб. пособие. Одеса. Триумф, 2016. 454 с.
- 2 Створення сайту і розробка web-дизайну: сайт. URL: dogma.ru/sozdanie-sajta_2.html (дата звернення 13.02.2022).
- 3 Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. М.: Эксмо, 2018. 480 с.
- 4 Бушманова В.Н. Использование on-line редакторов кода для обучения web-программированию. NovaInfo.Ru. 2017. Т.1. № 61.
- 5 Снелл. Абсолютно ясно о создании Web-страниц и Web-сайтов. М.: Триумф, 2015. 224 с.
- 6 Барисов Р.И. Постройте профессиональный сайт сами. Одеса, 2017. 304 с.
- 7 Дарнелл Р. HTML 4: энциклопедия пользователя (пер. с англ.). Киев: Диасофт, 2016. 688 с.
- 8 Дунаев В.В. HTML, скрипты и стили. БХВ, 2017. 208 с.
- 9 Лазаро Исси Коэн. Полный справочник по HTML, CSS и JavaScript. ЭКОМ Паблишерз, 2016. 233 с.
- 10 Прохоренок Н. М. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера. 2016. 912 с.
- 11 Чебыкин Р. Самоучитель HTML и CSS. Современные технологии. Одеса. Огни, 2016. 624 с.
- 12 Методическое пособие. Основы HTML, каскадные таблицы стилей CSS, создание сценариев на web-странице с помощью JavaScript.
- 13 КАК СОЗДАТЬ СОВРЕМЕННЫЕ ВЕБ-РЕСУРСЫ: сайт. URL: https://abaris.ru/articles/site-creation/site-creation-art_215.html (дата створення 13.03.2022).

14 Гарнаєв А. WEB-программирование на Java и JavaScript. Харьков, 2015. 524 с.

15 Bootstrap: сайт. URL: <https://getbootstrap.com/> (дата звернення 04.02.2020).

16 jQuery : сайт. URL: <https://jquery.com/> (дата звернення 28.03.2022).

17 Найкращі автосервіси URL:<https://info-parts.com.ua/biznes/po/item/4251-oglyad-crm-sistem-dlya-avtoservisu.html> (дата звернення 24.03.2022).