

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

Кафедра цифрової економіки та бізнес-аналітики

КУРСОВА РОБОТА

з дисципліни «Проектування та адміністрування БД і СД»

на тему:

«Інформаційна система для роботи з клієнтами автосалону.»

спеціальність: 051 «Економіка»
(код та найменування спеціальності)

спеціалізація: «Інформаційні технології в бізнесі»
(найменування спеціалізації)

освітній ступінь: бакалавр
(бакалавр/магістр)

Науковий керівник:

К.ф.-м.н., доц. Ярема О.Р.
(науковий ступінь, посада, прізвище, ініціали)

“ ” 2020 р.
(підпис)

Виконавець:

студент групи УФЕ-31с
Співак М. Д.
(прізвище, ініціали)

“ ” 2020 р.
(підпис)

Загальна кількість балів _____
(підпис, ППІ членів комісії)

ЛЬВІВ 2020

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ	5
1.1 Постановка завдання	5
1.2 Розробка моделі варіантів використання веб-сайту	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ	7
2.1 Опис моделі даних.....	7
2.2 Нормалізація реляційних відношень	8
2.3 Визначення типів даних	10
2.4 Обмеження цілісності даних.....	11
2.5 Реалізація SQL-скрипту	13
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ	14
3.1 Структура веб-сайту.....	14
3.2 Макети сторінок веб-сайту	15
3.3 Програмування серверної частини	15
3.4 Програмування клієнтської частини	17
3.5 Розміщення веб-сайту на локальному сервері	19
ВИСНОВКИ	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	24
ДОДАТКИ	25

ВСТУП

Актуальність теми курсової роботи. Розробка веб-додатків, на даний момент, одна з найбільш актуальних сфер ІТ. Будь-який бізнес потребує платформу для комунікації та різного роду взаємодії з клієнтом. Сайт один з найбільш популярних та зручних способів такої взаємодії.

Створення бази даних, дає змогу структурувати бізнес та автоматизувати бізнес-процеси. З іншого боку, клієнт отримує масу переваг, починаючи від можливості аналізу бренду, через сайт, та закінчуючи можливістю онлайн-запису. Сайт і база даних, може бути суттєвою конкурентною перевагою в більшості сфер бізнесу.

Метою даної роботи є реалізація web-орієнтованої інформаційної системи автомобільного сервісу.

Виходячи з цього можна сформулювати такі завдання:

- Створення інтерфейсу сайту.
- Розробка бази даних.
- Забезпечення взаємодії сайту та бази даних, шляхом розробки серверної частини.

Об'єктом курсової роботи, є особливості проектування веб-додатків на прикладі сервісу по ремонту авто.

Предметом даної роботи є дослідження та використання засобів розробки PHP, MySQL, HTML та CSS.

Практичне значення курсової роботи. Всі матеріали, створені в ході написання курсової роботи, можна використовувати в практичних цілях, як то навчальних або прикладних. Веб-додаток, створений у результаті курсової

роботи, може бути використаний автосервісом для створення клієнтської бази даних та автоматизації свого відділу продажів.

У ході роботи було використано локальний сервер MAMP, програмне середовище phpMyAdmin, спеціалізований текстовий редактор Sublime Text, мови web-розробки PHP, HTML5, CSS, система керування реляційними базами даних MySQL, мова програмування для взаємодії з базами даних SQL.

Структура роботи. Курсова робота складається з трьох розділів, а також вступу, висновку та додатків. Перший розділ, описує бізнес складову роботи. Другий розділ, показує процес побудови навчальної бази даних. Третій розділ, демонструє підходи по реалізації поставлених цілей.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

Бізнес потреба. Інформаційна система, у вигляді сайту, буде підвищувати ефективність процесу замовлення послуг. Основною ціллю даної інформаційної системи, буде створення бази даних клієнтів. В подальшому цю інформацію можна використовувати для створення промоакцій та будь-якої іншої комунікації з клієнтом. Також, форму на сайті можна використовувати як систему онлайн-замовлень. Клієнт може описати свою проблему, після чого представник сервісу зателефонує йому, проведе консультацію та повідомить дату, в яку може бути здійснений ремонт його авто.

Технічне завдання. Виходячи з даної бізнес потреби, нам потрібно створити:

- Зручний та простий інтерфейс сайту
- Базу даних, в яку ми будемо вносити інформацію про клієнтів.
- Форму зворотнього зв'язку, в яку буде вноситись інформація клієнтом.

У процесі створення web-системи використано наступні технології:

- MySQL — система керування базами даних, яка має набір функціональних можливостей, для зберігання, редагування та отримання даних.
- PHP — це мова програмування, яка дозволяє генерувати HTML-сторінки на стороні сервера.
- HTML — стандартна мова розмітки веб-сторінок в Інтернеті.

- PHPMyAdmin – сукупність скриптів написаних на PHP, що забезпечують роботу з базами даних MySQL, через вебінтерфейс.
- Розгортання програмного продукту відбувається за допомогою локального сервера MAMP.

1.2 Розробка моделі варіантів використання веб-сайту

Головним користувачем системи є клієнт. Основною його метою виступає здійснення замовлення на сайті. Клієнт заповнює форму, в якій вводить своє ім'я, телефон та описує проблему його автомобіля. Успішне заповнення форми сповістить клієнта про те, що з ним зконтактуються. Після замовлення, працівник буде бачити поповнення бази даних клієнтом (в PHPMyAdmin) і зможе використати цю інформацію у корисних для нього цілях.

Отже, метою інформаційної системи є, отримання заявки від клієнта, через форму зворотнього зв'язку, з додаванням його у базу даних.

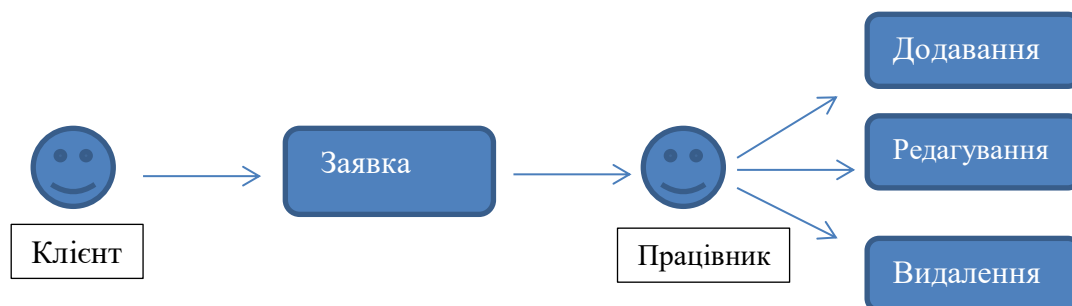


Рисунок 1.1 – Діаграма варіантів використання сайту

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Опис моделі даних

Основою бази даних є модель даних — фіксована система понять і правил для представлення даних структури, стану і динаміки проблемної області в базі даних.

В phpMyAdmin створено базу даних «test-db1», яка міститиме таблицю з даними про користувачів, які залишили заявку на сайті. При створенні бази даних необхідно обрати кодування, яке дозволяє зберігати в таблицях даних символи кирилиці - utf8_general_ci. При створенні бази даних, було створено таблиці з назвою «users», «employee», «services».

Поля, що входять до таблиці бази даних:

Для таблиці «users»:

- id;
- login;
- phonenumber;
- problem.

Для таблиці «employee»:

- id_e;
- name;
- id_s;

Для таблиці «service»:

- id_s;
- service;

2.2 Нормалізація реляційних відношень

Нормалізація відношень- покроковий процес розділення (декомпозиції) початкових відношень БД на простіші. Кроки цього процесу переводять схему відношення БД в послідовні нормальні форми. Кожна наступна форма володіє кращими властивостями ніж попередня.

У теорії реляційних баз даних прийнято виділяти таку послідовність нормальних форм:

- перша нормальна форма, 1NF (First Normal Form);
- друга нормальна форма, 2NF (Second Normal Form);
- третя нормальна форма, 3NF (Third Normal Form);
- нормальна форма Бойса – Кодда, BCNF (Brice – Codd Normal Form);
- четверта нормальна форма, 4NF (Fourth Normal Form);
- п'ята нормальна форма, 5NF (Fifth Normal Form).

Кожній нормальній формі відповідає певний набір обмежень. При переведенні структури відношення у форми вищого порядку досягають видалення з таблиць надмірної описової інформації.

Процес нормалізації заснований на понятті функціональної залежності атрибутів.

Атрибут В функціонально залежить від атрибуту А (позначають $A \rightarrow B$), якщо у будь-який момент часу кожному значенню атрибуту А відповідає не більше одного значення атрибуту В.

Якщо описовий атрибут залежить від усього складеного ключа і не залежить від його частин, то мова йде про повну функціональну залежність атрибуту від складеного ключа.

Якщо атрибут А залежить від атрибуту В, а В залежить від атрибуту С, але зворотна залежність відсутня, то кажуть, що атрибут А залежить від С транзитивно. Наприклад, відношення має два описові атрибути, причому один залежить від ключа, а другий – від першого описового атрибуту. Перша нормальна форма. Відношення представлене в першій нормальній формі (1НФ) тоді й тільки тоді, коли всі його атрибути містять тільки неподільні (атомарні) значення й у ньому відсутні групи атрибутів з однаковими за змістом значеннями, які повторюються у межах одного кортежу. Неподільність значення атрибуту говорить про те, що його не можна розділити на більш дрібні частини.

Перша нормальна форма. Відношення відповідає 1NF тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні (неподільні) значення атрибутів і не містяться групи, що повторюються.

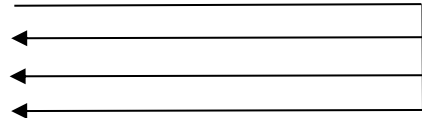
Друга нормальна форма. Відношення знаходиться в 2NF, якщо виконуються обмеження 1NF і кожен описовий атрибут функціонально повно залежить від первинного ключа (у тому числі і складеного).

Третя нормальна форма. Відношення знаходиться у 3NF, якщо виконуються обмеження 2NF і всі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут не транзитивно залежить від ключа.

Функціональні залежності між атрибутами в таблиці «users», котра містить усю інформацію про користувача зображено у Таблиці 2.1. У ній зображено, що ключовим полем є поле ідентифікаційний код користувача, тобто поле «id», із значенням «Primary». Поля «login», «phonenumber», «problem», залежні від поля «id».

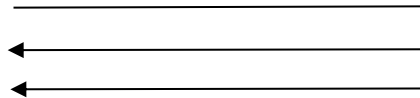
Таблиця 2.1

Функціональні залежності між атрибутами сутності «users»

Найменування атрибутів	Функціональні залежності
id login phonenumber problem	

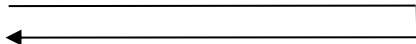
Таблиця 2.2

Функціональні залежності між атрибутами сутності «employee»

Найменування атрибутів	Функціональні залежності
Id_e name id_s	

Таблиця 2.3

Функціональні залежності між атрибутами сутності «service»

Найменування атрибутів	Функціональні залежності
Id_s service	

2.3 Визначення типів даних

В своїй структурі MySQL містить наступні складові:

- бази даних
- таблиці
- наповнення таблиць

Перш за все, ми створюємо базу даних «test-db1», за допомогою програмного середовища phpMyAdmin. Наступним кроком роботи буде створення таблиці «users» з наступними типами даних:

- INT – тип даних, що представляє ціле число, діапазон від -2 147 483 648 до 2 147 483 647;

- VARCHAR – набір символьних даних, може зберігати не більше 255 символів.

Таблиця 2.2

Типи даних таблиці «users»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	id (PK)	INT	10
2.	login	VARCHAR	10
3.	phonenumber	VARCHAR	10
4.	problem	VARCHAR	200

Типи даних таблиці «employee»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	Id_e(PK)	INT	10
2.	name	VARCHAR	255
3.	Id_s	INT	10

Типи даних таблиці «services»

№	Ім'я атрибуту	Тип даних	Розмір поля
1.	Id_s (PK)	INT	10
2.	login	VARCHAR	10

2.4 Обмеження цілісності даних

Правила цілісності бази даних – це правила, які забезпечують уникнення введення некоректних даних та дають можливість пов'язування декількох таблиць.

У роботі було використано наступні обмеження:

- not null;
- primary key;

- `auto_increment;`

Коротко про кожне з обмежень:

За замовчуванням стовпець може містити значення NULL. Обмеження NOT NULL призводить до того, що стовпець НЕ приймає значення NULL. Це призводить до того, що поле завжди містить значення, а це означає, що ви не можете вставити новий запис або оновити запис без додавання значення в це поле.

Обмеження PRIMARY KEY однозначно ідентифікує кожний запис у таблиці бази даних. Первинні ключі повинні містити UNIQUE значення і не можуть містити значення NULL.

У таблиці може бути тільки один первинний ключ, який може складатися з одного або декількох полів.

Auto-increment дозволяє створювати унікальний номер автоматично, коли новий запис вставляється в таблицю. Часто це поле основного ключа, яке ми хотіли б створити автоматично кожного разу, коли буде вставлена новий запис.

Зовнішні ключі (FK): `id_s` в таблиці `employee`.

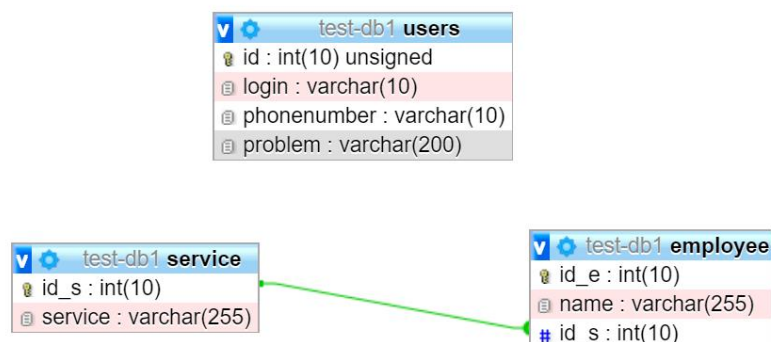


Рисунок 2.1 – Діаграма сутностей та зв'язків із зовнішніми ключами

2.5 Реалізація SQL-скрипту

Web-сайт реалізовує наступні запити до бази даних:

1. Запити типу SELECT:

`SELECT * FROM `users`` – запит виводить всі дані, з таблиці «users»

`SELECT * FROM `users` ORDER BY id ASC` – запит виконує виведення усіх послуг, впорядкованих по полю “id”;

2. Запити типу INSERT:

`INSERT INTO `users` (`login`, `phonenumner`, `problem`) VALUES ($login, $phonenumner, $problem)` – запит, що додає нового клієнта до бази даних.

3. Запити типу UPDATE:

`UPDATE `users` SET `phonenumner` = '1234567891' WHERE `users`.`id` = 2;` – запит змінює номер телефону клієнта з id=2.

4. Запити типу DELETE:

`DELETE FROM `users` WHERE `id` = 2` – запит, який видаляє дані з таблиці users, з id=2 .

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Інструменти для розробки веб-сайту

Для розробки сайту, були використані наступні технології:

- HTML - стандартизована мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузером, отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою.
- CSS - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.
- PHP - скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.
- MySQL - вільна реляційна система управління базами даних
- JavaScript – це мова програмування, що дозволяє зробити Web -сторінку інтерактивною, тобто такою що реагує на дії користувача.
- Для адміністрування СУБД було використане програмне середовище phpMyAdmin – це веб-додаток з відкритим кодом, написаний на мові PHP і представляє собою веб-інтерфейс для адміністрування СУБД MySQL.

Розгортання програмного продукту буде відбуватися за допомогою локального сервера MAMP, який являє собою найпопулярнішим середовищем розробки PHP. MAMP - це стек рішень, що складається з безкоштовного та відкритого коду та власного комерційного програмного забезпечення, що

використовується разом для розробки та запуску динамічних веб-сайтів на комп'ютерах.

3.2 Структура веб-сайту

Створений веб-сайт, складається з двох блоків та форми зворотнього зв'язку. Зверху сайту реалізовано меню, яке переносить відвідувача сайту на кожен з блоків. В кожному з блоків, реалізована кнопка, при натиску на яку, користувач переходить на наступний блок.

Форма зворотнього зв'язку містить три поля, які потрібно заповнити. Знизу форми розташовані дві кнопки, одна з яких підтверджує надсилання форми, інша ж очищує поля форми.

Після підтвердження надсилання форми, користувач бачить інформацію про заповнення форми. При успішній відправці форми, користувачу виведе повідомлення "Дякую за заявку. Очікуйте нашого дзвінка.". В разі некоректного введення інформації, користувач побачить повідомлення про одну з трьох проблем:

- Недопустима кількість букв в імені
- Неправильна кількість цифр або неправильний формат.
- Недопустимий опис проблеми.

Знизу сторінки, реалізовані кнопки, при натисканні на які, користувач переходить у соціальні мережі сервісу.

3.3 Макет сторінки веб-додатку

Блок перший, основної сторінки сайту. Кнопка переносить користувача на блок 2, «Чому ми?».

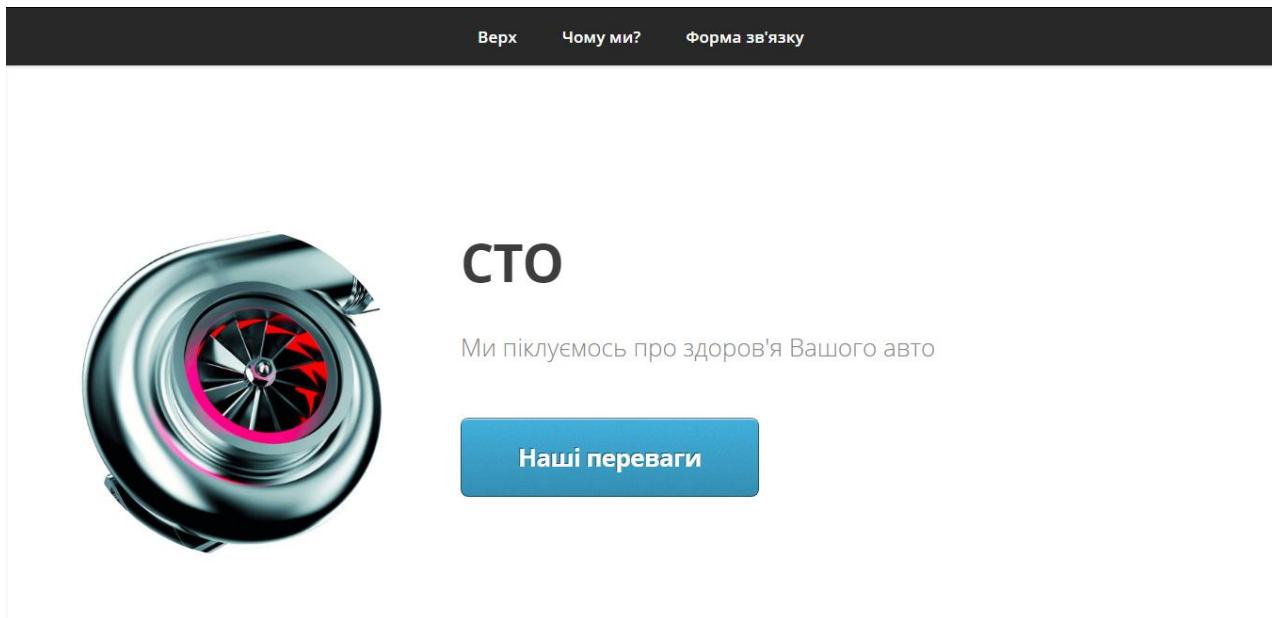


Рисунок 3.1 – Блок перший, основної сторінки сайту

Блок другий, короткий опис переваг, які отримує користувач при співпраці з автосервісом. Кнопка переносить користувача в блок з формою зворотнього зв'язку.

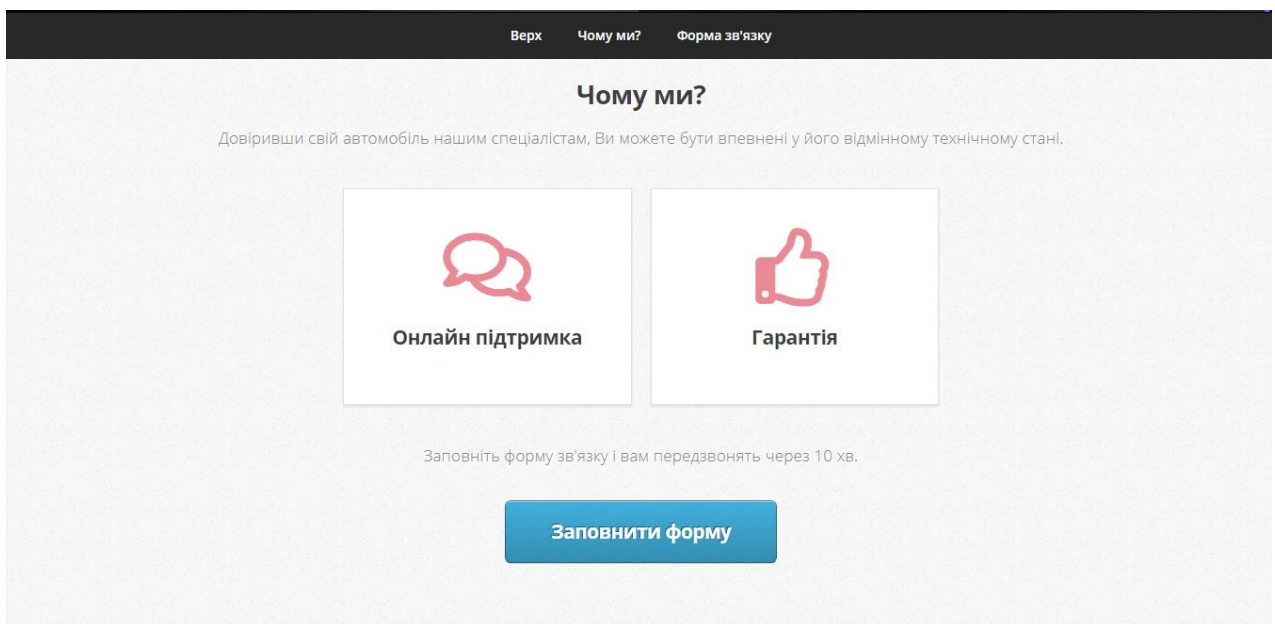


Рисунок 3.2 – Блок другий, основної сторінки сайту

Форма зворотнього зв'язку. Після введення неправильного введення даних, користувач може натиснути на кнопку очистити форму та всі поля стануть пустими. Якщо ж дані введені правильно, користувач нажимає підтвердити, тим самим надсилаючи дані в базу.

Рисунок 3.3 – Блок третій, основної сторінки сайту

В підвалі сайту реалізовано кнопки, натиснувши на які, можна перейти соціальні мережі автора.

Рисунок 3.4 – Блок четвертий, основної сторінки сайту

3.4 Програмування серверної частини

Серверна частина сайту написана скрипковою мовою програмування PHP, в спеціалізованому текстовому редакторі Notepad++. Зовнішній вигляд сайту, видимий для користувача, написаний мовами HTML та CSS.

Для забезпечення повноцінної роботи сайту створено наступні серверні скрипти:

1. Для обробки введених користувачем даних та правильного їх форматування:

```
<?php

$login = filter_var(trim($_POST['login']),
    FILTER_SANITIZE_STRING);
$phonenumber = filter_var(trim($_POST['phonenumber']),
    FILTER_SANITIZE_STRING);
$problem = filter_var(trim($_POST['problem']),
    FILTER_SANITIZE_STRING);

if (mb_strlen($login) < 2 || mb_strlen($login) > 15 ) {
    echo "Недопустима кількість букв в імені";
    exit();

} else if (mb_strlen($phonenumber) != 10 || is_numeric($phonenumber) !=
$phonenumber ) {
    echo "Неправильна кількість цифр або неправильний формат.";
    exit();

} else if (mb_strlen($problem) < 5 || mb_strlen($problem) > 200 ) {
    echo "Недопустимий опис проблеми.";
    exit();

} else {
    echo "Дякую за заявку. Очікуйте нашого дзвінка.";

}

?>
```

2. Для створення зв'язку між сайтом і базою даних.

```
<?php
```

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "root";
```

```
$dbname = "test-db1";
```

```
$conn = new mysqli ($servername, $username, $password, $dbname);
```

```
$conn -> query("INSERT INTO `users` (`login`, `phonenumner`, `problem`)
VALUES ('$login', '$phonenumner', '$problem')");
```

```
$conn-> close();
```

```
?>
```

3.5 Програмування клієнтської частини

При розробці клієнтської частини сайту, використано стандарти розмітки сторінок HTML, стилі CSS та JavaScript.

Вид кожного з блоків сторінки наведено в додатках. Опис наведено нижче:

1. Блок 1. Верх сторінки, створений для загального розуміння клієнтом чим займається компанія. Додаток А.
2. Блок 2. Цей елемент відповідає на питання, чому варто обрати, саме цей сервіс, тобто розкрити основні його конкурентні переваги порівняно з іншими автосервісами. Додаток Б.
3. Блок 3. Форма відправки повідомлень була створена, для автоматизації відділу продажів. Додаток В.
4. Блок 4. У підвалі сайту можна побачити кнопки з посиланнями на соціальні мережі, в даному випадку посилання переходять на автора сайту, проте у випадку автосервісу це можуть бути корпоративні сторінки в різноманітних соціальних мережах. Додаток Д.

3.5 Розміщення веб-сайту на локальному сервері

Розгортання php-сценаріїв здійснюється на стороні веб-сервера, щоб реалізувати це на своєму комп'ютері потрібно встановити локальний сервер.

За допомогою спеціального програмного забезпечення ви зможете розгорнути ресурс на своєму комп'ютері і працювати з ним. Потім сайт можна перенести на сервер хостинг-провайдера.

Йдеться про програму або набір програм, які імітують роботу реальних серверів хостинг-провайдерів. Локальний сервер забезпечує доступ до ресурсу через браузер. Набір програм і скриптів перетворює інформацію баз даних типу MySQL і мов програмування типу Perl і PHP в HTML- і CSS-код, який «розуміють» веб-переглядачі.

За допомогою локального сервера можна готувати сайт до публічного доступу. На етапі розробки ресурс бачите тільки ви і ваші колеги. Це позбавляє живих відвідувачів і пошукових роботів взаємодії з незаповненим і недопрацьованим сайтом. А ви можете перевірити працездатність ресурсу, освоїти адміністративну консоль і виконати інші необхідні дії в безпечних умовах.

У число найбільш популярних програмних рішень входять Open Server, Denwer, XAMPP, MAMP та інші.

Розгортання серверної частини сайту було здійснено на локальному сервері MAMP.

У пакет MAMP входять:

- веб-сервер Apache
- модуль PHP
- база даних MySQL.

- phpmyAdmin

Налаштування локального веб-сервера МАР не вимагає якихось спеціальних знань. Сервер Матр є на даний момент найбільш зручним і ефективним локальним сервером для розміщення сайтів на своєму комп'ютері, особливо для новачків.

ВИСНОВКИ

У результаті виконання курсової роботи досліджено можливості технологій PHP, MySQL, HTML та CSS. Було доведено актуальність теми та показано на практиці можливості використання технологій системи управління базами даних.

У ході роботи реалізована веб-орієнтована інформаційна система автомобільного сервісу.

Були виконані завдання по розробці інтерфейсу сайту та бази даних. Шляхом розробки серверної частини було забезпечено взаємодію сайту та бази даних.

В розділі 1, було сформовано основну бізнес ідею інформаційної системи. Також було описано технічне завдання, яке стоїть при розробці веб-додатку. Було розроблено моделі використання сайту та його складових.

В розділі 2, описаний процес створення реляційної бази даних, таблиць та запитів. Визначено предметну область, проаналізовано нормалізацію реляційних відношень, зазначено типи даних, сформовано правила для цілісності даних та складено запити до бази даних, які забезпечують повноцінну роботу сайту. Реалізація бази даних була здійснена завдяки програмному середовищу phpMyAdmin та системою управління базами даних MySQL. Окрім того, створено діаграми сутностей та зв'язків для демонстрації моделі даних.

В розділі 3, описано етапи розробки веб-орієнтованої інформаційної системи. Було поетапно показано структуру сайту, описано його функціонал та можливості. Надано серверні скрипти на яких базується сайт. Описано структуру клієнтської частини сайту. Було приділено ретельну увагу розгортанню локального серверу на комп'ютері та описані можливості програмного продукту MAMP.

Слід також зазначити, що розробка інформаційної системи це постійний процес, в якому завжди є куди рости та вдосконалюватись. Виявлення багів та збільшення функціоналу це нормальний процес розробки веб-додатків. При продовженні розробки, планується ввести новий функціонал, такий як:

- Створення можливості запису до майстра.
- Створення можливості запису на конкретну послугу.

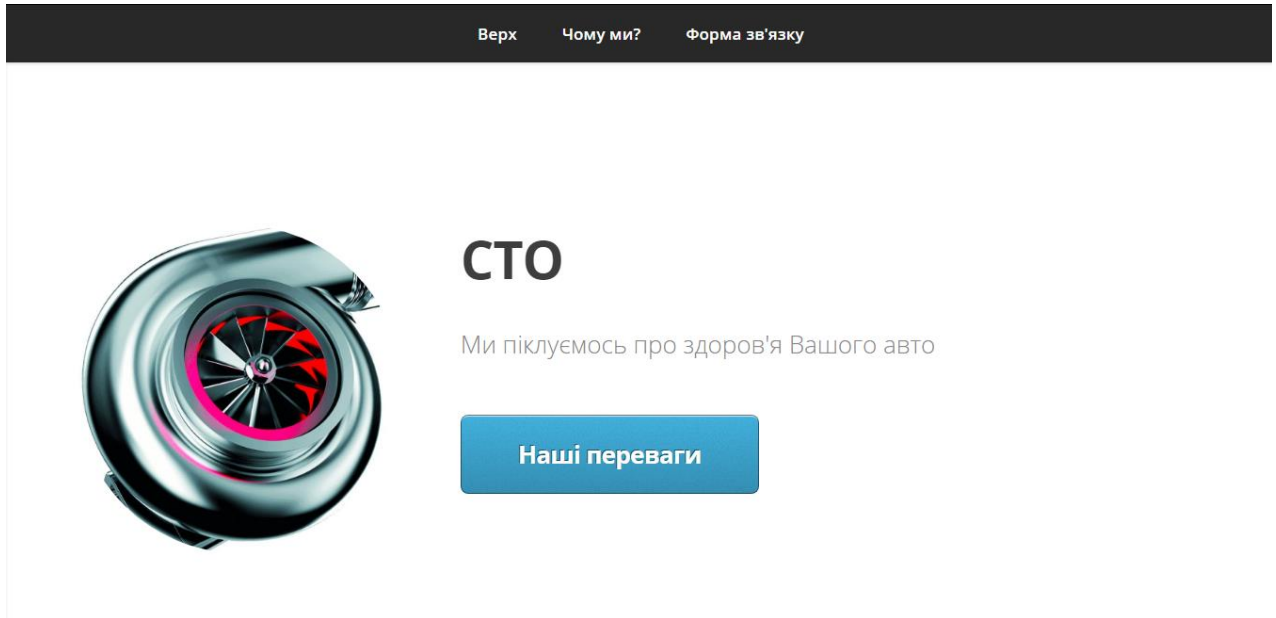
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Terry Felke-Morris. Web Development and Design Foundations with HTML5 / Terry Felke-Morris., 2018. – 720 с. – (9th Edition).
2. Jennifer Robbins. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics / Jennifer Robbins., 2018. – 810 с.
3. Mike O'Kane. A Web-Based Introduction to Programming / Mike O'Kane., 2017. – 628 с.
4. Larry Ullman. PHP and MySQL for Dynamic Web Sites / Larry Ullman., 2017. – 704 с.
5. Carlos Coronel. Database Systems: Design, Implementation, & Management / Carlos Coronel, Steven Morris., 2019. – 816 с.
6. Thomas Nield. Getting Started with SQL: A Hands-On Approach for Beginners / Thomas Nield., 2016. – 134 с.
7. CSS Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/css/default.asp>.
8. HTML Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/html/default.asp>.
9. SQL Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/sql/default.asp>.
10. PHP Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/php/default.asp>.
11. Создание PHP+MySQL сайта [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/sandbox/14484/>.
12. phpMyAdmin Tutorials [Електронний ресурс] – Режим доступу до ресурсу: <https://www.siteground.com/tutorials/phpmyadmin/>.

ДОДАТКИ

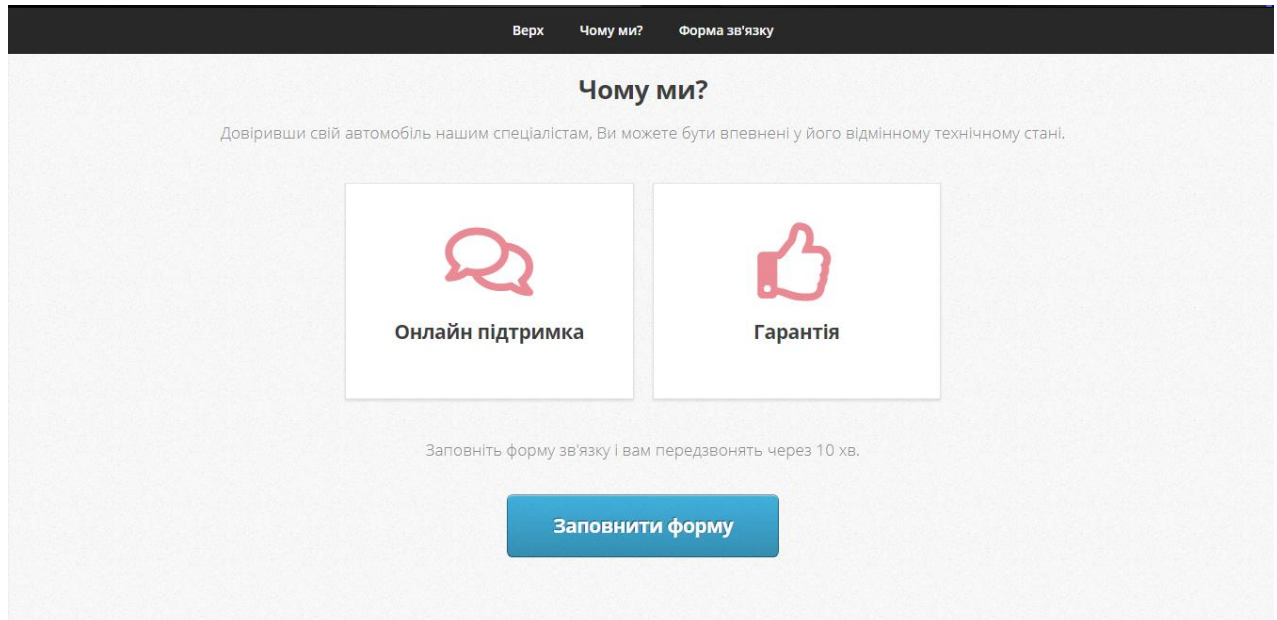
Додаток А

Блок перший, основної сторінки сайту.



Додаток Б

Блок другий, короткий опис переваг, які отримує користувач при співпраці з автосервісом.



Форма зворотнього зв'язку.

The image shows a dark-themed web form titled "Форма зв'язку" (Feedback Form). At the top, there is a navigation bar with three links: "Верх" (Top), "Чому ми?" (Why us?), and "Форма зв'язку" (Feedback Form). The main form area contains the following elements:

- Title:** "Форма зв'язку" (Feedback Form) in white text.
- Input Fields:**
 - A text input field labeled "Ваше ім'я" (Your name).
 - A text input field labeled "Номер телефону(Приклад 0971 150815)" (Phone number (Example 0971 150815)).
 - A large text area labeled "Опишіть вашу проблему..." (Describe your problem...).
- Buttons:**
 - A blue button labeled "Підтвердити" (Confirm).
 - A grey button labeled "Очистити форму" (Clear form).

Below the buttons, there is a horizontal line.

Підвал сайту.

