

ЕКОНОМІКО-ТЕХНОЛОГІЧНИЙ ІНСТИТУТ ІМЕНІ РОБЕРТА

ЕЛЬВОРТИ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

«Допущено до захисту»

Завідувач кафедри
інформаційних технологій

Бондар О.П.
« 19 » 06 2024 р.

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»
зі спеціальності 122 «Комп'ютерні науки»

на тему:

«Створення автоматизованої інформаційної системи для станції
технічного обслуговування автомобілів»

Зозуля Данило Євгенійович

Керівник кваліфікаційної роботи:

Неділько Віталій Миколайович, кандидат
технічних наук, завідувач кафедри
інформаційних технологій

Роботу рекомендовано до захисту
на засіданні кафедри інформаційних
технологій

Протокол № 10 від 06.06 2024 р.

Завідувач кафедри інформаційних
технологій

Бондар О.П.

Бондар О.П.

Роботу захищено на засіданні ЕК
з оцінкою

здоб. / 0 / 73

(за національною шкалою, шкалою
ECTS, бали)

Протокол № 8 від 06.06 2024 р.

Голова ЕК _____

Кропивницький – 2024

Керівник роботи

(підпис керівника)

Неділько В.М.

АНОТАЦІЯ

Зозуля Данило «Створення автоматизованої інформаційної системи для станції технічного обслуговування автомобілів». Дипломна робота з спеціальністю 122 «Комп'ютерні науки» складається з основної частини, що містить 3 розділи, 41с., 41 рис., 10 джерел використаної літератури.

Об'єкт дослідження: Процеси управління та організації роботи на станції технічного обслуговування автомобілів

Предмет дослідження: автоматизована інформаційна система, призначена для оптимізації та автоматизації процесів управління станцією технічного обслуговування автомобілів

Мета роботи: створення автоматизованої інформаційної системи для станції технічного обслуговування автомобілів, яка в свою чергу дозволить покращити організацію роботи підприємства, скоротити час на виконання операцій працівниками станції, та підвищить якість обслуговування клієнтів

Ключові слова: база даних, автоматизована інформаційна система, станція технічного обслуговування автомобілів

Зміст

Зміст	6
ВСТУП	7
РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА	9
1.1 Постановка задачі.....	9
1.2 Опис станції технічного обслуговування(СТО).....	10
1.3 Основні бізнес-процеси на СТО	11
1.4 Проблеми в управлінні станцією технічного обслуговування	12
1.5. Аналіз сучасних аналогів.....	12
РОЗДІЛ 2. ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СТО.....	21
2.1 Визначення вимог до системи.....	21
2.1.1 Вибір технологій та інструментів.....	22
2.1.2 Розробка бази даних.....	24
2.1.3 Розробка інтерфейсу користувача	26
2.1.4 Зв'язки між таблицями	33
2.3 Нові рішення у вирішенні задачі	36
РОЗДІЛ 3. ОПИС ТА ТЕСТУВАННЯ ФУНКЦІОНУВАННЯ СИСТЕМИ.....	38
3.1 Головна сторінка з навігаційним меню.....	38
3.2 Сторінки взаємодії з клієнтами.....	39
3.2 Сторінка взаємодії з автомобілями.....	41
3.3 Сторінки взаємодії з запчастинами	43
3.4 Сторінки взаємодії з записами на обслуговування.....	46
3.5 Тестування функціональності системи	47
3.6 Порівняння з існуючими аналогами.....	48
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТКИ	55

ВСТУП

Актуальність теми:

Сьогодні, коли інформаційні технології все більше інтегруються у всі сфери діяльності, автоматизація бізнес-процесів стає необхідністю для підвищення ефективності управління підприємствами. В умовах сучасного ринку, що характеризується високою конкуренцією та швидкими змінами, компанії змушені шукати нові способи підвищення своєї продуктивності та зниження витрат. Важливу роль у цьому відіграють системи управління підприємствами, які дозволяють централізовано зберігати дані, автоматизувати рутинні процеси та мінімізувати людський фактор.

Актуальність теми кваліфікаційної роботи полягає в тому, що впровадження автоматизованих систем управління на станціях технічного обслуговування (СТО) дозволяє значно покращити якість обслуговування клієнтів, знизити час на обробку інформації та зменшити кількість помилок, що виникають при ручному введенні даних. Це, в свою чергу, сприяє підвищенню конкурентоспроможності підприємства на ринку.

Сучасні інформаційні системи дозволяють ефективно управляти запасами запчастин, відслідковувати історію обслуговування автомобілів, автоматизувати процеси запису клієнтів на обслуговування та вести облік витрат. Вони забезпечують централізоване зберігання даних, що підвищує їхню цілісність та доступність для всіх користувачів системи.

В умовах зростаючого попиту на якісне та швидке обслуговування автомобілів, розробка та впровадження автоматизованої системи управління для СТО є актуальною та необхідною. Це дозволяє не тільки підвищити продуктивність роботи підприємства, але й забезпечити високий рівень задоволення клієнтів, що є ключовим фактором успіху у сучасному бізнесі.

Мета та завдання роботи:

Метою даної роботи є створення автоматизованої інформаційної системи для станції технічного обслуговування автомобілів, яка в свою чергу дозволить покращити організацію роботи підприємства, скоротити час на виконання операцій працівниками станції, та підвищить якість обслуговування клієнтів

Завдання роботи:

1. Провести аналіз існуючих рішень для автоматизації роботи СТО
2. Визначити основні вимоги для автоматизованої інформаційно\ системи СТО
3. Розробити архітектуру та прототип інформаційної системи
4. Провести тестування системи та зробити висновки що до її ефективності

Об'єкт та предмет дослідження:

Об'єктом дослідження в даному випадку є сама станція технічного обслуговування автомобілів, а саме процеси управління та організації роботи на цьому підприємстві

Предметом дослідження є спроектована автоматизована інформаційна система, призначена для оптимізації та автоматизації процесів управління станцією технічного обслуговування автомобілів.

Методи та інформаційна база проектування:

Методи:

1. Аналіз існуючих рішень
 - Аналіз існуючих програмних рішень, які можна знайти в інтернеті, та які використовуються на СТО. Їх переваги та недоліки
2. Методи системного аналізу
 - Розробка вимог до інформаційної системи на основі аналізу бізнес-процесів СТО

- Систематизація та класифікація даних, що використовуються в роботі СТО
3. Методи моделювання та проектування інформаційних систем
- Проектування бази даних та визначення її структури (таблиць, зв'язків між ними)
4. Програмування та тестування
- Розробка програмного забезпечення на базі PHP та MySQL
 - Тестування розробленого ПЗ

Інформаційна база дослідження:

- Наукові джерела
- Технічна документація
- Інтернет-ресурси
- Практичний досвід та емпіричні дані

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА

1.1 Постановка задачі

Метою мого дослідження є створення автоматизованої інформаційної системи для станції технічного обслуговування автомобілів, за допомогою якої можна буде вирішити основні проблеми управління СТО та підвищити ефективність її роботи. Для виконання цього завдання мені необхідно було вирішити такі завдання:

1. Провести аналіз бізнес-процесів на СТО та визначити ключові вимоги до інформаційної системи
2. Розробити модель даних та структуру бази даних, яка зможе забезпечити зберігання та управління інформацією про клієнтів, автомобілі, записи на обслуговування, послуги, запасні частини та інші об'єкти
3. Створити програмний інтерфейс користувача для зручної роботи з системою, включно з формами для введення та перегляду даних, також з інструментами для управління засобами
4. Реалізувати функціональні методи системи для автоматизації основних бізнес-процесів, таких як запис на обслуговування, облік виконаних робіт, управління запасами
5. Провести тестування системи для виявлення та усунення помилок

1.2 Опис станції технічного обслуговування(СТО)

Станція технічного обслуговування автомобілів це підприємство, що надає послуги з ремонту, комп'ютерної діагностики та обслуговування автомобілів. Основні види робіт притаманні більшості СТО:

- Регулярне технічне обслуговування автомобілів (заміна технічних рідин, фільтрів, гальмівних колодок тощо)
- Діагностика та ремонт основних агрегатів автомобіля (двигун, трансмісія, гальмівна система та підвіска)
- Ремонт заміна та фарбування деталей кузова автомобіля
- Шиномонтаж та балансування коліс
- Інші спеціалізовані послуги (роботи з електронікою, кондиціонуванням та інше)

1.3 Основні бізнес-процеси на СТО

Більшість процесів, що притаманні великій кількості станцій технічного обслуговування автомобілів можна розділити на наступні категорії:

1. Управління клієнтами та запис на обслуговування:
 - Реєстрація нових клієнтів
 - Запис клієнтів на технічне обслуговування або ремонт
 - Управління графіком обслуговування
2. Обслуговування та ремонт автомобілів:
 - Прийом автомобіля на обслуговування
 - Проведення діагностики та визначення обсягу робіт
 - Виконання ремонтних робіт та технічного обслуговування
 - Видача автомобіля клієнту після завершення робіт
3. Управління запасами та матеріалами:
 - Замовлення та прийом запасних частин і матеріалів
 - Облік і контроль запасів на складі
 - видача матеріалів для виконання робіт
4. Управління персоналом:
 - Розподіл завдань між робітниками
 - Облік робочого часу та продуктивність праці
 - підвищення кваліфікації та навчання персоналу
5. Фінансовий облік та звітність:
 - Ведення бухгалтерського обліку
 - Формування фінансових звітів
 - Розрахунок вартості робіт та видача рахунків клієнтам

1.4 Проблеми в управлінні станцією технічного обслуговування

Для виявлення проблем, що можуть виникнути на шляху автоматизації підприємства треба спочатку зрозуміти які проблеми існують на СТО ще до створення та впровадження автоматизованої інформаційної системи. Управління СТО включає в себе низку проблем, а саме:

1. Неорганізованість процесів

- відсутність чіткої структури автоматизації може призвести до плутанини та помилок у роботі

2. Низька ефективність управління запасами

- неправильний облік запасних частин може призвести до їх дефіциту або надлишку на складі, що в свою чергу негативно впливає на роботу СТО

3. Проблеми з обслуговуванням клієнтів

- відсутність єдиної бази клієнтів та історії їх обслуговування ускладнює роботу з клієнтами

1.5. Аналіз сучасних аналогів

Для визначення оптимальних рішень та функціональних можливостей автоматизованої інформаційної системи я провів аналіз сучасних аналогів, які вже використовуються на ринку. Розглянемо кілька популярних систем:

1. Autosoft

Опис: Autosoft є однією з найбільш відомих систем управління для СТО, яка пропонує широкий спектр функцій для управління

клієнтами, записами, запасами та фінансами. Система розроблена для задоволення потреб як малих, так і великих підприємств.[15]



Рис.1.1-Логотип Autosoft

Основні функції:

- Управління клієнтською базою та записами.
- Ведення історії обслуговування автомобілів.
- Управління запасами та поставаннями.
- Фінансовий облік та виставлення рахунків.
- Інтеграція з бухгалтерськими системами та іншими зовнішніми додатками.
- Потужні аналітичні інструменти для створення звітів та аналізу даних.

Переваги:

- Інтуїтивний інтерфейс, який легко освоїти.
- Хмарне рішення, що дозволяє доступ до даних з будь-якого місця.

- Можливість інтеграції з іншими системами (наприклад, бухгалтерськими програмами).
- Потужні аналітичні інструменти для створення звітів та аналізу даних.

Недоліки:

- Висока вартість впровадження та обслуговування.
- Складність налаштування для специфічних потреб окремих СТО.
- Обмежені можливості для кастомізації під унікальні вимоги підприємства.

2. Mitchell

1

Опис: Mitchell 1 пропонує повний набір інструментів для управління СТО, включаючи управління клієнтами, діагностику та ремонт, управління запасами та фінансовий облік. Система призначена для покращення ефективності та продуктивності роботи СТО[16].



Рис.1.2-Логотип Mitchell 1

Основні функції:

- Управління клієнтами та записами.
- Проведення діагностики та ремонтних робіт.
- Управління запасами та постачаннями.
- Фінансовий облік та звітність.
- Інтеграція з іншими продуктами Mitchell 1.
- Можливість масштабування та підтримка великої кількості користувачів.

Переваги:

- Комплексний підхід до управління всіма аспектами діяльності СТО.
- Вбудовані інструменти для діагностики та ремонтних процедур.
- Підтримка великої кількості користувачів та можливість масштабування.
- Регулярні оновлення програмного забезпечення.

Недоліки:

- Висока вартість ліцензій та технічної підтримки.
- Складний процес навчання персоналу для ефективного використання всіх функцій.
- Високі вимоги до технічних ресурсів для встановлення та експлуатації.

3. Shop

Boss

Опис: Shop Boss є онлайн платформою для управління СТО, яка пропонує функції для обліку клієнтів, управління записами,

фінансового обліку та звітності. Система спеціально розроблена для малого та середнього бізнесу[14].



Рис.1.3-Логотип Shop Boss

Основні функції:

- Управління клієнтами та записами.
- Ведення історії обслуговування автомобілів.
- Управління запасами та поставаннями.
- Фінансовий облік та виставлення рахунків.
- Онлайн доступ до системи з будь-якого пристрою.
- Підтримка мобільних пристроїв для роботи в режимі реального часу.

Переваги:

- Зручний інтерфейс з можливістю персоналізації.
- Онлайн доступ до системи з будь-якого пристрою.
- Можливість інтеграції з популярними платіжними системами та постачальниками запчастин.
- Підтримка мобільних пристроїв для роботи в режимі реального часу.

Недоліки:

- Відносно обмежені можливості для великих СТО з великою кількістю користувачів.
- Обмежена підтримка кастомізації під специфічні потреби.
- Деякі користувачі відзначають проблеми зі швидкістю роботи при великій кількості записів.

4. Manager SE (Manager Enterprise)

Опис: Manager SE від компанії Mitchell 1 - це розширене рішення для управління СТО, яке пропонує повний спектр інструментів для управління клієнтами, записами, запасами та фінансовим обліком. Система призначена для великих підприємств[13].



Рис.1.4-Логотип Mitchell 1

Основні функції:

- Управління клієнтами та записами.
- Проведення діагностики та ремонтних робіт.
- Управління запасами та постачаннями.
- Фінансовий облік та звітність.
- Інтеграція з іншими продуктами Mitchell 1.
- Високий рівень підтримки та регулярні оновлення.

Переваги:

- Потужний набір функцій для комплексного управління СТО.
- Можливість інтеграції з іншими продуктами Mitchell 1.
- Високий рівень підтримки та регулярні оновлення.
- Гнучкість у налаштуванні під специфічні потреби.

Недоліки:

- Висока вартість впровадження та підтримки.
- Складність налаштування та інтеграції з існуючими системами.
- Вимоги до висококваліфікованого персоналу для ефективного використання.

5. AutoFluent

Опис: AutoFluent пропонує рішення для управління СТО, що включає

управління клієнтами, записами, запасами, фінансами та аналітику. Система підходить для підприємств різного масштабу[17].



Рис.1.5-Логотип AutoFluent

Основні функції:

- Управління клієнтами та записами.
- Ведення історії обслуговування автомобілів.
- Управління запасами та поставаннями.
- Фінансовий облік та виставлення рахунків.
- Потужні аналітичні інструменти для створення звітів та аналізу даних.
- Інтеграція з різними сторонніми системами.

Переваги:

- Комплексний підхід до управління всіма аспектами СТО.
- Потужні аналітичні інструменти для створення звітів та аналізу даних.

- Висока гнучкість у налаштуванні під специфічні потреби бізнесу.
- Можливість інтеграції з різними сторонніми системами.

Недоліки:

- Відносно висока вартість ліцензій та технічної підтримки.
- Складність навчання персоналу для ефективного використання всіх функцій.
- Обмежені можливості для роботи в режимі реального часу з мобільних пристроїв.

РОЗДІЛ 2. ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СТО

2.1 Визначення вимог до системи

Перед початком створення автоматизованої інформаційної системи СТО, для її ефективної роботи мною були визначені наступні вимоги до системи:

1. Управління клієнтами, а саме додавання нових клієнтів, редагування та видалення інформації про старих відвідувачів та зберігання їх контактної інформації
2. Управління автомобілями: додавання нових автомобілів, редагування та видалення даних про клієнтські автомобілі, що вже відвідували станцію, зберігання інформації про марку, модель, рік випуску та номерний знак автомобіля
3. Записи на обслуговування: додавання нових, редагування та видалення існуючих записів на обслуговування, також зберігання інформації про дату, опис наданої послуги та статус виконання обслуговування
4. Управління запасами: тут так само повинні бути присутні операції з додавання нещодавно прибувших запчастин, редагування та видалення запчастин, що знаходяться на складі, зберігання даних про назву запчастин, їх кількість та ціну

2.1.1 Вибір технологій та інструментів

Для розробки моєї системи основною мовою програмування стала PHP. PHP — скриптова мова програмування (мова, яка розроблена для запису «сценаріїв», послідовностей операцій, які користувач може виконувати на комп'ютері.), була створена для генерації HTML-сторінок на стороні веб сервера. Її перевагами є:

1. Простий зрозумілий та логічний синтаксис, з чого випливає другий пункт
2. Легкість в опанування й використанні, командні функції легко зрозуміти, оскільки вони самі повідомляють розробнику, яку саме функцію вони виконують
3. Однією з найголовніших переваг PHP для використання його у моїй роботі є те, що він сумісний майже з усіма відомими видами баз даних
4. Ще одна перевага PHP полягає у тому, що він сумісний з HTML. У готову PHP сторінку можна запросто інтегрувати HTML-код за допомогою скрипту

У якості системи управління базами даних мною була вибрана MySQL. MySQL-це система управління реляційними базами даних(реляційні бази даних-це тип баз даних в кому самі дані організовані у вигляді таблиць, де рядки цих таблиць представляють собою унікальні елементи даних, стовпці ж представляють певний атрибут цих елементів), яка базується на мові структурованих запитів SQL(стандартна мова для роботи з реляційними базами даних)[9].

Переваги даної СУБД:

1. Висока продуктивність(вона оптимізована для швидкої обробки запитів та здатна ефективно працювати з великими обсягами даних)

2. Надійність та відмовостійкість(MySQL має функції реплікації та кластеризації, які забезпечують високу доступність і надійність даних. Це особливо важливо для критично важливих застосувань, де неперервність роботи є ключовою.)
3. Безпека (MySQL пропонує широкий набір інструментів для забезпечення безпеки даних, включаючи контроль доступу на основі ролей, шифрування даних, аутентифікацію користувачів і аудит операцій.)
4. Простота у використанні(MySQL має простий і інтуїтивно зрозумілий синтаксис SQL, що полегшує його вивчення і використання. Існує багато інструментів і утиліт для адміністрування та управління MySQL, таких як phpMyAdmin.)

Як інтерфейс для роботи з базою даних я вибрав phpMyAdmin.PhpMyAdmin-це безкоштовний інструмент для управління базами даних за допомогою веб-інтерфейсу, він дозволяє виконувати різні операції з базами даних без потреби знати деталі SQL, хоча функція виконання SQL-запитів вручну також підтримується цим інструментом.

Переваги phpMyAdmin:

1. Веб-інтерфейс(забезпечує зручний та інформативно зрозумілий інтерфейс для взаємодії з базами даних. Це дозволяє виконувати завдання з адміністрування баз даних через веб-браузер)
2. Управління базами даних(за допомогою phpMyAdmin можна створювати нові, редагувати та видаляти вже існуючі бази даних, також виконувати ті ж самі операції з таблицями, стовпцями, індексами, користувачами та іншими об'єктами баз даних)

3. Виконання SQL-запитів (phpMyAdmin підтримує можливість написання та виконання SQL-запитів вручну, що допомагає більш гнучко керувати даними)

Для створення інтерфейсу користувача я вибрав HTML та CSS через їх простоту та можливість легкого інтегрування HTML тегів у PHP код.

2.1.2 Розробка бази даних

Процес створення бази даних включав в себе розробку структури та таблиць, які б задовольняли вимоги до автоматизованої інформаційної системи:

1. Таблиця клієнтів:

- Таблиця клієнтів має містити в собі унікальний ідентифікатор клієнта, його ім'я, прізвище, номер телефону та адресу електронної пошти для подальшої взаємодії з клієнтами.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	first_name	varchar(50)	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
3	last_name	varchar(50)	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
4	phone	varchar(20)	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
5	email	varchar(100)	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше

Рис. 2.1-Структура таблиці клієнтів

2. Таблиця автомобілів:

- Таблиця автомобілів містить унікальний ідентифікатор кожного автомобіля, ідентифікатор клієнта, якому належить автомобіль, марку автомобіля, його модель, рік випуску та номерний знак автомобіля, також кожен автомобіль повинен бути пов'язаний з клієнтом у таблиці клієнтів і гарантувати що для кожного запису в таблиці автомобілів ідентифікатора клієнта буде відповідати існуючому клієнту в таблиці клієнтів

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	customer_id	int(11)			Так	NULL			Змінити Знищити Більше
3	make	varchar(50)	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
4	model	varchar(50)	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
5	year	int(11)			Так	NULL			Змінити Знищити Більше
6	plate_number	varchar(20)	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше

Рис. 2.2-Структура таблиці автомобілів

3. Таблиця записів на обслуговування:

- Таблиця записів на обслуговування повинна мати унікальний ідентифікатор для кожного запису про обслуговування, ідентифікатор автомобіля, до якого відноситься запис про обслуговування, дату обслуговування автомобіля, опис виконаних робіт або обслуговування, статус виконання замовлення (“Виконано” “На черзі” “В роботі”). Кожен запис у стовпці ідентифікатора автомобіля, в цій таблиці, повинен завжди відповідати значенню унікального ідентифікатора у таблиці автомобілів

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	car_id	int(11)			Так	NULL			Змінити Знищити Більше
3	service_date	date			Так	NULL			Змінити Знищити Більше
4	service_description	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
5	status	varchar(20)	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше

Рис. 2.3-Структура таблиці записів на обслуговування

4. Таблиця для запчастин

- Таблиця для запчастин також зобов'язана мати унікальний ідентифікатор для кожного запису в таблиці, назву запчастини, кількість конкретної запчастини або товару на складі та ціну на цю запчастину

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)		Ні	Немає			AUTO_INCREMENT	Змінити Знищити Більше
2	part_name	varchar(100)	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
3	quantity	int(11)		Так	NULL				Змінити Знищити Більше
4	price	decimal(10,2)		Так	NULL				Змінити Знищити Більше

Рис. 2.4-Структура таблиці запчастин

2.1.3 Розробка інтерфейсу користувача

Щоб створити веб-інтерфейс для взаємодії з системою, мені потрібно було розробити наступні сторінки:

1. Головна сторінка з навігацією:

- Головна сторінка повинна мати інтерфейс з навігаційним меню у якому міститься посилання на різні дії, такі як додавання та перегляд клієнтів, автомобілів, записів на обслуговування та запчастин

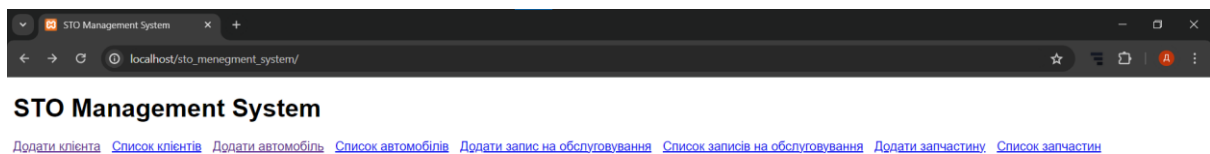


Рис.2.5-Головна сторінка з навігацією

2. Форми для додавання клієнтів, автомобілів, записів на обслуговування та запчастин

- Форма для додавання клієнтів містить ім'я нового клієнта, прізвище, телефон та його електронну пошту, також кнопку “Додати клієнта”

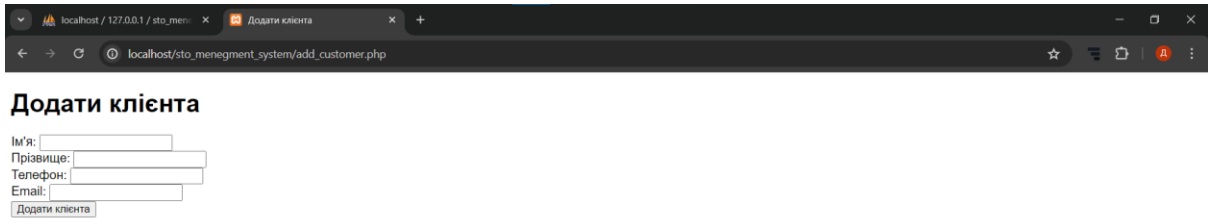


Рис. 2.6-Форма для додавання клієнтів

- Форма для додавання автомобіля містить ім'я вже зареєстрованого клієнта, марку, модель, рік випуску та державний реєстраційний номерний знак автомобіля

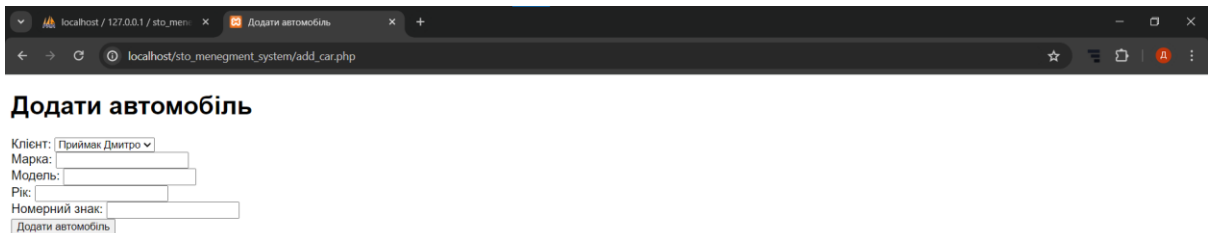
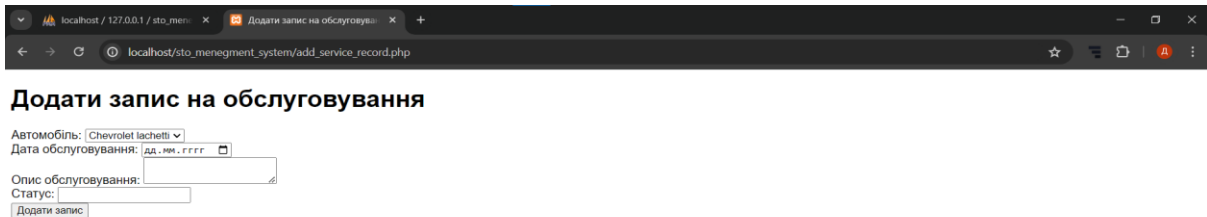


Рис.2.7-Форма додавання автомобіля

- Форма для додавання запису на обслуговування автомобілів має поле вибору вже зареєстрованого автомобіля, поле вводу дати запису на обслуговування, поле для опису обслуговування, поле з статусом замовлення, та кнопку додавання запису до таблиці



Додати запис на обслуговування

Автомобіль: Chevrolet Lachetti

Дата обслуговування: dd.mm.yyyy

Опис обслуговування:

Статус:

Додати запис

Рис.2.7-Форма додавання запису на обслуговування

- Форма для додавання запчастин містить в собі всього 3 поля: назва запчастини, кількість та ціна цієї запчастини, також кнопку додавання запчастини до таблиці

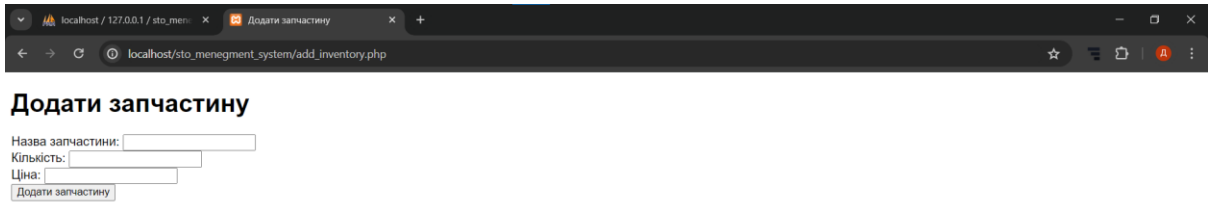


Рис.2.8-Форма для додавання запчастин

- Список клієнтів містить в собі таблицю в якій записан індивідуальний унікальний ідентифікатор клієнта (ID), прізвище та ім'я клієнта, його телефон та адреса електронної пошти, також в таблиці є окрема колонка у якій знаходиться кнопки видалення та редагування інформації з окремого рядка таблиці

ID	Ім'я	Прізвище	Телефон	Email	Дії
1	Приймак	Дмитро	0664367865	dmitro8838@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис.2.9-Список зареєстрованих клієнтів

- Список автомобілів також являє собою таблицю в якій знаходяться такі колонки, як: індивідуальний унікальний ідентифікатор автомобіля (ID), прізвище та ім'я власника, марка автомобіля, також його модель, рік випуску та державний реєстраційний номерний знак автомобіля, так само як і в списку клієнтів, список автомобілів має окрему колонку з двома кнопками для редагування та видалення записів в рядку, у якому знаходяться ці кнопки

Список автомобілів

ID	Клієнт	Марка	Модель	Рік	Номерний знак	Дії
2	Приймак Дмитро	Chevrolet	Lacetti	2012	BA3262EK	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис. 2.10-Список зареєстрованих автомобілів

- Список записів на обслуговування також представлений у вигляді таблиці з шістьма колонками та необмеженою кількістю рядків, колонки містять у собі таку інформацію: індивідуальний унікальний ідентифікатор номеру запису на обслуговування, марку та модель автомобіля, дату на яку заплановані роботи з обслуговування, сам опис робіт, які будуть проводитися з автомобілем, та статус виконання робіт, також як і інші списки в моїй роботі ця таблиця має колонку з двома кнопками, що відповідають за редагування та видалення записів

localhost / 127.0.0.1 / sto_men... x Список записів на обслугову... x +
localhost/sto_menagement_system/list_service_records.php

Список записів на обслуговування

ID	Автомобіль	Дата обслуговування	Опис	Статус	Дії
1	Chevrolet lachetti	2024-06-03	Заміна мастила, фільтрів та гальмівних колодок	На черзі	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис. 2.11-Список записів на обслуговування

- Список запчастин являє собою таблицю з п'ятьма колонками, перша як і в інших таблицях відповідає за індивідуальний унікальний ідентифікатор запису, далі йде назва запчастини, кількість цих запчастин, їх ціна та як і в інших таблицях остання колонка відповідає за редагування та видалення записів в даній таблиці

ID	Назва запчастини	Кількість	Ціна	Дії
2	Фільтр мастила(Lacetti)	2	234.00	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис.2.12-Список запчастин

2.1.4 Зв'язки між таблицями

Для правильної роботи базою даних та коректного додавання інформації і в подальшому виведенні її на екран користувача таблиці повинні бути зв'язані між собою за допомогою ключових полей.

Після створення бази даних мною була створена перша таблиця (таблиця клієнтів), структура якої була описана у пункті 2.1.3, ключовим полем для даної таблиці став унікальний ідентифікатор клієнта "id", за допомогою якого ця таблиця зв'язується з другою створеною таблицею (таблиця автомобілів), яка також містить у собі поле "customer_id" для визначення власника автомобіля.

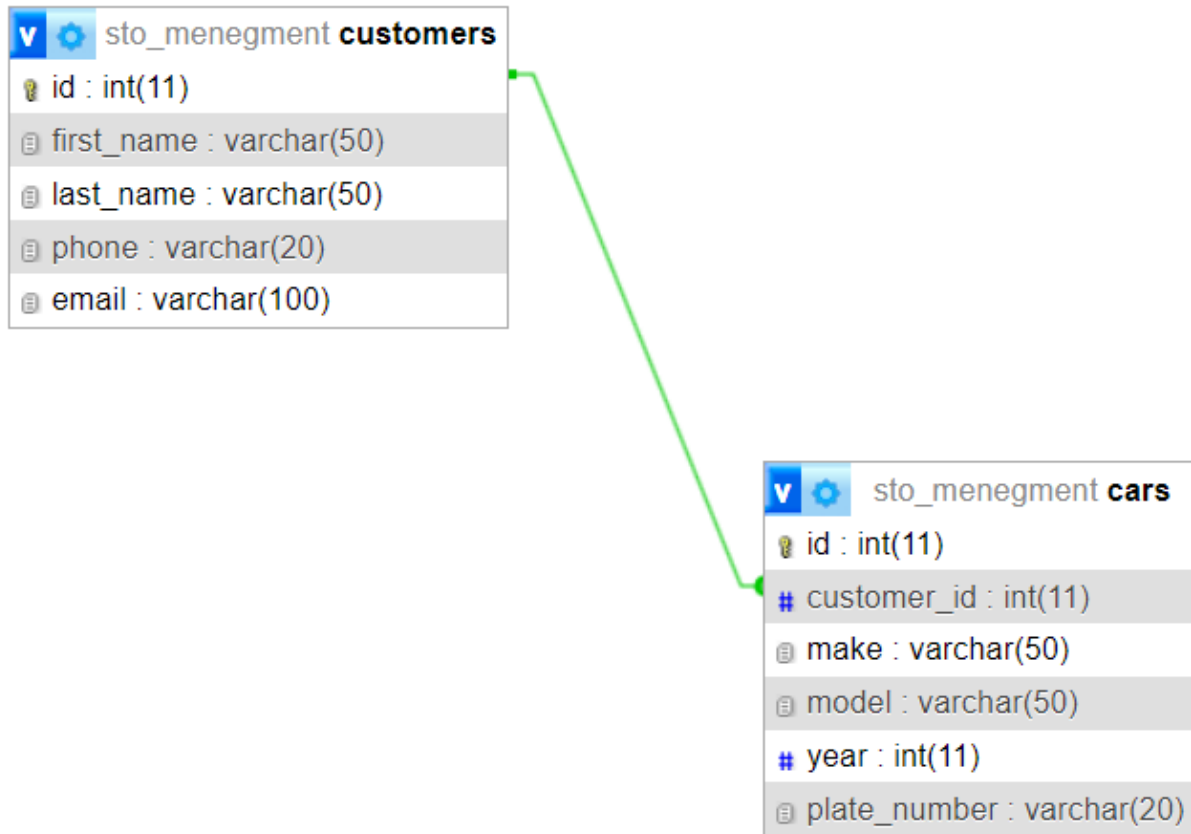


Рис.2.13-Зв'язок таблиці клієнтів з таблицею автомобілів

Друга таблиця (таблиця автомобілів) має вже два зв'язки, перший зв'язок з таблицею клієнтів, зображений на рисунку 2.13, а другий поєднує між собою таблицю автомобілів з таблицею записів на обслуговування за також за допомогою унікального ідентифікатора "id", але це вже ідентифікатор автомобіля, який далі буде використовуватися для створення запису на обслуговування автомобіля, також цей зв'язок є єдиним для третьої таблиці (таблиці записів на обслуговування)

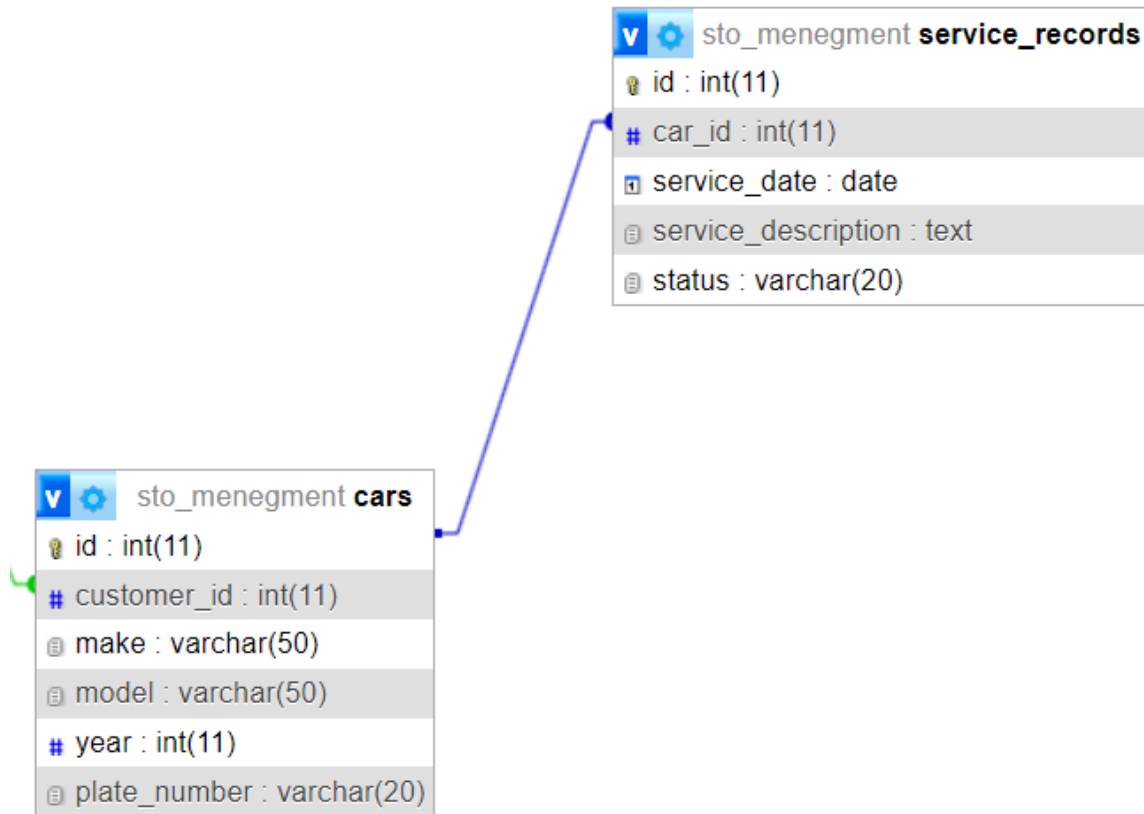


Рис. 2.14-Зв'язок таблиці автомобілів з таблицею записів на обслуговування

Остання таблиця (таблиця запчастин) не має зв'язків з іншими елементами системи, тому-що з особистої практики можу я можу зробити висновок, що в процесі виконання робіт з обслуговування та ремонту автомобілів можуть виникнути непередбачувані ситуації з запчастинами, тому на мій погляд простіше редагувати список запчастин вручну, після виконання робіт з автомобілем вже по факту використаних видів запчастин та їх кількості.

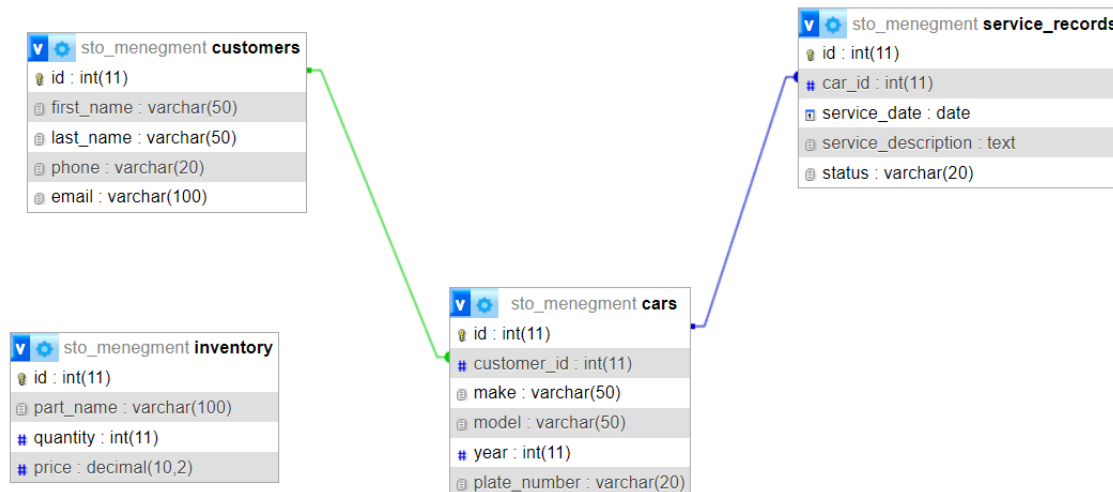


Рис.2.15-Всі зв'язки в базі даних автоматизованої інформаційної системи

2.3 Нові рішення у вирішенні задачі

Впровадження автоматизованої інформаційної системи дозволяє зменшити витрати часу на обробку інформації, та зменшити кількість помилок, що виникають при ручному введенні даних, а саме:

- Зменшення часу на обробку інформації досягається за рахунок швидкого доступу до даних (автоматизована система надає миттєвий доступ до всіх необхідних даних, що дозволяє знизити час, необхідний для їх обробки), та швидкого оновлення даних (усі зміни в даних автоматично зберігаються в реальному часі, забезпечуючи актуальність інформації без необхідності ручного введення)
- Зменшення кількості помилок при ручному введенні даних вдається досягти завдяки валідації даних (система може автоматично перевіряти правильність введених даних, запобігаючи помилкам ще на етапі введення), автоматичних підказок (під час введення даних система може надавати підказки та автозаповнення, що знижує

ймовірність помилок) та зниження людського фактора (автоматизація значно зменшує кількість ручних операцій, що в свою чергу мінімізує кількість помилок, викликаних людським фактором)

Поліпшення управління даними в автоматизованій інформаційній системі забезпечується завдяки:

1. Централізованому управлінню даними, що має на увазі под собою
 - Єдину базу даних (всі дані зберігаються в єдиній базі даних, що забезпечує їхню цілісність та узгодженість)
 - Легку інтеграці. (централізована система дозволяє легко інтегрувати дані з різних джерел та забезпечує узгоджене їх використання)
 - Забезпечення резервного копіювання (зберігання даних у централізованій базі дозволяє легко налаштувати резервне копіювання, що захищає від втрати даних)
2. Покращенню доступності до даних, в яке входить
 - Доступ з будь-якого місця (користувачі можуть отримувати доступ до даних з будь-якого місця за наявності підключення до інтернету)
 - Розширене управління правами доступу (система дозволяє гнучко налаштовувати права доступу для різних користувачів, забезпечуючи необхідний рівень безпеки)
 - Миттєвий пошук інформації (завдяки індексації та ефективним механізмам пошуку, користувачі можуть швидко знайти необхідну інформацію в базі даних)

РОЗДІЛ 3. ОПИС ТА ТЕСТУВАННЯ ФУНКЦІОНУВАННЯ СИСТЕМИ

3.1 Головна сторінка з навігаційним меню

Кінцевий вигляд головної сторінки навігаційного меню зображено на рисунку 3.1

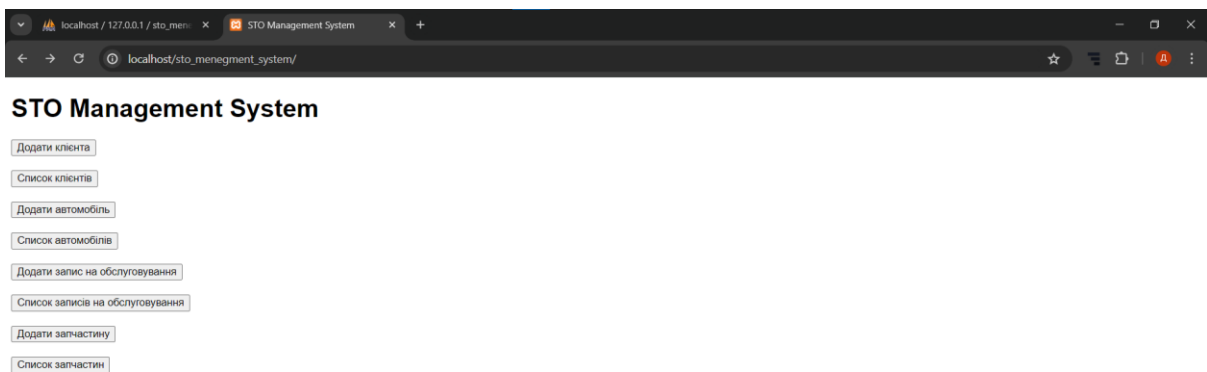


Рис.3.1-Головна сторінка з навігаційним меню

При потраплянні на головну сторінку користувача зустрічає назва системи написана чорними літерами на білому фоні, трошки нижче знаходиться саме навігаційне меню, яке складається з восьми кнопок, які в свою чергу можна умовно поділити на умовні 4 блоки по дві кнопки, одна з яких переміщає користувача на форму для додавання нових записів до бази даних, а інша відкриває інформацію що міститься в базі даних у вигляді таблиці.

3.2 Сторінки взаємодії з клієнтами

Для взаємодії з клієнтами станції технічного обслуговування було створено три сторінки:

1. Сторінка додавання нових клієнтів з полями для вводу ім'я, прізвища, номеру телефону та пошти клієнта, тож спробуємо додати нового клієнта. Якщо запит на додавання клієнта буде виконано, сторінка відзвітує про це.

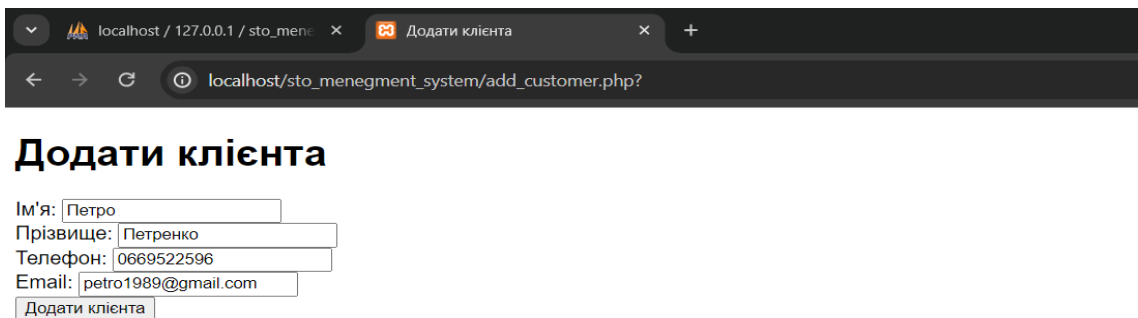


Рис.3.2-Додавання клієнта

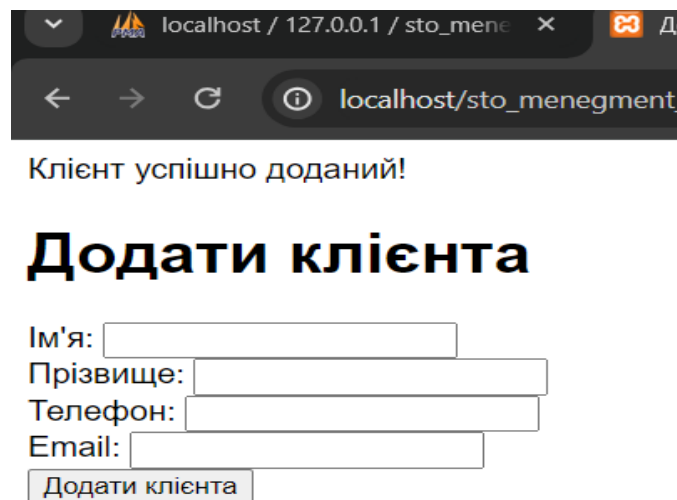


Рис.3.3-Інформація про виконання запиту

	id	first_name	last_name	phone	email
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	1	Приймак	Дмитро	0664367866	dmitro8838@gmail.com
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	2	Данило	Сібігатулін	0953857426	sibigatulln@gmail.com
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	3	Петро	Петренко	0669522596	petro1989@gmail.com

Рис.3.4-Клієнт у базі даних

- Друга таблиця відображає таблицю з інформацією про клієнтів, які вже знаходяться в базі даних, також таблиця має колонку з двома кнопками для видалення та редагування інформації про клієнта, при натисканні на кнопку редагування користувача переносить на сторінку редагування інформації, а при натисканні на кнопку видалення впливає повідомлення для підтвердження операції з записом.

Список клієнтів

Ім'я	Прізвище	Телефон	Email	Дії
Приймак	Дмитро	0664367866	dmitro8838@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
Данило	Сібігатулін	0953857426	sibigatulln@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
Петро	Петренко	0669522596	petro1989@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис.3.5-Таблиця інформації про зареєстрованих клієнтів

Список клієнтів

Ім'я	Прізвище	Телефон	Email	Дії
Приймак	Дмитро	0664367866	dmitro8838@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
Данило	Сібігатулін	0953857426	sibigatulln@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
Петро	Петренко	0669522596	petro1989@gmail.com	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Підтвердіть дійствие на localhost

Ви впевнені, що хочете видалити цього клієнта?

Рис.3.6-Повідомлення для підтвердження операції видалення

- Сторінка редагування складається з тих самих полів що й сторінка додавання нових клієнтів, але всі поля вже заповнені інформацією про клієнта яку ми хочемо редагувати, та знизу замість кнопки “Додати клієнта” розміщена кнопка “Оновити клієнта”

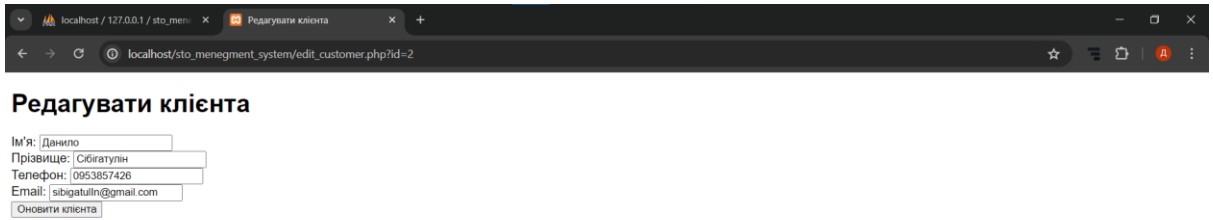


Рис. 3.7-Сторінка редагування інформації про клієнта

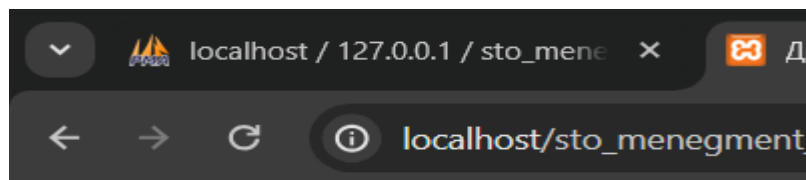
3.2 Сторінка взаємодії з автомобілями

Взаємодія з клієнтськими автомобілями відбувається також за допомогою трьох сторінок:

1. На сторінці, за допомогою якої відбувається додавання нового автомобіля знаходиться поле для вибору власника (зі списку вже зареєстрованих клієнтів), марка, модель, рік випуску та номерний знак автомобіля. При вдалому виконанні запиту сторінка також повідомить про додавання нового запису до таблиці



Рис.3.8-Сторінка додавання автомобілів

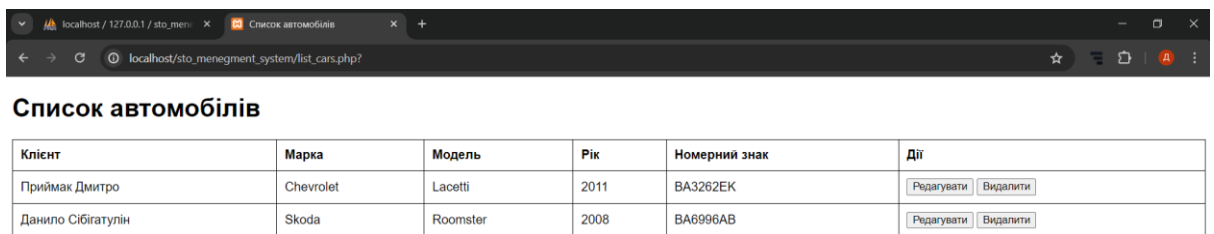


Автомобіль успішно доданий!

Додати автомобіль

Рис.3.9-Повідомлення про виконання запиту

2. Друга таблиця представляє собою таблицю в якій продубльована інформація про автомобілі з основної бази даних також як в таблиці з клієнтами, таблиця автомобілів має колонку з двома кнопками редагування та видалення записів. Після натискання кнопки видалення запису впливає повідомлення з підтвердженням операції видалення, а після натиску на кнопку редагування користувача перенесе на окрему форму для редагування вибраного запису



Клієнт	Марка	Модель	Рік	Номерний знак	Дії
Приймак Дмитро	Chevrolet	Lacetti	2011	BA3262EK	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
Данило Сібігатулін	Skoda	Roomster	2008	BA6996AB	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис.3.10-Таблиця зареєстрованих автомобілів

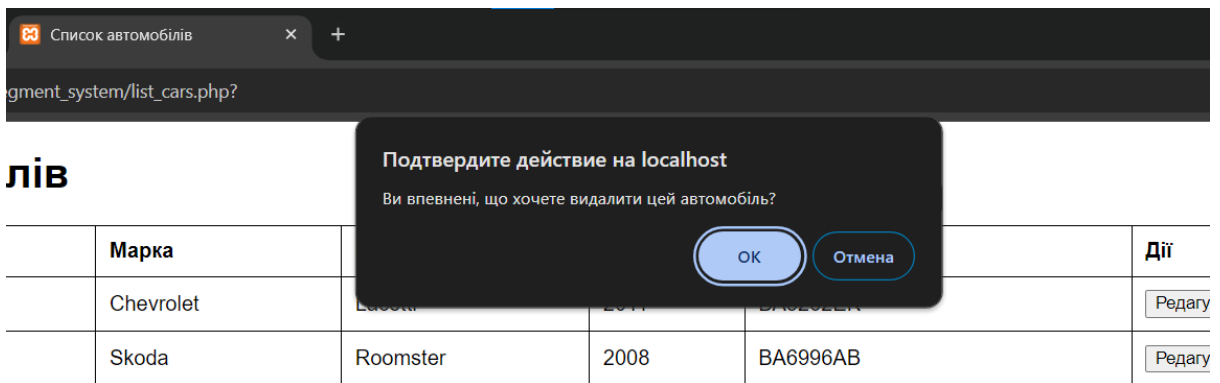


Рис.3.11-Спливаюче повідомлення підтвердження

3. На сторінці редагування також присутні ті ж самі поля що й на сторінці додавання, але вже заповнені інформацією для редагування, кнопка “додати автомобіль” замінена на “оновити автомобіль”

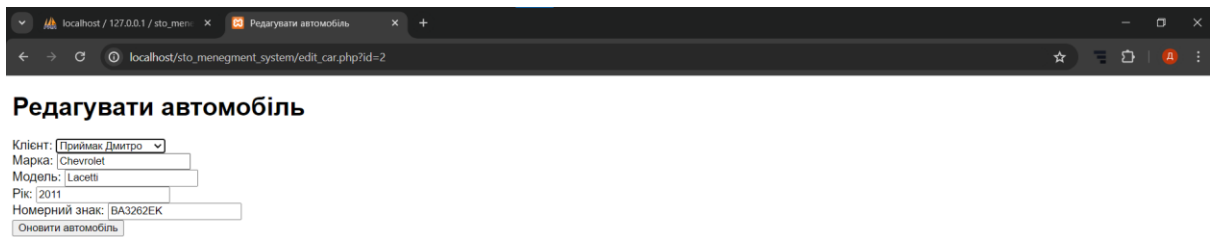


Рис.3.12-Сторінка редагування інформації про автомобілі

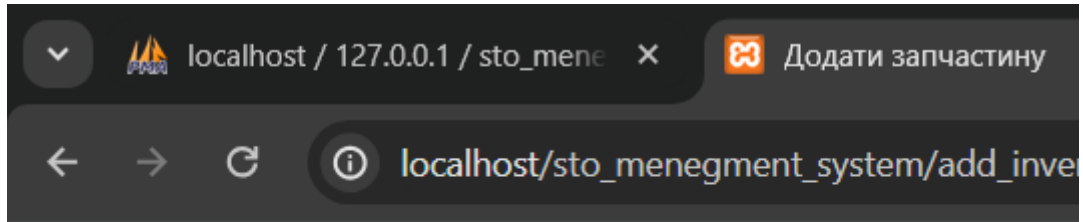
3.3 Сторінки взаємодії з запчастинами

Процес роботи з запчастинами також був реалізований через три сторінки:

1. Сторінка додавання нової запчастини - ця сторінка містить поля для вводу назви запчастини, кількості та ціни. Після успішного додавання запчастини система повідомляє про це.



Рис. 3.13 - Сторінка додавання запчастини



Запчастина успішно додана!

Додати запчастину

Рис. 3.14 - Повідомлення про виконання запиту

2. Сторінка перегляду запчастин - на цій сторінці зображена таблиця з інформацією про запчастини, що вже знаходяться в базі даних. Таблиця має колонку з двома кнопками для видалення та редагування інформації про запчастину. При натисканні на кнопку редагування користувача переносить на сторінку редагування інформації, а при натисканні на кнопку видалення впливає повідомлення для підтвердження операції з записом.

ID	Назва запчастини	Кількість	Ціна	Дії
2	Фільтр мастила(Lacetti)	6	234.00	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
4	Гальмівні колодки передні (lacetti)	4	812.00	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
5	Фільтр мастила(VW jetta)	5	452.00	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис. 3.15 - Таблиця інформації про запчастини

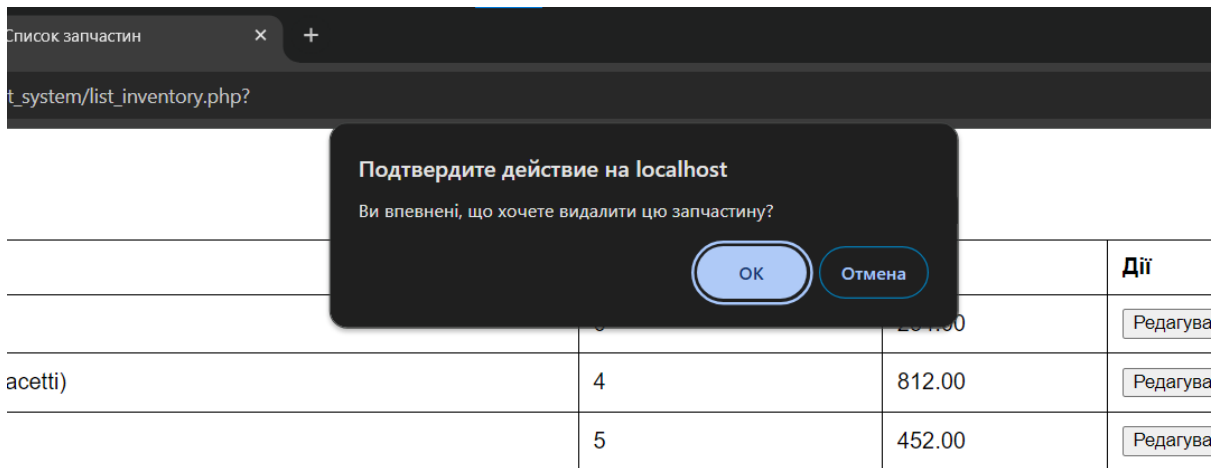


Рис. 3.16 - Повідомлення для підтвердження операції видалення

3. Сторінка редагування запчастин - ця сторінка складається з тих самих полів, що й сторінка додавання нових запчастин, але всі поля вже заповнені інформацією про запчастину, яку ми хочемо редагувати. Знизу замість кнопки “Додати запчастину” розміщена кнопка “Оновити запчастину”.



Рис. 3.17 - Сторінка редагування інформації про запчастину

3.4 Сторінки взаємодії з записами на обслуговування

Так само як і в інших варіантах, для взаємодії з записами на обслуговування в моїй системі створено три таблиці:

1. Сторінка додавання нового запису на обслуговування - ця сторінка містить поля для вибору клієнта, автомобіля, дати обслуговування, типу послуги та коментарів. Після успішного додавання запису система повідомляє про це.

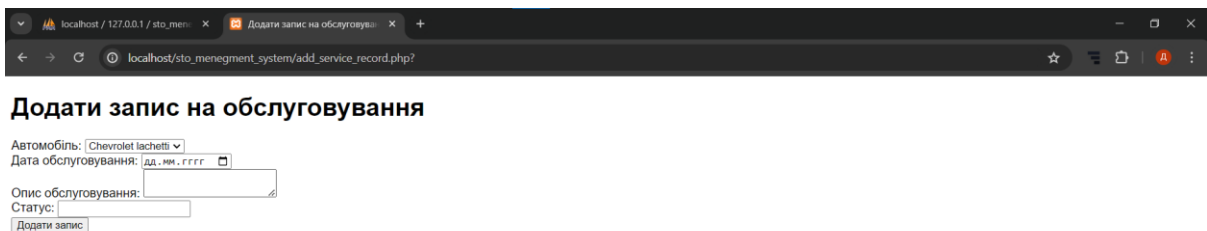
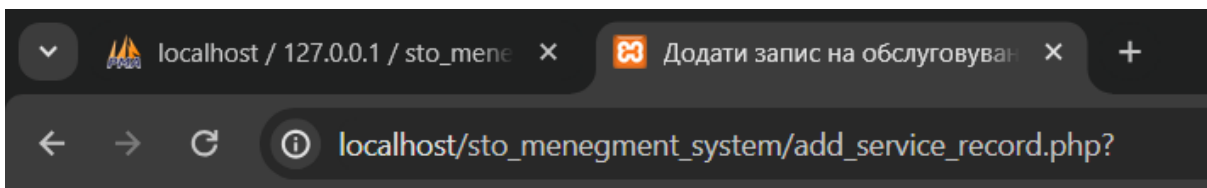


Рис. 3.18 - Сторінка додавання запису на обслуговування



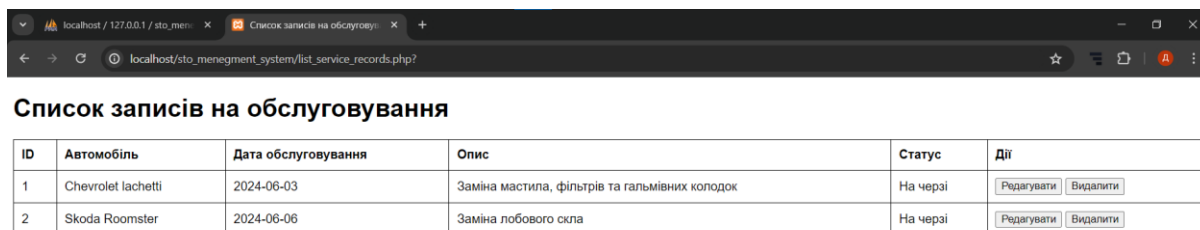
Запис на обслуговування успішно доданий!

Додати запис на обслуговування

Рис. 3.19 - Повідомлення про виконання запиту

2. Сторінка перегляду записів на обслуговування - на цій сторінці зображена таблиця з інформацією про записи на обслуговування, що вже знаходяться в базі даних. Таблиця має колонку з двома кнопками для видалення та редагування інформації про запис. При натисканні

на кнопку редагування користувача переносить на сторінку редагування інформації, а при натисканні на кнопку видалення впливає повідомлення для підтвердження операції з записом.



ID	Автомобіль	Дата обслуговування	Опис	Статус	Дії
1	Chevrolet lachetti	2024-06-03	Заміна мастила, фільтрів та гальмівних колодок	На черзі	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
2	Skoda Roomster	2024-06-06	Заміна лобового скла	На черзі	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рис. 3.20 - Таблиця інформації про записи на обслуговування

3. Сторінка редагування записів на обслуговування - ця сторінка складається з тих самих полів, що й сторінка додавання нових записів на обслуговування, але всі поля вже заповнені інформацією про запис, який ми хочемо редагувати. Знизу замість кнопки “Додати запис” розміщена кнопка “Оновити запис”.

3.5 Тестування функціональності системи

Для забезпечення коректної роботи системи було проведено тестування основних функціональних модулів:

1. Тестування додавання записів - перевірка коректності роботи форм для додавання клієнтів, автомобілів, запчастин та записів на обслуговування. Було перевірено, чи всі поля коректно зберігаються у базі даних, а також чи відображаються відповідні повідомлення про успішне додавання записів.
2. Тестування редагування записів - перевірка роботи форм для редагування інформації про клієнтів, автомобілі, запчастини та

записи на обслуговування. Було перевірено, чи всі зміни коректно зберігаються у базі даних, а також чи відображаються відповідні повідомлення про успішне оновлення записів.

3. Тестування видалення записів - перевірка функціональності видалення записів з бази даних. Було перевірено, чи відображаються відповідні повідомлення про підтвердження операції видалення, а також чи коректно видаляються записи з бази даних.
4. Тестування відображення даних - перевірка коректності відображення таблиць з інформацією про клієнтів, автомобілі, запчастини та записи на обслуговування. Було перевірено, чи всі дані коректно відображаються у відповідних таблицях, а також чи працюють кнопки для редагування та видалення записів.

3.6 Порівняння з існуючими аналогами

Для оцінки ефективності та функціональності розробленої системи STO Management System, проведемо порівняння з одним із існуючих аналогів на ринку — програмою "AutoRepair Cloud"[18].

1. Інтерфейс користувача
 - STO Management System: Інтерфейс простий та інтуїтивно зрозумілий, що полегшує роботу користувачів. Навігаційне меню дозволяє швидко переміщуватися між основними розділами системи.

- AutoRepair Cloud: Більш розширений інтерфейс з можливістю налаштування, підтримка різних тем та інтеграція з мобільним додатком для зручності користувачів.
2. Додаткові можливості
- STO Management System: Функції системи включають основні операції для управління клієнтами, автомобілями, запчастинами та записами на обслуговування. Відсутні можливості інтеграції з іншими системами та мобільний додаток.
 - AutoRepair Cloud: Пропонує розширені можливості, такі як інтеграція з іншими системами, підтримка мобільного додатку та хмарне зберігання даних, що забезпечує доступність даних з будь-якого місця.
3. Порівняння за вартістю
- STO Management System: Може бути розгорнута на локальному сервері, що дозволяє зменшити витрати на підписку та зберігання даних.
 - AutoRepair Cloud: Вимагає підписки для використання всіх функцій, включаючи хмарне зберігання та мобільний додаток.

Висновки

Розроблена система STO Management System забезпечує базовий набір функцій, необхідних для управління станцією технічного обслуговування, таких як додавання, редагування та видалення інформації про клієнтів, автомобілі, запчастини та записи на обслуговування. Вона відзначається простим та зрозумілим інтерфейсом, що робить її зручною у використанні.

Водночас, система AutoRepair Cloud пропонує більш розширені можливості, такі як інтеграція з іншими системами, мобільний додаток та

хмарне зберігання даних. Це робить її більш гнучкою та зручною для користувачів, які потребують доступу до своїх даних з будь-якого місця та з різних пристроїв.

Обидві системи мають свої переваги та недоліки, тому вибір залежить від конкретних потреб та ресурсів станції технічного обслуговування.

ВИСНОВКИ

У цій роботі було розглянуто впровадження автоматизованої системи управління клієнтами, автомобілями, записами на обслуговування та запасами в умовах автомобільного сервісного центру. Основна мета полягала в підвищенні ефективності роботи, зниженні витрат часу на обробку інформації та мінімізації помилок, що виникають при ручному введенні даних. Нижче наведено ключові результати та висновки.

Ключові результати:

1. Управління клієнтами:

- Реалізовано функції додавання, редагування та видалення клієнтів.
- Забезпечено централізоване зберігання контактної інформації клієнтів, що покращує її доступність та цілісність.

2. Управління автомобілями:

- Впроваджено функції додавання, редагування та видалення інформації про автомобілі клієнтів.
- Зберігання інформації про марку, модель, рік випуску та номерний знак автомобілів забезпечує повний контроль та облік транспортних засобів.

3. Записи на обслуговування:

- Реалізовано можливість додавання, редагування та видалення записів на обслуговування.
- Збереження інформації про дату, опис послуги та статус обслуговування дозволяє тримати в курсі всіх запланованих та виконаних робіт.

4. Управління запасами:

- Впроваджено функції додавання, редагування та видалення запчастин.
- Зберігання інформації про назву запчастини, кількість та ціну дозволяє ефективно управляти запасами та планувати закупівлі.

Нові рішення у вирішенні задачі:

1. Автоматизація процесів:

- Впровадження автоматизованої системи значно знизило витрати часу на обробку інформації.
- Автоматизація допомогла зменшити кількість помилок при ручному введенні даних завдяки валідації даних, автоматичним підказкам та зниженню людського фактора.

2. Поліпшення управління даними:

- Система забезпечила централізоване зберігання даних, що покращило їхню цілісність та доступність.
- Покращення доступності даних було досягнуто завдяки можливості доступу з будь-якого місця, розширеному управлінню правами доступу та миттєвому пошуку інформації.

Впровадження автоматизованої системи управління станцією технічного обслуговування автомобілів, дозволило значно підвищити ефективність роботи, покращити управління даними та зменшити кількість помилок. Ця система забезпечила централізоване зберігання інформації, що покращило її цілісність та доступність для всіх користувачів. Крім того, автоматизація процесів знизила витрати часу на обробку даних та підвищила точність введеної інформації. Таким чином, запровадження цієї системи має значний позитивний вплив на загальну продуктивність та якість послуг, що надаються автомобільним сервісним центром.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сільбершатц, А., Корф, Г. Ф., Сударшан, С. Системи управління базами даних: Концепції. – McGraw-Hill, 2010. – 1234 с.
2. Ульман, Дж. Д., Відом, Дж. Вступ до систем баз даних. – Pearson, 2007. – 987 с.
3. Коннолі, Т., Бегг, К. Системи баз даних: Практичний підхід до проектування, впровадження та управління. – Pearson, 2015. – 756 с.
4. Кодд, Е. Ф. Реляційна модель даних для великих спільних банків даних // Communications of the ACM. – 1970. – Т. 13, № 6. – С. 377-387.
5. Дейт, К. Дж. Вступ до систем баз даних. – Addison-Wesley, 2003. – 564 с.
6. PHP Tutorial // W3Schools. – Режим доступу: <https://www.w3schools.com/php/>. – Дата звернення: 01.06.2024.
7. PHP: Hypertext Preprocessor // PHP Manual. – Режим доступу: <https://www.php.net/manual/en/index.php>. – Дата звернення: 01.06.2024.
8. Як підключитися до бази даних MySQL за допомогою PHP // Stack Overflow. – Режим доступу: <https://stackoverflow.com/questions/46408137/how-to-connect-to-mysql-database-with-php>. – Дата звернення: 01.06.2024.
9. MySQL 8.0 Reference Manual // MySQL Documentation. – Режим доступу: <https://dev.mysql.com/doc/>. – Дата звернення: 01.06.2024.
10. Bootstrap 4.0 Documentation // Bootstrap. – Режим доступу: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>. – Дата звернення: 01.06.2024.
11. Sample Projects and Repositories // GitHub. – Режим доступу: <https://github.com/>. – Дата звернення: 01.06.2024.

12. Learn PHP // Codecademy. – Режим доступу: <https://www.codecademy.com/learn/learn-php>. – Дата звернення: 01.06.2024.
13. Manager SE (Manager Enterprise) // Mitchell 1. – Режим доступу: <https://www.mitchell1.com>. – Дата звернення: 29.05.2024.
14. Shop Boss // Shop Boss. – Режим доступу: <https://www.shopboss.net>. – Дата звернення: 29.05.2024.
15. Програма для обліку автомобілів "АвтоОблік" // AutoSoft. – Режим доступу: <https://autosoft.com.ua/autooblik>. – Дата звернення: 01.06.2024.
16. Mitchell 1 // Mitchell 1. – Режим доступу: <https://www.mitchell1.com>. – Дата звернення: 29.05.2024.
17. AutoFluent // AutoFluent. – Режим доступу: <https://www.autofluent.com>. – Дата звернення: 29.05.2024.
18. AutoRepair Cloud // AutoRepair Cloud. – Режим доступу: <https://www.autorepaircloud.com>. – Дата звернення: 29.05.2024.

ДОДАТКИ

ДОДАТОК А: config.php

```
<?php
```

```
$host = 'localhost';
```

```
$db = 'sto_management_system';
```

```
$user = 'root';
```

```
$pass = '';
```

```
$charset = 'utf8mb4';
```

```
$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
```

```
$options = [
```

```
    PDO::ATTR_ERRMODE          => PDO::ERRMODE_EXCEPTION,
```

```
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
```

```
    PDO::ATTR_EMULATE_PREPARES => false,
```

```
];
```

```
try {
```

```
    $pdo = new PDO($dsn, $user, $pass, $options);
```

```
} catch (\PDOException $e) {
```

```
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
```

```
}
```

```
?>
```

ДОДАТОК Б list_customers.php

```
<?php
```

```
require 'config.php';
```

```
$sql = 'SELECT * FROM customers';
```

```
$stmt = $pdo->query($sql);
```

```
$customers = $stmt->fetchAll();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Список клієнтів</title>
```

```
  <link rel="stylesheet" type="text/css" href="styles.css">
```

```
</head>
```

```
<body>
```

```
  <h1>Список клієнтів</h1>
```

```
  <table border="1">
```

```
    <tr>
```

```
      <th>ID</th>
```

```
      <th>Ім'я</th>
```

```
      <th>Прізвище</th>
```

```
      <th>Телефон</th>
```

```
      <th>Email</th>
```

```
      <th>Дії</th>
```

```
    </tr>
```

```
    <?php foreach ($customers as $customer): ?>
```

```
      <tr>
```

```
        <td><?php echo htmlspecialchars($customer['id']); ?></td>
```

```

<td><?php echo htmlspecialchars($customer['first_name']); ?></td>
<td><?php echo htmlspecialchars($customer['last_name']); ?></td>
<td><?php echo htmlspecialchars($customer['phone']); ?></td>
<td><?php echo htmlspecialchars($customer['email']); ?></td>
<td>
    <form method="POST" action="edit_customer.php">
        <input type="hidden" name="id" value="<?php echo
$customer['id']; ?>">
        <input type="submit" name="edit" value="Редагувати">
    </form>
    <form method="POST" action="delete_customer.php">
        <input type="hidden" name="id" value="<?php echo
$customer['id']; ?>">
        <input type="submit" name="delete" value="Видалити">
    </form>
</td>
</tr>
<?php endforeach; ?>
</table>
</body>
</html>

```

ДОДАТОК В : edit_customer.php

```
<?php
```

```
require 'config.php';
```

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['id'])) {  
    $stmt = $pdo->prepare('SELECT * FROM customers WHERE id = ?');  
    $stmt->execute([\$_POST['id']]);  
    $customer = $stmt->fetch();  
}
```

```
if      ($_SERVER['REQUEST_METHOD']      ===      'POST'      &&  
isset($_POST['update'])) {  
    $stmt = $pdo->prepare('UPDATE customers SET first_name = ?, last_name =  
?, phone = ?, email = ? WHERE id = ?');  
    $stmt->execute([\$_POST['first_name'],                \$_POST['last_name'],  
\$_POST['phone'], \$_POST['email'], \$_POST['id']]);  
    header('Location: list_customers.php');  
}  
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Редагування клієнта</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Редагування клієнта</h1>
```

```
    <form method="POST">
```



```
<input type="hidden" name="id" value="<?php echo
htmlspecialchars($customer['id']); ?>">
<label>Ім'я: <input type="text" name="first_name" value="<?php echo
htmlspecialchars($customer['first_name']); ?>"></label><br>
<label>Прізвище: <input type="text" name="last_name" value="<?php
echo htmlspecialchars($customer['last_name']); ?>"></label><br>
<label>Телефон: <input type="text" name="phone" value="<?php echo
htmlspecialchars($customer['phone']); ?>"></label><br>
<label>Email: <input type="text" name="email" value="<?php echo
htmlspecialchars($customer['email']); ?>"></label><br>
<input type="submit" name="update" value="Оновити">
</form>
</body>
</html>
```