

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні
системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Система автоматизації робочих процесів магазину з
продажу автомобілів»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІА-94

Світельський Лев Олегович

Керівник:

Асистент кафедри ІСТ,

Белоус Роман Володимирович

Рецензент:

Доцент, кандидат наук,

Ліхоузова Тетяна Анатоліївна

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Світельському Льву Олеговичу

1. Тема проєкту «Система автоматизації робочих процесів магазину з продажу автомобілів», керівник проєкту Белоус Роман Володимирович, асистент кафедри ІСТ, затверджені наказом по університету від «31» травня 2023 р. № 2101-с;
2. Термін подання студентом проєкту: «12» червня 2023 року;
3. Вихідні дані до проєкту: база даних інформації про користувачів та автомобілі з детальним описом і характеристиками що будуть використані в системі автоматизації;
4. Зміст пояснювальної записки: аналіз готових систем, порівняння наявних рішень, вибір необхідних технологій для розробки, опис особливостей технічної реалізації, огляд готової системи;
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма класів, діаграма прецедентів, карта сайту, діаграма бази даних;

6. Дата видачі завдання: «5» квітня 2023 року.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз існуючих рішень	До 19.04.2023	Виконано
2	Розробка концепції проєкту	До 21.04.2023	Виконано
3	Розробка макету веб застосунку	До 26.04.2023	Виконано
4	Вибір програмних ресурсів	До 28.04.2023	Виконано
5	Проектування бази даних	До 4.05.2023	Виконано
6	Розробка серверної частини застосунку	До 16.05.2023	Виконано
7	Розробка клієнтської частини застосунку	До 26.05.2023	Виконано
8	Налагодження взаємодії серверної та клієнтської частин застосунку	До 29.05.2023	Виконано
9	Передзахист дипломної роботи	29.05.2023	Виконано
10	Тестування розробленої системи	До 10.06.2023	Виконано
11	Захист дипломної роботи	3 19.06.2023	

Студент

Лев СВІТЕЛЬСЬКИЙ

Керівник

Роман БЕЛОУС

АНОТАЦІЯ

Світельський Л.О. Система автоматизації робочих процесів магазину з продажу автомобілів. КПП ім. Ігоря Сікорського, Київ, 2023.

Проект містить 65 сторінок тексту, 32 рисунки, 7 таблиць, посилання на 17 літературних джерел, додатки, 4 кресленники.

Ключові слова: автосалон, веб-застосунок, система автоматизації, Laravel, сайт.

Об'єкт дослідження – робочі процеси магазину з продажу автомобілів.

Метою роботи є створення досконалої системи для автоматизації продажу автомобілів.

Для реалізації проекту було використано PHP-фреймворк Laravel, а також мову гіпертекстової розмітки HTML та мови програмування PHP і JavaScript. У якості програмного забезпечення для розгортання сервера та бази даних використано XAMPP. Середовищем розробки виступило Visual Studio Code.

Було створено повноцінний веб-сайт автосалону, що забезпечує можливість перегляду каталогу автомобілів, роботу з замовленнями (створення, перегляд, видалення), а також адміністративні функції, такі як додавання нових автомобілів та зміна інформації про існуючі автомобілі.

Веб-застосунок забезпечує високу швидкість взаємодії з базою даних і відмінну продуктивність. Завдяки використанню PHP і Laravel, веб-сайт є масштабованим, надійним та легким для підтримки. Використання Bootstrap дозволяє забезпечити красивий і зручний інтерфейс.

Отримані результати можуть бути корисними при розробці систем автоматизації для автосалонів або інші компанії, які займаються продажем автомобілів.

ANNOTATION

Svitelskyi L.O. Car dealership's working processes automation system. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

The project contains 65 pages of text, 32 figures, 7 tables, references to 17 literature sources, appendices, 4 drawings.

Keywords: car dealership, web application, automation system, Laravel, website.

The subject of research – the operational processes of a car dealership.

The purpose of the work is to create an impeccable system for automating car sales.

The project was implemented using the PHP framework Laravel, as well as the HyperText Markup Language (HTML) and the programming languages PHP and JavaScript. XAMPP was used as the software for deploying the server and the database. The development environment was Visual Studio Code.

A full-fledged car dealership website was created, providing the ability to browse the car catalog, work with orders (creation, viewing, deletion), as well as administrative functions such as adding new cars and changing information about existing cars.

The web application ensures high-speed interaction with the database and excellent performance. Thanks to the use of PHP and Laravel, the website is scalable, reliable, and easy to maintain. The use of Bootstrap ensures a beautiful and convenient interface.

The results obtained can be useful in developing automation systems for car dealerships or other companies involved in car sales.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA94.160БАК.004 ПЗ	Пояснювальна записка	79		
6	A3	IA94.160БАК.004 Д1	Система автоматизації	1		
7			робочих процесів магазину			
8			з продажу автомобілів.			
9			Діаграма прецедентів			
10	A3	IA94.160БАК.004 Д2	Система автоматизації	1		
11			робочих процесів магазину			
12			з продажу автомобілів.			
13			Діаграма бази даних			
14	A3	IA94.160БАК.004 Д3	Система автоматизації	1		
15			робочих процесів магазину			
16			з продажу автомобілів.			
17			Діаграма класів			
18	A3	IA94.160БАК.004 Д4	Система автоматизації	1		
19			робочих процесів магазину			
20			з продажу автомобілів.			
21			Карта сайту			
22						
23						
24						
25						
26						
27						
28						
Зм.	Аркуш	№ докум.	Підпис	Дата	IA94.160БАК.004 ТП	
Розроб.		Світельський Л.О.				
Керівн.		Белоус Р.В.			Система автоматизації робочих процесів магазину з продажу автомобілів. Відомість проекту Літ. Аркуш Аркушів Т 1 1 КПІ ім. Ігоря Сікорського Група ІА-94	
Затв.						

**Пояснювальна записка
до дипломного проєкту
на тему: «Система автоматизації робочих
процесів магазину з продажу автомобілів»**

Київ – 2023 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Загальний огляд	9
1.2 Огляд існуючих рішень	17
Висновок до розділу 1.....	16
2 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ	17
2.1 Мова гіпертекстової розмітки HTML	17
2.2 Мова програмування JavaScript	18
2.3 Laravel.....	18
2.4 База даних MySQL	20
2.5 Мова програмування PHP	21
2.6 Eloquent ORM	22
2.7 Мова стилю сторінок CSS	23
2.8 Підходи до розробки веб-додатків	24
Висновок до розділу 2.....	26
3 РОЗРОБКА ТА АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	28
3.1 Архітектура веб-додатку	28
3.2 Проектування інформаційної системи	30
3.3 Розробка бази даних.....	33

					ІА94.160БАК.004 ПЗ										
Зм.	Лист	№ докум.	Підпис		Система автоматизації робочих процесів магазину з продажу автомобілів. Пояснювальна записка					Літ.		Арк.		Аркушів	
Розробив	Світельський	Л.О.		Т							2		68		
Перевірив	Белоус Р.В.														
Затв.										КПІ ім. Ігоря Сікорського Група ІА-94					

3.4 Розробка серверної частини	38
3.5 Розробка клієнтської частини	43
Висновок до розділу 3.....	45
4 ДЕМОНСТРАЦІЯ РОБОТИ СИСТЕМИ.....	46
4.1 Гість	46
4.2 Авторизований користувач	52
4.3 Адміністратор	59
Висновок до розділу 4.....	65
ВИСНОВКИ.....	64
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТОК А.....	68

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

VIN-код – Vehicle Identification Number, унікальний серійний номер автомобіля.

CRM-система – Customer Relationship Management, система управління взаємовідносинами з клієнтами.

БД – база даних.

API - Application Programming Interface, сукупність засобів та правил, що вможливають взаємодію між окремими складниками програмного забезпечення.

AJAX - Asynchronous JavaScript and XML, технологія асинхронних веб-запитів.

CSS - Cascading Style Sheets, мова стилізації веб-документів.

PHP - Hypertext Preprocessor, серверна мова програмування.

JS - JavaScript, мова програмування для веб-розробки.

HTTP - Hypertext Transfer Protocol, протокол передачі гіпертексту.

MVC – Model-View-Controller, паттерн проектування програмного забезпечення.

MySQL – відкрита система управління базами даних.

ORM - Object-Relational Mapping, технологія програмування, що зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування.

CLI - Command Line Interface, спосіб взаємодії людини і комп'ютера шляхом відправки текстових команд.

ВСТУП

Нам пощастило застати час, коли людство активно переходить від епохи ручної праці або машинної праці під наглядом людини до повністю автоматизованих процесів, які не потребують стороннього втручання. Фактично, технології вже у певній мірі проникли в усі сфери людського життя. І ми вимушені миритися з цим, незалежно від того, подобається це нам чи ні.

У сферу продажу автомобілів сучасні технології прийшли не в останню чергу.

Незважаючи на те, що звичайні автосалони все ще віддають перевагу особистій роботі з клієнтами, для успішного функціонування вони все одно повинні мати свій інформаційний ресурс в інтернеті для того, щоб потенціальні покупці могли ознайомитися з асортиментом автомобілів та умовами співпраці. А великим ритейлерам або безпосередньо відділам продажу конкретних автовиробників просто необхідно мати автоматизовану систему для того щоб мати можливість своєчасно та оперативно реалізовувати товар, при цьому мінімізуючи вірогідність людської помилки [1].

На сучасному ринку існує серйозна конкуренція, тому переможцем виходить той, хто здатен максимально швидко і точно підлаштовуватися під поточну ситуацію. Треба боротися за кожного покупця, тому що, якщо компанії не вдасться його втримати, то він одразу же потрапить до прямих конкурентів.

Об'єктом дослідження виступає комплекс робочих процесів, що зв'язані з салоном по продажу автомобілів. Вони включають в себе дії починаючи з надходження автомобіля до салону та реєстрації його в системі до його продажу та доставку клієнту. Даний об'єкт дослідження був обраний тому, що в результаті стрімкого збільшення користувачів Інтернету та переходу до формату онлайн продаж, автосалони теж змушені адаптуватися до формату онлайн роботи. Відповідно, перехід до такого формату створює необхідність вирішувати новий тип

проблем, що пов'язані з автоматизацією робочих процесів, забезпечення безпеки даних та зручністю використання сервісу клієнтами.

Предметом дослідження являється процес надання інформації про автомобілі та обробки замовлень в режимі онлайн. Важливими аспектами являються як питання реалізації ідеального користувацького інтерфейсу, так і можливість швидко обробляти замовлення та оновлювати всю необхідну інформацію про автомобіль.

Мета даної роботи – розробка та використання системи автоматизації робочих процесів магазину з продажу автомобілів, яка забезпечує зручну та ефективну взаємодію компанії з клієнтами та оптимізує керування асортиментом та замовленнями.

Для досягнення поставленої мети потрібно виконати наступні задачі:

1. Дослідження існуючих робочих процесів магазину, аналіз їх ефективності та оцінка потреб їх автоматизації;

2. Огляд існуючих рішень у сфері торгівлі автомобілями та виявлення їх сильних і слабких сторін, а також аналіз отриманої інформації та створення концепції досконалої системи;

3. Розробка технічного завдання для створення автоматизованої системи, що включає вимоги до функціоналу, дизайну, безпеки даних;

4. Розробка веб застосунку, що реалізує необхідний функціонал;

5. Тестування розробленої системи та доопрацювання виявлених недоліків;

6. Оцінка ефективності системи за допомогою аналізу статистики використання, відгуків користувачів та динаміки об'ємів продаж.

Практичне значення одержаних результатів обумовлене актуальністю та важливістю автоматизації робочих процесів у сучасній сфері торгівлі автомобілями, яка хоч і не являється новою, але на даний момент все ще продовжує розвиватися. Завдяки успішній роботі можна досягти наступних результатів:

1. Підвищення ефективності роботи за рахунок автоматизації процесів обробки замовлень, контролю наявності та доступності автомобілів і взаємодії з

клієнтами. В результаті скоротиться час обробки замовлень та підвищиться якість обслуговування;

2. Розширення клієнтської бази в результаті створення онлайн каталогу автомобілів та реалізації функціоналу для можливості онлайн замовлень. Таким чином, до списку клієнтів можуть приєднатися люди, які вже звикли робити покупки в Інтернеті або не мають змоги фізично прийти у салон;

3. Підвищення конкурентоспроможності досягається за рахунок наявності веб застосунку, який має сучасний дизайн та зручний інтерфейс, а також можливість легко зробити замовлення або отримати необхідну інформацію по автомобілю чи деталям угоди. Це дозволить автосалону вигідно виділитися серед конкурентів та залучити більше клієнтів;

4. Підвищення доходів в результаті зменшення витрат на офлайн обслуговування та розширення клієнтської бази, яка дозволить отримати більше прибутків;

5. Широкі можливості для масштабування через те, що розроблена система може використовуватися у магазинах з продажу автомобілів різних масштабів, навіть у найбільш популярних дилерів та може бути масштабована у будь-який момент в залежності від потреб компанії.

У першому розділі описується важливість реалізації системи автоматизації магазину з продажу автомобілів у вигляді веб застосунку. Наводиться статистика з об'єму продаж в Інтернеті за останні роки. Проводиться порівняння та оцінка найвідоміших вітчизняних та зарубіжних рішень у даній сфері.

У другому розділі описуються основні використані технології під час розробки веб-додатку та обґрунтовується їх використання. Розглядаються підходи до розробки веб-додатків. В результаті складається уявлення про майбутню систему.

В третьому розділі проводиться розробка застосунку. Описується структура його серверної та клієнтської частин. Розглядається призначення окремих

компонентів Laravel проєкту на прикладі розроблюваної програми. Окремо описана структура бази даних, а також загальна архітектура даного додатку.

В четвертому розділі демонструється готова система. Огляд проводиться від лица різних акторів, передбачених діаграмою прецедентів. Фактично наводиться інструкція з користування системою, а також її тестування та огляд можливостей.

					ІА94.160БАК.004 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний огляд

Через стрімкий розвиток інформаційних технологій та збільшення кількості людей на Землі, а відповідно й користувачів Інтернету, системам автоматизації робочих процесів тепер відводиться важлива роль. Сфера автомобільного ринку також не є виключенням. Сьогодні такі системи зазвичай реалізуються у вигляді веб-застосунків.

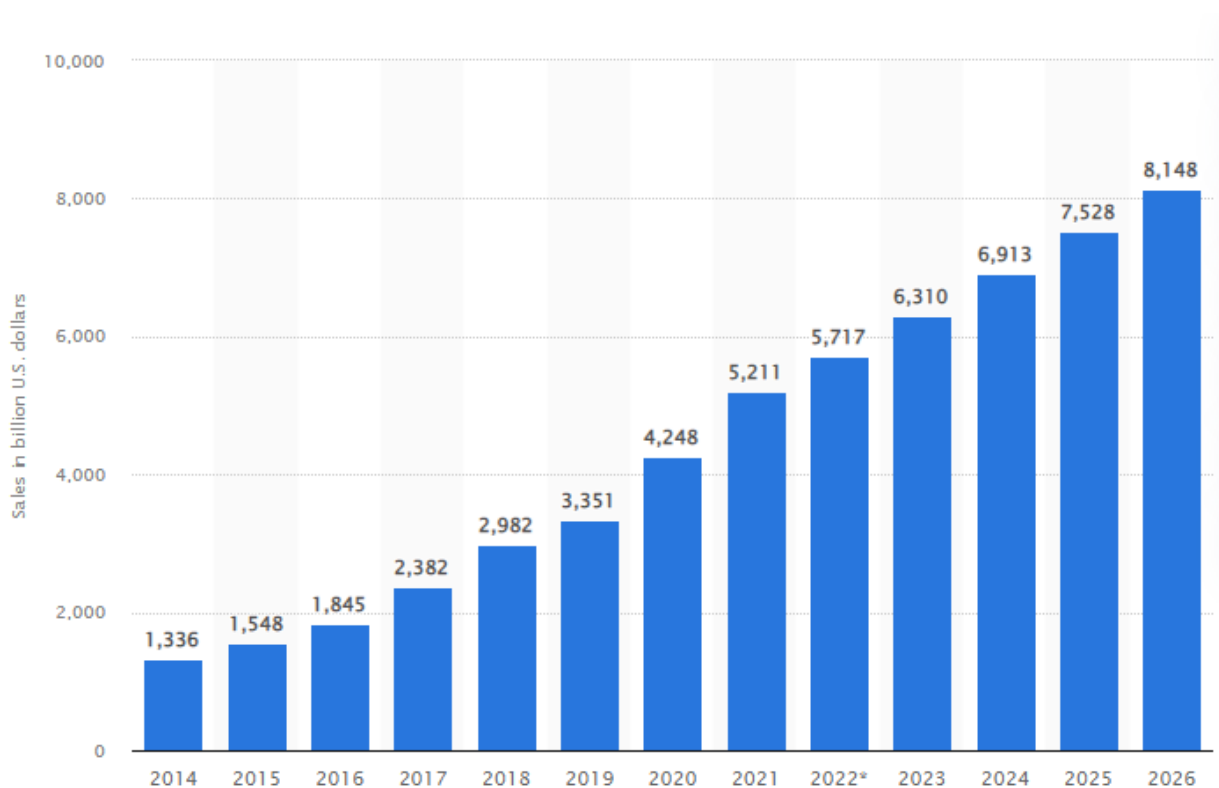


Рисунок 1.1 – Загальний дохід компаній за продажі в Інтернеті (у доларах США) [2]

Зараз веб-застосунки замінюють звичайні програми тому що їх переваги занадто очевидні для того щоб це ігнорувати. Завдяки набутого за останні роки розвитку, веб-сайти стали потужним інструментом, який вже мало чим поступається звичайним програмам. Сучасні фреймворки дозволяють

реалізовувати складну логіку, яку можна без особливих зусиль поєднати з приємним для ока та зручним інтерфейсом. До того ж важливим аспектом являється те, що один веб застосунок можуть використовувати безліч людей з усіх куточків світу незалежно від того, яким браузером вони користуються та яка операційна система в них встановлена, і навіть того, який пристрій вони використовують для доступу в Інтернет. Веб-сайти надають можливість використовувати один продукт на різних пристроях з однаковим комфортом при умові дотримання певних правил при їх розробці.

Говорячи про переваги веб-сайтів перед стаціонарними застосунками, не можна не згадати про легкість оновлення та підтримки. Для того щоб оновити певний контент, або навіть повністю весь сайт, адміністратору достатньо зберегти зміни на сервері, а всі користувачі надалі просто будуть завантажувати оновлені сторінки. Одним словом, оновлення для них пройде абсолютно непомітно. У той час як для оновлення застосунку треба щоб кожен користувач завантажив та встановив нову версію, при тому, не факт, що вона буде безпомилково працювати на усіх пристроях.

Важливо усвідомлювати, що перед початком розробки системи у даному напрямі, треба більш детально дослідити специфіку роботи автомобільних салонів. Для більш детального огляду цієї теми варто більш детально розглянути деякі пункти:

– робочі процеси у автомобільному бізнесі. З першого погляду здається, що автосалони працюють по принципу: отримали товар, виставили на продаж, продали та отримали дохід. Але при більш детальному аналізі виявляється що все є дещо складнішим. Кожна поставка має бути строго задокументована як у постачальника, так і у самого автосалону. Після надходження кожен автомобіль повинен бути внесений у базу даних, включаючи його VIN-код. Усі автомобілі в обов'язковому порядку повинні бути застраховані до моменту продажу клієнту. Документи повинні бути оформлені виключно за встановленими правилами. При продажу клієнту повинні видати тимчасовий номерний знак, зареєстрований в системі;

					ІА94.160БАК.004 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

– сучасні технології в автосалонах. Дослідження технологій, які на даний момент використовуються в автосалонах допоможе виділити головний напрям дослідження, а також оцінити можливості їх автоматизації. Варто розглянути CRM-системи, фінансові системи, системи контролю товару;

– існуючі рішення в області автоматизації, що використовуються в автосалонах. Огляд існуючих рішень допоможе зрозуміти принцип їх роботи, а також виділити позитивні та негативні сторони кожного з них;

– ключові вимоги до системи автоматизації роботи автосалону. Визначення ключових вимог таких як зручність використання, безпека даних, можливості звітності, масштабованість допоможе оцінити об'єм роботи та визначити необхідні технології для використання при розробці системи;

– вплив автоматизації на показники прибутків компанії. Прогноз очікуваного впливу на показники ефективності роботи автосалону, такі як продажі, рівень обслуговування клієнтів, якість контролю асортименту та ін.

1.2 Огляд існуючих рішень

У цьому підрозділі будуть розглядатися системи автоматизації на основі веб-сайтів різних типів для того, щоб можна було оцінити різні підходи до створення систем автоматизації. Так як для компанії такого типу в першу чергу важлива зацікавленість покупця, а також через те, що ми не маємо можливості дізнатися всі тонкощі роботи певної системи, то надалі будемо розглядати веб-застосунки насамперед з точки зору користувача.

1.2.1 Веб-застосунок AUTO.RIA

Сайт AUTO.RIA – це один з найбільш відомих та активно використовуваних ресурсів в Україні для продажу та купівлі автомобілів. Він надає широкий спектр можливостей, як для продавців, так і для покупців. Насамперед варто зазначити,

що AUTO.RIA фактично не являється магазином з продажу автомобілів, а лише виступає площадкою для оголошень від різних приватних та юридичних осіб. Незважаючи на це, даний веб-застосунок все одно представляє для нас інтерес [3].

The screenshot displays the AUTO.RIA website interface. At the top, there is a navigation bar with the 'auto RIA' logo, links for 'Вживані авто', 'Нові авто', 'Новини', and 'Все для авто', along with a green button labeled '+ Продати авто'. Below this, a banner encourages users to 'Підтримати збір на авто для ЗСУ'. The main search area features a red sidebar with filters for vehicle type (Всі, Вживані, Нові, Під пригон), body type (Легкові, Марка, Модель), region (Регіон), year (Рік випуску), and price (Ціна, \$). It includes buttons for 'Розширений пошук' and 'Пошук'. To the right, a light blue box promotes 'Запчастини' (Spare parts) with an image of car parts and an 'Обрати' button. Below these is a large blue banner with the text 'СМІЛИВІСТЬ будувати авторинок довіри' and statistics: '200 000+ авто з усієї України • за годину + 1 094 • перевірено по VIN-коду + 184 509'. At the bottom, there are category filters (Сімейні, 100-150 тис км, Мінівени, Джипи, Електро, Для подорожей) and three small image thumbnails.

Рисунок 1.2 – Інтерфейс веб-застосунку AUTO.RIA

На головній сторінці сайту нас одразу зустрічає великий блок для вибору фільтрів та пошуку необхідного автомобіля. Нижче розташована невелика підбірка різноманітних автомобілів, які пропонуються ще до того, як будуть налаштовані фільтри. Можна припустити, що власники цих оголошень скористалися додатковою платною опцією для того, щоб туди потрапити. Крім продажу автомобілів, на сайті також містяться посилання на каталог СТО та автозапчастини, але 99% уваги зосереджено саме на автомобілях. Час від часу можуть зустрічатися рекламні банери невеликих розмірів, які в цілому не заважають користуванню застосунком.

Переваги даного сервісу:

- величезний набір фільтрів для пошуку автомобіля (при використанні розширеного пошуку);
- гарна швидкодія навіть на слабких пристроях;
- відсутність набридливої реклами;
- дуже великий вибір автомобілів, які включають як нові, так і вживані;
- можливість безкоштовного перегляду історії певного автомобіля, що включає наявність ДТП, або попередніх продаж;
- можливість перевірки репутації продавця;
- можливість зв'язатися з продавцем за допомогою чату на сайті;
- наявність інформації про ступінь перевірки автомобіля та останній зафіксований пробіг.

Недоліки даного сервісу:

- застарілий дизайн;
- багато екземплярів із сумнівною історією;
- відслідковується пряма залежність вірогідності продажу автомобіля від використання платних опцій.

1.2.2 Веб-сайт автосалону Winner Automotive

Сайт Winner Auto – це офіційний веб-ресурс одного з ведучих автомобільних дилерів в Україні. На відміну від AUTO.RIA, ця компанія займається не тільки продажем автомобілів, а й наданням різноманітних послуг, таких як сервіс і запчастини, фінансові послуги, тест драйв. Автосалон Winner Automotive в першу чергу займається продажем нових автомобілів деяких виробників (Ford, Volvo, Jaguar, Land Rover, Bentley, Porsche, Renault, MG), але також паралельно допомагає продавати автомобілі з пробігом від попередніх власників [4].

					ІА94.160БАК.004 ПЗ	Арк.
						13
Зм.	Лист	№ докум.	Підпис	Дата		

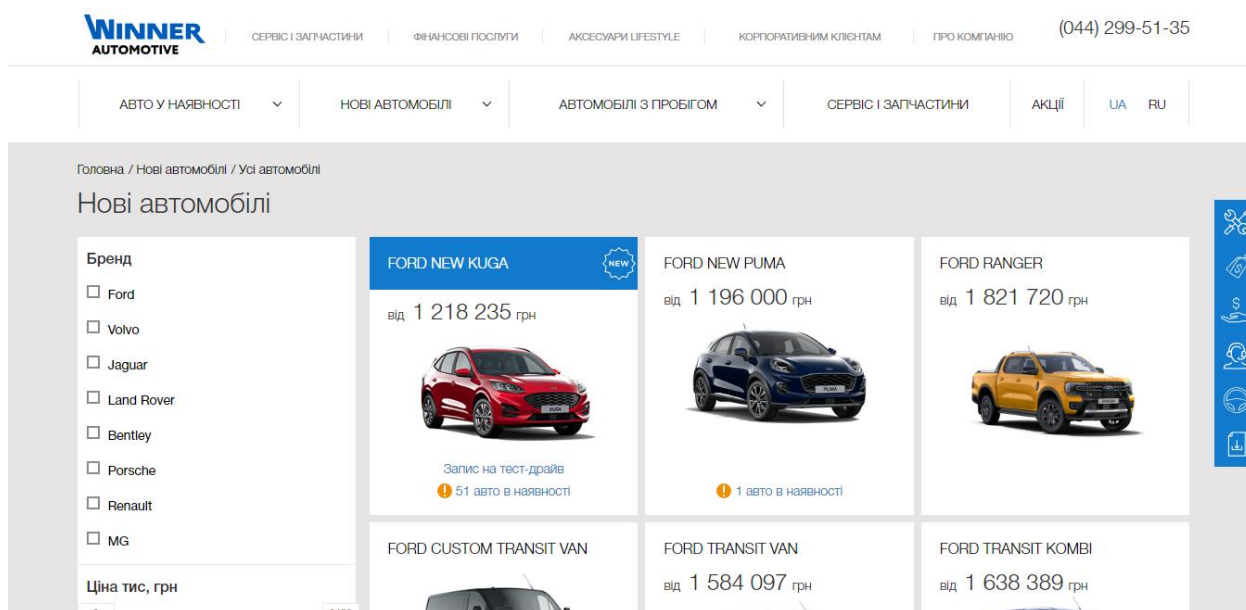


Рисунок 1.3 – Інтерфейс веб-застосунку Winner Auto

В цілому можна відмітити, що веб-сайт має відносно непоганий інтерфейс, Автомобілі фактично представлені у вигляді трьох каталогів: у першому (рисунок 1.3) відображений весь перелік моделей, які можна придбати в даному автосалоні; у другому розміщуються автомобілі, які є в наявності з чітко визначеними цінами, які варіюються в залежності від комплектації; у третьому представлені автомобілі з пробігом, для кожного з яких додатково вказаний пробіг та стан, який був визначений експертами. Серед цікавих особливостей користувацького інтерфейсу можна відмітити плаваюче меню справа, яке розкривається при наведенні курсора та згортається назад при прокручуванні сторінки.

Переваги даного сервісу:

- можливість отримати автомобіль та весь пакет послуг в одному місці;
- можливість отримати консультацію від кваліфікованого спеціаліста;
- компанія займається продажем тільки надійних та перевірених автомобілів;
- компанія надає гарантію на автомобілі, що продає;
- клієнт має можливість проходити офіційне технічне обслуговування в даному автосалоні.

Недоліки даного сервісу:

- кількість параметрів для фільтрації автомобілів значно менша ніж на AUTO.RIA;
- сайт хоч і непоганий, але занадто простий станом на 2023 рік;
- застосунок перенавантажений непотрібними для більшості клієнтів послугами.

1.2.3 Веб-сайт автосалону F1rst Motors

Сайт F1RST Motors – це веб-платформа, що забезпечує повний спектр послуг, пов'язаних з продажем нових та вживаних автомобілів. Сайт пропонує широкий асортимент автомобілів, розширені опції пошуку та детальну інформацію про кожний автомобіль, що полегшує вибір потрібного автомобіля для користувача. Власне даний автосалон знаходиться в Дубаї, Об'єднані Арабські Емірати, та представляє найдорожчі автомобілі, відповідно й розрахований на найбагатші верстви населення [5].

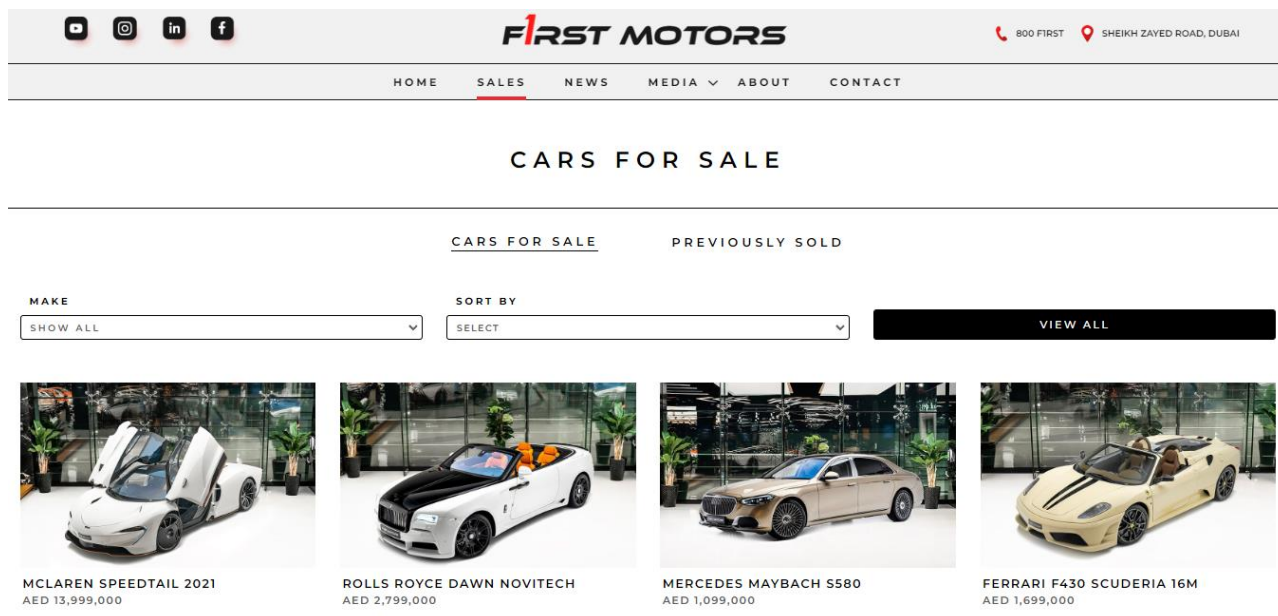


Рисунок 1.4 – Інтерфейс веб-застосунку F1rst Motors

З першого погляду видно, що сайт розроблений в доволі мінімалістичному дизайні, але при цьому широко використовуються анімації (однотипні, але виглядає непогано). Стає очевидно, що вся увага зосереджена саме на автомобілях. Серед цікавих особливостей варто окремо відмітити можливість віртуального туру по автосалону, а також зйомка деяких автомобілів у режимі 360°.

Переваги даного сервісу:

- зручний інтерфейс без зайвих опцій та сторінок;
- сучасне оформлення;
- великий асортимент автомобілів, кожен із яких доступний для придбання у будь-який момент;
- кожен автомобіль має багато зображень найвищої якості, на яких можна роздивитися усі деталі.

Недоліки даного сервісу:

- всі ціни виключно у дирхамах ОАЕ;
- можливість фільтрації автомобілів тільки по марці, а сортування тільки по ціні;
- веб-сайт може працювати повільно на слабких пристроях;
- система недостатньо автоматизована та виступає в першу чергу як інформаційний ресурс, а не як повноцінний застосунок.

Висновок до розділу 1

У даному розділі були більш детально розглянуті робочі процеси магазину з продажу автомобілів, а також проведене порівняння для визначення переваг веб-застосунку перед стаціонарним у сфері автоматизації торгівлі автомобілями. Крім того, було проведене порівняння популярних застосунків у даній сфері та виділені сильні та слабкі сторони кожного з них.

2 ВИБІР ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ

2.1 Мова гіпертекстової розмітки HTML

HTML (HyperText Markup Language) є основною мовою для створення веб-сторінок і веб-застосунків. Саме за допомогою HTML визначається структура та зміст веб-сторінки.

Важливо пам'ятати, що HTML – це мова розмітки, а не програмування. Вона не виконує жодних обчислювальних операцій, а використовується для визначення структури та відображення інформації на веб-сторінці, наприклад: тексту, зображень, відео, форм та інших елементів.

Однією з основних переваг HTML є універсальність. Всі сучасні веб-браузери підтримують HTML, що означає, що веб-сторінка, створена з використанням HTML, буде відображатися коректно на будь-якому пристрої з будь-яким веб-браузером [6].

Крім того, HTML є відкритим стандартом, який розвивається і підтримується консорціумом W3C. Це означає, що розробники можуть використовувати HTML без оплати ліцензійних зборів і бути впевненими, що їх код буде працювати в майбутньому.

HTML також легко інтегрується з іншими технологіями веб-розробки, такими як CSS для стилізації та JavaScript для додавання інтерактивності. З цієї причини, HTML часто використовується разом з CSS і JavaScript для створення повноцінних веб-застосунків.

У зв'язку з вищезгаданими перевагами, вибір HTML як мови розмітки для створення веб-застосунку є фактично безальтернативним варіантом.

В нашому випадку HTML буде використовуватися в рамках Blade сторінок, які відповідають за відображення інформації у Laravel.

2.2 Мова програмування JavaScript

JavaScript є основною мовою програмування для розробки інтерактивних веб-застосунків. Ця мова була створена для виконання на клієнтській стороні, прямо в браузері, що дозволяє зробити веб-сторінки більш динамічними та «живими».

Аналогічно до HTML, важливою перевагою JS є її підтримка всіма сучасними веб-браузерами без потреби встановлення додаткових плагінів. Код JS виконується безпосередньо в браузері, що дозволяє забезпечити швидку взаємодію з користувачем та миттєву відповідь на його дії.

JavaScript має потужні об'єктно-орієнтовані можливості, які дозволяють створювати складні структури даних та алгоритми. Окрім того, ця мова підтримує функціональне програмування, що дозволяє використовувати високорівневі абстракції та створювати більш модульний та масштабований код.

JavaScript надає можливість маніпулювати елементами HTML та CSS на веб-сторінці, що дає можливість динамічно змінювати зміст та оформлення сторінки відповідно до дій користувача, стану додатку або вхідних даних [7].

JavaScript активно використовується для створення асинхронних веб-застосунків, що дозволяє забезпечити високу швидкість відклику інтерфейсу, незалежно від того, очікується відповідь від сервера чи виконується важке обчислення.

2.3 Laravel

Laravel є одним з найпопулярніших PHP фреймворків, який використовується для розробки веб-застосунків. Він пропонує надійний спосіб побудови веб-застосунків, який дозволяє зосередитися на основних завданнях, замість вирішення низькорівневих проблем.

					ІА94.160БАК.004 ПЗ	Арк.
						18
Зм.	Лист	№ докум.	Підпис	Дата		

Laravel надає широкий спектр інструментів для розробки, включаючи засоби для роботи з базами даних, маршрутизації, сесіями, кешуванням, аутентифікацією, авторизацією, відправкою електронних листів, завантаженням файлів, і багато іншого.

Однією з ключових переваг Laravel є використання принципів MVC (Model-View-Controller). Це забезпечує чітке розділення бізнес-логіки додатку та відображення інформації, що сприяє розподілу відповідальності, полегшує тестування та підтримку коду [8].

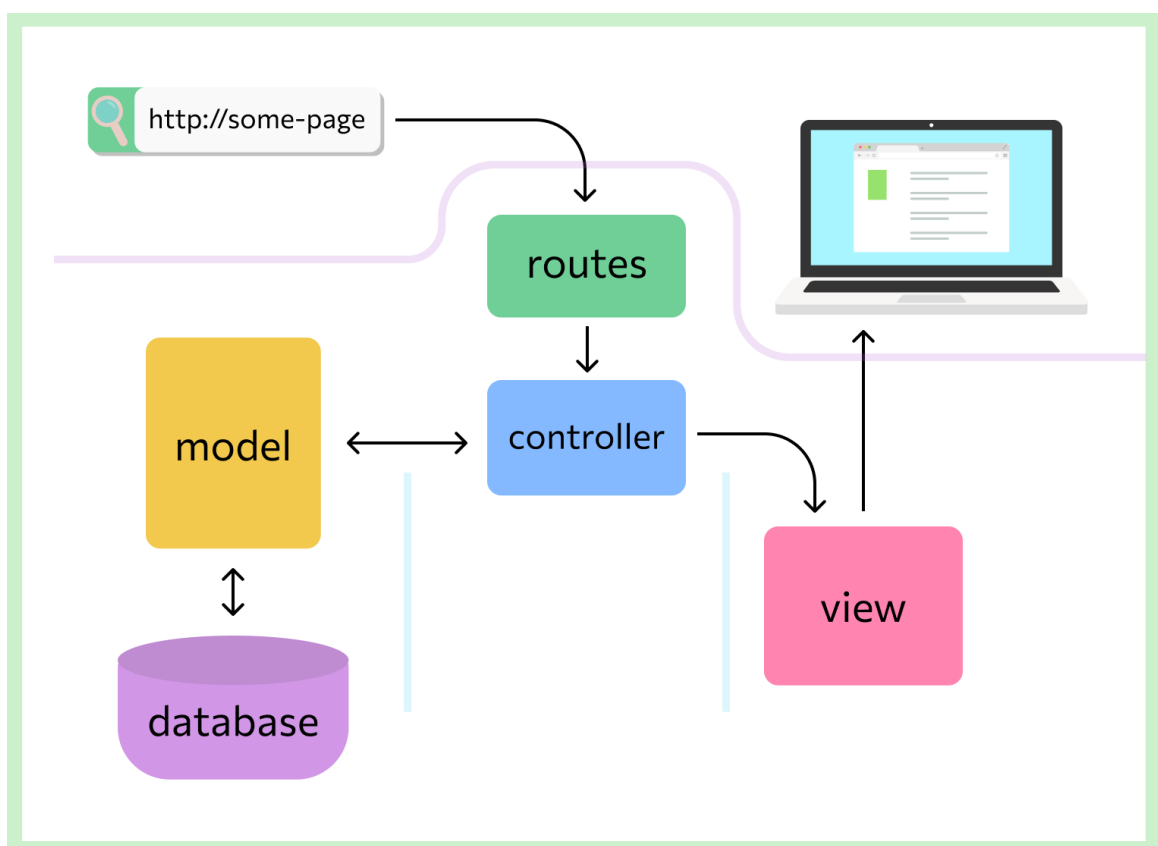


Рисунок 2.1 – Схематичне відображення архітектури MVC

Laravel також включає в себе систему шаблонів Blade, яка дозволяє створювати складні і гнучкі представлення даних, використовуючи простий і зручний синтаксис.

Laravel активно розвивається і підтримується спільнотою, що гарантує його актуальність і відповідність сучасним технологічним трендам.

Можна також виділити такі особливі компоненти Laravel:

- моделі (models) – представляють таблиці баз даних, і використовуються для взаємодії з таблицею. Кожна модель в Laravel пов'язана з однією таблицею бази даних. Для відправлення запитів до бази даних використовується ORM Eloquent;

- контролери (controllers) – використовуються для обробки бізнес-логіки додатку. Коли користувач здійснює запит до веб-застосунку, маршрутизатор перенаправляє запит на відповідний контролер, і контролер відповідає за обробку цього запиту та відправку відповіді. Контролери можуть бути різними за об'ємом, починаючи від простого посилання на файл відображення, закінчуючи повним набором CRUD операцій та користувацьких функцій;

- маршрутизація (routing): - відповідає за узгодження шляхів з контролерами, які відповідатимуть на певні запити. Також надає можливість накладати обмеження для доступу до деяких шляхів або передавати певні дані до контролера;

- представлення (views) – використовуються для генерації HTML відповіді і зазвичай містять HTML, JavaScript і CSS, а також Laravel Blade шаблонізатор;

- middleware – використовується для фільтрації HTTP-запитів до вашого додатку. Наприклад, Laravel включає в себе middleware, який перевіряє, чи є користувач авторизованим, перш ніж дозволити йому доступ до певних ресурсів;

- artisan CLI – могутній інструмент командного рядка. Цей інструмент дозволяє автоматизувати багато рутинних задач, таких як створення контролерів, моделей, міграцій бази даних, тестів та ін.

2.4 База даних MySQL

MySQL – це одна з найпопулярніших відкритих систем управління базами даних, що використовує SQL (Structured Query Language) для роботи з даними. Вона є компонентом стеку технологій LAMP (Linux, Apache, MySQL, PHP), відомого своєю надійністю, гнучкістю і широкою підтримкою.

Серед переваг MySQL можна виділити:

					ІА94.160БАК.004 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

– багатокористувацька система: MySQL підтримує одночасну роботу декількох користувачів. Це означає, що багато користувачів можуть виконувати різні запити до однієї бази даних одночасно;

– ACID-сумісність: MySQL дотримується принципів ACID (Atomicity, Consistency, Isolation, Durability), що є важливим для транзакційних систем. Це означає, що всі транзакції в MySQL є атомарними, консистентними, ізольованими і стійкими;

– висока продуктивність: MySQL оптимізована для високої продуктивності і є в змозі обробляти великі набори даних;

– безпека: MySQL надає потужні механізми безпеки, включаючи підтримку SSL для шифрування даних під час передачі, а також різні рівні доступу для користувачів;

– гнучкість: MySQL підтримує велику кількість типів даних, включаючи числові, строкові, дати/час, а також бінарні дані.

Використання MySQL разом з PHP і Laravel дозволяє розробникам створювати масштабовані, надійні та безпечні веб-застосунки. MySQL легко інтегрується з Laravel через ORM Eloquent, що робить процес розробки дуже зручним.

2.5 Мова програмування PHP

PHP (Hypertext Preprocessor) є популярною серверною мовою програмування, яка активно використовується в розробці веб-додатків. PHP спочатку був створений для роботи з гіпертекстом, але в подальшому його можливості були значно розширені, і тепер він включає в себе повноцінні можливості мови програмування.

PHP є відкритим програмним забезпеченням, що дозволяє розробникам вільно використовувати його для створення веб-додатків. Основною особливістю PHP є його інтеграція з HTML, що спрощує розробку веб-застосунків [9].

Мова PHP дозволяє працювати з базами даних, файлами, забезпечує обробку форм і розсилку електронних листів, а також має засоби для управління користувацькими сесіями і куки.

Однією з ключових переваг PHP є його простота і доступність. Мова має простий синтаксис і легко інтегрується з різними веб-серверами і платформами.

В Laravel PHP виступає основною мовою програмування та використовується для написання моделей, контролерів та інших серверних частин коду.

2.6 Eloquent ORM

Eloquent ORM (Об'єктно-реляційне відображення) є основою фреймворка Laravel і надає зручний, простий у використанні інтерфейс для роботи з базами даних. Заснований на патерні «Active Record», Eloquent дозволяє працювати з базами даних в об'єктно-орієнтованому стилі, де кожен рядок в таблиці відповідає об'єкту в додатку.

Eloquent дозволяє використовувати всі переваги об'єктно-орієнтованого програмування з базами даних: наслідування, поліморфізм, інкапсуляцію та ін.

Однією з найбільш цінних особливостей Eloquent є його здатність керувати відношеннями між різними моделями. Він підтримує всі основні типи відношень, такі як «один до одного», «один до багатьох», «багато до багатьох», «поліморфні відношення» та ін. Це дозволяє створювати більш гнучкі і ефективні структури даних.

Також Eloquent має засоби для роботи з подіями, які відбуваються при створенні, оновленні або видаленні моделі. Це можна використовувати для того, щоб автоматично виконувати додаткові дії, коли відбувається певна подія.

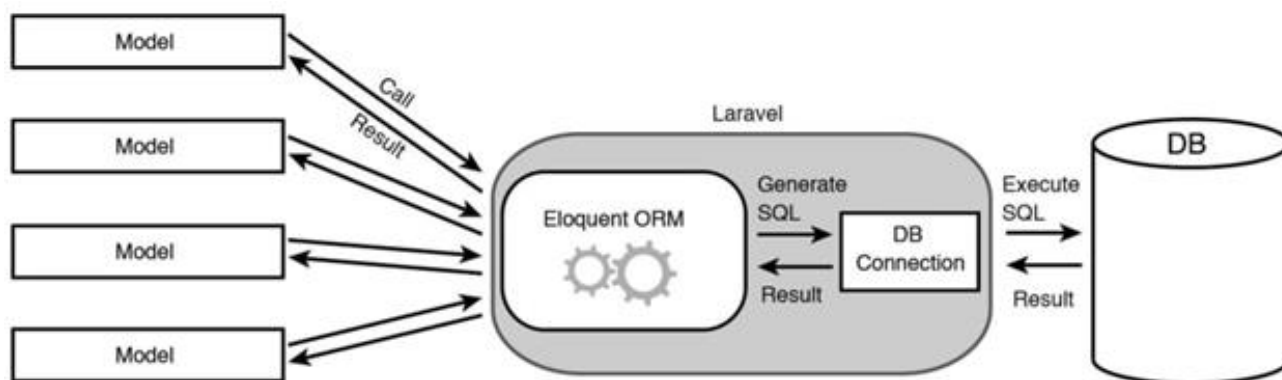


Рисунок 2.2 – Роль Eloquent ORM в Laravel

На рисунку 2.2 відображені процеси, у яких бере участь Eloquent. Таким чином, можна побачити, що він виступає зв'язуючою ланкою між моделлю та базою даних.

За допомогою Eloquent можна також працювати з колекціями моделей. Це дозволяє виконувати різні операції з групами моделей, такі як фільтрація, сортування, перебір.

Так як у системі автоматизації роботи автосалону взаємодія веб-інтерфейсу з базою даних буде необхідною на багатьох етапах, Eloquent ORM відіграватиме важливу роль у роботі застосунку.

2.7 Мова стилю сторінок CSS

CSS, або Cascading Style Sheets (каскадні таблиці стилів), є мовою, яка використовується для опису вигляду документів, написаних на мові розмітки, таких як HTML. CSS дозволяє контролювати такі аспекти веб-сторінки, як шрифти, кольори, відступи, вирівнювання і так далі. Окрім цього, CSS дає змогу створювати адаптивні веб-дизайни, що автоматично налаштовуються під різні розміри дисплеїв та пристроїв [10].

Основна перевага CSS полягає в тому, що він дозволяє відокремлювати зовнішній вигляд веб-сайту від його структури і змісту. Це означає, що можна змінювати дизайн сайту за межами HTML-коду. Крім того, це дозволяє

використовувати одні й ті ж стилі для різних сторінок вашого веб-сайту, виділивши стилі в окремий файл, що призводить до більшої узгодженості і меншого обсягу коду.

CSS підтримує різні види селекторів, що дозволяє обирати конкретні елементи на сторінці для стилізації. Таким чином можна обирати елементи за назвою тега, класу, ідентифікатору, атрибуту.

CSS також включає можливості для створення анімацій та переходів, що дозволяє додавати більше взаємодії та динаміки на веб-сайти.

Препроцесори CSS, такі як Sass або Less, дозволяють використовувати змінні, змішування, вкладеність та інші можливості, що спрощують написання та підтримку CSS-коду.

2.8 Підходи до розробки веб-додатків

2.8.1 Традиційний підхід до розробки веб-додатків

Традиційний підхід до розробки веб-додатків, також відомий як «водоспадний» підхід, характеризується послідовним процесом, що включає ряд визначених етапів. Ці етапи, як правило, включають аналіз вимог, проектування, реалізацію, тестування та обслуговування.

На етапі аналізу вимог розробники веб-додатків вивчають потреби користувачів та бізнес-вимоги до додатка. Вони також визначають основні функції та можливості, які має мати веб-додаток.

Етап проектування включає розробку архітектури додатка, вибір технологій для його реалізації, а також проектування інтерфейсу користувача.

На етапі реалізації розробники пишуть код веб-додатку, реалізуючи всі заплановані функції та можливості.

Тестування – це етап, на якому додаток перевіряється на виявлення помилок та проблем з процесами. Також на цьому етапі перевіряється, чи відповідає додаток всім вимогам, встановленим на початковому етапі аналізу вимог.

					ІА94.160БАК.004 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

Нарешті, на етапі обслуговування веб-додаток регулярно оновлюється та модифікується згідно з вимогами користувачів та змінами в бізнес-вимогах.

Традиційний підхід до розробки веб-додатків пропонує чітку структуру та послідовність, яка дозволяє розробникам систематизувати процес розробки. Однак він також може бути жорстким і не ефективним у проектах, де вимоги можуть змінюватися протягом часу.

2.8.2 Гнучкий (Agile) підхід до розробки веб-додатків

Серед основних принципів гнучкого підходу до розробки веб-додатків, відомого також як Agile, виділяють гнучкість, взаємодію та готовність до змін. Замість використання строго визначеного, послідовного процесу, як в традиційному підході, Agile фокусується на постійній взаємодії, регулярних оновленнях та зворотному зв'язку.

Гнучкий підхід включає в себе ітераційний процес, де розробка веб-додатку розбивається на серії коротких «спринтів». Кожен спринт, як правило, триває від одного до чотирьох тижнів і містить повний цикл розробки – від проектування до тестування [11].

Одним з ключових аспектів Agile є активна участь замовника або користувача у процесі розробки. Це означає, що замовник часто залучається до перегляду прогресу, надає зворотний зв'язок та допомагає визначати пріоритети для наступного спринта. Такий підхід допомагає команді розробників краще розуміти потреби та очікування замовника.

Agile підхід сприяє більш швидкій адаптації до змін. Замість зосередження на дотриманні початкового плану, Agile-команди готові до змін, що дозволяє їм краще реагувати на нові вимоги або виклики, які можуть виникнути під час процесу розробки.

Хоча Agile має багато переваг, таких як гнучкість, здатність швидко адаптуватися до змін, а також більш тісна взаємодія з замовником, цей підхід також

потребує добре організованої команди та активної участі замовника. Також важливо врахувати, що Agile не може стати ідеальним рішенням для всіх проектів, особливо для тих, де вимоги є стабільними і добре визначеними з самого початку.

2.8.3 Порівняння традиційного та гнучкого підходів

При виборі підходу до розробки веб-додатку важливо розглянути відмінності між традиційними і гнучкими методами.

Традиційний підхід, як правило, включає в себе високу ступінь планування та структуризації. Цей підхід вимагає детального початкового проектування та визначення вимог до продукту до початку роботи. Він зосереджується на строго послідовних етапах розробки, зокрема аналіз, проектування, реалізація, тестування та підтримка. Однак, традиційний підхід може бути неефективним при змінних вимогах, через його низьку гнучкість та пізній зворотній зв'язок.

Навпроти, гнучкий підхід, часто описується як Agile, зосереджується на ітераційній розробці та постійному зворотному зв'язку. Він дозволяє команді розробників часто перевіряти прогрес та вносити корективи в процесі розробки. Гнучкий підхід має набагато більшу гнучкість для адаптації до змін, що може бути важливим для проектів, де вимоги можуть змінюватися або уточнюватися.

Обидва підходи мають свої переваги та недоліки. Вибір між ними повинен базуватися на конкретних потребах проекту, доступних ресурсах, часових рамках та ступені необхідної гнучкості.

Висновок до розділу 2

В даному розділі були розглянуті перспективи актуальних мов програмування та їх потенціальна роль в розробці веб-застосунку. В результаті було прийняте рішення про розробку застосунку на PHP фреймворку Laravel.

Для визначення розмітки сайту використовуватиметься мова гіпертекстової розмітки HTML разом з таблицею стилів CSS. У разі необхідності планується використання Java Script.

В рамках особливостей Laravel, для написання серверної частини застосунку, використовується PHP, а для взаємодії з базою даних MySQL– Eloquent ORM.

У якості середовища розробки було обране Visual Studio Code по причині універсальності та наявності можливостей для встановлення розширень, які дозволяють з комфортом розробляти додатки на будь-якій мові програмування.

					ІА94.160БАК.004 ПЗ	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

3 РОЗРОБКА ТА АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

3.1 Архітектура веб-додатку

3.1.1. Компоненти архітектури веб-додатку

Архітектура веб-додатку є ключовим елементом його проектування та розробки. Це структура системи, що включає в себе різні компоненти та забезпечує їх взаємодію [12]. Загальна архітектура веб-додатку зазвичай складається з наступних основних компонентів:

- клієнтська сторона (Frontend) – інтерфейс, з яким взаємодіє користувач. Він зазвичай реалізований з використанням технологій як HTML, CSS, та JavaScript. Завдяки цим технологіям можлива розробка динамічних та інтерактивних веб-сторінок;

- серверна сторона (Backend) – займається обробкою даних, запитів від клієнта, а також управлінням даними в базі даних. Мови програмування для серверної сторони включають PHP, Python, Ruby, Java та інші;

- база даних – зберігає всю інформацію, яку використовує веб-додаток. Це може включати дані користувача, інформацію про товари, пости блогу тощо;

- вебсервер – приймає HTTP-запити від клієнтів (браузерів користувачів), обробляє їх і відправляє HTTP-відповіді назад. Він забезпечує взаємодію між клієнтською стороною та серверною стороною;

- middleware – це програмне забезпечення, що забезпечує взаємодію між різними компонентами системи. Воно може включати авторизацію, обробку помилок, логування та інші функції.

3.1.2. Взаємодія компонентів архітектури

Взаємодія компонентів в архітектурі веб-додатку є центральним аспектом їх успішного функціонування. Щоб забезпечити надійну роботу додатка, необхідно правильно налаштувати співпрацю між клієнтською та серверною сторонами,

базою даних та іншими компонентами. У веб-застосунку можна виділити наступні види взаємодій:

- взаємодія клієнта і сервера: Клієнт (браузер користувача) надсилає HTTP-запит до сервера. Сервер обробляє цей запит, виконує необхідні дії (наприклад, взаємодію з базою даних), а потім відправляє HTTP-відповідь назад до клієнта;

- взаємодія сервера з базою даних: Коли сервер отримує запит на роботу з даними (наприклад, отримання, створення, оновлення або видалення записів), він виконує ці дії в базі даних. Після виконання дії сервер отримує результати з бази даних і відправляє їх назад до клієнта;

- використання Middleware: Middleware зазвичай використовується для обробки специфічних задач між клієнтом та сервером. Наприклад, Middleware може перевірити, чи має користувач право доступу до певного ресурсу, обробити помилку або логувати інформацію про запит;

- взаємодія з іншими компонентами: В залежності від складності та вимог веб-додатка, можуть використовуватися додаткові компоненти, такі як сервіси електронної пошти, оплати, системи кешування тощо. Ці компоненти також взаємодіють з основною архітектурою веб-додатка.

3.1.3. Особливості архітектури MVC (Model-View-Controller)

Архітектурний шаблон Model-View-Controller (MVC) дуже популярний у розробці веб-додатків, оскільки він допомагає зберегти код чистим і організованим, зокрема в проектах з великою кількістю коду.

Model-View-Controller розбиває додаток на три основні компоненти: Model, View і Controller.

Model відповідає за усі дані, бізнес-логіку і правила додатка. Це можуть бути, наприклад, класи для роботи з базою даних, класи, які реалізують бізнес-логіку, та інші компоненти, що виконують обчислення або обробку даних.

View відповідає за відображення даних на веб-сторінках, презентує їх у зрозумілому вигляді. Цей компонент не містить у собі бізнес-логіки, він лише отримує дані від моделі і відображає їх. Використовуючи HTML, CSS та JavaScript, View гарантує, що користувачі побачать певні зміни в результаті взаємодії з веб-сторінкою.

Controller слугує мостом між Model і View. Контролер реагує на дії користувача, взаємодіє з моделлю, отримує дані та передає їх до View для відображення.

Однією з ключових особливостей архітектури MVC є відокремлення логіки від представлення, що покращує організованість коду і дозволяє розробникам працювати над різними частинами додатка паралельно.

Також, завдяки цій архітектурі, можна легко масштабувати додаток, оскільки можна модифікувати або розширювати кожний компонент незалежно від інших.

3.2 Проектування інформаційної системи

3.2.1 Ролі користувачів

У веб-додатку використовуються різні ролі, щоб забезпечити рівень доступу користувачів до різних функцій та вмісту. Основні ролі, які зазвичай визначаються в системі, включають гостя, авторизованого користувача та адміністратора.

Гість – це користувач, який відвідує сайт без виконання входу в систему. Гості мають обмежений доступ до вмісту сайту, зазвичай вони можуть переглядати публічний вміст, але не можуть виконувати дії, які потребують авторизації, такі як коментування або редагування вмісту.

Авторизований користувач – це користувач, який зареєструвався та увійшов до системи. Авторизовані користувачі мають більше прав і можливостей, ніж гості. Вони можуть брати участь в інтерактивних функціях, таких як публікація вмісту, коментування та інше. Вони також мають доступ до своїх особистих налаштувань профілю.

					ІА94.160БАК.004 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

Адміністратори – це користувачі, які мають повний доступ до функцій сайту. Вони мають право налаштовувати вміст сайту, користувачів, ролі та дозволи. Вони можуть переглядати, редагувати, видаляти і додавати вміст, керувати користувачами, видаляти коментарі та виконувати інші адміністративні функції.

Ці ролі важливі для підтримки безпеки сайту та управління доступом до функцій та вмісту. За допомогою Middleware у Laravel можна забезпечити те, що користувачі можуть виконувати тільки ті дії, до яких у них є доступ, та забезпечують контроль над тим, хто може втручатися у вміст бази даних та структуру сайту.

Діаграма прецедентів наведена на кресленику ІА94.160БАК.004 Д1.

На діаграмі прецедентів чітко визначений перелік можливостей для кожної ролі [13]. Проаналізувавши дану діаграму, можна зробити висновок, що користувач, який зацікавлений лише в перегляді основної інформації про автосалон, а також каталогу автомобілів, може залишатися гостем, але якщо відвідувач веб-застосунку також хоче мати доступ до підтримки, коментування або замовлень, то йому слід зареєструватися в системі, або авторизуватися у тому випадку, якщо він вже має власний аккаунт.

На діаграмі видно, що у застосунку буде присутня так звана ієрархія ролей. Тобто актори розміщені як: Гість, Авторизований користувач, Адміністратор. Відповідно, кожен наступний актор наслідує всі можливості попереднього.

Варто більш детально розглянути можливості кожного актора.

Гість:

– переглянути основні сторінки: головна, каталог, контакти. Фактично, ці сторінки доступні будь-якому відвідувачу веб-сайту та містять лише загальну інформацію;

– переглянути кожний автомобіль більш детально. Також будь-який користувач може подивитися інформацію про автомобіль, що його зацікавив;

– фільтрувати або сортувати автомобілі в каталозі. Всім доступна функція сортування та фільтрації авто, щоб підібрати те, що найкраще підходить;

– зареєструватися або авторизуватися в системі. Безперечно, гості повинні мати можливість зареєструватися або авторизуватися в системі, якщо вони вже мають обліковий запис для того, щоб мати можливість користуватися привілеями інших груп.

Авторизований користувач:

– керувати налаштуваннями облікового запису. Авторизовані користувачі можуть змінювати ім'я в обліковому записі, електронну пошту та пароль. А також можна видалити свій аккаунт;

– залишати коментарі на сторінці автомобіля. Користувачі можуть від свого імені коментувати автомобілі для того щоб поділитися досвідом з іншими, або залишити таким чином відгук;

– використовувати чат з підтримкою. Також користувачі можуть консультуватися зі штучним інтелектом щодо автомобілів, таким чином отримуючи поради або необхідну інформацію;

– оформлювати замовлення. На сторінці з автомобілем можна натиснути кнопку «Замовити автомобіль». В результаті, після додаткового підтвердження, замовлення буде оформлене;

– створювати власні конфігурації автомобіля. Для деяких автомобілів доступна можливість налаштування власної конфігурації перед замовленням;

– отримувати електронні листи про зміну статусу замовлення. При оформленні нового замовлення або зміні його статусу адміністратором застосунку, користувач отримує електронний лист на пошту, що зв'язана з аккаунтом з детальною інформацією;

– переходити на сторінку власних замовлень та мати можливість їх скасувати. Одразу після успішного оформлення замовлення, користувач потрапляє на сторінку з власними замовленнями, де він може слідкувати за статусом виконання замовлення, або скасувати його, якщо воно ще не виконане.

					ІА94.160БАК.004 ПЗ	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

Адміністратор:

- додавати нові автомобілі в каталог. Адміністратор може додавати нові автомобілі шляхом заповнення спеціальної форми з усіма даними про автомобіль;
- додавати додаткові фотографії на сторінку автомобіля. При необхідності, окрім обов'язкової основної фотографії, можна додати ще додаткові на власний розсуд;
- змінювати інформацію про автомобілі в каталозі. Адміністратор може змінити будь-які дані про певний автомобіль;
- змінювати (додавати та видаляти) додаткові фотографії на сторінці автомобіля. Адміністратор може видалити одну або декілька додаткових фотографій на сторінці автомобіля та, за бажанням, додати інші;
- видаляти автомобілі з каталогу. Можна в одне натискання видалити будь-який автомобіль, при чому, замовлення, що були зроблені на нього, залишаються активними;
- змінювати статус всіх замовлень. Можна обрати будь-який статус для будь-якого оформленого замовлення у системі;
- додавати нові опції для автомобіля. За допомогою кнопки у адміністраторському каталозі автомобілів є можливість додати певні опції для обраного авто;
- дізнаватися про нові замовлення через Телеграм. Завдяки використанню спеціального плагіна, при оформленні нового замовлення користувачем, у канал адміністрації автосалону в Телеграм надходить інформація про нове замовлення.

3.3 Розробка бази даних

База даних відіграє важливу роль у роботі веб-застосунку. Для функціонування системи автоматизації робочих процесів автосалону використовуватиметься БД MySQL [14], яка відповідатиме за наступні сутності:

- автомобілі;

					ІА94.160БАК.004 ПЗ	Арк.
						33
Зм.	Лист	№ докум.	Підпис	Дата		

- користувачі;
- коментарі до автомобілів;
- замовлення;
- опції для автомобілів;
- зв'язок між замовленнями та опціями;
- зв'язок між опціями та категоріями.

Діаграма бази даних наведена на кресленику ІА94.160БАК.004 Д2.

Розглянемо структуру бази даних більш детально. У таблиці «cars» зберігається вся необхідна інформація про автомобілі. Вона використовується насамперед для автоматичної генерації сторінки кожного автомобіля, а також для доповнення інформації у замовленнях. Окремо варто виділити відношення композиції additional_images та main_images до таблиці «cars». Для зручності, вони були виділені в окремі сутності. Також варто наголосити, що власне фотографії знаходяться окремо на сервері, а у базі даних зберігаються лише їх імена. Відповідно, у полі додаткових фотографій зберігається текстовий масив з необхідними іменами. Це зроблено для того, щоб оптимізувати швидкість взаємодії з базою даних, а також не жертвувати якістю фотографій.

Таблиця 3.1 – Структура таблиці автомобілів

Атрибут	Тип	Опис
id	Integer	Ідентифікатор автомобіля
make	Varchar	Марка автомобіля
model	Varchar	Модель автомобіля
year	Integer	Рік випуску
price	Integer	Ціна
body_type	Varchar	Тип кузова
transmission	Varchar	Привід
doors	Integer	Кількість дверей
engine_type	Varchar	Тип двигуна

engine_power	Integer	Потужність двигуна
torque	Integer	Крутний момент двигуна
acceleration	Double	Розгін до 100 км/год
top_speed	Integer	Максимальна швидкість
description	Text	Опис
main_image	Text	Основне зображення
additional_images	Text	Додаткові зображення

У наступній таблиці «users» зберігається вся інформація про облікові записи користувачів. У сфері торгівлі не дуже важливі біологічні дані клієнта, тому таблиця має нескладну структуру. Це зроблено також для того, щоб не навантажувати клієнтів вводом зайвої інформації.

Отже, фактично у даній таблиці використовується стандартна структура, визначена за замовчуванням Laravel Breeze, що відповідає за реєстрацію та авторизацію. Особливої уваги заслуговує поле `is_admin`, яке може приймати 1 або 0, що відповідає булевим значенням `True` або `False` відповідно. За замовчуванням зареєстрованому користувачу присвоюється значення 0, але воно може бути змінено вручну через панель бази даних. Відповідно до значення цього поля у користувача, йому відображаються різні сторінки, а також надаються різні можливості, що було описано на діаграмі прецедентів.

Таблиця 3.2 – Структура таблиці користувачів

Атрибут	Тип	Опис
id	Integer	Ідентифікатор користувача
name	Varchar	Ім'я користувача
email	Varchar	Електронна пошта
password	Varchar	Пароль
is_admin	Integer	Є адміністратором або ні

У таблиці «comments» містяться коментарі, які користувачі залишають на сторінці автомобілів. Коментарі не можуть існувати без автомобіля, тому вони автоматично видаляються при видаленні відповідного екземпляра.

Таблиця 3.3 – Структура таблиці коментарів

Атрибут	Тип	Опис
id	Integer	Ідентифікатор коментаря
body	Varchar	Текст коментаря
user_id	Integer	Ідентифікатор автора
car_id	Integer	Ідентифікатор автомобіля

У таблиці «orders» зберігаються замовлення, які були оформлені користувачами. Замовлення ніколи не видаляються з системи тому що, за логікою сфери продаж, завжди повинен бути доступ до історії замовлень для звітності та загальної статистики.

Таблиця 3.4 – Структура таблиці замовлень

Атрибут	Тип	Опис
id	Integer	Ідентифікатор замовлення
user_id	Integer	Ідентифікатор клієнта
car_id	Integer	Ідентифікатор автомобіля
status	Varchar	Статус замовлення
total_price	Decimal	Сума замовлення

Таблиця «options». Вона містить дані про опції кожного автомобіля та не може існувати окремо. Всі опції поділяються на обмежену кількість категорій. Категорія певної опції вказана у стовпці category. Таким чином ми можемо обирати поступово опції з кожної категорії для відображення.

Таблиця 3.5 – Структура таблиці опцій

Атрибут	Тип	Опис
id	Integer	Ідентифікатор замовлення
car_id	Integer	Ідентифікатор автомобіля
category	Varchar	Категорія опції
name	Varchar	Назва опції
price	Decimal	Ціна

Таблиця «option_categories» має обмежену кількість записів та слугує для додавання більш читабельних імен для категорій. Тому що в таблиці опцій вони позначені скорочено англійською. Тому треба подавати їх в більш звичному та зручному вигляді для користувача. Для цього встановлений зв'язок у моделі опцій, який дозволяє отримувати нормальну назву категорії, звертаючись до опції.

Таблиця 3.6 – Структура таблиці відповідності категорій

Атрибут	Тип	Опис
id	Integer	Ідентифікатор
system_name	Varchar	Назва опції в системі
display_name	Varchar	Назва опції для користувача

Остання таблиця – «order_option» слугує для реалізації зв'язку типу багато до багатьох. Відповідно, її структура обмежена зв'язуючими ідентифікаторами.

Таблиця 3.7 – Структура таблиці зв'язку замовлень та опцій

Атрибут	Тип	Опис
id	Integer	Ідентифікатор
order_id	Integer	Ідентифікатор замовлення
option_id	Integer	Ідентифікатор опції

3.4 Розробка серверної частини

Серверна частина веб-застосунку для автоматизації процесів роботи автосалону відповідає за реалізацію бізнес-логіки. Там містяться класи для обробки та перетворення даних, відображення представлень, відправки електронних листів, роботи з базою даних та ін.

Діаграма класів наведена на кресленику ІА94.160БАК.004 ДЗ.

Розглянемо більш детально структуру серверної частини Laravel-додатку [15], яка міститься в каталозі «app»:

– controllers: Цей каталог містить контролери, які є важливою частиною архітектури MVC. Контролери — це класи, що управляють бізнес-логікою додатку. Вони приймають вхідні дані від користувача через запити, обробляють ці дані, а також взаємодіють з моделями для зміни або отримання даних. Контролери відповідають за координацію різних ресурсів: моделей, представлень, сервісів, вони об'єднують ці компоненти разом;

– requests: Каталог Requests використовується для зберігання класів запитів. Ці класи слугують для валідації вхідних даних від користувача, перш ніж контролер обробить ці дані. Це допомагає забезпечити, що дані, які доходять до бізнес-логіки додатку, відповідають вимогам та високим стандартам якості;

– middleware: Тип фільтраційного програмного забезпечення в Laravel, що виконує певні задачі, такі як перевірка аутентифікації, перевірка ролей та дозволів, перед тим як запит досягне контролера. Middleware може бути застосований до окремих маршрутів або груп маршрутів, що забезпечує гнучкість управління доступом. Зазвичай Middleware застосовується у файлі маршрутів web.php;

– mail: В цьому каталозі розташовані класи, що відповідають за створення та відправку електронних листів;

– models: Моделі — це одна з найважливіших частин кожного Laravel-додатку, що відповідають за взаємодію з базою даних. Вони представляють

структуру таблиці в базі даних і дозволяють виконувати різноманітні операції, такі як вставка, видалення, оновлення та вибірка даних;

– providers: Провайдери сервісів — це основний спосіб завантаження компонентів в Laravel. Вони розташовані в каталозі Providers і відповідають за ініціалізацію різних сервісів, таких як роутинг, події, щоденники тощо.

Варто розглянути більш детально вміст каталогу Controllers. Спочатку зробимо огляд підкаталогу Auth. Він містить контролери, що відповідають за реалізацію різних аспектів аутентифікації в Laravel. Breeze — це пакет, що забезпечує прості сторінки аутентифікації, зокрема вхід, реєстрацію, відновлення пароля та верифікацію електронної пошти. Він був спеціально розроблений Laravel, тому його інтеграція забезпечується розробниками фреймворку, на відміну від аналогічних пакетів, розроблених сторонніми компаніями або особами. Розглянемо детальніше призначення кожного контролера:

– AuthenticatedSessionController.php – відповідає за реалізацію входу в систему і виходу з неї. Він містить методи для відображення форми входу, обробки запиту на вхід, а також обробки запиту на вихід;

– ConfirmablePasswordController.php – використовується для підтвердження пароля користувача перед виконанням дій, що вимагають підвищеного рівня безпеки;

– EmailVerificationNotificationController.php – відповідає за відправку нового листа з повідомленням про верифікацію електронної пошти користувачеві;

– EmailVerificationPromptController.php: відображає відповідне повідомлення користувачеві, якщо його електронна пошта ще не була підтверджена;

– NewPasswordController.php – слугує для обробки форми встановлення нового пароля після скидання;

– PasswordController.php – відповідає за зміну пароля авторизованого користувача;

– PasswordResetLinkController.php – відповідає за відображення форми запити на скидання пароля і відправку посилання для скидання пароля користувачу;

– RegisteredUserController.php – потрібен для обробки реєстрації нового користувача;

– VerifyEmailController.php – відповідає за обробку посилання верифікації електронної пошти.

Також окрему увагу необхідно виділити користувацьким контролерам:

– CarController.php – найбільш навантажений контролер у застосунку. Відповідає за всю логіку роботи з автомобілями: відображення карток у каталозі, а також їх сортування та пошук; відображення адміністраторського списку автомобілів; валідація отриманих зі сторінки додавання нового автомобіля даних та запис їх у базу даних, а також збереження картинок на сервері; відображення сторінки кожного автомобіля; редагування та видалення автомобілів; робота зі списком опцій;

– CatalogController.php – відповідає за виклик відображення каталогу автомобілів;

– ChatController.php – отримує повідомлення від користувача та займається взаємодією з мовною моделлю GPT-3.5 Turbo. В результаті чого повертає відповідь на повідомлення користувача, який відображається у відповідному представленні;

– CommentController.php – відповідає за збереження та видалення коментарів на сторінках з автомобілями;

– ConfigurationController.php – займається зберіганням та збереженням опції для конфігурації автомобілів;

– ContactController.php – відповідає за виклик відображення сторінки контактів;

– Controller.php – базовий клас, який розширюють усі інші контролери в каталозі;

					ІА94.160БАК.004 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

– `HomeController.php` – відповідає за виклик відображення домашньої сторінки;

– `OrderController.php` – ще один навантажений контролер, що реалізує логіку замовлень. Він дозволяє зберігати нові замовлення, скасовувати власні замовлення користувачам, змінювати статус замовлення адміністраторам, виводити окремі списки замовлень звичайному користувачеві та адміністратору, а також відображати сторінку з підтвердженням замовлення;

– `ProfileController.php` – відповідає за редагування та видалення профіля користувача.

Також важливо пам'ятати про `Middleware`. Хоч воно і має доволі примітивну структуру, його використання дійсно допомагає при розробці. Достатньо лише вказати потрібне `Middleware` у файлі роутинга, і цей шлях одразу стає обмежений для певних груп користувачів. Завдяки цьому можна зекономити час для написання коду для обмеження кожної сторінки.

У нашому випадку використовується два класи: `Authenticate.php` та `IsAdmin.php`. Перший дозволяє доступ до маршруту тільки для авторизованих користувачів, а другий – тільки для адміністраторів. Це дозволяє реалізувати представлену раніше діаграму прецедентів, яка відображає ієрархію ролей у системі. Відповідно, гість може відвідувати сторінки, які не захищені `Middleware`, авторизований користувач – які також захищені `Authenticate`, а адміністратор усі попередні, а також захищені `IsAdmin`.

Laravel має вбудований функціонал для відправки електронних листів. Він реалізований у вигляді класу `Mailable`.

Варто детальніше роздивитися структуру `Mailable` класу. За замовчуванням він складається з функцій `envelope`, `content` та `__construct`. Останній метод з перелічених вказує що очікується у якості вхідних даних під час виклику даного класу. `Envelope` визначає тему листа, а також може вказувати відправника, якщо це необхідно. `Content` вказує яке представлення використовуватиметься як шаблон для листа, а також те, які дані та в якому вигляді будуть туди передані.

					ІА94.160БАК.004 ПЗ	Арк.
						41
Зм.	Лист	№ докум.	Підпис	Дата		

Як вже було сказано раніше, користувачські класи, які розширюють базовий клас Model та знаходяться в однойменному каталозі, відповідають за зв'язок з відповідними таблицями у базі даних. За замовчуванням це працює так, що, наприклад, класу з назвою Car відповідає таблиця cars, але це можна змінити, вказавши відповідний параметр при ініціалізації моделі за допомогою Artisan або власне у коді класу

На прикладі еталонної моделі Cars також можна описати базові можливості класу Model. Насамперед, у створених моделях вказується список атрибутів, яким доступне масове присвоєння. Для цього списку передбачена зарезервована змінна \$fillable. Навпаки, можна вказати атрибути, яким прямо заборонене масове присвоєння. Це робиться для безпеки застосунку, щоб користувач, наприклад, не зміг змінити значення атрибуту is_admin для власного облікового запису та не отримав таким чином права адміністратора.

Також важливим моментом є пряме вказання відношень сутностей. Це можна побачити на прикладі відношення між сутностями автомобілів та коментарів. Таким чином, Car використовує метод hasMany, що посилається на клас Comment, який, у свою чергу, використовує метод belongsTo, що посилається на клас Car. Це дозволяє отримувати доступ до властивостей дочірнього або батьківського об'єкта, початково звертаючись до зв'язаної з ним сутності.

У каталозі Notifications містяться класи повідомлень. За замовчуванням цей каталог відсутній та з'являється лише після додавання першого класу повідомлення за допомогою Artisan. Взагалі Notifications викликаються при певній дії. В даному випадку він слугує для відправки повідомлень про нові замовлення в робочий канал в Telegram та викликається під час створення нового замовлення разом з відправкою електронних листів.

3.5 Розробка клієнтської частини

Клієнтська частина представлена Blade шаблонами – простою, але потужною системою шаблонів, вбудованою в Laravel. Blade не обмежується використанням простого PHP в представленнях. На відміну від багатьох інших систем шаблонів, він не розробляє свої власні структури контролю потоку або проміжні змінні. Замість цього, Blade використовує директиви для перекладу ванільного PHP, що дозволяє легко створювати динамічні веб-сторінки.

Файли, що відносяться до клієнтської частини, знаходяться в каталозі `views`, тобто представлення. Подальше сортування представлень за підкаталогами носить суто логічний характер. Зазвичай це роблять відповідно до смислового розподілу.

В каталозі `admin` розміщуються представлення, до яких має доступ лише адміністратор, що гарантується використанням `Middleware`. Тобто це сторінка створення нового автомобіля, редагування вже створеного автомобіля, адміністраторський список автомобілів, сторінка керування замовленнями та додавання нових опцій.

В каталозі `emails` знаходяться представлення, які викликаються класом `Mailable` та являються шаблонами для електронних листів покупцям. Від них зазвичай потребується максимально мінімалістичний дизайн, тому що різні поштові клієнти мають різні можливості для відображення листів користувачам. Відповідно, чим складніший буде дизайн електронного листа, тим більше вірогідність того, що користувач побачить лист не так, як потрібно. У даному застосунку використовуються два шаблони: для підтвердження нового замовлення та для повідомлення покупця про зміну статусу оформленого раніше замовлення.

В каталозі `layouts` знаходяться шаблони, які являються основою для інших представлень. Це допомагає значно скоротити об'єм коду у кожному файлі за рахунок того, що спільні частини для кожної сторінки знаходяться в батьківському файлі, а інші файли просто розширюють його, додаючи власні дані у вигляді секцій.

					ІА94.160БАК.004 ПЗ	Арк.
						43
Зм.	Лист	№ докум.	Підпис	Дата		

Основний шаблон розмічений таким чином, що він очікує певні секції, для яких вже визначені місця в загальному коді HTML-сторінки. В даному застосунку в першу чергу використовуються шаблони `app.blade.php` та `user_app.blade.php`. Перший слугує базою для панелі користувача та інших сторінок з налаштуваннями. Він має зовсім інше оформлення для того щоб не навантажувати клієнтів та адміністратора зайвими декоративними елементами там, де це не потрібно. У свою чергу, він використовує `navigation.blade.php` для додавання спадаючого меню у верхньому правому кутку.

Шаблон `user_app.blade.php` являється головним та використовується для генерації всіх сторінок, які повинні мати встановлене оформлення з шапкою, меню та рештою елементів. В даному випадку залишається лише розширити його іншим файлом, додавши відповідний код до тегу `<body>`.

В каталозі `order` розміщені представлення, що стосуються замовлень. Файл `index.blade.php` показує користувачеві його замовлення та надає можливість скасувати ті, які ще не виконані. Представлення `confirm.blade.php` і `thankyou.blade.php` мають нескладну структуру та відповідають за підтвердження замовлення та подяку за оформлене замовлення.

Каталог `partials` призначений для зберігання фрагментів представлень, які використовуються неодноразово. Таким чином, замість того щоб писати один шматок коду кілька разів, він просто виноситься в окремий файл та використовується там, де потрібно. Важливо усвідомлювати різницю між `partials` та `layouts`. Вони являються схожими за смислом, але `layouts` являють собою готовий шаблон для веб-сторінки, а `partials` лише випадкову частину коду. Тобто, `header.blade.php` та `footer.blade.php` містять шапку сайту з меню та нижню частину сайту зі службовою інформацією відповідно. А `scripts.blade.php` зберігає повний список залежностей, таких як Bootstrap, Ajax, css та js файлів, слайдера.

Значна частина представлень не належить до підкаталогів та знаходиться прямо в каталозі `views`. Вони відповідають за відображення загальних сторінок,

					ІА94.160БАК.004 ПЗ	Арк.
						44
Зм.	Лист	№ докум.	Підпис	Дата		

таких як головна, контакти, каталог автомобілів, сторінка автомобіля, центр допомоги, профіль користувача тощо.

Висновок до розділу 3

У даному розділі був виконаний значний обсяг розробки та аналізу, що включає опис ролей користувачів, створення моделі бази даних і детальне розглядання структури файлів серверної та клієнтської частин веб-застосунку.

Були визначені ключові ролі в системі – Гість, Авторизований користувач та Адміністратор, і описані їхні можливості. Визначення цих ролей є важливим кроком для правильного розуміння функціональності системи та можливих сценаріїв використання.

Також була приділена увага створенню моделі бази даних та визначенню основних сутностей та зв'язків між ними. Таким чином була отримана добре структурована та гнучка основа для зберігання та керування даними в веб-застосунку.

Крім цього, була розглянута структура файлів серверної частини проєкту, зокрема в контексті фреймворку Laravel. В рамках цього були розглянуті способи організації коду в Laravel, такі як контролери, моделі, middleware, об'єкти класу Mailable та інше.

Нарешті, була описана структура файлів клієнтської частини веб-застосунку. Тобто був розглянутий принцип побудови веб-сторінок в Laravel та особливості реалізації представлень в рамках даного проєкту.

					ІА94.160БАК.004 ПЗ	Арк.
						45
Зм.	Лист	№ докум.	Підпис	Дата		

4 ДЕМОНСТРАЦІЯ РОБОТИ СИСТЕМИ

Наразі веб-застосунок знаходиться на локальному сервері за адресою “<http://127.0.0.1:8000>”. У якості програмного забезпечення для розгортання локального сервера та бази даних MySQL використовується XAMPP.

Як вже було вказано раніше, а також відображено на діаграмі прецедентів, у застосунку присутні 3 ролі: Гість, Авторизований користувач та Адміністратор. Кожна з ролей має доступ до обмеженого переліку сторінок. Тож варто розглянути кожен з ролей для кращого уявлення.

Карта сайту наведена на кресленику ІА94.160БАК.004 Д4. Відповідно до неї, далі розглядатимуться конкретні сторінки веб-застосунку.

4.1 Гість

Роль гостя призначена в першу чергу для ознайомлення з застосунком та для перегляду каталогу автомобілів. При вході на веб-сайт за адресою “<http://127.0.0.1:8000>”, користувач потрапляє на домашню сторінку.

Ласкаво просимо до автосалону Lev Motors!

Ми пропонуємо вам широкий вибір автомобілів від провідних світових виробників.

Широкий вибір автомобілів

У нас ви знайдете автомобілі на будь-який смак: від компактних міських автомобілів до розкішних спортивних машин.

Гнучка система оплати

Ми пропонуємо різні варіанти оплати, включаючи кредит і лізинг. Наша мета - зробити процес покупки автомобіля якомога зручнішим для вас.

Якісний сервіс

Наша команда професіоналів завжди готова допомогти вам з вибором автомобіля і відповісти на всі ваші питання.

Про нас

Автосалон LevMotors був заснований у 2023 році з метою надати клієнтам найкращий вибір автомобілів та забезпечити найвищий рівень обслуговування. Ми постійно вдосконалюємо свої навички і розширюємо асортимент, щоб кожен клієнт міг знайти ідеальний автомобіль. Разом з цим ми прагнемо до постійного поліпшення якості нашого сервісу, тому ваші відгуки та пропозиції завжди важливі для нас.

Рисунок 4.1 – Початкова сторінка

					ІА94.160БАК.004 ПЗ	Арк.
						46
Зм.	Лист	№ докум.	Підпис	Дата		

Початкова сторінка призначена для першого знайомства користувача з застосунком та містить загальну інформацію про автосалон.

Окрему увагу варто приділити шапці сайту:



Рисунок 4.2 – Шапка сайту

У верхньому правому кутку для гостей відображається панель, яка надає можливість авторизуватися під власним обліковим записом або зареєструватися, якщо він ще не створений. Нижче розміщене меню для простої та зручної навігації по основним сторінкам веб-застосунку.

Рисунок 4.3 – Сторінка реєстрації

Сторінка реєстрації не перенавантажена зайвими даними та орієнтована на те, щоб надати користувачеві можливість якнайшвидше зареєструватися та отримати доступ до всіх можливостей авторизованого користувача. Логін використовується для відображення імені користувача у верхньому правому кутку сторінки, а також для вказання автора коментаря. Крім того, він потрібен для звернення до користувача під час відправки електронних листів для підтвердження замовлення та зміни статусу замовлення.

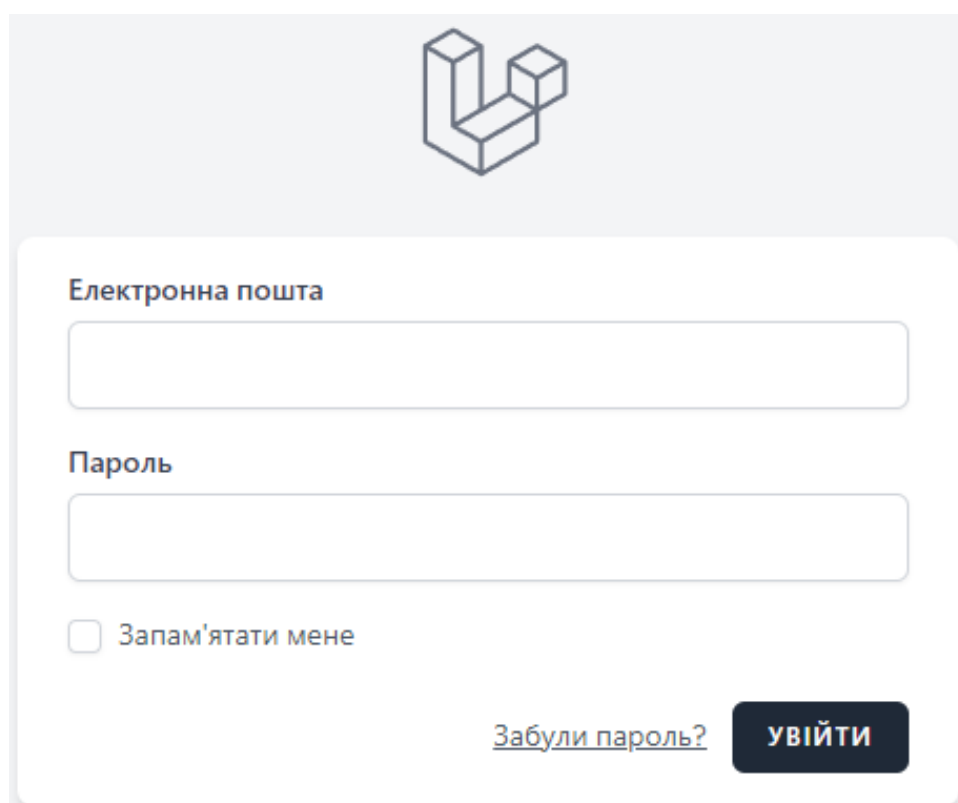


Рисунок 4.4 – Сторінка авторизації

Для авторизації достатньо ввести правильний логін та пароль. Проте, якщо користувач забув свій пароль, то не варто засмучуватися, тому що є можливість скинути пароль за допомогою прив'язаної адреси електронної пошти. В такому випадку, на електронну пошту надійде лист з унікальним посиланням, при переході по якому користувач матиме змогу встановити новий пароль для свого облікового запису.



Забули свій пароль? Без проблем. Просто повідомте нам свою електронну адресу, і ми надішлемо вам електронною поштою посилання для скидання пароля, за яким ви зможете вибрати новий.

Електронна пошта

ВІДПРАВИТИ ФОРМУ ДЛЯ СКИДАННЯ ПАРОЛЯ

Рисунок 4.5 – Сторінка скидання пароля

Повертаючись до основної частини веб-застосунку, окрему увагу варто приділити каталогу автомобілів.

Каталог автомобілів

Введіть марку або модель для пошуку...

Сортувати за... Порядок...

Шукати



Dodge Challenger 2022
Ціна: 34570 \$
Потужність: 357 к.с.
Трансмісія: RWD
[Детальніше](#)



Lamborghini Huracan STO 2021
Ціна: 331000 \$
Потужність: 640 к.с.
Трансмісія: RWD
[Детальніше](#)



Ferrari SF90 Stradale 2020
Ціна: 650000 \$
Потужність: 780 к.с.
Трансмісія: AWD
[Детальніше](#)

Рисунок 4.6 – Каталог автомобілів

Картки автомобілів створюються автоматично на основі інформації з таблиці «cars» у базі даних застосунку. У якості основного стилістичного шаблону використовується Bootstrap, який має зарезервовані класи саме для карток.

Окрім простого перегляду каталогу, користувачеві доступний пошук автомобілів, який фактично працює як фільтрація за маркою або моделлю автомобіля. Також присутня можливість сортувати картки за ціною автомобіля та потужністю двигуна у порядку зростання чи спадання.

Dodge Challenger 2022

Назад

Ціна від 34570 \$

(залежить від комплектації)



• • • •

Конфігурувати автомобіль

Марка:	Dodge
Модель:	Challenger

Опис автомобіля:

Dodge Challenger — спортивне купе компанії Dodge класу так званих Pony car, який з перервами в різноманітних модифікаціях

Рисунок 4.7 – Сторінка автомобіля

На сторінці певного автомобіля розміщений слайдер з фотографіями даного авто, а також усі технічні характеристики, які можуть зацікавити клієнта. Окремо розміщений опис автомобіля, який у текстовій формі описує переваги даного авто. У верхньому лівому кутку розміщена кнопка «Назад» для швидкого переходу назад до каталогу. Ціна відображається в залежності від можливості конфігурації. Для конфігуруємих автомобілів вказана початкова ціна з базовим набором опцій, для решти статична. Також на сторінці присутня кнопка «Замовити автомобіль» або «Конфігурувати автомобіль», але, при натисканні на неї звичайним гостем, він потрапляє на сторінку авторизації, тому що, згідно діаграмі прецедентів, гість не

може оформити замовлення. У нижній частині сторінки також розміщений розділ з коментарями, але гість не може залишати коментарі та отримує повідомлення «Увійдіть або зареєструйтесь, щоб залишити коментар.» замість поля для вводу, тому це буде розглянуто пізніше.

Контакти

Контактна інформація

Адреса: м. Київ, проспект Перемоги, 37К18

Телефон: +380 111111111

Електронна пошта: info@levmotors.ua

Години роботи

Понеділок - П'ятниця: 9:00 - 18:00

Субота: 10:00 - 16:00

Неділя: вихідний

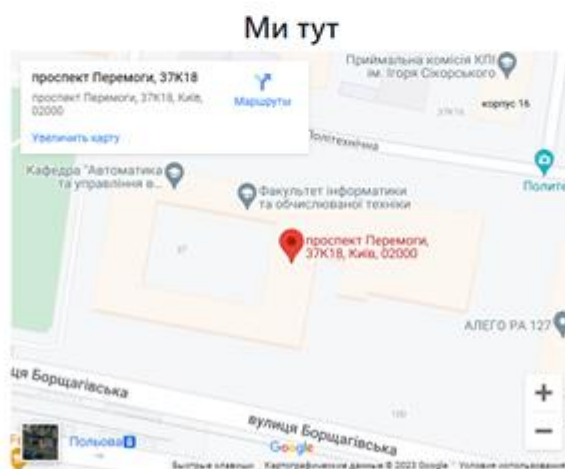


Рисунок 4.8 – Сторінка контактів

На сторінці контактів розміщена інформація для зв'язку з автосалоном, така як телефон, електронна пошта та мапа його фізичного розміщення для того, щоб користувач міг зв'язатися з командою спеціалістів будь-яким зручним способом.

4.2 Авторизований користувач

Окрім базового функціоналу гостя, авторизованому користувачеві також доступна можливість залишати коментарі на сторінках автомобілів:

Максимальна швидкість:	315 км/год
------------------------	------------

Lev Administrator
Коментар

User
Коментар користувача

Опублікувати коментар

Опублікувати коментар

Рисунок 4.9 – Секція коментарів

Як можна побачити на рисунку, коментарі від звичайних користувачів та від адміністраторів відрізняються наявністю помітки «адміністратор» поруч. Проте, на відміну від гостя, авторизованому користувачеві вже показується текстове поле для коментаря.

При натисканні на кнопку «Конфігурувати автомобіль» або «Замовити автомобіль» (в залежності від того, чи присутня у автомобіля можливість конфігурації) на сторінці бажаного автомобіля, користувач потрапляє на сторінку конфігурації, або одразу сторінку підтвердження:

Діаметр дисків

☒ 18 - 0.00 \$

☐ 19 - 550.00 \$

Колір кузова

☐ синій - 1000.00 \$

☐ червоний - 1100.00 \$

☒ срібний - 0.00 \$

Тип дисків

☒ базові - 0.00 \$

☐ титанові - 270.00 \$

Тип сидінь

☒ тканеві - 0.00 \$

☐ шкіряні - 420.00 \$

☐ спортивні ковши - 640.00 \$

Додаткові опції

☒ пакет "Карбон" - 1350.00 \$

☒ комфорт-пакет - 970.00 \$

Оформити замовлення

Рисунок 4.10 – Сторінка конфігурації автомобіля

У випадку наявності можливих конфігурацій у базі даних, сторінка конфігурації генерується динамічно. У даному випадку це працює наступним чином: є обмежений перелік категорій, до яких належать опції (діаметр дисків, колір кузова, тип дисків, тип сидінь, додаткові опції). Якщо автомобіль має хоч одну створену адміністратором опцію, для нього автоматично з'являється сторінка конфігурації перед потраплянням на сторінку підтвердження. На цій сторінці відображаються лише ті категорії, які мають хоч одну додану опцію, а пусті пропускаються. Для кожної категорії, окрім додаткових опцій, можна обрати лише одну зі списку. Для останньої навпаки доступний вибір будь-якої кількості з перелічених. Після підтвердження вибору дані зберігаються у сесії до моменту підтвердження замовлення або початку оформлення нового.

Підтвердження замовлення

Dodge Challenger

Трансмсія: RWD

Рік випуску: 2022р.

Вибрані опції:

Колір кузова: червоний (1100.00\$)

Діаметр дисків: 19 (550.00\$)

Тип дисків: базові (0.00\$)

Тип сидінь: шкіряні (420.00\$)

Вартість автомобіля: 36640 \$

Підтвердити замовлення

Відмінити замовлення

Рисунок 4.11 – Сторінка підтвердження замовлення

На цій сторінці клієнту надається можливість ще раз подумати перед оформленням замовлення. Якщо він впевнений у своєму виборі, то натискає кнопку «Підтвердити замовлення», після чого замовлення додається до бази даних, а користувач отримує повідомлення про успішне оформлення замовлення.

Сторінка підтвердження, відповідно, також генерується відповідно до автомобіля. Тобто, якщо автомобіль не розрахований на конфігурацію, то користувач натискає кнопку «Замовити автомобіль» та потрапляє одразу на цю сторінку. Очевидно, що при цьому область з вибраними опціями не відображається.

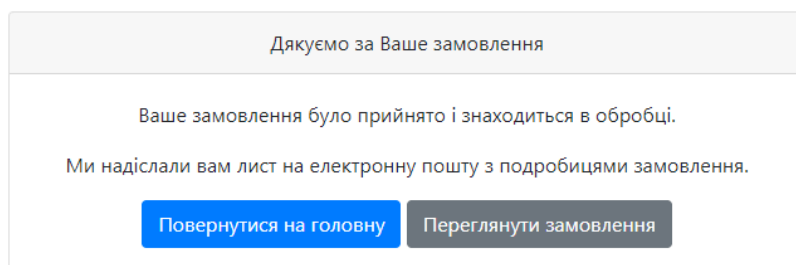


Рисунок 4.12 – Сторінка подяки за замовлення

Крім того, користувач також отримує лист на електронну пошту, вказану при реєстрації.

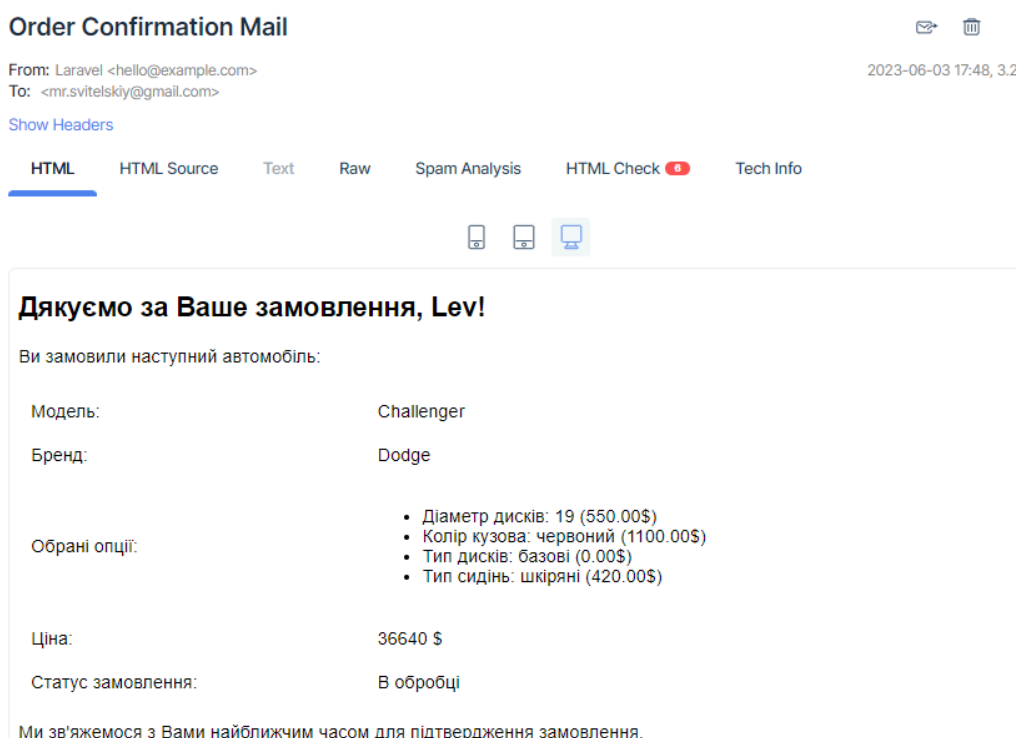


Рисунок 4.13 – Зовнішній вигляд електронного листа про підтвердження замовлення

					ІА94.160БАК.004 ПЗ	Арк.
						54
Зм.	Лист	№ докум.	Підпис	Дата		

При оформленні нового замовлення користувачем, в службовий канал автосалону в Telegram надходить повідомлення про те, що певний користувач залишив нове замовлення на певний автомобіль та обрав деякі опції:

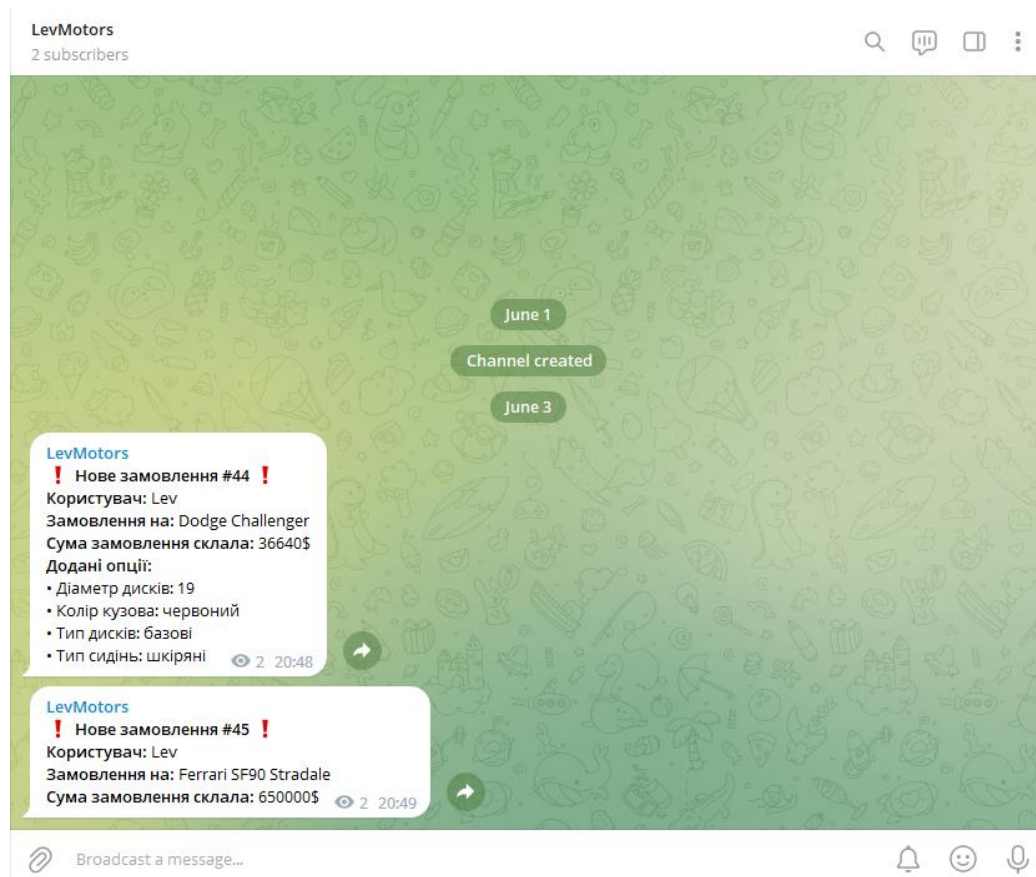


Рисунок 4.14 – Канал в телеграм, куди надходить інформація про нові замовлення

Для реалізації даного функціоналу використовується телеграм-бот, що надсилає повідомлення у канал. Для цього використовується пакет «Telegram Notifications Channel for Laravel» [17].

Якщо після підтвердження замовлення користувач натискає кнопку «Переглянути замовлення», то потрапляє на сторінку з усіма своїми замовленнями.

Мої замовлення						
Номер замовлення	Дата замовлення	Автомобіль	Опції	Вартість замовлення	Статус	
16	2023-05-31 21:15:26	Lamborghini Huracan STO, 2021р.	-	331000.00 \$	В обробці	Скасувати замовлення
27	2023-06-01 11:02:59	Dodge Challenger, 2022р.	Діаметр дисків: 19 Колір кузова: сірий	35120.00 \$	В обробці	Скасувати замовлення
28	2023-06-01 11:14:57	Dodge Challenger, 2022р.	Діаметр дисків: 19 Колір кузова: червоний	36220.00 \$	В обробці	Скасувати замовлення
29	2023-06-01 11:15:30	Lamborghini Huracan STO, 2021р.	-	331000.00 \$	Відмінений	
30	2023-06-01 11:41:22	Dodge Challenger, 2022р.	Діаметр дисків: 19 Колір кузова: синій	36120.00 \$	В обробці	Скасувати замовлення

Рисунок 4.15 – Список замовлень користувача

Користувацькі замовлення відображаються у вигляді таблиці, у якій вказано унікальний номер замовлення у системі, дату та час його оформлення, автомобіль та його ціну, а також статус і список обраних опцій (якщо є). Користувач має можливість скасувати будь-яке своє замовлення, яке ще не виконане за допомогою кнопки поруч.

Інноваційним рішенням в даному веб-застосунку виступає чат з підтримкою. Його особливість полягає в тому, що у якості агента підтримки виступає штучний інтелект на базі мовної моделі GPT-3.5 Turbo [16]. Для взаємодії з ним на серверній частині застосунку встановлений спеціальний пакет, який дозволяє отримати доступ до API. Для того щоб штучний інтелект відповідав у потрібній формі та на потрібну тему, для нього задана спеціальна інструкція, що містить список наявних автомобілів у автосалоні та вказівки щодо комунікації з клієнтами.

User

Доброго дня, допоможіть мені, будь ласка, обрати автомобіль

Менеджер:

Залежно від ваших потреб і бюджету можливо варто розглянути такі варіанти: - Якщо вам потрібен автомобіль для міських поїздок, то можна дивитися на хетчбеки чи седани. Наприклад, Mazda 3 або Toyota Corolla. - Якщо ви часто їздите за місто або на тривалу відстань, то варто звернути увагу на кросовери. Наприклад, Honda CR-V або Toyota RAV4. - Якщо вам потрібен автомобіль для великої сім'ї або для перевезення великих вантажів, то можна розглянути мінівени або SUV. Наприклад, Toyota Sienna або Ford Expedition. При виборі автомобіля також враховуйте його технічний стан, рівень споживання палива, марку та якість запчастин.

User

Я би хотів придбати люксовий суперкар, що можете порадити?

Менеджер:

Якщо ви шукаєте люксовий суперкар, то можу порадити наступні моделі: 1. Lamborghini Aventador - один з найбільш вражаючих і потужних суперкарів з V12-двигуном. 2. Bugatti Chiron - найшвидший, найпотужніший і найдрібніший суперкар, який створювався з моменту запуску марки. 3. Ferrari LaFerrari - гібридний суперкар з F1-технологією, який є одним з найшвидших і найбільш престижних автомобілів на світі. 4. Pagani Huayra - ручної роботи суперкар з неповторним дизайном і потужним двигуном V12. 5. Koenigsegg Jesko - найшвидший серійний автомобіль на світі з максимальною швидкістю понад 480 км / год. Звичайно, ці суперкари не є доступними для більшості людей, вони дуже дорогі і вимагають великого вміння водіння. Тому перш ніж придбати суперкар, переконайтеся, що він відповідає вашим потребам і вмінням.

Введіть повідомлення тут...

Відправити

Рисунок 4.16 – Інтерфейс чату з підтримкою

Чат реалізований у приємному нескладному дизайні та розміщений на окремій сторінці для зручності використання. Для початку діалогу достатньо зайти на сторінку та ввести перше повідомлення у поле для вводу, після чого надіслати його за допомогою клавіші «Enter» або кнопки «Відправити». Через невеликий проміжок часу користувач отримає повідомлення від віртуального менеджера, який завжди готовий допомогти з вирішенням проблеми.

Головна сторінка

Панель користувача

User

Панель користувача

Профіль

Панель користувача

Вийти

Мої замовлення

Рисунок 4.17 – Панель користувача

На панелі керування користувача відображені доступні додаткові можливості, до яких не передбачений доступ з інших місць. На даний момент, для звичайного користувача тут передбачена кнопка для переходу на сторінку всіх замовлень, яка також доступна після оформлення нового замовлення.

Рисунок 4.18 – Налаштування профілю

Остання доступна сторінка користувачеві – налаштування профілю. Тут він має змогу змінити свій логін, адресу електронної пошти, пароль, або взагалі видалити свій аккаунт. При тому, якщо користувач побажає видалити аккаунт, то для безпеки встановлений додатковий фактор захисту – у спливаючому вікні він повинен ввести пароль та підтвердити вибір.

4.3 Адміністратор

Серед усіх акторів адміністратор має найбільший спектр можливостей. В першу чергу це проявляється у керування списком автомобілів та замовлень. Розглянемо можливості адміністратора ближче:

The screenshot shows a user interface for an administrator. At the top, there is a box with the text 'Максимальна швидкість:' and '315 км/год'. Below this, there are two sections: 'Lev адміністратор' with 'Коментар' and 'Видалити коментар' link, and 'User' with 'Коментар користувача' and 'Видалити коментар' link. At the bottom, there is a text input field with the placeholder 'Опублікувати коментар' and a button labeled 'Опублікувати коментар'.

Рисунок 4.19 – Секція коментарів для адміністратора

Перша особливість проявляється в тому, що адміністратор має можливість видаляти будь-які коментарі на сторінках автомобілів.

The screenshot shows a user interface for an administrator. At the top, there is a navigation bar with 'Головна сторінка' and 'Панель користувача' (highlighted). To the right is a dropdown menu labeled 'Lev'. Below the navigation bar, there is a section titled 'Панель користувача' containing three buttons: 'Створити новий автомобіль' (blue), 'Перейти до списку автомобілів' (grey), and 'Перейти до списку замовлень' (green).

Рисунок 4.20 – Секція коментарів для адміністратора

Крім того, панель керування адміністратора має інший функціонал – замість списку замовлень, тут знаходяться кнопки для створення нового автомобіля, переходу до адміністраторського списку автомобілів або замовлень користувачів.

Додати автомобіль

Марка

Модель

Рік випуску

Ціна

Тип кузова

Трансмісія

Оберіть трансмісію



Кількість дверей

Тип двигуна

Потужність двигуна (к.с.)

Крутний момент (Нм)

Розгін до 100 км/год (с)

Максимальна швидкість (км/год)

Основне зображення

Выберите файл

Файл не выбран

Додаткові зображення

Выбрать файлы

Файл не выбран

Опис автомобіля

Додати автомобіль

Рисунок 4.21 – Сторінка додавання нового автомобіля

Для додавання нового автомобіля адміністратор повинен заповнити приємну для ока форму, яка складається переважно з текстових полів, а також зі спадаючого списку для вибору приводу та двох вікон для зображень. При чому перше вікно обов'язково очікує одну фотографію, а друге можна пропустити, або додати декілька фотографій одразу.

					ІА94.160БАК.004 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис	Дата		

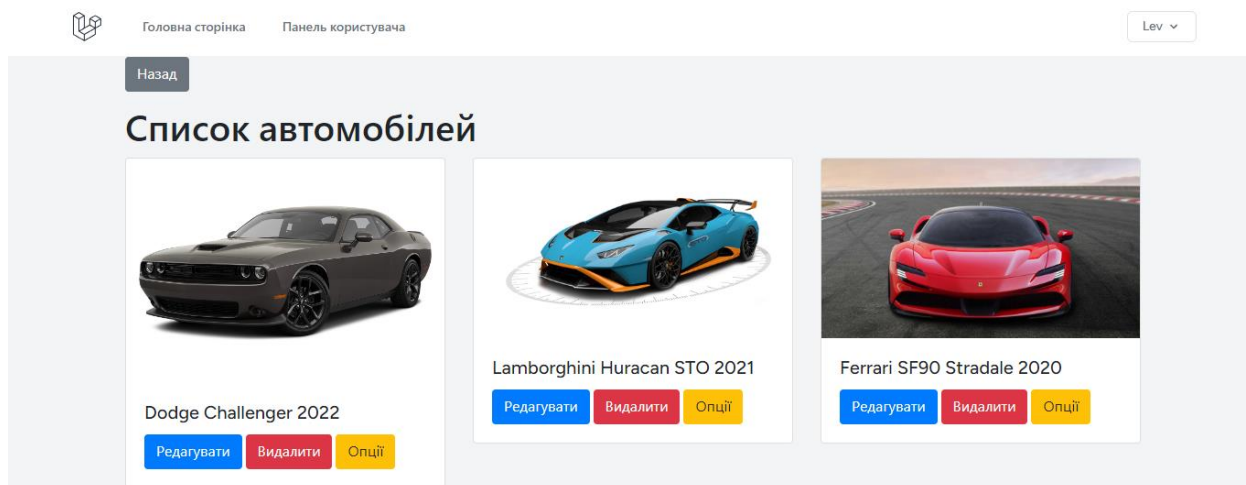


Рисунок 4.22 – Адміністраторський список автомобілів

Як вже було зазначено раніше, адміністратор має окремий список автомобілів, розширений додатковими можливостями. Замість можливості перегляду сторінки автомобіля, як у звичайного користувача, адміністратору доступна можливість редагувати дані про автомобіль, видалити його (але не повністю, тому що для сутності автомобіля використовується метод «Soft delete», який фактично не видаляє запис з бази даних, а додає поле про видалення, таким чином зв'язані записи про замовлення зберігаються) або додати нові опції.

Додати опцію для Dodge Challenger

Категорія

Назва

Ціна

Рисунок 4.23 – Сторінка для додавання опцій

Опції поділяються на 5 типів: колір кузова, тип дисків, діаметр дисків, тип сидінь та додаткові опції. Строго визначений список категорій потрібен для коректного відображення конфігурацій на відповідній сторінці.

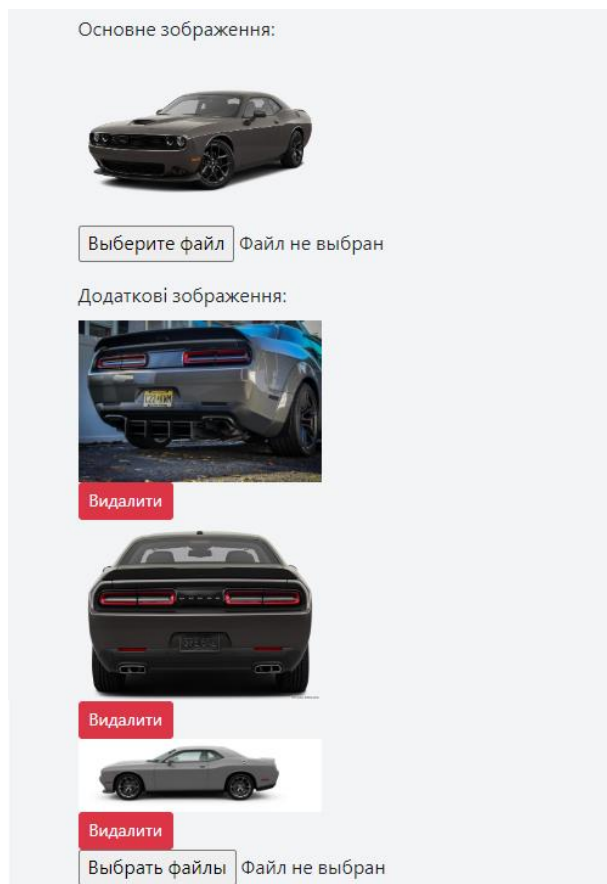


Рисунок 4.24 – Секція для керування зображеннями на сторінці редагування автомобіля

В цілому, сторінка для редагування автомобіля виглядає так же, як і для створення, основна відмінність полягає в логіці фотографій. Усі додані на даний момент фотографії відображаються на сторінці в мініатюрі. Адміністратору надається можливість заміни головного зображення – для цього достатньо просто обрати інше зображення з файлової системи. Також він має змогу видалити додаткові зображення та додавати нові. Для цього поряд з кожним зображенням розміщена кнопка «Видалити».

Назад					
Список замовлень					
#	Ім'я замовника	Електронна пошта замовника	Марка та модель автомобіля	Обрані опції	Статус замовлення
14	User	user@gmail.com	Dodge Challenger, 2022р.	Колір кузова: червоний	В обробці
16	Lev	mr.svitelskiy@gmail.com	Lamborghini Huracan STO, 2021р.	-	В обробці
27	Lev	mr.svitelskiy@gmail.com	Dodge Challenger, 2022р.	Діаметр дисків: 19 Колір кузова: сірий	В обробці
28	Lev	mr.svitelskiy@gmail.com	Dodge Challenger, 2022р.	Діаметр дисків: 19 Колір кузова: червоний	В обробці
29	Lev	mr.svitelskiy@gmail.com	Lamborghini Huracan STO, 2021р.	-	Відмінений
30	Lev	mr.svitelskiy@gmail.com	Dodge Challenger, 2022р.	Діаметр дисків: 19 Колір кузова: синій	В обробці

Рисунок 4.25 – Сторінка керування замовленнями

Сторінка керування замовленнями відіграє ключову роль в системі автоматизації. Вона дозволяє адміністратору змінити статус будь-якого замовлення. Це працює наступним чином: він натискає на спадаючий список поряд з потрібним замовленням та обирає новий статус. Після цього поряд зі списком з'являється кнопка «Зберегти». При натисканні на неї зміни записуються в базу даних, адміністратору відображається зелена табличка про успішну зміну статусу, а замовнику на електронну пошту надходить відповідний лист:

Статус вашого замовлення змінений

Вітаємо, User,

Статус вашого замовлення #4 на Dodge Challenger був змінений.

Попередній статус: Затриманий

Поточний статус: Прийнятий до роботи

Ми продовжимо тримати вас в курсі про будь-які подальші оновлення вашого замовлення.

Наша компанія завжди на зв'язку!

Рисунок 4.26 – Електронний лист про зміну статусу замовлення

Висновок до розділу 4

В даному розділі був розвернутий на локальному сервері веб-застосунок, реалізацію якого було описано в розділі 3. В результаті чого можна було пересвідчитися в його працездатності та ступені реалізації запланованого функціоналу. Окрім того, були виконані наступні дії:

- наочно розглянуті можливості всіх трьох акторів: Гостя, Авторизованого користувача та Адміністратора;
- був повністю описаний функціонал системи автоматизації;
- була продемонстрована у вигляді рисунка кожна створена сторінка (або, принаймні, її фрагмент);
- були уточнені деякі тонкощі програмної реалізації певних компонентів;
- фактично наведено інструкцію по користуванню застосунком.

Отже, можна зробити висновок, що весь запланований функціонал був реалізований та розглянутий.

					ІА94.160БАК.004 ПЗ	Арк.
						64
Зм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

В ході виконання дипломного проєкту були досліджені системи автоматизації робочих процесів автосалонів. При аналізі існуючих рішень виділені особливості кожного з них та визначити оптимальний формат веб-застосунку, а також основні вимоги для його розробки та реалізації.

В якості основи для веб-застосунку був обраний PHP-фреймворк Laravel. Він надає інструменти для зручної та ефективної реалізації задуманого функціоналу, а також дозволяє додавати до проєкту та використовувати сторонні технології та мови програмування. Зокрема, завдяки гнучкому налаштуванню середовища, яке надає Composer, вдалося додати пакети для використання OpenAI API, що дозволило отримати доступ до штучного інтелекту для додавання його у чат підтримки.

В якості програмного забезпечення для розгортання веб серверу та бази даних MySQL був обраний XAMPP. Він надає всі необхідні ресурси для працездатності веб-сайту, при цьому сильно не навантажуючи систему, а також швидко розгортається та має простий і зручний інтерфейс.

В результаті був розроблений веб-застосунок, що дозволяє автоматизувати робочі процеси автосалону. Також була детально розглянута структура серверної та клієнтської частини системи для кращого розуміння принципів функціонування Laravel застосунку та особливостей розробки даного проєкту. Для коректного розподілу можливостей користувачів було виділено три ролі: Гість, Авторизований користувач та Адміністратор. Таким чином, можна вводити обмеження для певних дій або перегляду інформації для окремих груп користувачів.

Крім того, був проведений детальний огляд створених веб сторінок для кожної ролі користувачів, який також може слугувати інструкцією для роботи з застосунком та загальним описом його функціоналу. Отже, система автоматизації справно функціонує і вирішує поставлені перед нею завдання.

					ІА94.160БАК.004 ПЗ	Арк.
						65
Зм.	Лист	№ докум.	Підпис	Дата		

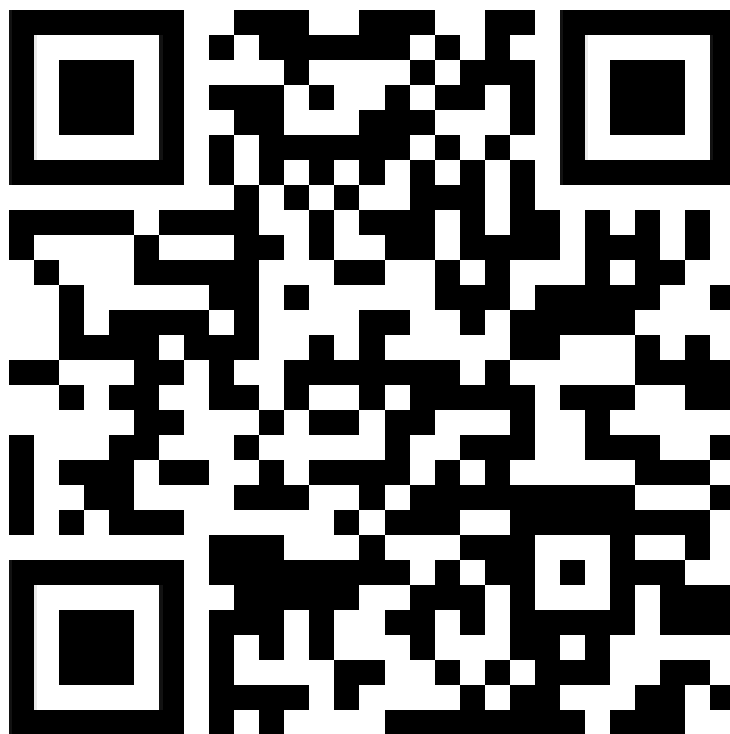
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. JM&A Group. Посібник по розвитку бізнесу у сфері торгівлі автомобілями. JM&A Group Resources, 2022 [Електронний ресурс]. – Режим доступу: <https://www.jmagroup.com/resources/operations/fiawa/ultimate-guide-to-dealership-growth>.
2. Chevalier, Stephanie. Global retail e-commerce sales 2014-2026. Statista, 2022 [Електронний ресурс]. – Режим доступу: <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>.
3. AUTO.RIA [Електронний ресурс]. – Режим доступу: <https://auto.ria.com/uk/>.
4. Winner Automotive [Електронний ресурс]. – Режим доступу: <https://www.winnerauto.ua/>.
5. FIRSTMOTORS [Електронний ресурс]. – Режим доступу: <https://f1rstmotors.com/>.
6. HTML [Електронний ресурс]. – Режим доступу: <https://devdocs.io/html/>
7. JavaScript [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
8. Laravel [Електронний ресурс]. – Режим доступу: <https://laravel.com/docs/10.x>.
9. PHP Documentation [Електронний ресурс]. – Режим доступу: <https://www.php.net/docs.php>.
10. CSS Reference [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/cssref/index.php>.
11. Jimmy Molina Ríos, Nieves Pedreira-Souto. Approach of Agile Methodologies in the Development of Web-Based Software [Електронний ресурс]. – Режим доступу до ресурсу: https://www.researchgate.net/publication/336522596_Approach_of_Agile_Methodologies_in_the_Development_of_Web-Based_Software.

12. Matthias Noback. Advanced Web Application Architecture. Leanpub, 2020. 384 pp.
13. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Print2print, 2016. 208 pp.
14. Jamie Chan. Learn SQL (using MySQL) Fast and Learn It Well. LCF Publishing, 2018. 164 pp.
15. Daniel Correa, Paola Vallejo. Practical Laravel: Develop clean MVC web applications. Independently published, 2022. 177 pp.
16. OpenAI Platform Documentation [Електронний ресурс]. – Режим доступу: <https://platform.openai.com/docs/>.
17. Laravel Notification Channels: Telegram (documentation) [Електронний ресурс]. – Режим доступу: <https://github.com/laravel-notification-channels/telegram>.

ДОДАТОК А

Код програми (репозиторій на GitHub): <https://github.com/Mochilnic/car-dealership-diploma>



					ІА94.160БАК.004 ПЗ	Арк.
						68
Зм.	Лист	№ докум.	Підпис	Дата		