

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**Факультет економіки і підприємництва**

**Кафедра інформаційних технологій**

**Допущено до захисту**

**" \_\_\_\_\_ " \_\_\_\_\_ 2025 р.**

**Зав. кафедри, к.т.н., доцент**

**\_\_\_\_\_ Ліщук Р.І.**

**КВАЛІФІКАЦІЙНА РОБОТА**  
**БАКАЛАВРА**

**Розробка інформаційної підсистеми для прогнозування та  
планування проведення технічного обслуговування  
автомобілів**

**Виконав - студент 4 курсу, групи 41-кн  
освітньої програми «Комп'ютерні науки»  
спеціальності 122 Комп'ютерні науки**

**Клімович Максим Віталійович**

**Науковий керівник - д.т.н., професор**

**Кулаков Павло Ігорович**

**Умань - 2025**

## Зміст

|                                                                                                                        |    |
|------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....                                                                   | 8  |
| ВСТУП.....                                                                                                             | 9  |
| РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....                                                                    | 11 |
| 1.1. Об'єкт та предмет дослідження.....                                                                                | 11 |
| 1.2. Особливості програм для прогнозування технічного обслуговування автомобілів на СТО.....                           | 13 |
| 1.3. Постановка задачі.....                                                                                            | 16 |
| РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ.....                           | 17 |
| 2.1. Аналіз вимог користувача до підсистеми прогнозування технічного обслуговування автомобілів на основі пробігу..... | 17 |
| 2.2. Математична постановка задачі.....                                                                                | 19 |
| 2.3. Вибір та обґрунтування програмних засобів.....                                                                    | 20 |
| 2.4. Архітектура підсистеми прогнозування технічного обслуговування.....                                               | 26 |
| 2.5. Алгоритм функціонування алгоритму ТО.....                                                                         | 29 |
| РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОДУЛІВ ПІДСИСТЕМИ АВТОМАТИЗОВАНОГО СКЛАДСЬКОГО ОБЛІКУ .....               | 32 |
| 3.1. Структура бази даних.....                                                                                         | 32 |
| 3.2. Інтерфейс користувача.....                                                                                        | 35 |
| 3.3. Реалізація обробки даних в базі даних.....                                                                        | 40 |
| 3.4. Прогнозування та вивід повідомлень.....                                                                           | 40 |
| ВИСНОВОК.....                                                                                                          | 44 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....                                                                                    | 45 |

## Вступ

Сучасні вимоги до інформаційних систем управління підприємствами передбачають поєднання класичних функціональних можливостей з новітніми технологічними підходами, серед яких використання реляційних баз даних, архітектурних рішень клієнт-серверних систем, сумісність та інтеграція з існуючими програмними й апаратними платформами.

Велике значення у сфері оптимізації бізнес-процесів мають системи планування ресурсів підприємства (ERP – Enterprise Resource Planning), які забезпечують комплексне управління діяльністю організацій. Структурно такі системи складаються з основного функціоналу, засобів керування даними та спеціалізованих модулів, що забезпечують повний цикл роботи підприємства. ERP-системи автоматизують облік ресурсів, планування виробничих процесів, логістики, а також управління взаємодією з клієнтами. У той же час, для підприємств, пов'язаних з наданням послуг з технічного обслуговування транспортних засобів, важливим є впровадження підсистем, які дозволяють прогнозувати технічні роботи на основі фактичного пробігу автомобілів.

На ринку існує низка ERP-рішень, таких як Acumatica ERP, Oracle ERP, Sage X3, Unit4, які, хоча й мають потужний функціонал, часто орієнтовані на великі корпорації та не враховують специфіку малого та середнього бізнесу в сфері автосервісу. Їх висока вартість, складність інтеграції та обмежена адаптивність спонукають до розробки більш вузькоспеціалізованих рішень.

**Актуальність** теми кваліфікаційної роботи бакалавра зумовлена необхідністю впровадження сучасних інформаційних технологій у сферу обслуговування транспортних засобів. Попри наявність комплексних ERP-рішень та спеціалізованих програм для автосервісу, таких як 1С:Управління автосервісом, Autosoft, CarSys чи Bosch Car Service Software, більшість із них не забезпечують гнучкого та адаптивного прогнозування технічного обслуговування на основі аналізу реального пробігу автомобіля. Це обмеження обумовлює необхідність розробки спеціалізованих підсистем, які дозволяють

більш точно та своєчасно планувати обслуговування транспортних засобів, враховуючи індивідуальні особливості їх експлуатації.

**Об'єкт дослідження** - процес автоматизації прогнозування технічного обслуговування транспортних засобів на станціях технічного обслуговування.

**Предмет дослідження** - програмна реалізація інформаційної підсистеми прогнозування технічного обслуговування автомобілів на основі аналізу їх пробігу.

### **Мета дослідження**

Метою кваліфікаційної роботи бакалавра є розробка інформаційної підсистеми для прогнозування технічного обслуговування транспортних засобів на основі аналізу пробігу автомобіля з метою підвищення ефективності роботи станції технічного обслуговування.

Завданням кваліфікаційної роботи бакалавра є:

- проаналізувати підходи до автоматизації процесів технічного обслуговування транспортних засобів у системах управління автосервісами;
- здійснити аналіз та вибір інформаційних технологій для реалізації підсистеми прогнозування технічного обслуговування на основі пробігу;
- розробити програмне забезпечення модулів інформаційної підсистеми прогнозування технічного обслуговування автомобілів.

## РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Об'єкт та предмет дослідження

В умовах стрімкого розвитку автомобільного транспорту та збільшення кількості автотранспорту, що перебуває в експлуатації, особливого значення набуває питання ефективної організації технічного обслуговування (ТО) транспортних засобів. Надійність, безпека, економічність експлуатації автомобілів значною мірою залежать від своєчасного та якісного виконання ТО. Для підприємств, що надають послуги у сфері обслуговування автомобілів (СТО), постає завдання не лише оперативного реагування на поточні заявки, а й ефективного прогнозування майбутніх потреб в обслуговуванні.

Класичні підходи до ТО передбачають виконання робіт за фіксованими міжсервісними інтервалами, встановленими виробниками, однак вони не враховують реальних умов експлуатації автомобіля. Наприклад, інтенсивність використання, стиль водіння, кліматичні умови чи дорожня обстановка мають прямий вплив на зношення агрегатів, але залишаються поза увагою в стандартних схемах обслуговування. Це призводить до того, що одні автомобілі обслуговуються занадто часто, що економічно невигідно, а інші - несвоєчасно, що підвищує ризик поломок.

Відповіддю на цю проблему є створення інформаційних підсистем прогнозування ТО на основі аналізу пробігу транспортних засобів. Такі системи дозволяють більш точно планувати обслуговування, оптимізувати витрати підприємства, забезпечувати безпеку і комфорт клієнтів, а також підвищувати якість обслуговування.

На сучасному етапі автоматизації бізнес-процесів у сфері обслуговування транспорту, поряд з глобальними ERP-системами, зростає потреба у спеціалізованих рішеннях для СТО. Відомі ERP-системи (Acumatica ERP, Oracle ERP, Sage X3, 1С:Управління автосервісом) надають базовий функціонал для ведення обліку клієнтів, управління запасами, фінансами та документообігом. Проте, їх функціонал часто є універсальним і не враховує специфіку роботи

автосервісів з точки зору прогнозування технічного обслуговування на основі фактичного пробігу.

Існують також окремі галузеві рішення, такі як Autosoft, CarSys, Bosch Car Service Software, які частково автоматизують процеси обслуговування. Вони дозволяють вести облік виконаних робіт, формувати нагадування про планове ТО за фіксованими інтервалами, але їх можливості в частині динамічного прогнозування потребують доповнення. Зокрема, більшість таких систем орієнтуються на статичні графіки обслуговування, не враховуючи індивідуальні показники пробігу та умов експлуатації конкретного транспортного засобу.

Об'єктом дослідження даної кваліфікаційної роботи є процес автоматизації прогнозування технічного обслуговування автомобілів на основі пробігу у рамках діяльності станції технічного обслуговування. Предметом дослідження є програмна реалізація інформаційної підсистеми, що дозволяє здійснювати аналіз даних про пробіг автомобілів, формувати персоналізовані рекомендації щодо ТО та інтегруватися в інформаційну систему СТО.

Архітектура такої підсистеми передбачає три ключових рівні:

- Рівень бази даних: відповідальний за зберігання історії пробігу, виконаних робіт, параметрів транспортних засобів, обліку клієнтів та ресурсів;
- Аналітичний рівень: реалізує алгоритми аналізу пробігу, визначення оптимальних термінів ТО, побудову прогнозів на основі історичних даних, середніх навантажень, умов експлуатації;
- Рівень користувацького інтерфейсу: забезпечує доступ до функціоналу системи для працівників СТО, відображення прогнозів, формування графіків робіт, інтерактивну взаємодію з клієнтами (нагадування, повідомлення).

Важливою характеристикою підсистеми є її гнучкість та можливість адаптації під потреби конкретного підприємства, що передбачає можливість налаштування алгоритмів прогнозування, масштабування та інтеграції з іншими інформаційними системами.

Реалізація підсистеми здійснюється із застосуванням мови програмування

PHP, локального серверного середовища XAMPP та стандартних веб-технологій HTML і CSS. Таке технологічне рішення обумовлене його доступністю, простотою впровадження та підтримки, а також широкими можливостями інтеграції у корпоративне середовище.

Використання підсистеми прогнозування ТО дозволяє підприємствам досягти наступних переваг:

1. зниження експлуатаційних витрат за рахунок оптимізації обслуговування;
2. підвищення рівня безпеки та надійності транспортних засобів;
3. поліпшення якості клієнтського сервісу;
4. зменшення кількості аварійних ситуацій;
5. автоматизація планування робіт та обліку ресурсів;
6. покращення внутрішнього управління на СТО.

Попри очевидні переваги, впровадження таких рішень також супроводжується низкою викликів, серед яких можна виділити необхідність інтеграції з наявними системами обліку, забезпечення захисту даних, навчання персоналу та технічну підтримку. Проте ці труднощі є вирішуваними завдяки грамотному проектуванню та розробці програмного забезпечення.

Таким чином, розробка інформаційної підсистеми прогнозування технічного обслуговування транспортних засобів на основі аналізу пробігу є актуальною задачею, яка дозволяє підвищити ефективність діяльності станцій технічного обслуговування, забезпечити економічну доцільність процесів обслуговування та покращити якість послуг для клієнтів.

## **1.2 Особливості програм для прогнозування технічного обслуговування автомобілів на СТО**

Зростаючий попит на якісне обслуговування транспортних засобів, підвищення конкуренції серед станцій технічного обслуговування (СТО), а також необхідність оптимізації витрат формують потребу у впровадженні сучасних інформаційних систем. Одним із перспективних напрямів є

автоматизація процесів прогнозування технічного обслуговування (ТО) на основі реальних даних про пробіг та умови експлуатації транспортних засобів.

На сьогодні більшість рішень для автоматизації роботи СТО забезпечують лише базовий функціонал: облік клієнтів, замовлень, складських залишків та фінансів. Функціонал динамічного прогнозування ТО на основі пробігу реалізовано недостатньо або взагалі відсутній.

«1С: Управління автосервісом» є найбільш поширеним рішенням серед українських СТО завдяки своїй доступності та широкому функціоналу. Програма дозволяє вести облік клієнтів, історії обслуговування, управляти залишками на складі, формувати фінансову звітність. Проте прогнозування ТО обмежується фіксованими інтервалами за регламентами, без урахування реального пробігу транспортного засобу чи особливостей його експлуатації. Це знижує ефективність планування та може призводити до зайвих витрат або несвоєчасного обслуговування.

Autosoft - хмарна система управління автосервісами, орієнтована на малий та середній бізнес. Вона інтегрує CRM, складський облік, фінанси та документообіг в єдину платформу. До особливостей Autosoft належить можливість доступу з будь-якого пристрою через браузер, автоматизація робочих процесів, а також інтеграція з платіжними системами. Проте, як і в 1С, прогнозування ТО засноване на статичних інтервалах, без динамічного аналізу пробігу автомобіля.

Bosch Car Service Software - професійне рішення, що поєднує функціонал обліку з інтеграцією діагностичного обладнання Bosch. Програма забезпечує автоматизоване створення заявок на ремонт, доступ до технічної документації Bosch, облік витратних матеріалів. Втім, прогнозування ТО в цій системі обмежується лише стандартними регламентами. Вартість ліцензії та необхідність використання сертифікованого обладнання є додатковими факторами, які знижують доцільність її впровадження для більшості українських СТО.

Cartrack, Samsara, Fleet Complete - це телематичні платформи для управління автопарками, які дозволяють відстежувати пробіг у реальному часі, контролювати витрати палива, здійснювати дистанційну діагностику технічного стану авто та будувати аналітичні звіти. Попри потужний функціонал, такі системи є надмірно дорогими для невеликих СТО та мають складну архітектуру, орієнтовану на корпоративний сектор.

Програми типу Simply Auto, Drivvo орієнтовані на приватних автовласників. Вони дозволяють вручну вести облік пробігу, фіксувати витрати на обслуговування, отримувати нагадування про планові ТО. Проте для використання у бізнес-процесах СТО ці рішення не підходять через відсутність інтеграції з обліковими системами та обмежений функціонал.

Інформаційна система «ПлазмІС», розроблена для ПрАТ «ПлазмаТек», є прикладом модульної ERP-системи, яка враховує специфіку підприємства. Вона включає облік виробничих потужностей, управління складом, кадровий облік та документообіг. Особливістю «ПлазмІС» є можливість додавання нових модулів та адаптації під потреби замовника. Цей підхід можна застосувати і для розробки підсистеми прогнозування ТО на СТО, забезпечивши гнучкість, масштабованість та економічну доцільність.

Аналіз ринку дозволяє виділити три основні категорії рішень. По-перше, ERP-рішення з модулем автосервісу, які є універсальними, але обмежені у функціоналі прогнозування ТО (1С, Autosoft). По-друге, професійні системи, інтегровані з обладнанням, які є потужними, але дорогими і складними (Bosch Car Service Software). По-третє, IoT-рішення для автопарків з високим рівнем аналітики, які недосяжні для малого бізнесу (Cartrack, Samsara).

Водночас відсутні рішення середнього рівня, які б враховували пробіг і умови експлуатації, мали адекватну вартість для невеликих СТО та інтегрувалися з іншими модулями інформаційної системи.

До основних проблем сучасних рішень належать відсутність аналізу динамічного пробігу, обмеження по інтеграції, недостатня адаптація під ринок

України, висока вартість впровадження (особливо Bosch та IoT-системи), а також статичне планування без врахування індивідуальних даних.

Потреба у розробці спеціалізованої підсистеми прогнозування технічного обслуговування автомобілів на основі аналізу пробігу є очевидною. Власна розробка дозволить врахувати особливості діяльності конкретного СТО, забезпечити гнучкість налаштувань, інтеграцію з наявними обліковими системами, та при цьому залишитися економічно доцільною.

### **1.3 Постановка задачі**

Метою кваліфікаційної роботи бакалавра є розширення функціональних можливостей інформаційної системи станції технічного обслуговування шляхом розробки інформаційної підсистеми прогнозування технічного обслуговування автомобілів на основі аналізу їхнього пробігу.

Завданням розробки є аналіз підходів до автоматизації процесів технічного обслуговування транспортних засобів в інформаційних системах СТО, аналіз та вибір технологій для реалізації інформаційної підсистеми прогнозування ТО, а також розробка програмного забезпечення модуля, який дозволяє здійснювати аналіз пробігу автомобілів та формувати персоналізовані рекомендації щодо проведення технічного обслуговування.

Запропоноване рішення передбачає реалізацію таких функціональних можливостей, як зберігання та обробка даних про пробіг транспортних засобів, аналіз динаміки пробігу з урахуванням реальних умов експлуатації, формування прогнозів потреби в технічному обслуговуванні для кожного автомобіля, інтеграція з існуючими модулями інформаційної системи СТО, а також забезпечення доступного та зручного інтерфейсу для користувачів.

## **РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ ПІДСИСТЕМИ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ**

### **2.1 Аналіз вимог користувача до підсистеми прогнозування технічного обслуговування автомобілів на основі пробігу**

Інформаційні системи у сфері обслуговування транспорту повинні забезпечувати не лише облік виконаних операцій, але й допомагати приймати рішення щодо планування технічного обслуговування (ТО) на основі реальних даних. Актуальність створення підсистеми прогнозування ТО полягає у підвищенні ефективності діяльності станцій технічного обслуговування, зменшенні простоїв автотранспорту та запобіганні аварій через несвоєчасне обслуговування.

Змістовна постановка задачі базується на практичній потребі забезпечити підтримку прийняття рішень щодо планування ТО транспортного засобу, використовуючи історичні дані про пробіг і частоту обслуговування. Основними функціями розроблюваної підсистеми є: збирання та оновлення даних про пробіг автомобілів, аналіз динаміки змін пробігу, формування прогнозів щодо майбутніх ТО, збереження результатів у базі даних, а також відображення цих прогнозів у зручному інтерфейсі.

Умови функціонування підсистеми передбачають її інтеграцію до вже існуючої інформаційної системи СТО. Вона повинна відповідати низці вимог, зокрема:

- відображення актуального стану технічного обслуговування кожного автомобіля;
- можливість візуалізації графіка планових ТО;
- автоматичне формування повідомлень про наближення ТО;
- збереження історії обслуговування та пробігу.

Діалог між користувачем і підсистемою організується через веб-інтерфейс, розроблений із використанням HTML/CSS, PHP та JavaScript.

Інтерфейс має бути зручним як для майстра, так і для адміністратора, з можливістю швидкого пошуку автомобіля за VIN-кодом або маркою. Для локального розгортання застосовується XAMPP, що забезпечує функціонування системи в умовах невеликого СТО без необхідності додаткових витрат на серверне обладнання.

Вхідна інформація включає:

- реєстраційні дані автомобіля;
- поточний пробіг;
- історію технічного обслуговування (дата, тип робіт, пробіг на момент ТО).

Вихідна інформація:

- повідомлення про майбутнє обслуговування;
- зведений графік стану автопарку.

Форма подання результатів – таблична або у вигляді картки транспортного засобу. Частота оновлення прогнозу – кожне внесення або зміну пробігу. Періодичність повідомлень – щоденно при вході користувача до системи.

Система має забезпечити надійне зберігання та захист даних, а також можливість ручного коригування прогнозів або пробігу у разі потреби. Передбачено базовий контроль коректності введених даних та ведення логів змін.

Обмеження на час виконання задачі визначаються очікуванням відгуку користувача - оновлення даних і відображення прогнозу повинні здійснюватися в межах 1–2 секунд. Інформація має бути доступною цілодобово в межах локальної мережі.

Підсистема має враховувати можливість масштабування та доопрацювання - наприклад, додавання функції підключення до телематичних систем (GPS, OBD2) у майбутньому. Усі рішення в процесі проєктування базуються на сучасних принципах розробки інформаційних систем, відкритих технологіях і принципах модульності.

## **2.2. Математична постановка задачі**

Підсистема прогнозування технічного обслуговування (ТО) в рамках CRM-системи для СТО створена з метою визначення ймовірності необхідності певного виду сервісних робіт для кожного автомобіля клієнта. Основна мета — вчасно попереджати користувача про потенційне технічне обслуговування, що дозволяє підвищити безпеку, надійність експлуатації транспортного засобу та ефективність роботи СТО.

### Вихідні параметри

Для кожного автомобіля та кожного виду ремонту використовується низка параметрів:

$M_{now}$  — поточний пробіг автомобіля (в км),

$M_{last}$  — пробіг при останньому проведенні відповідного ремонту,

$M_{norm}$  — нормативний пробіг між обслуговуванням,

$T_{since}$  — кількість місяців з останнього обслуговування цього типу,

$T_{norm}$  — нормативний інтервал між обслуговуванням за часом (в місяцях),

$A$  — вік автомобіля у роках (визначається як різниця між поточним роком та роком випуску),

$E$  — поправочний коефіцієнт залежно від типу двигуна:

бензиновий —  $E = 0$ ,

дизельний —  $E = 0,03$ ,

електричний —  $E = -0,03$ , (через менше навантаження на гальма, мастила та інші класичні системи).

### Побудова моделі ризику

Оскільки ризик зростає нерівномірно з часом і пробігом, доцільним є використання експоненційної моделі. Вона враховує, що на початкових етапах відхилення від нормативу не критичні, але з часом небезпека суттєво зростає:

$$R_{km} = 1 - e^{-k_1 \cdot r_{km}}, R_{time} = 1 - e^{-k_2 \cdot r_{time}}$$

де:

$k_1 = 2.0$  — коефіцієнт чутливості до пробігу,

$k_2 = 1.5$  — коефіцієнт чутливості до часу,

Ці значення визначено експериментальним шляхом на основі аналізу різних сценаріїв.

### **Підсумкова формула ризику**

Інтегральна оцінка ризику обчислюється за зваженою сумою:

$$R = 0.55 \cdot R_{km} + 0.35 \cdot R_{time} + 0.01 \cdot A + E$$

Це дозволяє збалансовано враховувати всі ключові фактори. Отримане значення нормалізується до діапазону  $[0, 1]$ , після чого множиться на 100 для зручності виводу у вигляді відсотків. Наприклад, значення  $R = 0.68$  буде представлено як **68% ризику** необхідності ТО.

### **Додаткові умови:**

- Якщо дані про попереднє обслуговування відсутні, система імітує, що обслуговування не виконувалося давно (наприклад, пробіг зменшується на 80% від нормативу), щоб уникнути помилкової впевненості.

- Якщо ризик нижче порогового значення (30%), він вважається несуттєвим і не відображається для користувача.

Запропонована модель легко адаптується під різні види ремонту або категорії автомобілів. Також можливе динамічне налаштування коефіцієнтів на основі зібраної статистики реальних поломок чи обслуговувань.

## **2.3 Вибір та обґрунтування програмних засобів**

У процесі розробки підсистеми прогнозування технічного обслуговування автомобілів ключовим етапом стало визначення інструментів, які найкраще відповідають технічним вимогам, обмеженням та можливостям майбутніх користувачів. Особливістю цього проєкту є спрямованість на малий та середній бізнес, зокрема на СТО, які не завжди мають доступ до висококваліфікованих ІТ-фахівців або спеціального серверного обладнання. Тому особливий акцент робився на простоту впровадження, локальне використання, відкритість технологій та подальшу підтримуваність проєкту.

Інформаційна система проєкту має класичну багаторівневу архітектуру: інтерфейс користувача (frontend), серверна логіка (backend), система збереження даних (база даних). Для кожного з цих рівнів було підібрано оптимальний набір технологій, що відповідають сучасним стандартам веб-розробки.

Інтерфейс користувача у розроблюваній інформаційній системі побудований на основі сучасних веб-технологій, зокрема HTML5, CSS3 та JavaScript. Цей набір технологій обрано через їхню широковживаність, сумісність з різними пристроями та можливість створення зручного, інтуїтивно зрозумілого і адаптивного користувацького досвіду.

HTML5 - це остання версія стандарту розмітки веб-сторінок, яка відповідає за структуру й семантику контенту. Завдяки новим семантичним тегам (наприклад, <header>, <footer>, <section>, <article>), HTML5 значно покращує доступність веб-ресурсів та полегшує роботу пошукових систем і допоміжних технологій. Цей стандарт підтримується всіма сучасними браузерами, що гарантує коректне відображення інтерфейсу незалежно від платформи користувача. Крім того, HTML5 включає в себе розширені можливості для роботи з мультимедійним контентом та інтерактивністю без необхідності встановлення додаткових плагінів.

CSS3 використовується для стилізації веб-сторінок і визначає візуальний вигляд інтерфейсу. Важливою перевагою CSS3 є підтримка адаптивного дизайну (responsive design), що дозволяє автоматично підлаштовувати розташування елементів і їхні розміри під різні розміри екрану — від великих моніторів до смартфонів і планшетів. Це забезпечує комфортне користування системою на різноманітних пристроях. За допомогою CSS3 можна задавати кольорові схеми, шрифти, тіні, анімації, переходи та інші візуальні ефекти, що покращують естетику і зручність інтерфейсу.

JavaScript відповідає за динамічну поведінку інтерфейсу і взаємодію користувача з системою в реальному часі. Завдяки JavaScript інтерфейс стає інтерактивним - він може реагувати на дії користувача (клацання, наведення,

введення тексту) без потреби перезавантаження сторінки. Для спрощення розробки та підвищення стабільності можна використовувати різні популярні бібліотеки, такі як jQuery, які надають готові інструменти для маніпуляції DOM, обробки подій і реалізації асинхронних запитів.

Застосування AJAX (Asynchronous JavaScript and XML) дає змогу обмінюватися даними з сервером у фоновому режимі, що значно покращує продуктивність і швидкодію системи. Користувачі отримують відчуття миттєвого відгуку на свої дії, оскільки не потрібно чекати повного оновлення сторінки при кожній взаємодії. Це особливо важливо для систем, що вимагають активної роботи з даними в реальному часі.

Для реалізації серверної логіки було обрано мову програмування PHP. Цей вибір зумовлений кількома чинниками. PHP (Hypertext Preprocessor) - це мова сценаріїв загального призначення, спеціально створена для веб-розробки і широко застосовувана для розробки динамічних веб-сторінок та серверних додатків. Основна причина вибору PHP полягає в його поширеності, простоті, гнучкості та широкій підтримці в індустрії.

PHP є інтерпретованою мовою, що дозволяє швидко запускати і тестувати код без необхідності компіляції. Завдяки своїй архітектурі PHP легко інтегрується з HTML, CSS і JavaScript, що робить його ідеальним для створення серверної логіки, яка тісно взаємодіє з клієнтським інтерфейсом. Крім того, PHP підтримує роботу з великою кількістю баз даних, включаючи MySQL, PostgreSQL, SQLite, що є важливим для реалізації зберігання, пошуку і обробки даних у веб-додатку.

Важливою перевагою PHP є його відкритий код та величезна спільнота розробників, які підтримують і розвивають мову, що забезпечує доступність великої кількості бібліотек, фреймворків і готових рішень. Серед найбільш популярних фреймворків для PHP - Laravel, Symfony, CodeIgniter, які дозволяють значно прискорити процес розробки та покращити організацію коду. В даному

проекті, враховуючи простоту та масштаби задачі, застосування фреймворку було не обов'язковим, проте можливе у разі подальшого розвитку системи.

З погляду продуктивності, PHP показує високу швидкість обробки запитів завдяки оптимізованим механізмам кешування, підтримці багатопотокової роботи на сервері та постійному вдосконаленню мови. Вона також сумісна з більшістю серверних операційних систем: Windows, Linux, MacOS, що забезпечує гнучкість у виборі інфраструктури.

PHP підтримує сесійну роботу і механізми безпеки, такі як шифрування, захист від SQL-ін'єкцій, XSS, CSRF-атак, що дозволяє забезпечити безпечну обробку користувацьких даних і підвищує довіру до системи.

PHP добре підходить для малих і середніх проєктів, адже не потребує складного налаштування серверного середовища і може працювати на звичайному персональному комп'ютері або локальному сервері. У поєднанні з інструментами, які входять до складу XAMPP, PHP дозволяє швидко розгорнути повноцінний сервер, що важливо для організацій без власної IT-інфраструктури.

Для організації локального середовища розробки було обрано пакет XAMPP, який включає Apache HTTP Server, MySQL (MariaDB), PHP та Perl. XAMPP — це популярний кросплатформенний інструмент, який надає повний набір необхідних компонентів для запуску веб-сервера локально на комп'ютері розробника.

Вибір XAMPP пояснюється кількома ключовими чинниками:

- Легка установка та налаштування. XAMPP забезпечує швидке і зручне розгортання веб-сервера і бази даних, не вимагаючи складних конфігурацій чи додаткових пакетів. Це дозволяє почати розробку майже миттєво, що особливо важливо на початкових етапах проєкту.

- Всі необхідні компоненти в одному пакеті. Apache служить веб-сервером для обробки HTTP-запитів, MySQL - для зберігання та керування даними, PHP - для реалізації серверної логіки. Наявність всього цього під одним інтерфейсом спрощує робочий процес.

- Кросплатформеність. ХАМРР працює на Windows, Linux та MacOS, що дозволяє використовувати однакове середовище розробки на різних пристроях і системах без додаткових змін у конфігурації.

- Активна спільнота та підтримка. На офіційному сайті та численних форумах доступна велика кількість навчальних матеріалів, інструкцій та порад, що допомагає розв'язувати технічні питання швидко та ефективно.

- Відкритий код та безкоштовність. ХАМРР є безкоштовним і відкритим програмним продуктом, що відповідає вимогам бюджетних проектів та не створює додаткових фінансових навантажень.

- Гнучкість у розгортанні. Хоча ХАМРР призначений для локальної розробки, його компоненти можуть бути використані і в продакшн-середовищі, а розроблені на ньому додатки можна легко перенести на віддалені сервери, які підтримують PHP та MySQL.

Для розробки серверної частини системи були сформовані такі вимоги:

- Надійність та стабільність роботи. Серверне ПЗ повинно коректно обробляти запити, стабільно працювати під навантаженням та не допускати збоїв.

- Забезпечення безпеки даних. Захист від несанкціонованого доступу, верифікація користувачів, захист від типових веб-атак.

- Продуктивність. Швидка обробка запитів та мінімальні затримки у взаємодії з базою даних.

- Легкість налаштування та розгортання. Можливість швидко змінювати конфігурації та запускати серверне середовище.

- Масштабованість. Система повинна мати потенціал для подальшого розвитку та розширення функціоналу.

- Підтримка інтеграції з іншими сервісами. Можливість реалізації API, імпорту/експорту даних, взаємодії із зовнішніми системами.

Враховуючи вищезазначені вимоги, поєднання PHP та ХАМРР цілком відповідає поставленим завданням. PHP забезпечує гнучкість та

функціональність серверної логіки, а XAMPP створює комфортне середовище для розробки, тестування та локального розгортання додатку.

Одним із ключових компонентів будь-якої інформаційної системи є система управління базами даних (СУБД), яка відповідає за організацію збереження, пошуку та обробки даних. Для реалізації зберігання інформації в рамках цього проєкту було обрано реляційну СУБД MySQL.

MySQL є однією з найпопулярніших та найпоширеніших реляційних систем управління базами даних у світі. Вона підтримує стандарт мови структурованих запитів SQL (Structured Query Language), що дозволяє ефективно працювати з різноманітними типами структурованих даних, здійснювати пошук, фільтрацію, оновлення та агрегацію інформації.

Вибір MySQL обґрунтовується кількома ключовими факторами. По-перше, це висока продуктивність і стабільність роботи, які забезпечують швидкий доступ до даних навіть при великому навантаженні. По-друге, MySQL має розвинений механізм забезпечення цілісності даних, зокрема, підтримку транзакцій, зовнішніх ключів, індексів та обмежень, що дозволяє уникати некоректних або неповних записів у базі даних.

Крім того, MySQL відзначається широкою підтримкою та великою спільнотою розробників, що забезпечує регулярні оновлення, покращення безпеки та наявність багатьох готових інструментів і бібліотек для інтеграції з іншими програмними засобами.

Важливою перевагою MySQL є її гнучкість у налаштуванні і можливість масштабування, що дозволяє адаптувати систему до зростаючих обсягів даних та розширення функціоналу проєкту.

Серед альтернатив, які були розглянуті, варто відзначити Python із Django, Node.js, а також CMS-системи. Django як потужний фреймворк для Python надає великі можливості, але потребує складнішого налаштування, наявності рір-середовища, віртуальних середовищ та розгортання через WSGI. Node.js працює з JavaScript і дозволяє створювати бекенд на тій же мові, що й frontend, проте

вимагає налаштування середовища, npm, окремих фреймворків, таких як Express. Усі ці варіанти вимагають більшої кількості кроків для налаштування, що ускладнює впровадження в умовах малого СТО. Саме тому було обрано перевірене, просте у використанні та достатньо гнучке рішення - PHP + MySQL + XAMPP.

## **2.4 Архітектура підсистеми прогнозування технічного обслуговування**

Розробка інформаційної підсистеми прогнозування технічного обслуговування (ТО) транспортних засобів потребує ретельно продуманої архітектури, яка б забезпечувала ефективну взаємодію користувача з системою, обробку запитів та зберігання інформації.

Розроблена інформаційна підсистема прогнозування технічного обслуговування транспортних засобів реалізована відповідно до трирівневої архітектурної моделі, яка забезпечує гнучкість, надійність та легкість у подальшій підтримці й масштабуванні системи. Такий підхід дозволяє логічно розділити функціональні обов'язки між трьома основними рівнями: інтерфейсом користувача (frontend), серверною логікою (backend) та базою даних (database). Кожен з цих рівнів має свою зону відповідальності та реалізований з використанням відповідних технологій.

### **1. Клієнтський рівень (Frontend)**

Клієнтська частина реалізована за допомогою сучасних вебтехнологій - HTML5, CSS3 та JavaScript. Для покращення інтерактивності використовується бібліотека jQuery, яка значно спрощує обробку подій, маніпуляції з DOM, а також роботу з AJAX-запитами.

Інтерфейс спроектовано таким чином, щоби бути максимально зрозумілим для кінцевого користувача - працівника СТО, менеджера автопарку або адміністратора. Сторінки мають логічно організовану структуру, завдяки чому користувач швидко орієнтується в основних функціях системи.

Основний функціонал клієнтського рівня включає:

- введення та редагування інформації про автомобілі;
- заповнення форм прогнозування ТО;
- перегляд історії ремонтів;
- відображення майбутніх ТО на основі обчислень;
- інформування користувача про заплановані роботи через спливаючі повідомлення або змінені елементи інтерфейсу.

Використання AJAX дозволяє значно зменшити навантаження на сервер і зробити взаємодію із системою швидкою та безперервною - дані оновлюються динамічно без повного перезавантаження сторінки.

## 2. Серверний рівень (Backend)

Серверна частина побудована на мові програмування PHP, що широко застосовується у веб-розробці завдяки своїй простоті та широкому набору бібліотек. Саме тут зосереджена уся бізнес-логіка підсистеми.

Ключові завдання, які виконує серверна частина:

- обробка даних, надісланих із клієнтської частини (в тому числі перевірка на коректність і безпеку);
- реалізація алгоритму прогнозування технічного обслуговування на основі середнього пробігу, інтервалів обслуговування та історичних даних;
- аутентифікація та авторизація користувачів системи з урахуванням ролей (адміністратор, механік тощо);
- створення, оновлення та видалення записів у базі даних;
- логування подій та важливих змін.

Сервер обробляє HTTP-запити, отримані з клієнтської частини, формує SQL-запити до бази даних, виконує розрахунки й повертає результати назад у браузер.

## 3. Рівень бази даних (Database)

Для зберігання всієї інформації використовується реляційна система керування базами даних MySQL, що зарекомендувала себе як стабільне та продуктивне рішення. Структура бази даних сформована з урахуванням

принципів нормалізації до третьої нормальної форми, що дозволяє уникнути надлишковості даних і забезпечити логічні зв'язки між сутностями.

База даних містить такі основні таблиці:

- users - містить дані користувачів (ім'я, email, роль, пароль), забезпечуючи контроль доступу;
- cars - зберігає інформацію про транспортні засоби (марка, модель, рік, VIN, пробіг тощо);
- repair\_types - визначає типи обслуговування з прив'язкою до рекомендованих інтервалів;
- repairs - фіксує заплановані ТО, пов'язані з певним автомобілем;
- service\_history - зберігає фактичні дані про виконані ремонти, включаючи дату, вартість, опис та пробіг.

Усі таблиці мають первинні ключі, індекси для прискорення пошуку, а також зовнішні ключі для забезпечення цілісності даних. Зовнішні ключі забезпечують зв'язок між таблицями, зокрема user\_id, car\_id, repair\_type\_id, що дозволяє будувати комплексні запити для відображення стану автопарку, індивідуальних прогнозів, історії обслуговування.

Система локально розгортається за допомогою середовища ХАМРР, яке включає:

- Apache - веб-сервер для обробки HTTP-запитів;
- PHP - мова програмування для серверної логіки;
- MySQL - СУБД для зберігання інформації;
- phpMyAdmin - зручний графічний інтерфейс для управління базою даних.

Такий підхід дозволяє швидко протестувати й запустити систему на локальному комп'ютері без додаткових витрат на хостинг. Це ідеальний варіант для невеликих СТО, які хочуть використовувати програмне забезпечення локально без залежності від інтернет-з'єднання.

Можливості подальшого розвитку:

Архітектура підсистеми спроектована з урахуванням майбутнього розширення. У разі потреби вона може бути модифікована, наприклад:

- реалізація REST API для доступу до даних із мобільного додатку або зовнішніх систем;
- перенесення бази даних на віддалений сервер або в хмару;
- підключення телематичних датчиків для автоматичного збору пробігу;
- інтеграція із сервісами push-сповіщень для нагадувань про ТО.
- за потреби система може бути інтегрована з CRM-системою або модулем обліку витрат, що дозволить об'єднати технічне обслуговування з управлінням клієнтськими запитами та фінансами.

## **2.5 Алгоритм функціонування прогнозу ТО**

Підсистема прогнозування технічного обслуговування виконує аналіз пробігу автомобіля та попередніх сервісів для оцінки ймовірності потреби в новому обслуговуванні. Основна мета - своєчасно попереджати користувача про необхідність ТО.

Алгоритм функціонує наступним чином, користувач самостійно ініціює розрахунок ризиків натисканням на відповідну кнопку біля обраного автомобіля. Після цього система:

- 1) Завантажує список усіх типів ремонтів, які можуть бути застосовані до авто.
- 2) Для кожного типу ремонту система:
  - перевіряє останній запис про виконання відповідної роботи;
  - визначає нормативний інтервал у км та місяцях;
  - оцінює, скільки часу пройшло з моменту останнього ТО, та скільки км пройдено;
  - обчислює ризик технічного зносу або пропущеного ТО за формулою (лінійною або експоненційною).

3) Якщо ризик перевищує поріг (наприклад, 0.6), генерується попередження.

4) Результати виводяться користувачу на екран (таблиця або картка).

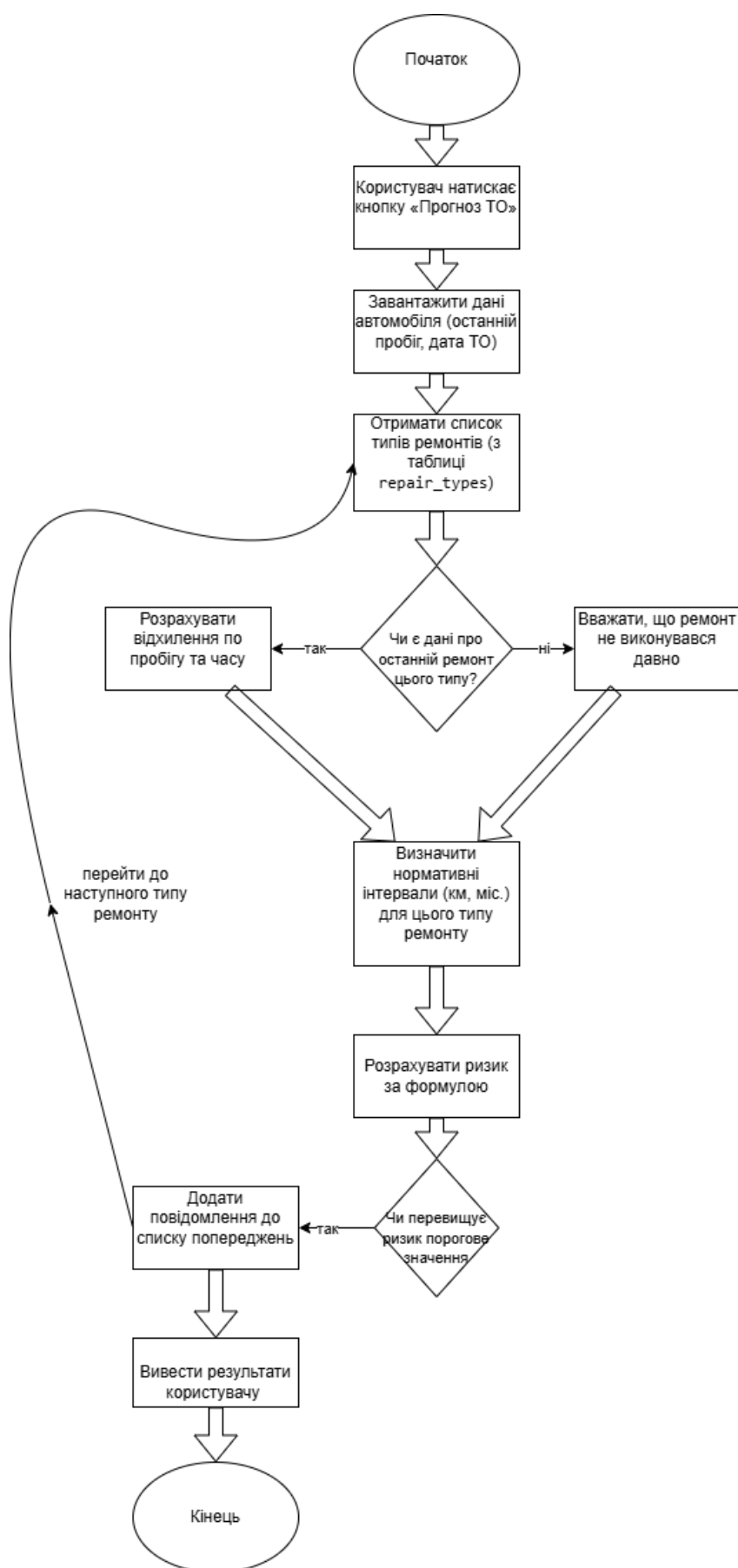


Рисунок 2.1 - Блок-схема алгоритму прогнозу ТО

## РОЗДІЛ 3. РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ

### 3.1. Структура бази даних

У розробленій CRM-системі для СТО з прогнозуванням технічного обслуговування використовується реляційна база даних під керуванням системи управління базами даних MariaDB. База даних має назву sto та містить кілька основних таблиць, кожна з яких виконує специфічну функцію у зберіганні даних про користувачів, транспортні засоби, історію обслуговування, заплановані ремонти тощо.

1. cars - Автомобілі (таблиця зберігає дані про автомобілі, які належать користувачам):

id - унікальний ідентифікатор автомобіля;

user\_id - зовнішній ключ до таблиці користувачів;

brand, model - марка та модель авто;

odometer - поточний пробіг;

VIN - ідентифікаційний номер транспортного засобу;

year - рік випуску;

engine - тип двигуна (petrol, diesel, petrol\_gas, electric);

engine\_volume, power - об'єм двигуна та потужність;

created\_at - дата додавання запису;

km\_per\_month - середній щомісячний пробіг;

last\_mileage\_update - дата останнього оновлення пробігу;

last\_service\_odometer - пробіг під час останнього обслуговування;

notify\_threshold - поріг пробігу для нагадування про ТО;

predicted\_odometer - прогнозований пробіг.

## 2. repairs - Заплановані ремонти (зберігає записи про майбутні ремонти):

id, user\_id, car\_id - ідентифікатори;

repair\_name - назва роботи;

description - опис;

appointment\_date, time - дата та час запису;

created\_at - дата створення запису.

## 3. repair\_types - Типи ремонтів (містить список усіх можливих видів ремонтів і нормативні інтервали їх виконання):

id, name - ідентифікатор і назва;

interval\_km, interval\_months - рекомендований інтервал заміни в кілометрах та місяцях відповідно.

## 4. service\_history - Історія обслуговування

Зберігає інформацію про виконані ремонти:

id, user\_id, car\_id - ідентифікатори;

service\_date - дата ремонту;

odometer\_at\_service - пробіг під час ремонту;

description - короткий опис робіт;

cost - вартість послуг;

repair\_type\_id - тип проведеного обслуговування;

created\_at - дата створення запису.

Усі таблиці пов'язані між собою через зовнішні ключі, що забезпечує цілісність даних і можливість реалізації логіки прогнозування техобслуговування. Зокрема, таблиця repair\_types використовується для розрахунку ризику техобслуговування, а таблиця service\_history - для аналізу минулих ремонтів. Така структура дозволяє ефективно управляти даними та автоматизувати процес обслуговування транспортних засобів.

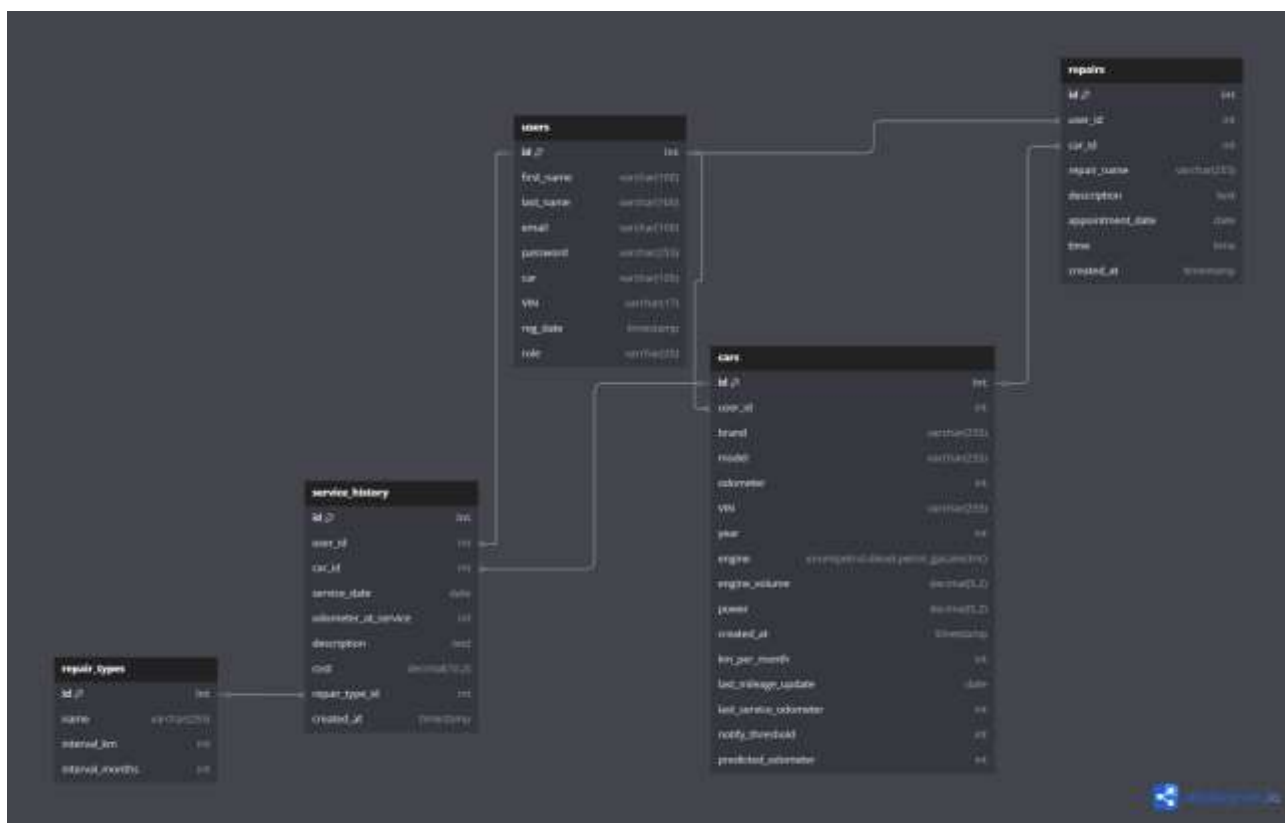


Рисунок 3.1 - ER- Діаграма бази даних автосервісу

## 3.2 Інтерфейс користувача

Інтерфейс користувача є критичним елементом для підсистеми станції технічного обслуговування, оскільки саме через нього здійснюється щоденна взаємодія працівників СТО та клієнтів з системою. Основне завдання – забезпечити інтуїтивно зрозуміле, просте у використанні середовище для роботи з даними про автомобілі, ремонти та технічне обслуговування.

Інтерфейс побудовано за принципами сучасного веб-дизайну з використанням HTML5, CSS (у тому числі Bootstrap) та JavaScript. Уся взаємодія відбувається у браузері.

Зовнішній вигляд сайту витриманий у світлому, мінімалістичному стилі для кращої читабельності. Основні компоненти інтерфейсу включають навігаційне меню, заголовки сторінок, інтерактивні таблиці та форми.

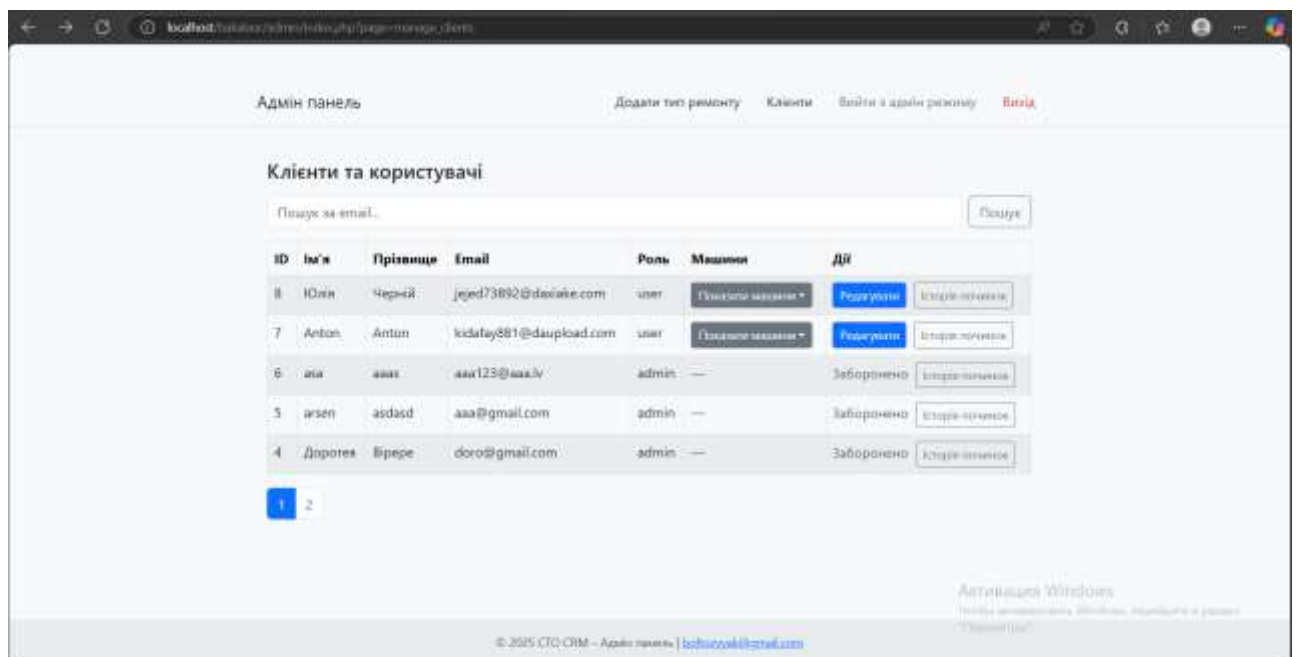


Рисунок 3.2 - Скріншот адмін-панелі

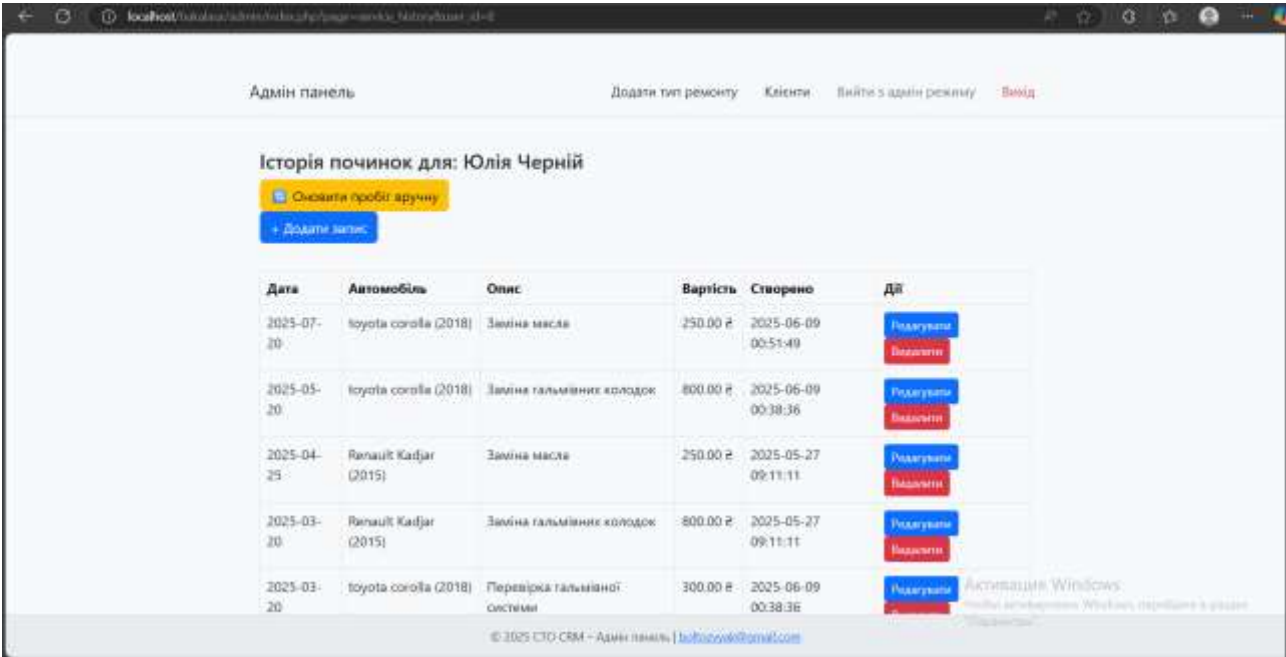


Рисунок 3.3 - Скріншот історії обслуговування в адмін панелі

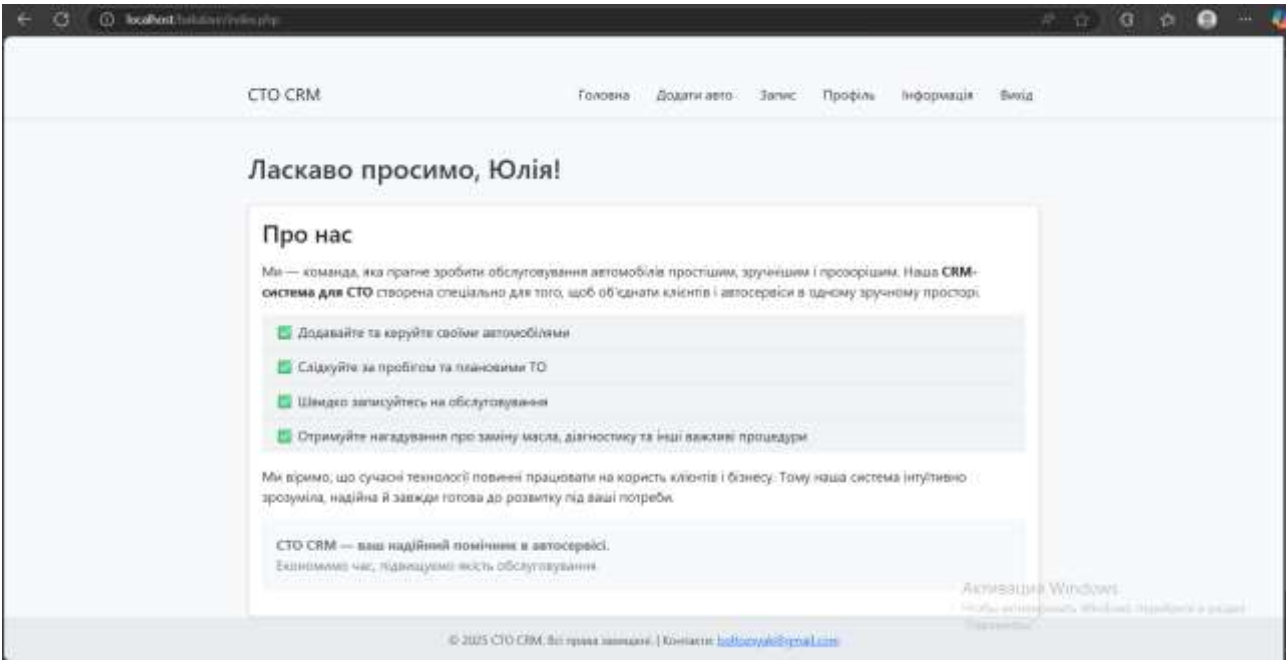


Рисунок 3.4 - Скріншот головної сторінки

Вікна та сценарії взаємодії:

### 1. Панель користувача (кабінет)

Користувач після авторизації потрапляє до свого кабінету, де відображаються всі його автомобілі. Кожен запис містить назву авто, рік випуску, пробіг, тип двигуна та кнопку для оцінки ризику обслуговування.

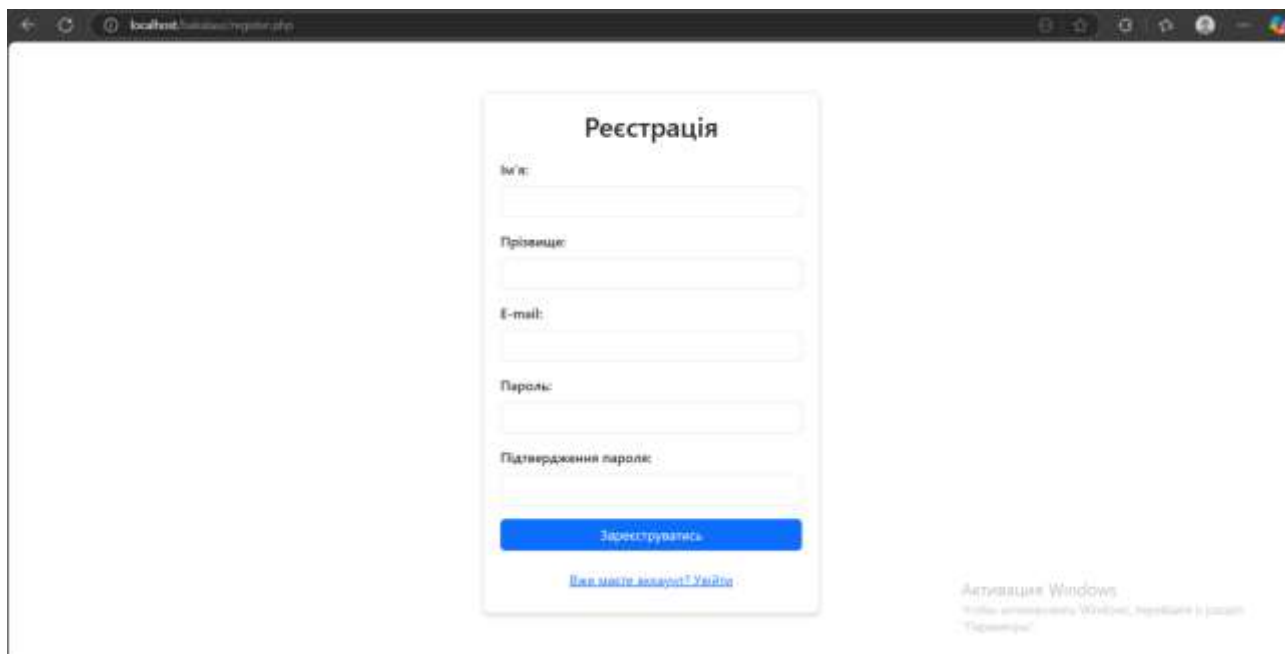


Рисунок 3.5 - Скріншот реєстрації користувача

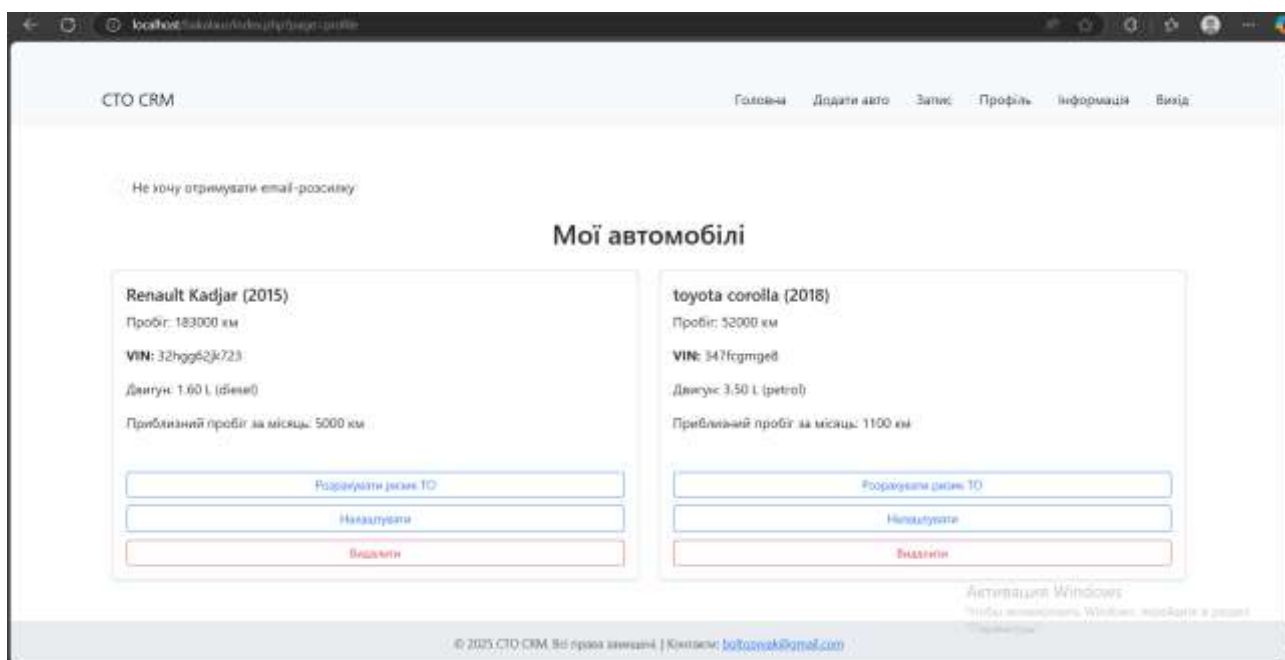


Рисунок 3.6 - Скріншот профілю користувача

## 2. Форма додавання авто

Надається форма для створення нового запису. Вона включає обов'язкові поля: марка, модель, рік, тип двигуна, поточний пробіг. Система здійснює базову валідацію полів.

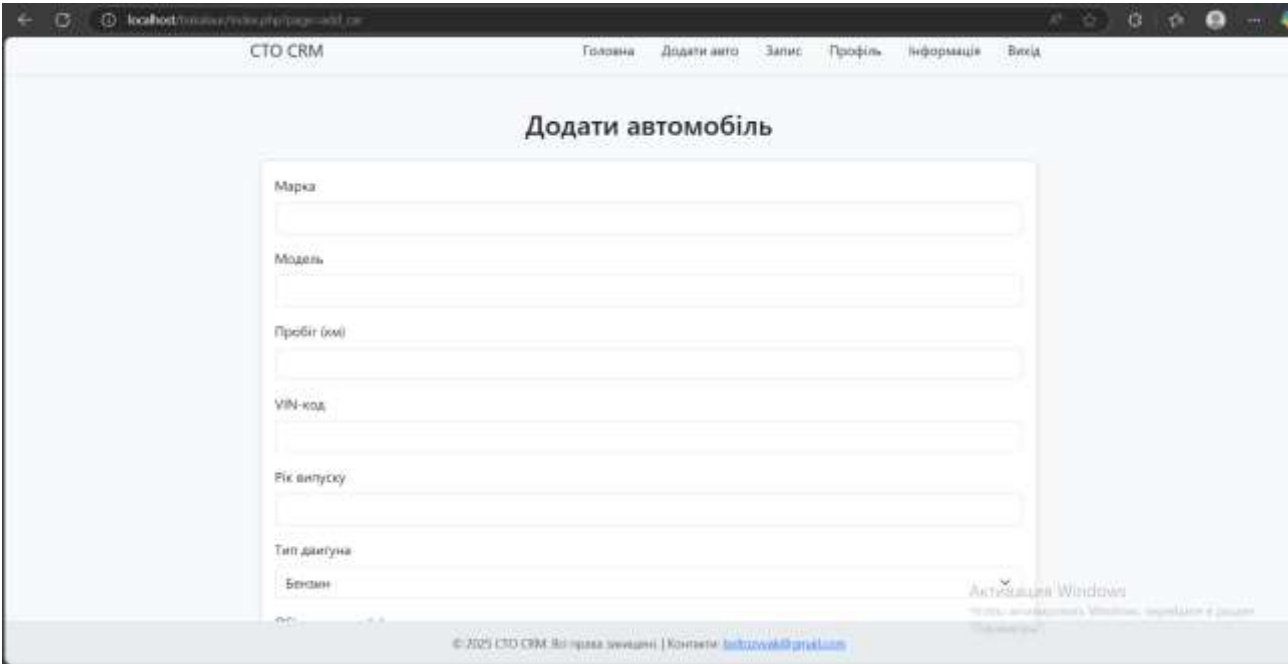


Рисунок 3.7 - Скріншот форми додавання автомобіля

3. Сторінка оцінки ризику ТО

Користувач може натиснути кнопку поруч із конкретним авто, після чого відображається таблиця з оцінкою ризику для кожного типу ремонту. Таблиця включає останню дату обслуговування, кількість місяців з моменту ТО та кольорову шкалу ризику (зелений - низький, жовтий - помірний, червоний - високий).

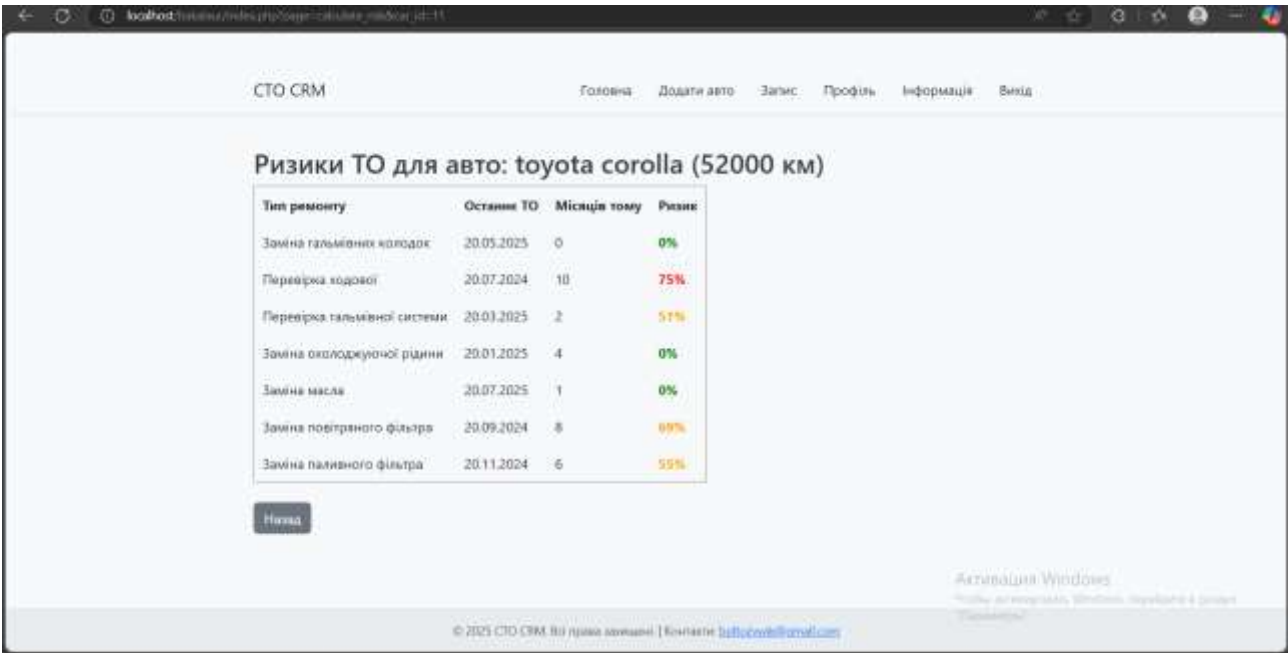


Рисунок 3.8 - Скріншот оцінки ризиків поломки

#### 4. Сторінка запису на ремонт

Користувач може натиснути кнопку на кнопку “Запис”, де він може записатись на ремонт

Рисунок 3.9 - Скріншот сторінки запису на ремонт

Інтерфейс побудований згідно з принципами UX-дизайну:

- Мінімальна кількість дій: уся важлива інформація згрупована на одному екрані
- користувачеві не потрібно переходити між великою кількістю сторінок.
- Візуальна ієрархія: таблиці, кнопки та кольори дозволяють швидко зорієнтуватися у статусі обслуговування.
- Інтуїтивна навігація: усі дії підписані, кнопки не дублюються і розміщені логічно.
- Миттєвий зворотний зв'язок: після оцінки ризику користувач одразу бачить результат без перезавантаження сторінки.

Для реалізації інтерфейсу було використано такі інструменти:

1. Bootstrap - для побудови адаптивної сітки та стилізації елементів (кнопки, таблиці, форми);
2. JavaScript - для обробки подій (наприклад, обробка натиснення кнопки «Оцінити ризик»);
3. PHP - для генерації HTML-вмісту та обробки серверної логіки.

### 3.3 Серверна логіка та обробка даних

Серверна частина системи реалізована за допомогою мови PHP. Вона забезпечує обробку запитів користувача, взаємодію з базою даних та реалізацію основної бізнес-логіки, включаючи валідацію введених даних, авторизацію, обробку подій та формування відповідей.

Основні компоненти логіки:

- Авторизація та автентифікація: перевірка прав доступу на основі ролі користувача (адміністратор, клієнт).
- Валідація даних: при збереженні або редагуванні записів система перевіряє правильність введених даних (наприклад, формат email, числові поля, дата випуску авто).
- Додавання та редагування записів: реалізовано через форми, які взаємодіють з контролерами PHP, що надсилають SQL-запити до відповідних таблиць (cars, users, repairs тощо).
- Обробка запитів щодо технічного обслуговування: при перегляді профілю авто користувача система виконує запит до таблиці service\_history, знаходить останні виконані роботи, зіставляє їх з нормативними інтервалами (repair\_types) і передає ці дані у функцію прогнозування.

Розрахунок ризику відбувається окремо в логічному модулі (див. підрозділ 3.4), а результати інтегруються в інтерфейс користувача. Для цього PHP-скрипт збирає вхідні параметри (пробіг, дата останнього обслуговування, тип двигуна), викликає функцію оцінки ризику та формує HTML-відповідь.

Окремо також реалізовано механізм формування email-повідомлень: користувачам, які не вимкнули розсилку (email\_opt\_out = false), система надсилає попередження про ймовірну необхідність технічного обслуговування.

### 3.4 Прогнозування та вивід повідомлень

Однією з ключових функцій підсистеми для СТО є прогнозування необхідності проведення технічного обслуговування для транспортних засобів

на основі наявних даних про пробіг, дату останнього ремонту, нормативні інтервали та технічні характеристики автомобіля. Це дозволяє формувати інтелектуальні попередження користувачам про необхідність сервісу до появи проблем.

### **Розрахунок пробігу та відхилень**

Для кожного типу ремонту (наприклад, «Заміна гальмівних колодок», «Заміна масла» тощо) у базі даних зберігаються нормативні інтервали: по пробігу (наприклад, 10 000 км) та по часу (наприклад, 12 місяців). Також у таблиці `service_history` зберігається інформація про дату останнього технічного обслуговування та показник одометра на той момент.

Поточний пробіг береться з таблиці `cars`, де він може оновлюватись вручну користувачем. Якщо дані про попередній ремонт відсутні, система вважає, що обслуговування давно не проводилось, і розраховує ризик відповідно до допустимих норм із запасом (вважаючи, що останнє ТО було приблизно на 80% інтервалу назад).

### **Алгоритм розрахунку ризику**

Для кожного типу ремонту обчислюється два основних показники:

- Кілометрове перевищення: наскільки поточний пробіг перевищує останній зафіксований пробіг ремонту.
- Часова затримка: скільки місяців минуло з останньої дати обслуговування.

Ці показники нормуються до відповідних нормативних інтервалів і далі підставляються у формулу ризику. У поточній версії використовується експоненціальна модель:

$$\text{risk} = 0.55 \cdot (1 - e^{-2\text{km\_ratio}}) + 0.35 \cdot (e^{-1.5 \cdot \text{time\_ratio}}) + \text{age\_correction} + \\ + \text{engine\_factor}$$

де `km_ratio` - перевищення пробігу відносно нормативу, `time_ratio` — перевищення часу, `age_correction` - надбавка за вік автомобіля (до 15%), а `engine_factor` зменшує ризик для електромобілів і трохи підвищує для дизельних авто.

У майбутньому можливо перейти на лінійну або комбіновану модель із машинним навчанням для більш точного прогнозу.

### Вивід повідомлень та система сповіщення

Після обчислення ризиків для всіх типів ремонтів система формує перелік ТО з високими ризиками (понад 75%) для кожного автомобіля. Якщо користувач не вимкнув email-розсилку (перевіряється поле `email_opt_out` у таблиці `users`), йому надсилається повідомлення з рекомендаціями пройти сервісне обслуговування. У повідомленні вказується марка автомобіля, тип рекомендованого ремонту, дата або пробіг останнього обслуговування, а також значення ризику.

Функція відправки реалізована окремим PHP-файлом (`update_mileage_and_notify.php`), який може запускатись вручну або через планувальник (`cron`). Це дозволяє автоматизувати процес моніторингу стану автопарку і попереджати користувача, навіть якщо він рідко заходить у систему.

Також, на сторінці профілю користувача доступна кнопка для миттєвого розрахунку ризиків вручну, яка дозволяє побачити поточні значення без очікування автоматичної перевірки.

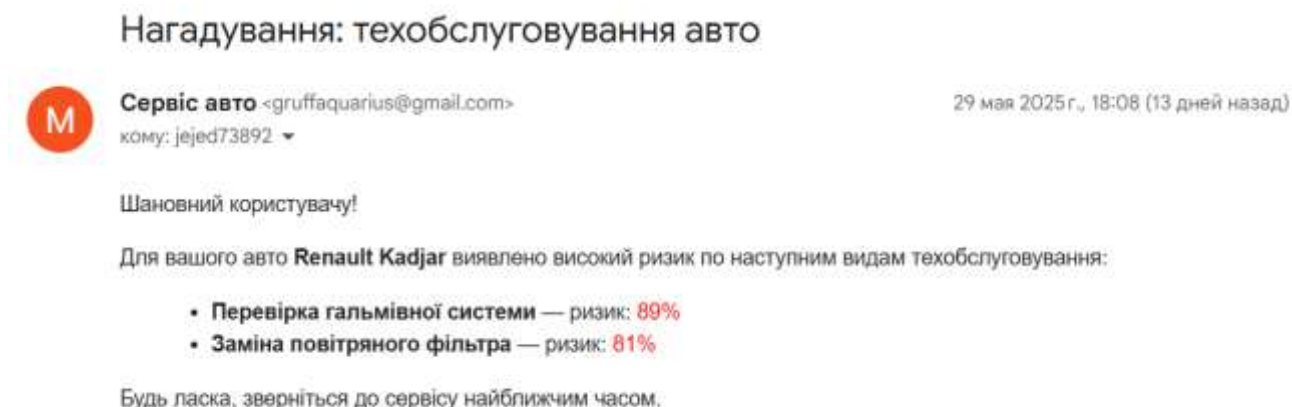


Рисунок 3.10 - Скріншот E-mail повідомлення



Рисунок 3.11 - Діаграма потоків даних

## ВИСНОВОК

У результаті виконання кваліфікаційної роботи було реалізовано підсистему прогнозування технічного обслуговування автомобілів у складі CRM-системи для станцій технічного обслуговування (СТО). У першому розділі проведено аналіз предметної області, зокрема вивчено існуючі інформаційні рішення для СТО, їх функціональні можливості та обмеження. Встановлено, що більшість систем базуються на фіксованих інтервалах обслуговування і не враховують динаміку пробігу чи інші змінні.

У другому розділі обґрунтовано вибір технологій для реалізації системи, описано структуру бази даних і математичну модель розрахунку ризику. Розроблено алгоритм, що враховує пробіг, вік автомобіля, тип двигуна та інтервали ТО. Це дозволяє отримати приблизну оцінку наближення необхідного обслуговування.

У третьому розділі реалізовано серверну логіку обробки запитів користувача, виконання розрахунку ризику для кожного типу робіт, а також створено інтерфейс для відображення цих результатів. Розрахунок здійснюється при натисканні кнопки у профілі користувача. Передбачено можливість email-сповіщення, якщо користувач не відмовився від розсилки. Додатково створено діаграму алгоритму прогнозу, яка наочно відображає логіку прийняття рішення.

Запропонована підсистема допомагає СТО краще планувати обслуговування клієнтів, зменшує ризик запізненого ремонту та формує основу для впровадження автоматизованих нагадувань і прогнозів у майбутньому.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Moovago. 10 ERP Statistics In 2022. URL: <https://moovago.com/en/blog-en/erp-statistics/>
2. BlueCart. Top ERP Systems: List of ERP Systems. URL: <https://www.bluecart.com/blog/top-erp-systems>
3. jQuery. What is jQuery? URL: <https://jquery.com/>
4. Apteian. The benefits of ERP. URL: <https://apteian.com/en-US/insights/blog/15-benefits-implementing-erp-software>
5. Thomas Hamilton. Guru99. What is Software Testing? Definition, Basics & Types in Software Engineering. URL: <https://www.guru99.com/software-testing-introduction-importance.html>
6. FIRMAO. Online CRM software. URL: <https://firmao.net/software-for-the-company.php>
7. Wikipedia. Algorithm. URL: <https://en.wikipedia.org/wiki/Algorithm>
8. BELITSOFT. Front-end development services URL: <https://belitsoft.com/front-end-development-services>
9. Tom Butler, Kevin Yank. PHP & MySQL: Novice to Ninja: Get Up to Speed With PHP the Easy Way.  
URL:  
<http://repo.darmajaya.ac.id/4984/1/PHP%20%26%20MySQL%20Novice%20to%20Ninja%2C%206th%20Edition.pdf>
10. Max Bramer. Web programming with PHP and MySQL. URL: <https://github.com/dimateos/eBook-1/blob/master/Web%20Programming%20with%20PHP%20and%20MySQL.pdf>
11. Kyle Simpson. You Don't Know JS: Async & Performance. URL: <https://github.com/azat-io/you-dont-know-js-ru>
12. Yurgen Wolf. HTML and CSS: The Comprehensive Guide. URL: <https://dokumen.pub/qdownload/html-and-css-the-comprehensive-guide-1nbsped-1493224220-9781493224227-q-2748246.html>

13. Paul McFedries. HTML, CSS, & JavaScript All-in-One For Dummies. URL: <https://dokumen.pub/html-css-amp-javascript-all-in-one-for-dummies-for-dummies-team-ira-1nbsped-1394164688-9781394164684.html>
14. Jennifer Niederst Robbins. Learning Web Design, A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. URL: [https://github.com/CodesdaLu/Programming-Books/blob/main/Front-End/Learning/Learning%20Web%20Design%20\\_%20A%20Beginner%E2%80%99s%20Guide%20to%20HTML%2C%20CSS%2C%20JavaScript%2C%20and%20Web%20Graphics.pdf](https://github.com/CodesdaLu/Programming-Books/blob/main/Front-End/Learning/Learning%20Web%20Design%20_%20A%20Beginner%E2%80%99s%20Guide%20to%20HTML%2C%20CSS%2C%20JavaScript%2C%20and%20Web%20Graphics.pdf)
15. Jacob Lett. Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4. URL: <https://github.com/JacobLett/bootstrap-4-book>