

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Інформаційна система оренди автомобілів»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-72 Літній С.В.

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«___» червня 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Шендрик В.В

Голова комісії

(підпис)

Шифрін Д.М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«___» _____ 2021 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра студентіві

Літній Сергій Володимирович

(прізвище, ім'я, по батькові)

1 Тема проекту Інформаційна система оренди автомобілів

затверджена наказом по університету від «14» квітня 2021 р. №0181-VI

2 Термін здачі студентом закінченого проекту « 07 » червня 2021 р.

3 Вхідні дані до проекту _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) _____

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка

Студент _____

Літній С.В.

Керівник роботи _____

к.т.н., доц. Шендрик В.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Інформаційна система оренди автомобілів».

Пояснювальна записка складається із вступу, трьох основних розділів, висновку, списку використаних джерел із 21 найменування та трьох додатків. Загальний обсяг пояснювальної записки складає 83 сторінки, 41 сторінка з яких основного тексту, 3 сторінки списку використаних джерел, 39 сторінок додатків.

В першому розділі проведено аналіз обраної предметної області. У цьому розділі виконано: огляд актуальних досліджень та публікацій, аналіз аналогів, постановку задачі.

В другому розділі було сформовано мету та задачі дослідження. Даний етап також включає в себе вибір засобів реалізації. В даному розділі описано також проектування інформаційної системи. Було виконано IDEF0, її декомпозиція, діаграма варіантів використання функціоналу інформаційної системи та спроектована база даних.

В останньому, третьому розділі, розділі було виконано реалізацію та детальний опис використання інформаційної системи зі сторони замовника та адміністратора.

Результатом проведеної роботи є розроблена інформаційна система для оренди автомобілів.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, АВТОСЕРВІС, СЕРВІС, АВТО, ЗАМОВЛЕННЯ, ЗАВКА.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Дослідження актуальності проблеми.....	7
1.2 Аналіз сайтів комерційних компаній по оренді автомобілів.....	8
1.3 Постановка задачі проекту.....	16
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКА.....	19
2.1 Вибір засобів реалізації додатка.....	19
2.2 Проектування web-додатку по оренди автомобілів.....	180
2.3. Проектування бази даних.....	21
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ.....	23
3.1 Архітектура веб додатку.....	23
3.2 Використання web-додатку зі сторони користувача.....	25
3.4 Адміністрування сайту.....	34
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТОК А.....	43
ДОДАТОК Б.....	51
ДОДАТОК В.....	64

ВСТУП

Бізнес з прокату автомобілів рухається до самообслуговування та цифрового підходу. Частина причин, чому люди їздять більше за кермом, полягає в тому, що вони відпочивають ближче до дому, оскільки не можуть їздити за кордон. Однак клієнти також повідомляють, що на даний момент почувають себе безпечніше, подорожуючи автомобілем, порівняно з використанням громадського транспорту.

Туристичні страхові компанії також повідомляють, що страховки на прокат автомобілів зростають. Такі компанії, як Allianz Travel , спостерігають більший інтерес до таких продуктів прокату автомобілів, як Rental Car Damage Protector, котрі за 9 доларів на день коштують значно менше, ніж деякі страхові послуги з оренди автомобілів.

Метою кваліфікаційної є розробка web-додатку для по підбору та оренди автомобілю у місті Суми. Інформаційна система допоможе оцінити якість послуг, перевірити репутацію компанії та ознайомитися із діапазоном цін.

У Web-додатку можливість оформити оренду автомобіля матиме певна кількість користувачів. До даних користувачів відноситься адміністратор та звичайний користувач.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

2020 рік став жорстоким роком для багатьох галузей, включаючи автомобільну, з рентабельність та загальна прибутковість опинилась під серйозною загрозою [1-3]. З іншого боку, прокат автомобілів - дуже надійна та приваблива ніша на сьогоднішній час.

Українські послуги з прокату автомобілів розвиваються дуже динамічно, виділяючи їх серед інших сегментів автомобільного ринку. Сильна конкуренція терпить вибивання деяких, й навпаки. Яскравий приклад успіху - прокат автомобілів у Києві «7Cars». З кінця 2013 року він збільшив флот до понад 100 екземплярів у різних класах та відкрив повномасштабні філії у Львові, Одесі та Дніпрі.

У зв'язку з цим набувають популярності такі способи заробітку, як оренда власного автомобіля в довгострокову оренду в успішної орендної компанії. Насправді, багато людей рідко користуються автомобілем. Однак паркування, гараж, страховка повинні сплачуватися регулярно. У той же час не всі хочуть продавати машину[4].

Оренда автомобіля – просто ідеальне рішення в таких випадках. Адже власник отримує стабільну частку доходу, заробленого його «сталевим конем», без будь-яких зусиль. Управлінська компанія обробляє все, починаючи від пошуку клієнтів, створення всіх документів, догляду за автомобілями тощо.

Середній термін окупності оренди нових автомобілів у Києві становить близько двох років. Тому є багато людей, які перетворили суборенду на бізнес. Потім купують нову і знову позичають. Знання, де і як його приймати, є великою користю.

Тенденції на українському ринку на сьогоднішній день свідчать, що попит на оренду автомобілів стабільно зростає у великих містах. Це означає, що, орендуючи машину компанії, можна почуватися впевнено.

Крім того, підвищення інтересу до оренди не означає, що люди перестали думати про переваги покупки автомобіля чи нерухомості, а навпаки. Однак у контексті нової реальності споживачі активно переосмислюють свої звички, а економічна невизначеність, пов'язана з пандемією, змусила споживачів продовжувати витратити більш обережно та ставити пріоритети в оренду.

1.2 Аналіз сайтів комерційних компаній по оренді автомобілів

З часом з'являються все більше і більше угод щодо автомобілів. Якщо користувач шукає вигідну ціну, то це не потребує великих зусиль. На сьогоднішній час є велика конкуренція у даній сфері [5,6].

Для досягнення мети було проведення дослідження сторінки веб-сайтів по оренді автомобілів у різних містах України:

- Автопрокат Суми [6];
- LUXSERVIS[7];
- DriveUp [8].

«Автопрокат Суми» [6] демонструє велику варіативність на своїй домашній сторінці за допомогою слайд-шоу із демонстрацією різних машин, що підлягають оренді та експлуатації.

Це сучасна тенденція у веб-дизайні, яка дуже добре виконана на веб-сайті завдяки використанню професійних фотографій (рис.1.1).



Рисунок 1.1 – Головна сторінка сайту «Автопрокат Сумы»

На сайті також представлений список із цінами на автомобілі, що можна орендувати на даний момент (рис.1.2). Головним недоліком є те, що дана інформація не є динамічною та користувач не має можливості переглянути на сайті, чи є в наявності той чи інший автомобіль.

Автомобиль	1-3 суток	4-7 суток	8-14 суток	15-30 суток	Залог
Dacia Sandero	600 грн.	550 грн.	500 грн.	450 грн.	2000 грн.
Dacia Logan MCV	650 грн.	600 грн.	550 грн.	500 грн.	2000 грн.
Hyundai i20	650 грн.	600 грн.	550 грн.	500 грн.	2000 грн.
Chevrolet Spark AT	750 грн.	700 грн.	650 грн.	600 грн.	2000 грн.
Hyundai Accent AT	850 грн.	800 грн.	750 грн.	700 грн.	2000 грн.
Peugeot 301 AT	850 грн.	800 грн.	750 грн.	700 грн.	2000 грн.

Рисунок 1.2 – Цінова політика на «Автопрокат Сумы»

Також на сайті представлений, власне, сам список автомобілів із зображенням їх зовнішнього вигляду, що є досить зручним (рис.1.3). Переходячи на сторінку з описом будь-якого автомобілю можна побачити детальну інформацію про обрану машину (рис.1.4).

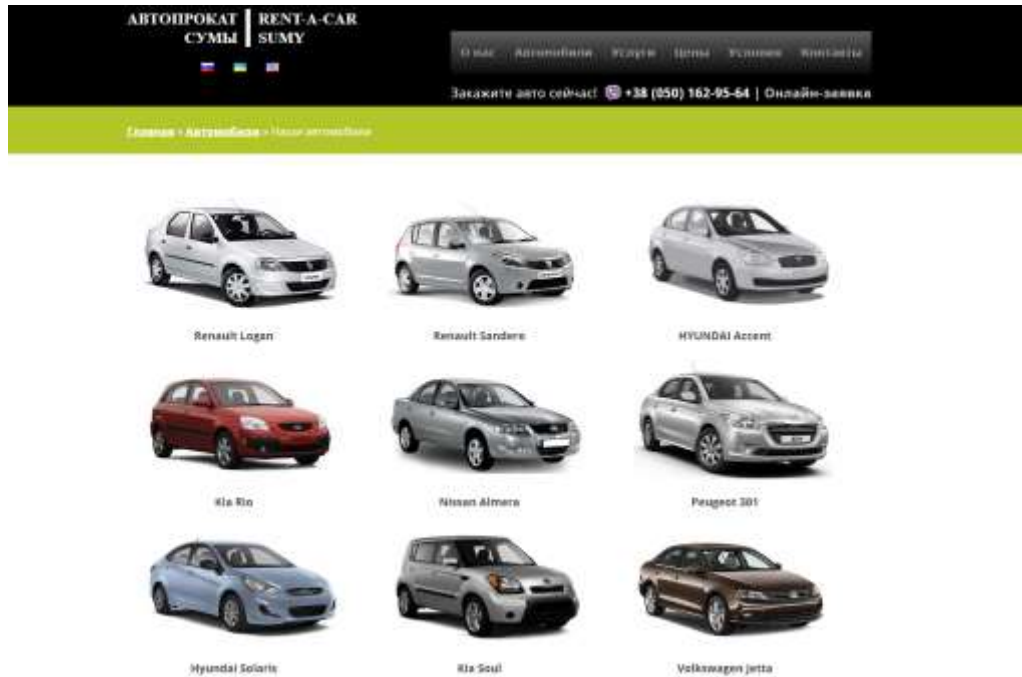


Рисунок 1.3 – Перелік доступних автомобілів на сайті «Автопрокат Суми»

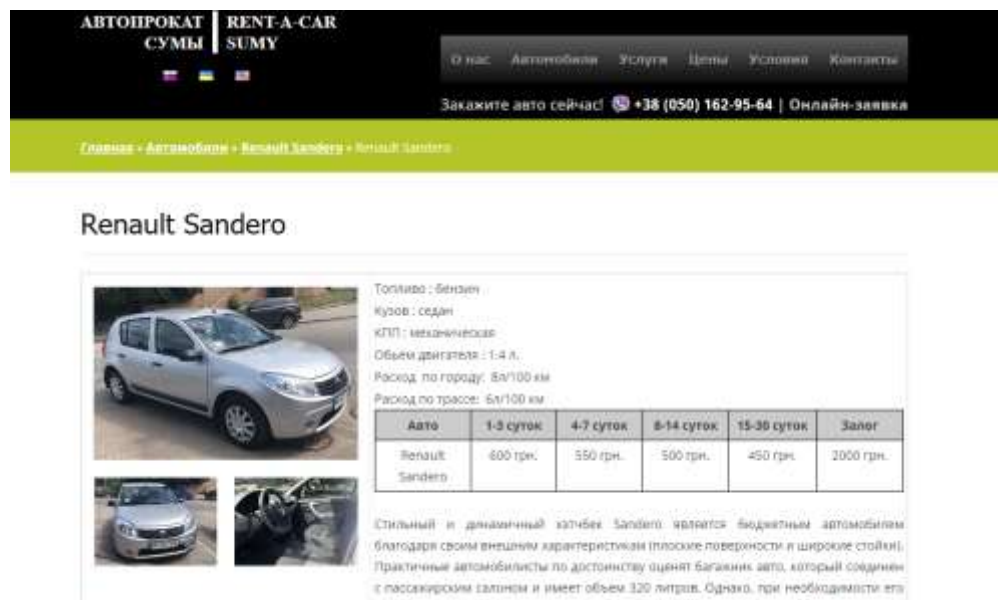


Рисунок 1.4 – Детальна інформація про автомобіль на сайті «Автопрокат Суми»

Головний недолік даного прикладу – відсутність оренди автомобілю безпосередньо на сайті компанії. Для виконання даної операції потрібно зв'язатися із працівниками через номер телефону чи електронну адресу, що знаходиться у розділі «Контакти».

Розглянемо сторінку «LUXSERVIS» [7]. На головній сторінці знаходиться інформація про найпопулярніші авто та функціонал з можливістю фільтрації, щоб дізнатися певну інформацію цікаву користувачу. Звичайно, веб-сайт повинен бути візуально привабливим, читабельним та викликати почуття довіри та лідерства, важливо не використовувати занадто багато елементів. Хоча розробники даного сайту притримувалися таких правил, представлений дизайн не складає позитивного враження (рис.1.5).



Рисунок 1.5 – Головна сторінка сайту «LUXSERVIS»

Можна також відмітити надокучливе модальне вікно, що з'являється посеред екрану через певні проміжки часу (рис.1.6).

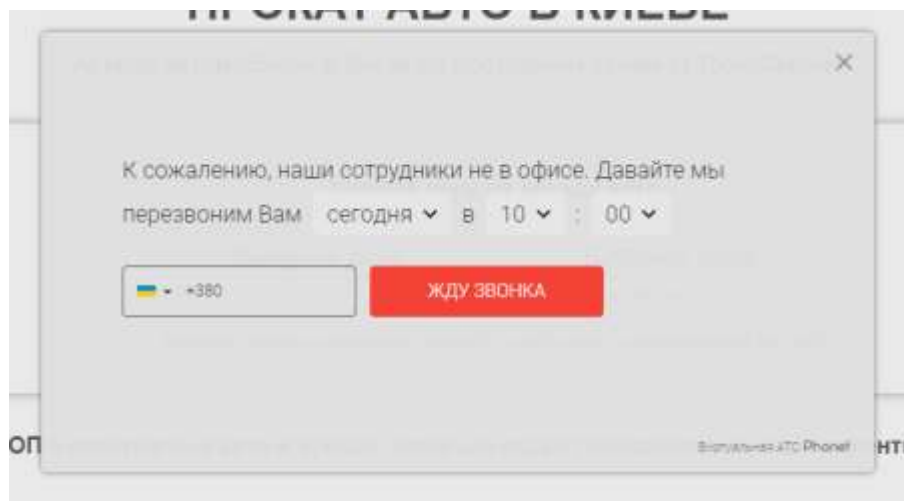


Рисунок 1.6 – Модальне вікно на сайті «LUXSERVIS»

Хоча сайт має окрему сторінку із можливістю перегляду автомобілів для оренди, на сайті відсутня фільтрація саме у даному розділі, що не дозволяє зручно використовувати додаток та виконувати пошук за обраними параметрами (рис. 1.7).



Рисунок 1.7 – Каталог автомобілів на сайті «LUXSERVIS»

Компанія «DriveUp» [8] пишається тим, що не є типовою автомобільною компанією. На домашній сторінці компанія відразу показує результати їх праці у вигляді високоякісної фотографії та професійної анімації.

На головній сторінці представлений блок із можливістю оформити бронювання (рис.1.8).

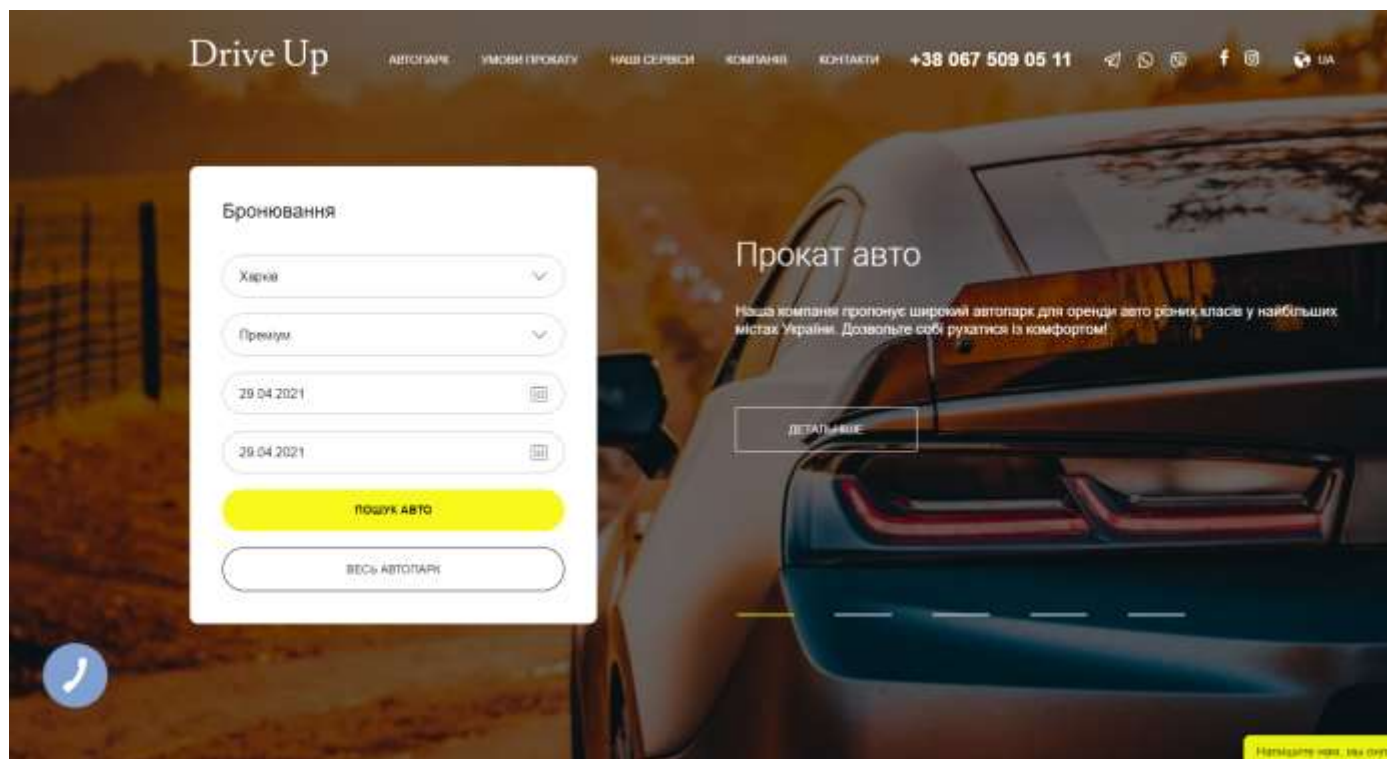


Рисунок 1.8 – Головна сторінка сайту «DriveUp»

Можна також позитивно відмітити реалізацію можливості фільтрації за місцем проживання та оренди авто на сторінках сайту, що представлено на сторінці із умовами прокату автомобілів (рис.1.9).

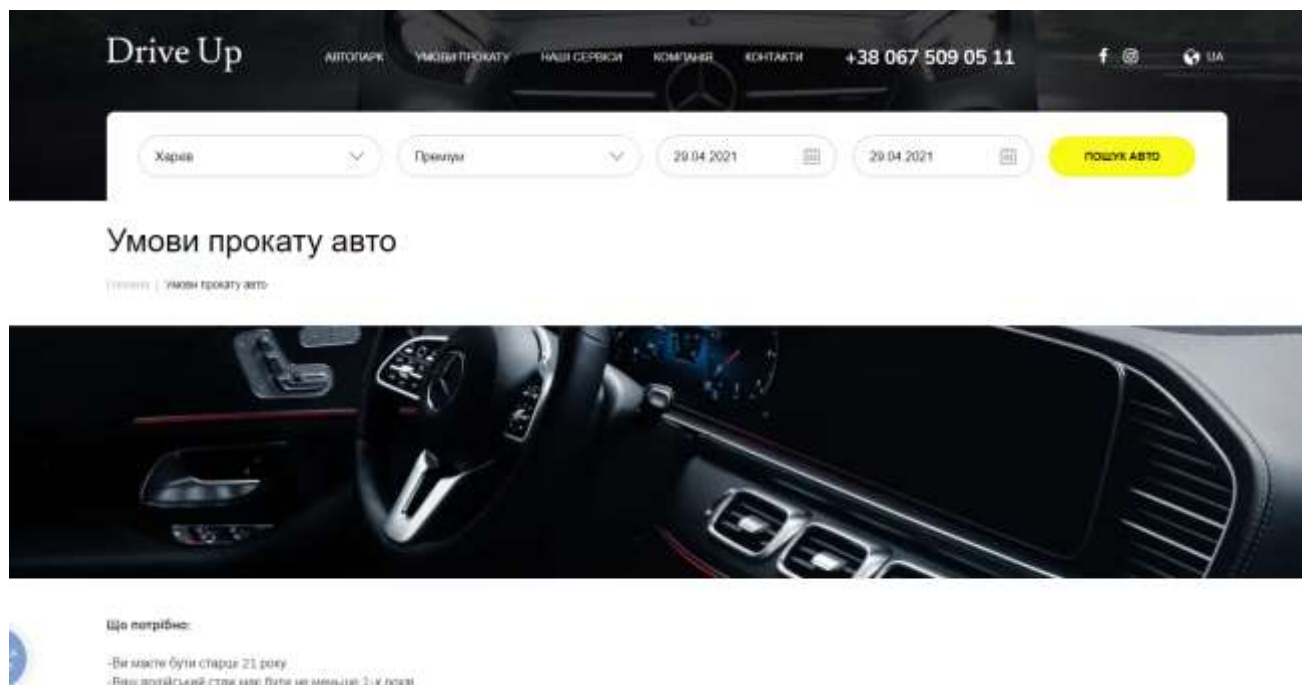


Рисунок 1.9 – Умови прокату на сайті «DriveUp»

На сторінці «Контакти» надана карта, що дозволяє переглянути розміщення компанії в залежності від обраного місця проживання (рис.1.10)

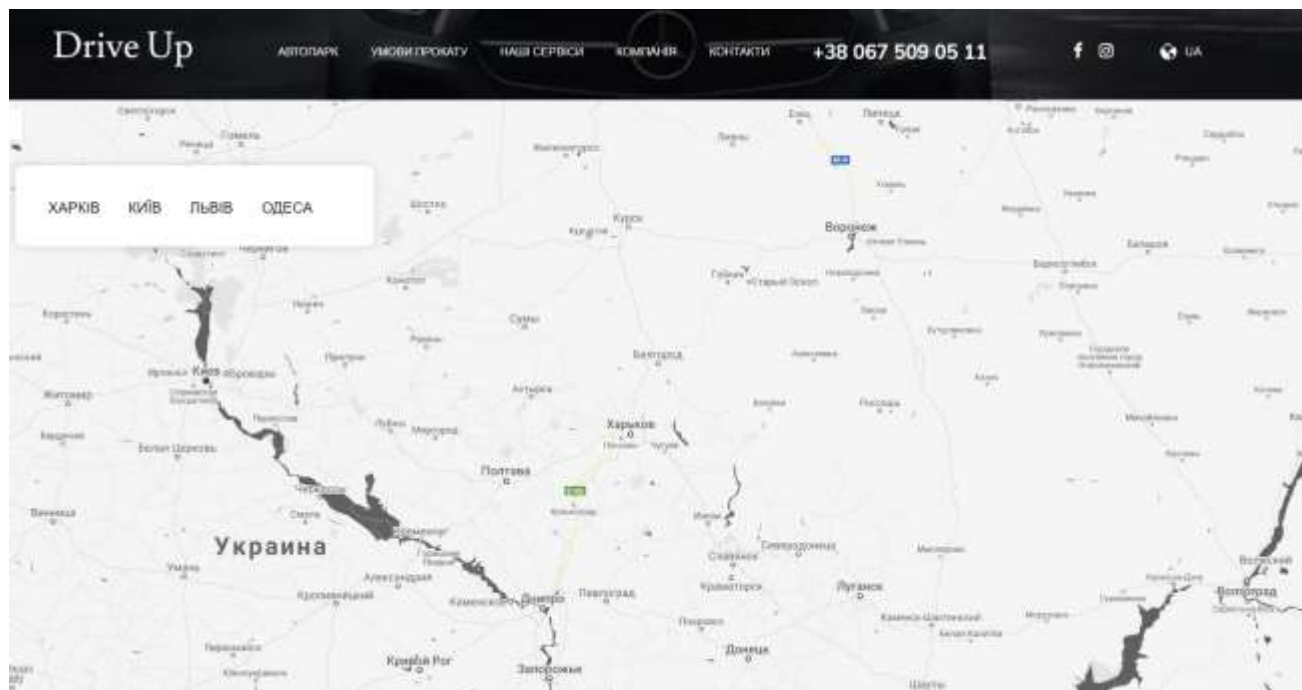


Рисунок 1.10 – Сторінка із контактами на сайті «DriveUp»

Проаналізувавши веб-сторінки по оренді автомобілів в різних містах України, було проведено порівняльний аналіз (табл.1.2), де було виявлено позитивні та негативні сторони. Усі аспекти будуть враховані при роботі над дипломним проектом.

До недоліків можна віднести:

- відсутність фільтрації при пошуку;
- не чітке розподілення інформації на сайті;
- застарілий дизайн;
- наявність надокучливої реклами;
- використання статичних даних на сайті.

До переваг сайтів-аналогів можна віднести:

- можливість зворотного зв'язку;
- можливість детального перегляду інформації про автомобіль;
- зручність навігації.

Таблиця 1.2 – Аналіз сайтів по прокату автомобілів

Критерії	Автопрокат Суми	LUXSERVIS	DriveUp
Можливість зворотного зв'язку	+	+	+
Можливість детального перегляду інформації про автомобіль	+	+	+
Зручність навігації	+	–	+
Застарілий дизайн	+	–	+
Відсутність надокучливої реклами	–	+	+
Динамічна інформація на сайті	–	–	–
Фільтрації при пошуку	–	–	–
Можливість оренди автомобілю онлайн	–	–	–

Отже, при аналізі сайтів-конкурентів по прокату автомобілів можна зробити висновок, що більшість додатків не реалізують функціонал для безпосередньої оренди, а використовуються, як інформативні сторінки.

Головні функції, що повинні бути реалізовані при розробці проекту:

- Оренда автомобілю на сайті.
- Реалізація фільтру для пошуку.
- Можливість реєстрації користувачів.
- Динамічна зміна інформації на сайті.

1.3 Постановка задачі проекту

Метою кваліфікаційної бакалавра є розробка додатку по підбору та оренді автомобілю у місті Суми. Інформаційна система допоможе оцінити якість послуг, перевірити репутацію компанії та ознайомитися із діапазоном цін.

Web-додаток із надання послуг по оренді автомобілів має певну кількість типів користувачів, а саме такі як адміністратор, звичайний користувач.

Головні задачі проекту:

- створення технічного завдання;
- прототипування та розробка макету сайту;
- front-end розробка;
- back-end розробка;
- заповнення бази даних;
- тестування.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКА

2.1 Вибір засобів реалізації додатка

Взаємодія з користувачем є пріоритетом номер один для кожного бізнесу на ринку сьогодні [10-12]. Front-end фреймворки - це перші блоки процесу розробки програмного забезпечення. Але є багато варіантів на вибір, коли мова заходить про створення візуально привабливих програм.

Для створення інтерфейсу інженери використовують комбінацію HTML для базової структури сторінки та вмісту, CSS, для візуального редагування та JavaScript, для створення веб-сайтів інтерактивними [13-15]. Той самий набір інструментів використовується для створення прогресивних веб-програм - мобільних додатків, але створюються із використанням інтерфейсних технологій.

Vue.js – найпопулярніший фреймворк з відкритим кодом JavaScript. Він полегшує розробникам процес створення привабливих та простих у використанні інтерфейсів.

Переваги використання обраного фреймворку:

- швидкий розвиток;
- маленький розмір;
- легкий в обслуговуванні та навчанні;
- може інтегруватися з іншими програмами;
- віртуальний рендеринг і продуктивність dom;
- багаторазове використання компонентів;
- гарна інструментальна екосистема.

Недоліки Vue.js:

- Не має стабільних інструментів;
- Менше плагінів та компонентів.

Загалом back-end складається з трьох основних частин: база даних, застосування, сервер.

На стороні сервера, back-end підсилює роботу веб-сайту. Крім того, програми використовуються для передачі інформації бази даних браузеру [16-19].

Отже, для розробки даної частини буде використовуватися PHP. Дана мова програмування має вбудовані можливості розробки веб-сайтів. PHP сприяє швидкій розробці веб-сайтів, спрощуючи його базовою структурою. Розробники можуть додавати функції в додаток без написання складних і довгих кодів.

2.2 Проектування web-додатку по оренди автомобілів

2.2.1 Моделювання процесу роботи нотації IDEF

Дана діаграма відображає всі дані та вхідну інформацію [20], яка використовується в інформаційній системі для оренди автомобілів. Діаграма нульового рівня відображена на рис. 2.1.



Рисунок 2.1 – Модель IDEF0

Представлений варіант діаграми IDEF1 [21], із детальним розкриттям поетапності, для даного проекту показаний на рис. 2.2.

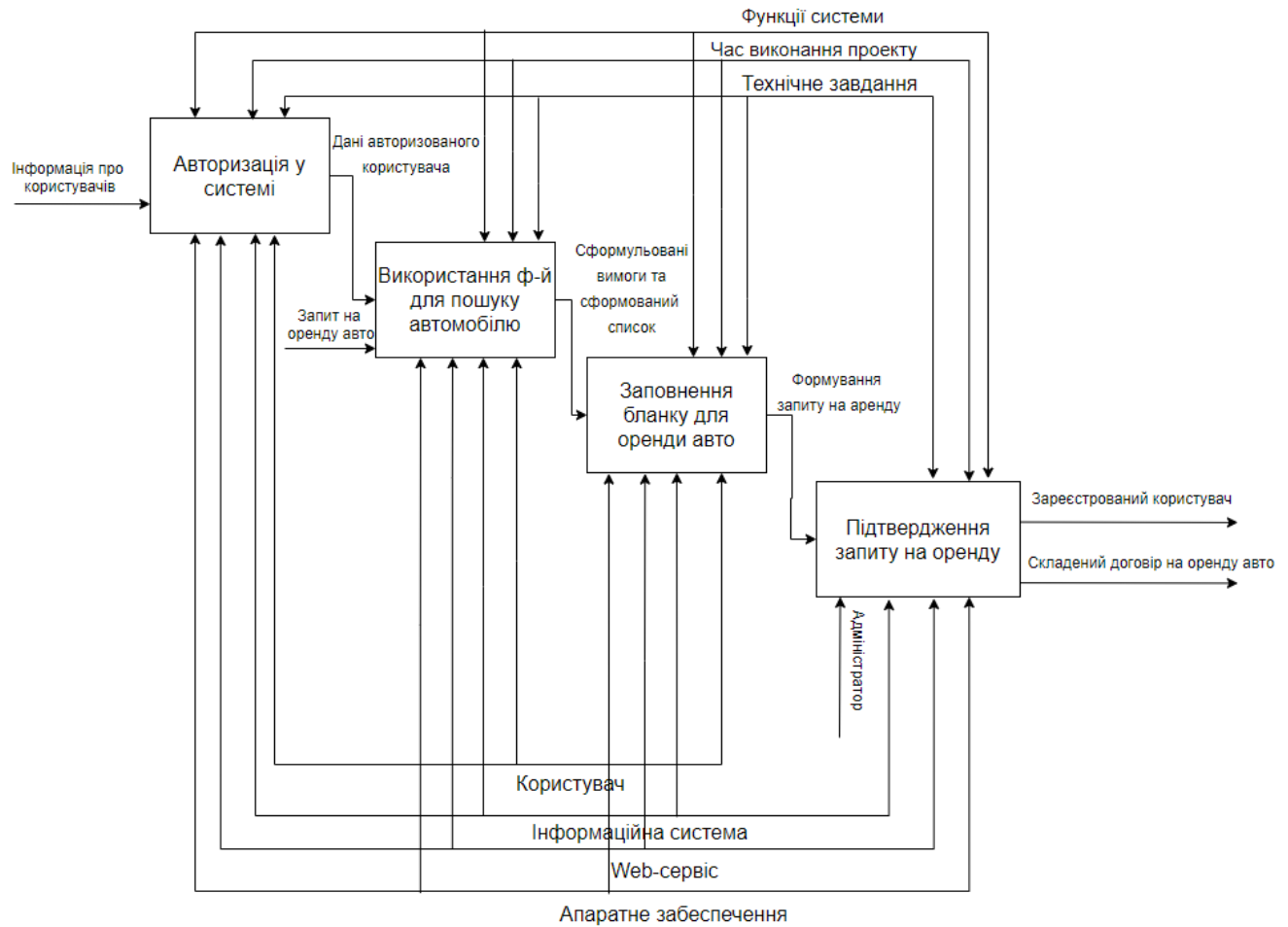


Рисунок 2.2 – Модель IDEF1

2.2.2 Модель варіантів використання веб-додатку

Розглянемо інформацію про акторів створеної системи (табл.2.1) та опис варіантів використання (табл.2.2).

Таблиця 2.1 – Опис акторів-користувачів

Назва	Опис
Користувач	Користувач інформаційної системи для оренди авто.
Адміністратор	Користувач системи у ролі «адміністратора».

Таблиця 2.2 – Опис варіантів використання

Назва	Опис
Авторизація	Функція входу в систему.
Реєстрація	Користувач має можливість реєстрації у системі.
Редагування інформації на сайті	Адміністратор має можливість редагувати дані на сайті.
Додання авто на сайт	Адміністратор може додати інформацію про нові автомобілі на сайт.
Підтвердження оренди	Адміністратор може підтверджувати запит на оренду авто.
Перегляд новин та загальної інформації	Перегляд користувачами інформації на сайті.
Редагування даних на сторінці	Можливість редагування користувачем власних даних.
Пошук на сайті	Користувач має можливість пошуку авто з використанням фільтру
Оренда обраного автомобілю	Можливість оформлення запиту на оренду обраного автомобілю.

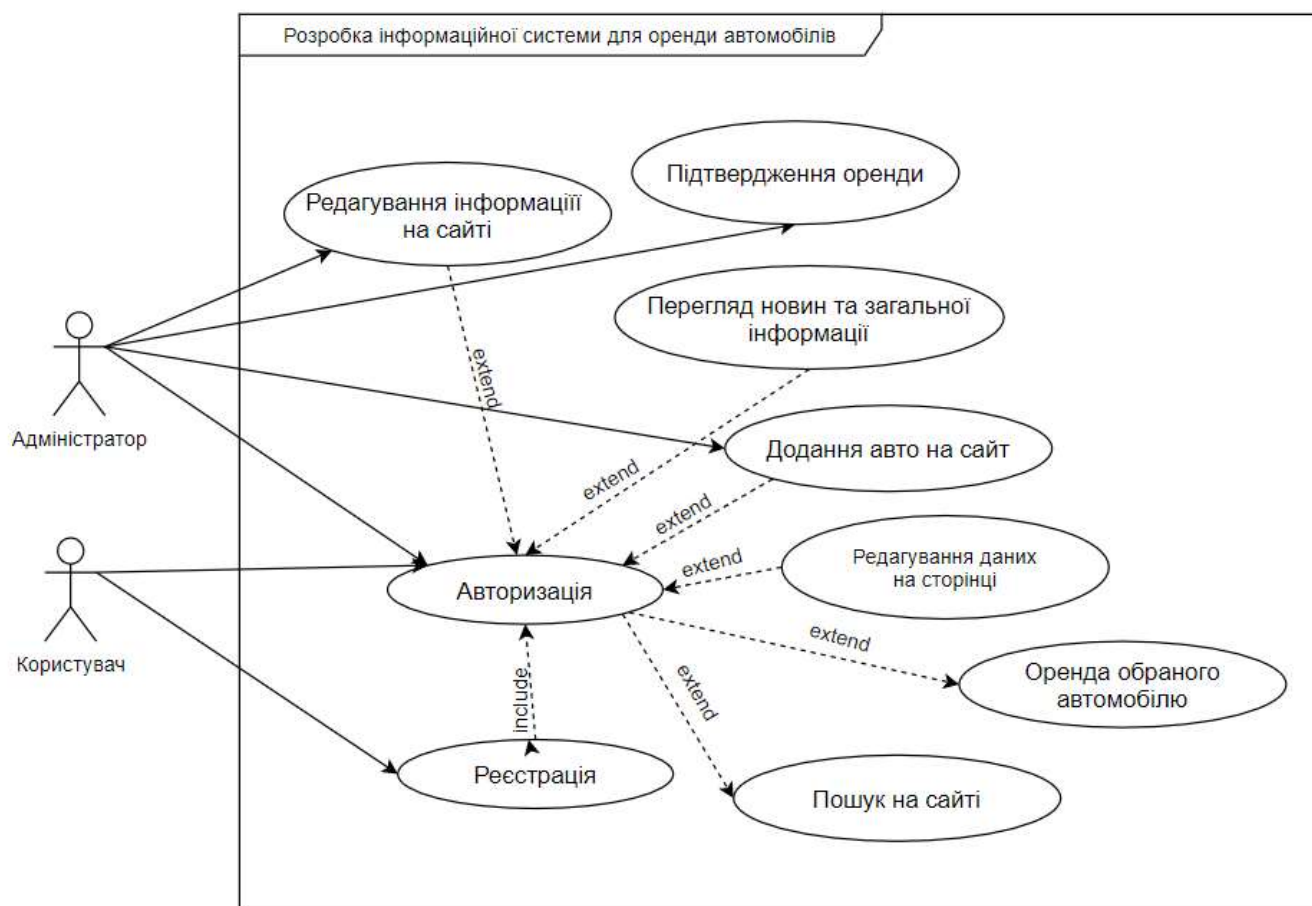


Рисунок 2.3 – Діаграма варіантів використання

2.3. Проектування бази даних

Основний користувач системи – замовник. Використовуючи web-додаток користувач має можливість обрати товар та зробити замовлення.

Зобразимо базу даних web-додатку в ER-діаграмі (рис.2.4).

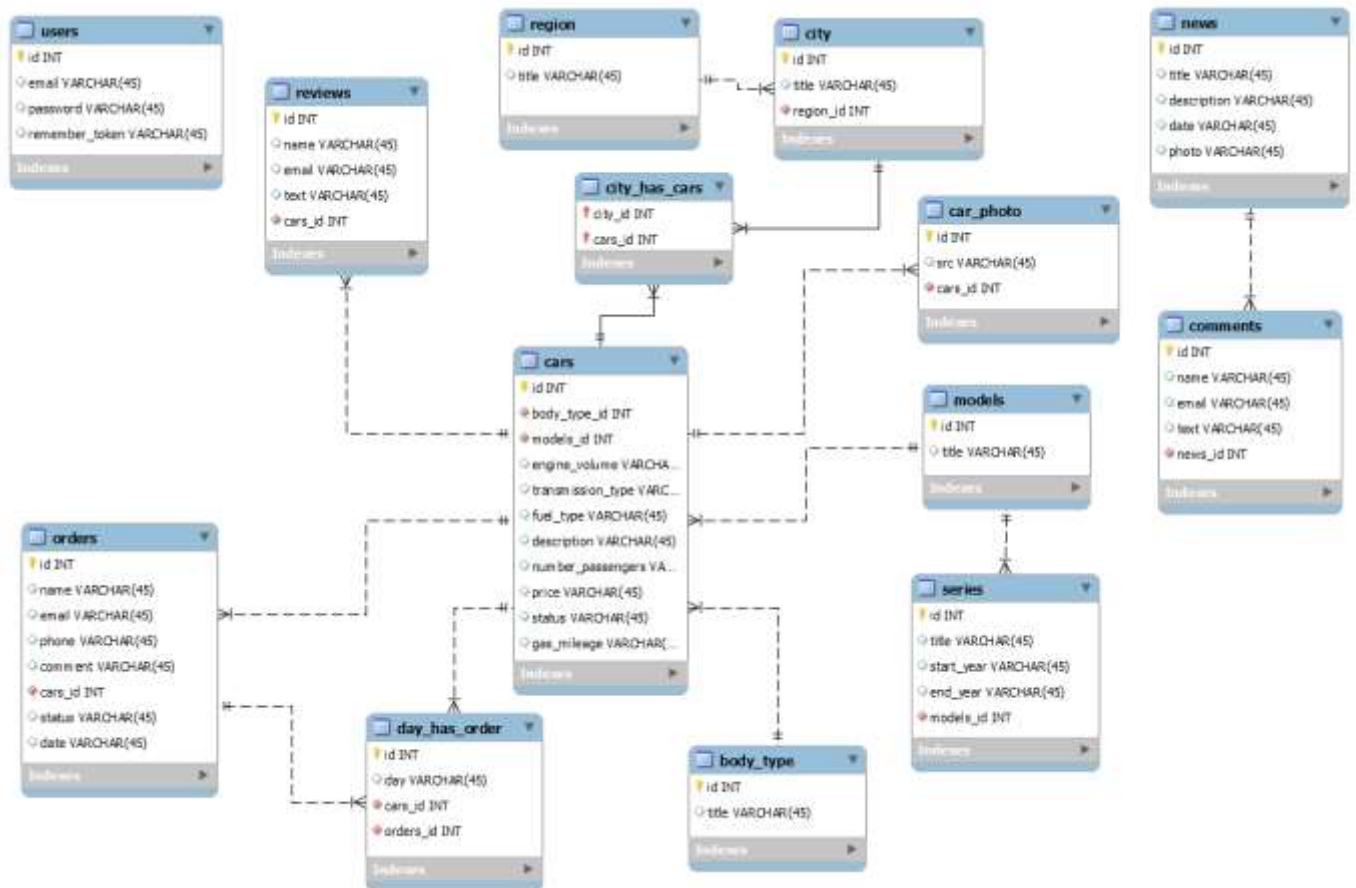


Рисунок 2.4 – ER-діаграма

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ

3.1 Архітектура веб додатку

Для побудови web-додатку було використано фреймворк php – laravel. Обраний спосіб реалізації дозволяє створювати web-додаток на основі MVC архітектури.

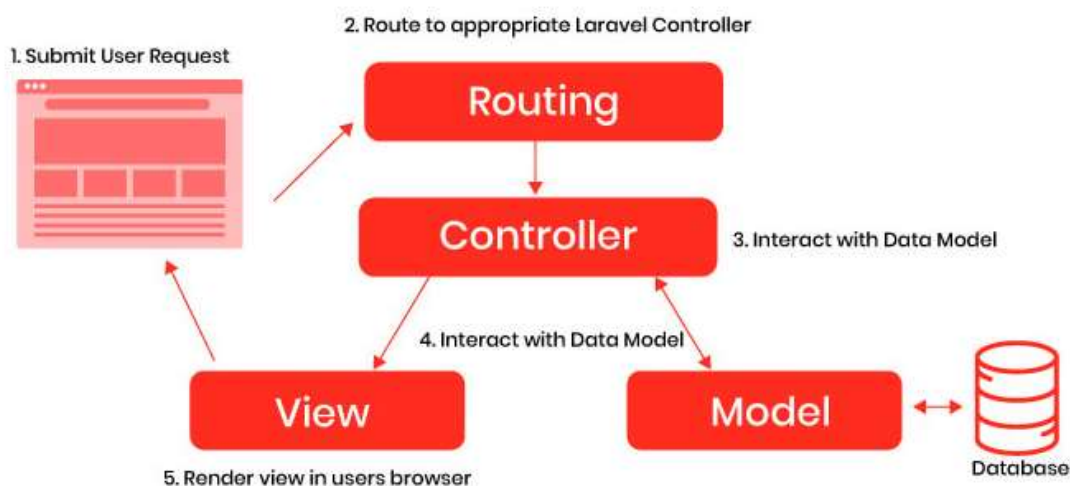


Рисунок 3.1 - MVC архітектура з використанням laravel

Найпоширеніша архітектура веб-додатків. Сервер генерує HTML-вміст і відправляє його клієнту як повноцінну HTML-сторінку. Іноді цю архітектуру називають "Web 1.0", оскільки вона з'явилася першою і в даний час домінує у сфері веб-розробки. Це найстаріша архітектура веб-розробки, тому можна вибрати будь-яку мову сервера та фреймворк для певних потреб.

Якщо ми подивимось на генерацію HTML, під зростаючим навантаженням настає момент, коли потрібен баланс навантаження. Ситуація з масштабуванням баз даних набагато складніша, але це завдання однакове для цих трьох прикладів архітектури програмного забезпечення.

Тісно пов'язаний з можливістю масштабування. Продуктивність порівняно низька, тому що потрібно передавати велику кількість даних, що містить HTML,

дизайн та бізнес-дані. Тому необхідно генерувати дані для всієї сторінки, не тільки для змінених бізнес-даних, а також усієї супровідної інформації.

3.2 Використання web-додатку зі сторони користувача

Переходячи за посиланням на сайт, в якості користувача, потрапляємо на головну сторінку, зміст якої можемо бачити на рис. 3.2-5. тут можна побачити і знайти посилання на інші сторінки сайту, а також деяку інформацію і новини. Прогорнувши вниз можна ознайомитись з усіма представленими варіантами авто для оренди по посилання на соціальні мережі цього ж сайту.



Рисунок 3.2 – Головна сторінка сайту

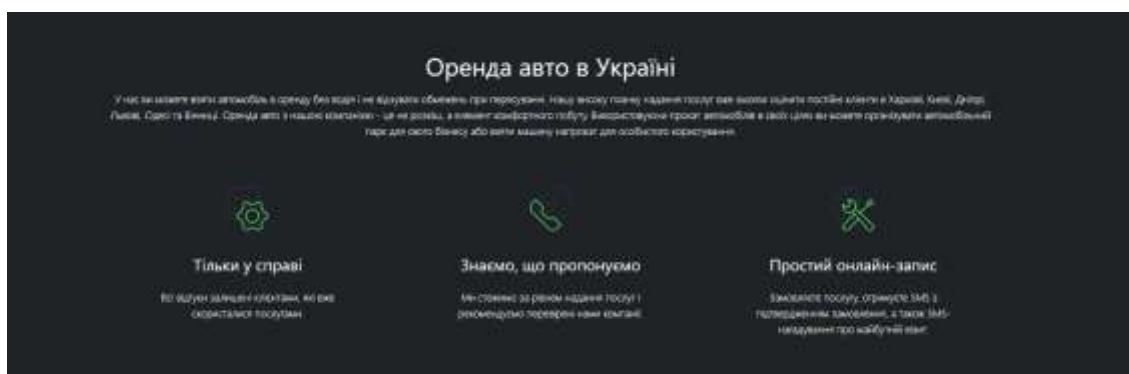


Рисунок 3.3 – Короткий опис-реклама сайту

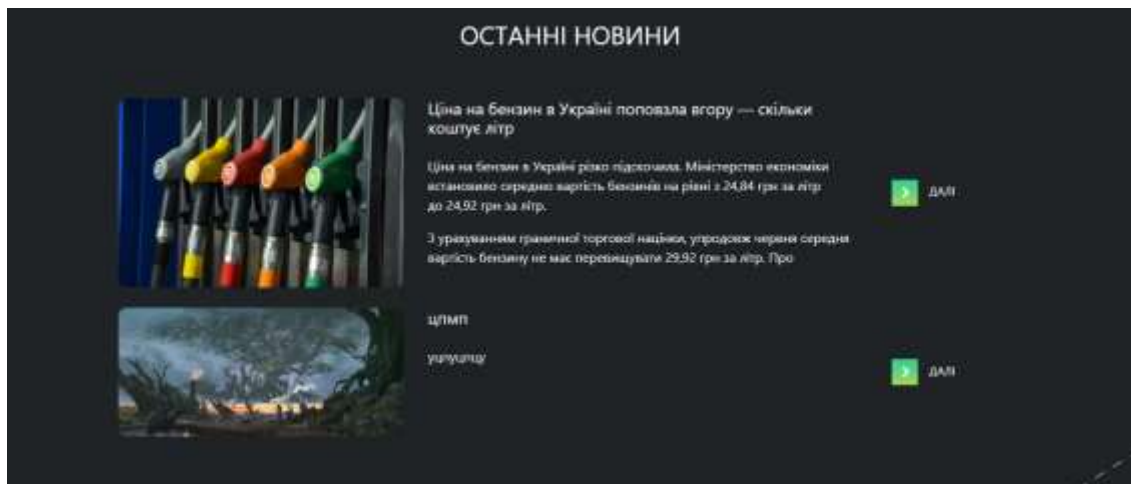


Рисунок 3.4 – Останні новини стосовно автомобілів

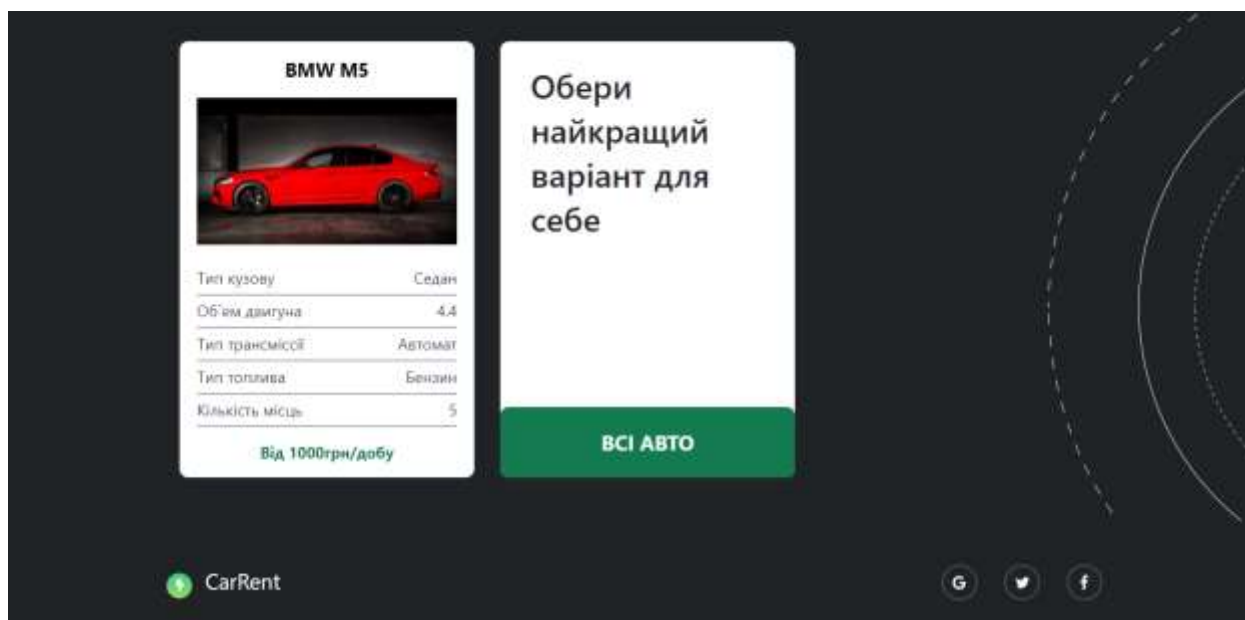


Рисунок 3.5 – Посилання на наявні авто для оренди

Переходимо на вкладку «ОРЕНДА» (рис.3.6). Тут користувач вказує необхідні критерії, такі як модель, серія авто та область, в якій йому потрібно орендувати автівку. Після чого бачить всі можливі варіанти та обирає ту, яка до вподоби.

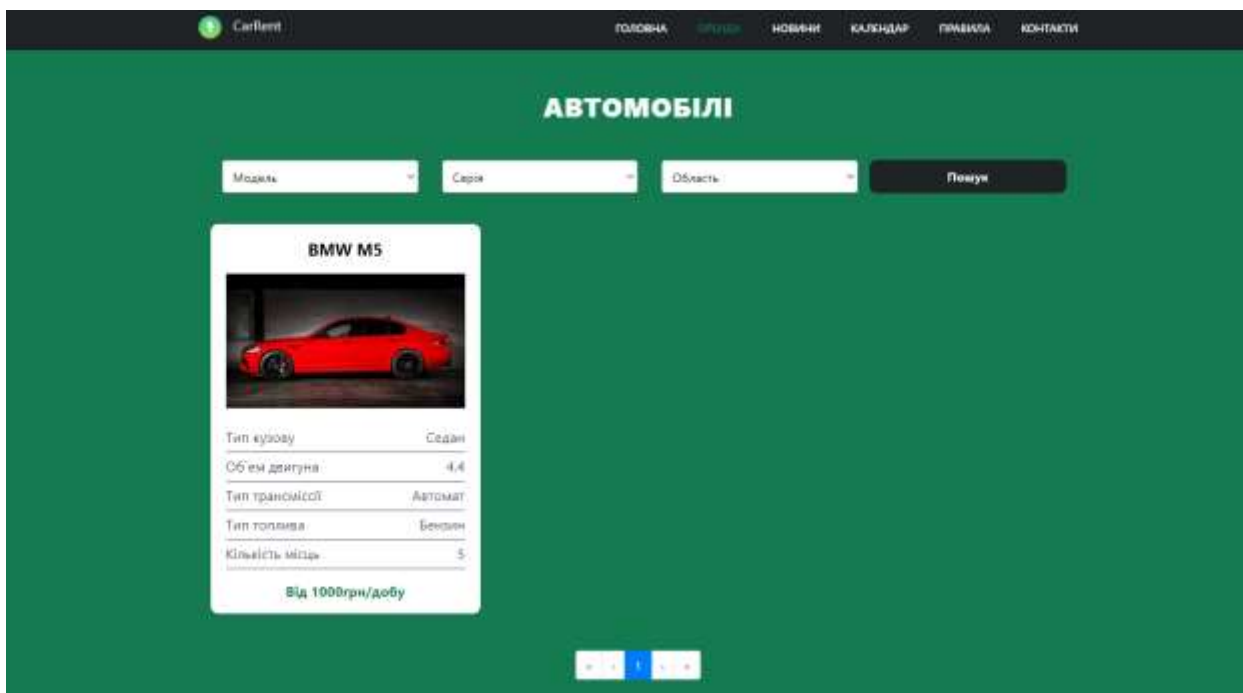


Рисунок 3.6 – Фільтр для пошуку

Обравши авто, що найбільше сподобалось можна переглянути більше інформації про нього (рис.3.7).



Рисунок 3.6 – Анкета автомобіля

В анкеті автомобіля можна побачити короткий опис автомобіля обраної моделі, технічні параметри, фото самого автомобіля, відгуки та, власне, оформити оренду (рис.3.8-10).

BMW M5

BMW M5 Competition (F90) - убійця спорткаров в класі седан. Найшвидша BMW M5 Competition тепер доступна в аренду без водія в Москві.

Фірмовий дизайн, урешаджана світлодіодна оптика і прострели високошвидкісної системи заставляють тремтати і дарят яркі емоції от вождіння! 625 л.с. видают 750 Н*м і разгоняют автомобиль от 0 до 100 за 3.3 секунды! А упрямление 100 - 200 выполняется за 7 секунд! 1/4 мили BMW M5 Competition преодолевает за 11.2 секунды, эта динамика сопоставима с динамикой суперкара! В потоке машина узнается по фирменному шильдику M5 Competition и оригинальным дискам M-Ривер, эту отличительная черта удоставляют вниманию как и у знатоков BMW, так и у обычных водителей в потоке.

Отдельного внимания заслуживает салон автомобиля, наша BMW M5 Competition в максимальной комплектации. Массаж сидений, панорамная крыша, проекция и система управления жестами добавляют ощущения качества, удобства и технологичности. Обилие внутренней светодиодной подсветки, ароматизация и ионизация воздуха создают особую атмосферу комфорта и респектабельности. И несмотря на заряженность версии M5 Competition, в режиме комфорт, автомобиль - мягкий, плавный и на каждый день. Но стоит включить режим M и мир преобразуется: в вместе с ним и вы!

Технические параметры

BMW M5 Competition - четырехдверный суперспортивный седан с динамикой спорткара. Особенности BMW M5 Competition:

- 4.4 битурбо мотор с мощностью в 625 л.с.
- восьмиступенчатая АКП;
- крутящий момент - 750 Нм;
- полный привод.

Аренда BMW M5 Competition - это уникальная возможность испытать всю палитру эмоций от вождения комфортного автомобиля с динамикой суперкара!

Рисунок 3.8 – Інформація про обране авто

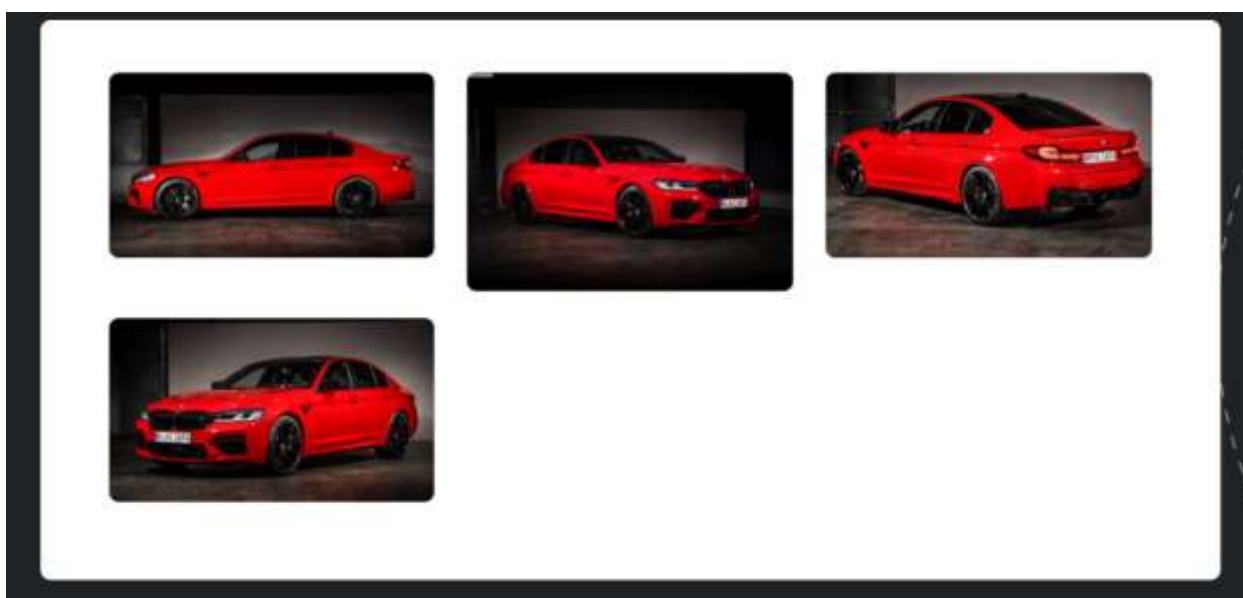


Рисунок 3.9 – Фотографії автомобіля



Рисунок 3.10 – Відгуки

Після користування автомобілем, протягом певного часу, можна залишити свій коментар, заповнивши для цього форму залишення коментаря, що представлена на рис.3.11.

A screenshot of a feedback form titled 'Залишити відгук' (Leave a review) on a dark green background. The form contains two input fields: 'Прізвище та ім'я' (Surname and name) and 'Електронна адреса' (Email address). Below these fields is a large white text area for the review. At the bottom center, there is a green button with the text 'Залишити відгук' (Leave a review).

Рисунок 3.11 – Форма залишення коментаря

На рис.3.12 та 3.13 можна побачити приклад заповнення форми та, власне, результат, як він відобразатиметься і блоці відгуків.

Залишити відгук

ТЕСТ

ТЕСТ@gmail.com

Цікаво

Залишити відгук

Рисунок 3.12 – Приклад заповнення форми

BMW M5 1000грн за добу

ІНФОРМАЦІЯ

ФОТО

ВІДГУКИ

ОРЕНДУВАТИ

Тип кузова	Седан
Об'єм двигуна	4.4
Тип трансмісії	Автомат
Тип палива	Бензин
Кількість місць	5

DFDSFDS
бомба!

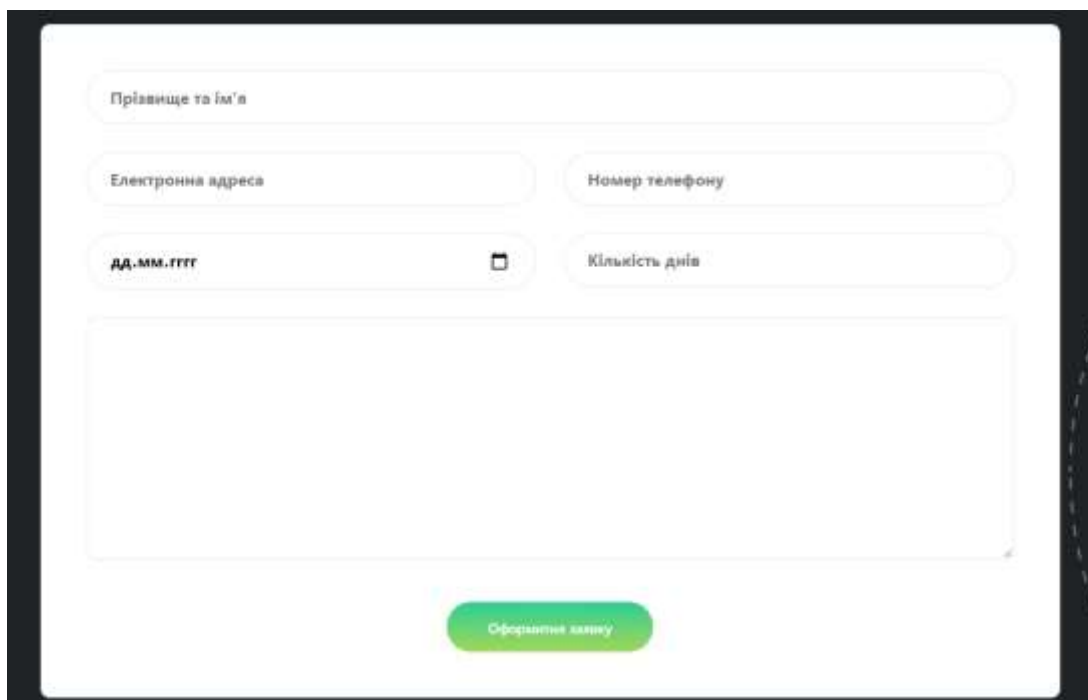
02.06.2021

ТЕСТ
Цікаво

02.06.2021

Рисунок 3.13 – Відображення коментарів

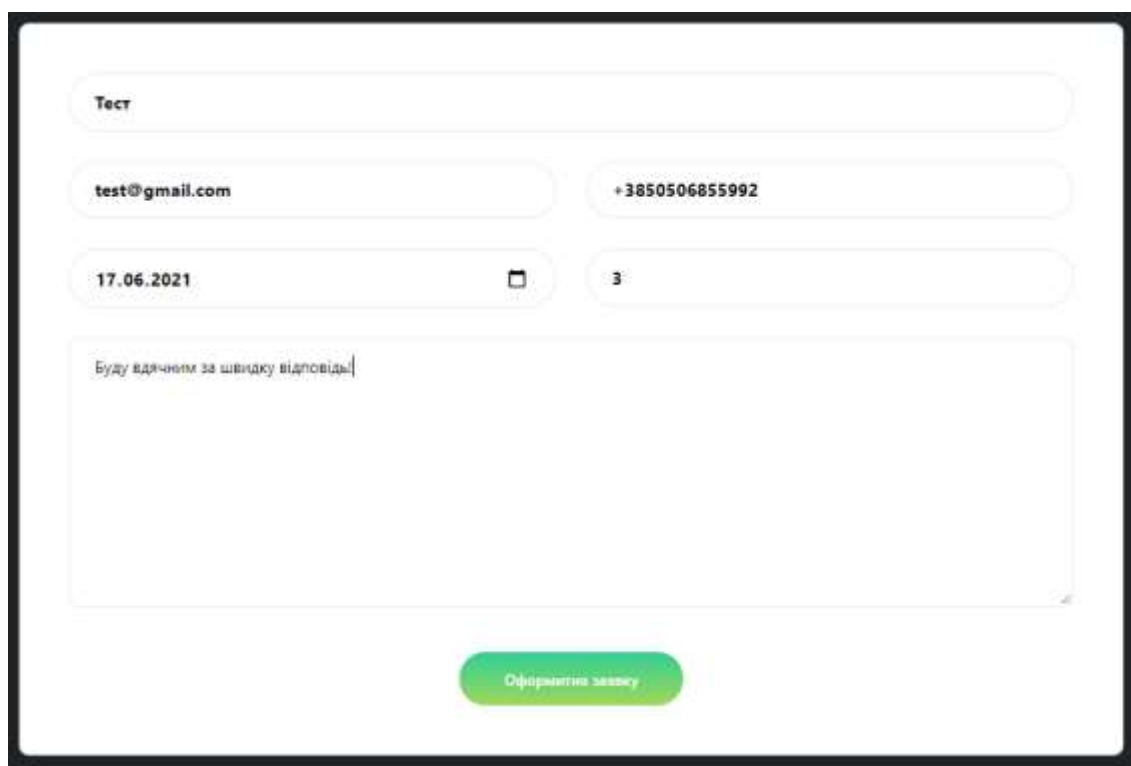
Форма заявки на оренду представлена на рис. 3.14.



The image shows a web form for a rental application. It consists of several input fields: a single-line text field for 'Прізвище та ім'я', two single-line text fields for 'Електронна адреса' and 'Номер телефону', a date field with a calendar icon for 'ДД.ММ.РРРР', and a single-line text field for 'Кількість днів'. Below these fields is a large, empty text area. At the bottom center is a green button labeled 'Оформити заявку'.

Рисунок 3.14 – Форма заявки на оренду

Приклад заповнення заявки та результат її обробки можна побачити на рис.3.15-16.



The image shows the same rental application form as in Figure 3.14, but now it is filled out. The fields contain the following information: 'Тест' in the name field, 'test@gmail.com' in the email field, '+3850506855992' in the phone number field, '17.06.2021' in the date field, and '3' in the days field. The text area contains the message 'Буду вдячним за швидку відповідь'. The green button at the bottom is still labeled 'Оформити заявку'.

Рисунок 3.15 – Приклад заповнення заявки

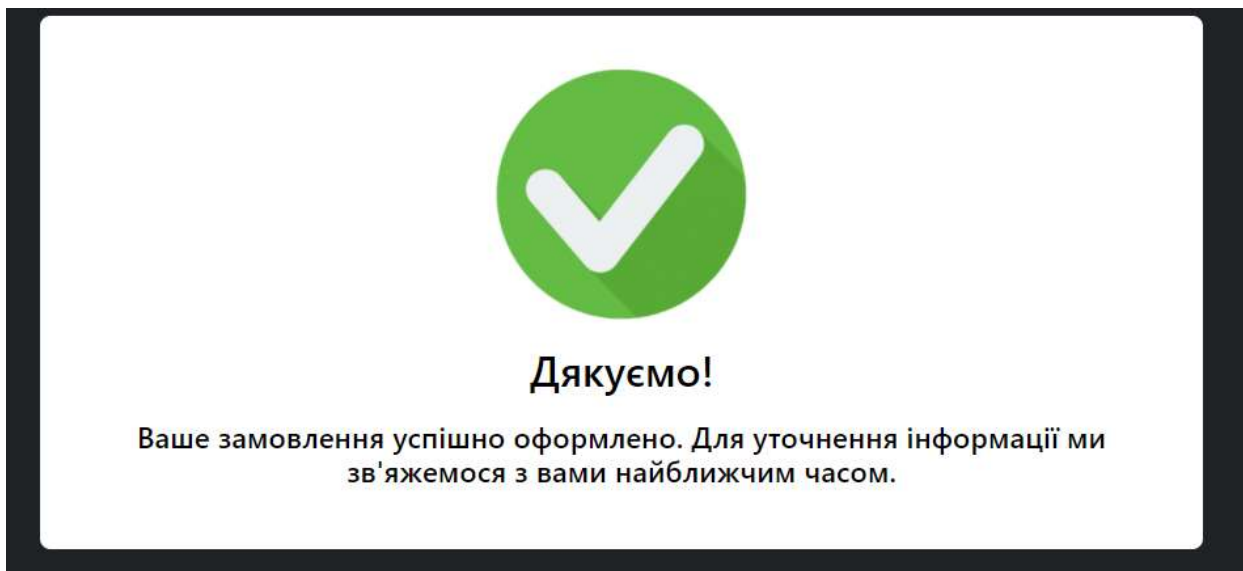


Рисунок 3.16 – Результат обробки поданої заявки

Також можна ознайомитись з правилами оренди, що включають вікові обмеження та наявність певних документів (рис. 3.17).

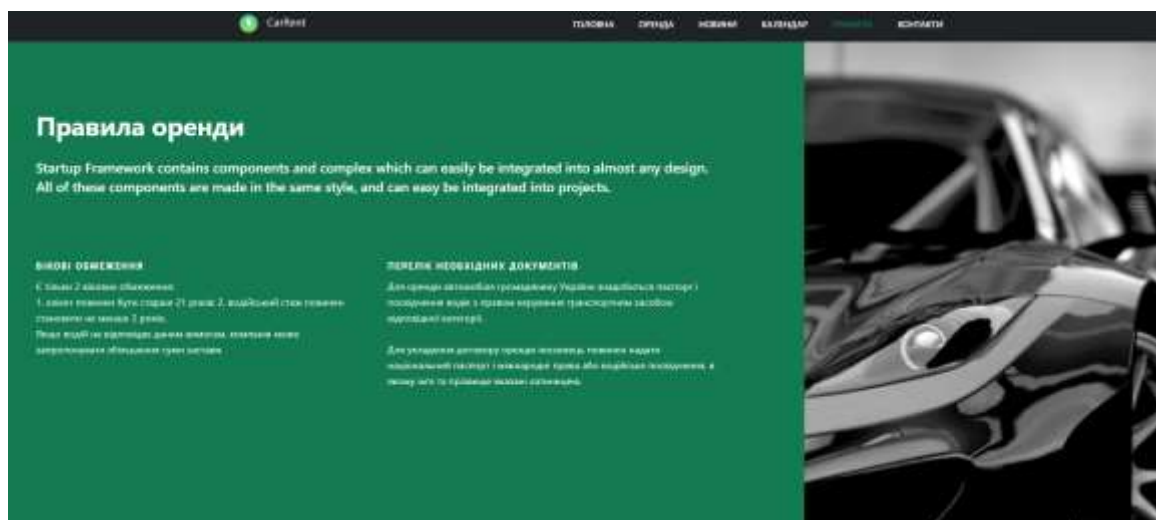


Рисунок 3.17 – Правила

В розділі «КОНТАКТИ» можна знайти номер телефону, електронну адресу та адресу знаходження офісу (рис. 3.18).

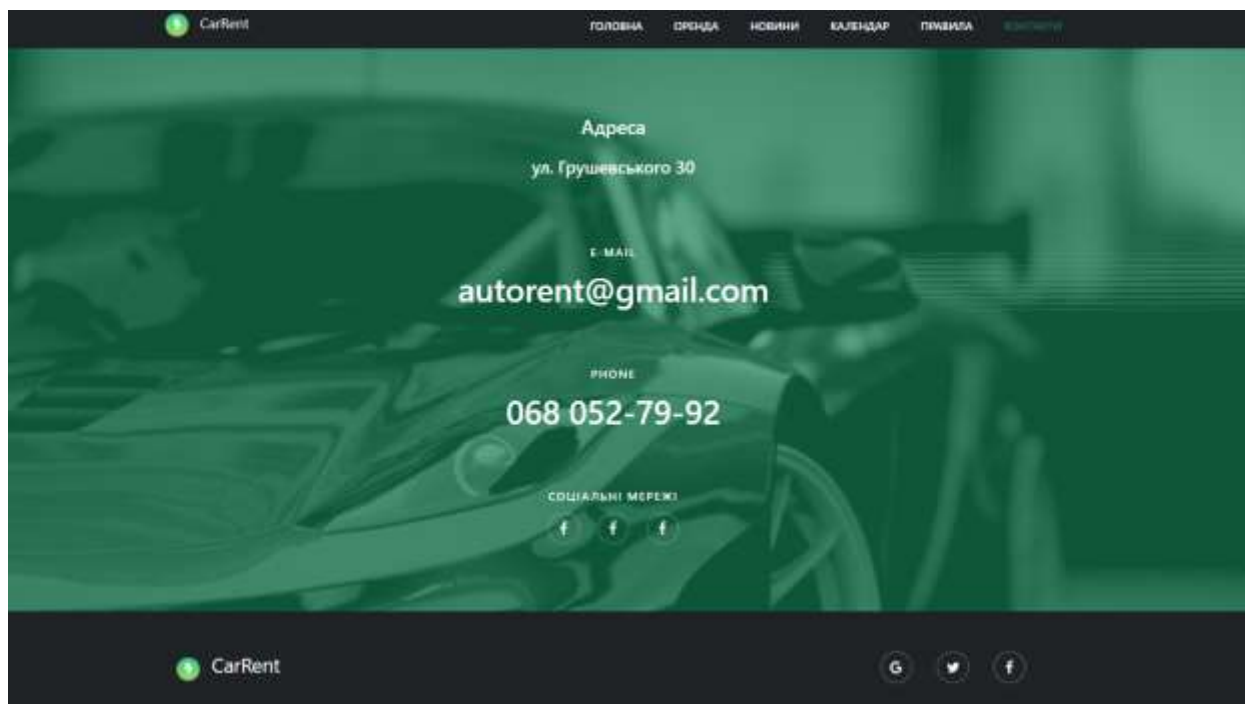


Рисунок 3.18 – Контакти

3.4 Адміністрування сайту

Подивимось як виглядає сайт з боку адміністратора. Для початку виконуємо авторизацію (рис.3.19)



АВТОРИЗАЦІЯ

Увійдіть за допомогою електронної пошти та паролю.

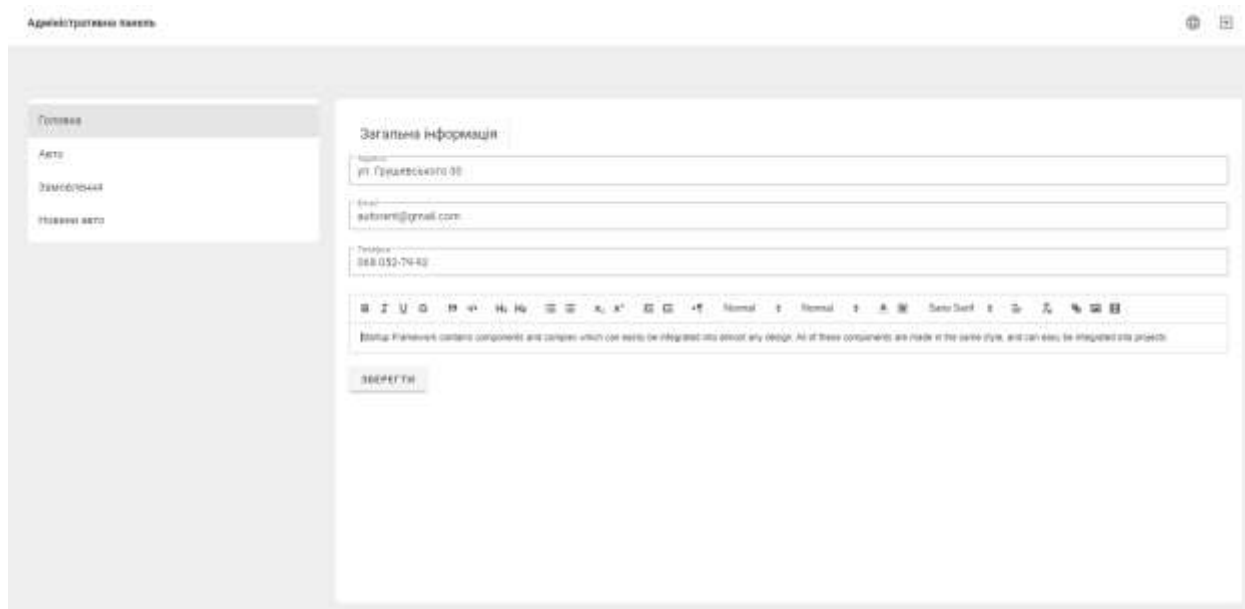
Email

Пароль

Вхід

Рисунок 3.19 – Авторизація в якості адміна

Адміністратор має змогу змінювати інформацію на головній сторінці (рис.3.20).



Адміністрування контенту

Головна

Авто

Замовлення

Нові авто

Зміна інформації

Назва: ул. Пушкінського 98

Е-пошта: autotam@gmail.com

Телефон: 068-052-79-42

Normal Normal Seto Default

Зберегти

Рисунок 3.20 – Зміна контактної інформації

В права та можливості адміністратора входить можливість додавати, видаляти або змінювати інформацію про наявні автомобілі для оренди (рис. 3.21-24).

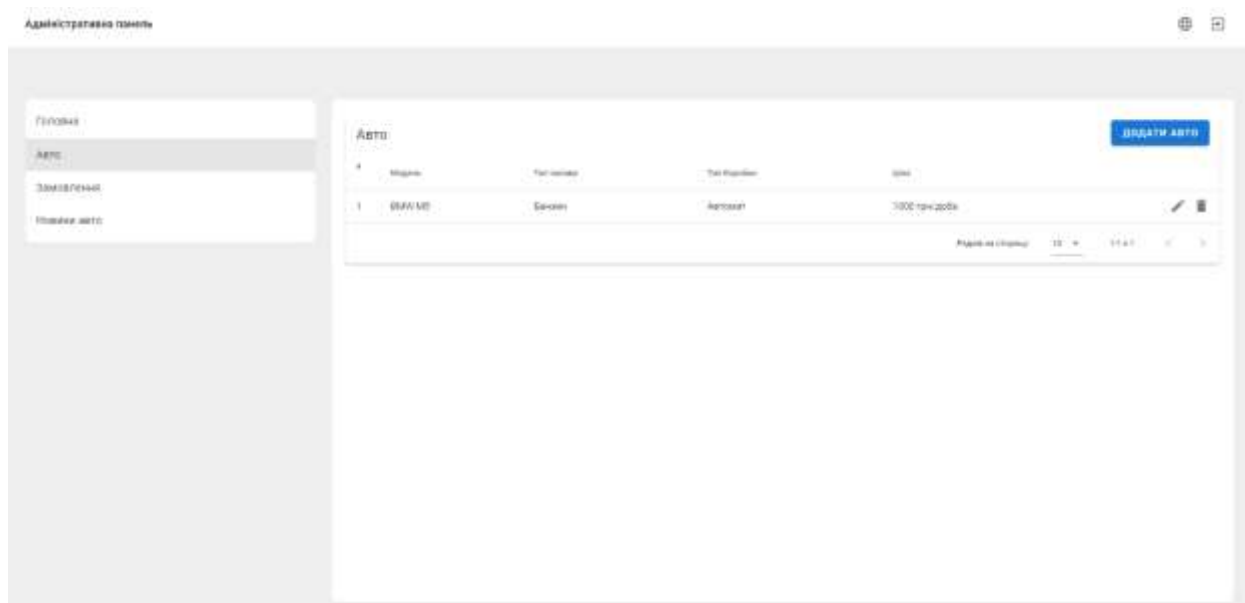


Рисунок 3.21 – Додавання авто

Модель: BMW

Серія: M5

Регіон: Сумська область X Харківська область X

Тип кузова: Седан

Об'єм двигуна: 4.4

Тип коробки: Автомат

Тип палива: Бензин

Витрати палива: 12

Кількість пасажирів: 5

Ціна оренди за добу: 1000

Наявність

Рисунок 3.22 - Редагування авто

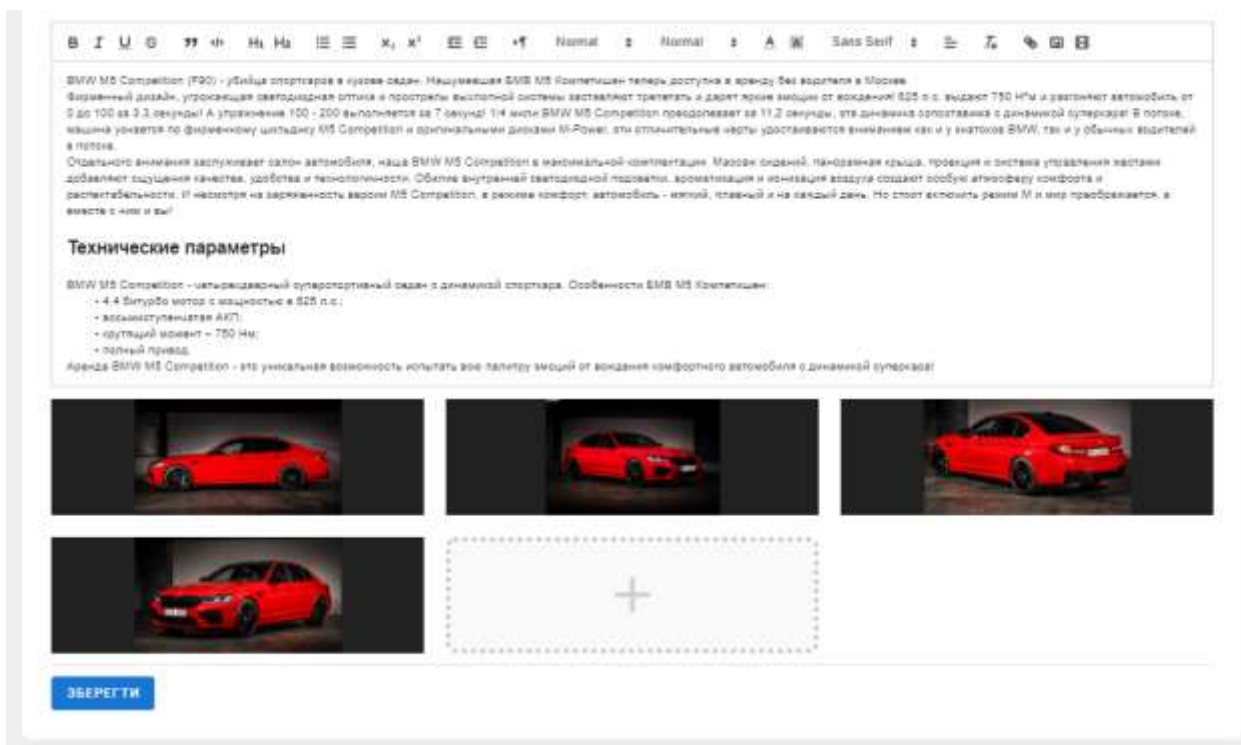


Рисунок 3.23 - Редагування авто

Form for adding a new car listing with the following fields:

- Модель (Model)
- Серія (Series)
- Регіон (Region)
- Тип кузова (Body Type)
- Об'єм двигуна (Engine Volume)
- Тип коробки (Gear Type)
- Тип палива (Fuel Type)
- Витрати палива (Fuel Consumption)
- Кількість пасажирів (Number of Passengers)
- Ціна оренди за добу (Daily Rental Price)
- наявність (Availability)

Below the form is a rich text editor with a toolbar and a description field. A blue 'ЗБЕРЕГТИ' (Save) button is located at the bottom left.

Рисунок 3.24 - Форма додавання нового авто

Адміністратор також оброблює замовлення (рис. 3.25-26).

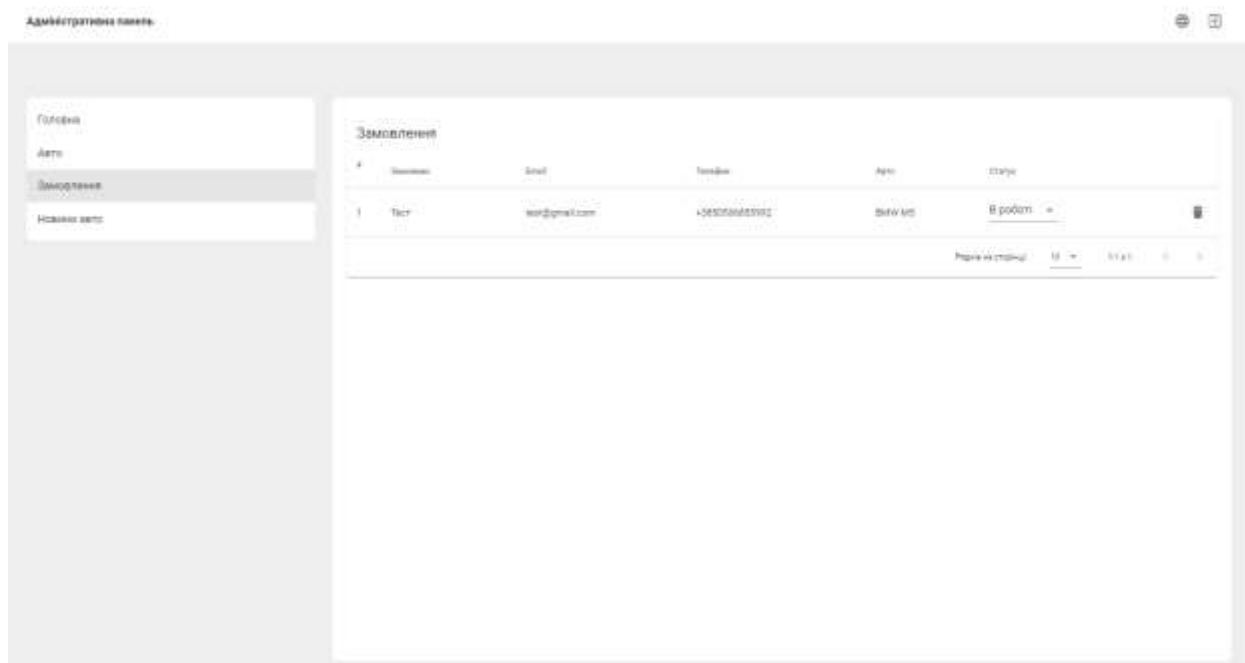


Рисунок 3.25 - Замовлення

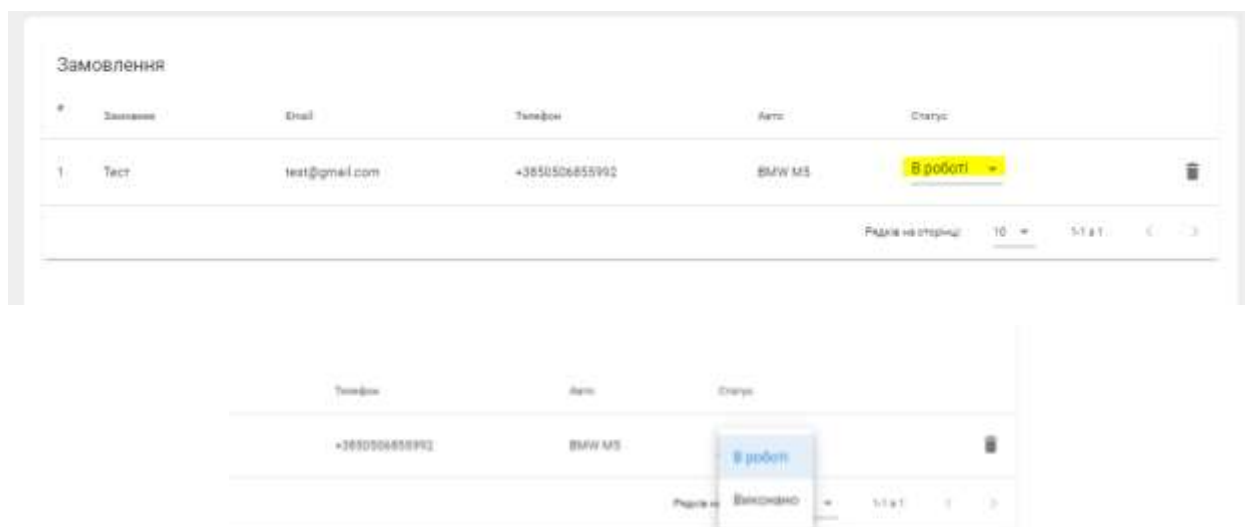


Рисунок 3.26 - Зміна статусу

В права адміністратора також входить можливість редагування та додавання новин, оновлення інформації (рис. 3.267-29).







Рисунок 3.27 – Наявні новини



Рисунок 3.28 - Редагування новини

The image shows a two-part screenshot of a web application interface. The top part is a rich text editor with a toolbar containing various icons for text formatting (bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, undo, redo) and font settings (font size, font family: Sans Serif). Below the toolbar is a text input field containing the word "Тест". Underneath is an image selection area with a small thumbnail of a woman's face and a text box containing the filename "dd1aaf046e...e076e8.jpg". The bottom part of the screenshot shows a "Новини" (News) section with a "ДОДАТИ НОВИНУ" (Add News) button. It contains a table with two rows of news items:

#	Новина	Дата публікації	
1	Ціна на бензин в Україні дозволяє знову – скільки коштує літр	2021.06.02	 
3	Тест	2021.06.02	 

At the bottom right of the news list, there is a pagination control: "Рядків на сторінці: 10" and "1-2 з 2".

Рисунок 3.29 - Додавання нової новини

ВИСНОВКИ

Отже, було досліджено та проаналізовано всі головні аспекти при оренді автомобілів. На базі аналогів інформаційних систем були виявлені недоліки та переваги інформаційних сторінок, які допоможуть в проектуванні сайту та розробити його більш універсальним й зручним у використанні.

Було вирішено наступні задачі:

- Пошук та аналіз інформації згідно тематики, а саме використання та оренда автомобілів;
- Ознайомлення з подібними реалізаціями у вигляді сайтів й виділення їх переваг та недоліків;
- Було розроблено технічне завдання (Додаток А);
- Проаналізовано можливі ризики та шляхи їх попередження або усунення (Додаток Б).

У результаті роботи було реалізовано можливості перегляду контенту, представленого у web-додатку, перегляд останніх новин та оголошень на сайті. Пристуня можливість перегляду коментарів та загальної інформації на сайті. Користувач також має можливість заповнення заявки для оренди обраного автомобіля.

Редагування текстової інформації, що відображається на сторінках web-додатку, додавання нової інформації про автомобілі на сайт та робота із інформаційною системою ведеться через адміністративну панель.

;

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Top 10 Global Automotive Trends [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.forbes.com/sites/sarwantsingh/2021/01/27/top-10-global-automotive-trends-2021/?sh=7adcc997605b> (дата звернення: 25.10.2020);
2. M. Fitzgerald, Building a better car company with analytics. MIT Sloan Management Review. 57, 5–5 (2016).
3. TRENDS IN THE AUTOMOTIVE INDUSTRY FOR 2021 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.endava.com/en/blog/Business/2020/Trends-in-the-automotive-industry-for-2021> (дата звернення: 25.12.2020);
4. Як пандемія вплинула на ринок оренди [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://ain.ua/2020/06/13/vliyanie-pandemii-na-rynok-arendy/> (дата звернення: 25.12.2020);
5. Прокат автомобілів [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.beboss.ru/bplans-rent-a-car> (дата звернення: 25.12.2020);
6. 7Cars [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.beboss.ru/bplans-rent-a-car> (дата звернення: 25.12.2020);
7. Автопрокат Суми [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://7cars.com.ua/sdat-avto-v-prokat/> (дата звернення: 25.04.2021);
8. LUXSERVIS [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://luxservis.com.ua> (дата звернення: 25.04.2021);
9. DriveUp [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.driveup.com.ua> (дата звернення: 26.04.2021);
10. Оренда авто в Україні під час карантину [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://narscars.com.ua/blog/nuzhnoznat/arenda-avto-ukraine-karantina> (дата звернення: 26.04.2021);

11. Automotive Software [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.eetasia.com/2021-auto-industry-outlook-what-can-we-expect-from-bevs-connected-cars-software/> (дата звернення: 26.04.2021);

12. TOP 5 TECH TRENDS THAT WILL TRANSFORM THE AUTOMOTIVE INDUSTRY [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.analyticsinsight.net/top-5-tech-trends-that-will-transform-the-automotive-industry-in-2021/> (дата звернення: 26.04.2021);

13. J. Yang, L. Sun, Q. Wu, Parts inventory information integrated security technology of automobile industry chain SaaS platform. Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS. 26, 1277–1285 (2020).

14. C. Legacy, D. Ashmore, J. Scheurer, J. Stone, C. Curtis, Planning the driverless city. Transport Reviews. 39, 84–102 (2019).

15. The best new electric and petrol/diesel/hybrid cars coming in 2021-2022 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.carwow.co.uk/news/4498/best-new-cars-coming-2020-2021-2022>

16. The Front-End Web Developer Roadmap for 2021 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://javascript.plainenglish.io/the-front-end-web-developer-roadmap-for-2021-bcf88c5d4ccd> (дата звернення: 26.04.2021);

17. Best Front-End Web Development Frameworks of 2021 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://dev.to/codedthemes/best-front-end-web-development-frameworks-of-2021-56hk> (дата звернення: 26.04.2021);

18. J. Peng, G. Zhang, S. Zhang, X. Dai, J. Li, Effects of online advertising on automobile sales. Management Decision. 52, 834–851 (2014).

19. Top 5 picks for backend development in 2021 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.thirdrocktechkno.com/blog/top-5-picks-for-backend-development-in-2021/> (дата звернення: 29.04.2021);

20. Лешек, А. Мацяшек Аналіз і проектування інформаційних систем за допомогою UML 2.0 / Лешек А. Мацяшек. - М.: Вільямс, 2016. - 816 с.

21. Путілін, А. Б. компонентний моделювання та програмування на мові UML. Практичний посібник з проектування інформаційних систем: монографія. / А.Б. Путілін, Е.А. Юрагов. - М .: НТ Пресс, 2005. - 664 с.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ на розробку інформаційної системи «Інформаційна система оренди автомобілів»

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ ДОДАТКУ

1.1 Призначення інформаційної системи

Дана інформаційна система допоможе в оренді автомобіля. Користувач матиме можливість переглядати інформацію про автомобілі, відгуки, ціни та інше.

1.2 Мета створення інформаційної системи

Підбір та оренда автомобіля у місті Суми. Інформаційна система допоможе оцінити якість послуг, перевірити репутацію компанії та ознайомитися із діапазоном цін.

1.3 Цільова аудиторія

Переважною цільовою аудиторією для інформаційної системи є повнолітні люди, що зацікавлені в оренді автомобілю.

2 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Вимоги до web-додатку в цілому

Перелік головних функцій web-додатку:

1. Авторизація;
2. Реєстрація;
3. Редагування інформації на сайті
4. Додання авто на сайт;
5. Підтвердження оренди;
6. Перегляд новин та загальної інформації;
7. Редагування даних на сторінці;
8. Пошук на сайті;
9. Оренда обраного автомобілю;

2.2. Розподіл функцій за групами користувачів

Web-додаток надання послуг з оренди автомобілів має певну кількість типів користувачів, а саме такі як адміністратор, звичайний користувач.

Розглянемо розподіл функціоналу детальніше в табл.А.1.

Таблиця А.1 – Функціонал за групами користувачів

№	Користувач	Можливості користувача
1	Адміністратор	Перегляд загальної інформації;
		Підтримка сайту;
		Блокування користувачів;
2	Звичайний користувач	Створення власної інформаційної сторінки;
		Редагування даних;

Продовження таблиці А.1.

№	Користувач	Можливості користувача
2	Звичайний користувач	Використання вбудованого калькулятора для обрахування ціни;
		Оформлення замовлення послуги або консультації;
		Залишити відгук виконавцю за виконану роботу;
		Написання повідомлення адміністратору.

2.2.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у табл.А.2.

Таблиця А.2 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Перегляд контенту, представленого у web-додатку;	Клієнт
UN-02	Перегляд останніх новин та оголошень на сайті;	Клієнт
UN-03	Заповнення заявки для оренди обраного автомобіля;	Клієнт
UN-04	Можливість зворотного зв'язку;	Клієнт
UN-05	Реєстрація як звичайний користувач;	Клієнт
UN-06	Редагування текстової інформації, що відображається на сторінках web-додатку	Адміністратор
UN-07	Додавання нової інформації про автомобілі на сайт;	Адміністратор
UN-08	Перегляд коментарів та загальної інформації на сайті;	Клієнт
UN-09	Додавання фото;	Клієнт

2.2.2 Функціональні вимоги

На основі аналізу інформації були визначені такі функціональні вимоги:

1. Пошук інформації.
2. Перегляд основної інформації у web-додатку.
3. Перегляд інформації та стану заявки.
4. Додання коментарю чи фото після оренди автомобілю.

2.2.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в табл. А.3.

Таблиця А.3 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність модуля реєстрації користувача у web-додатку	M	Надає можливість користувачу зареєструватися у web-додатку
SR-02	Список автомобілів, які можливо орендувати	M	Список автомобілів, які користувач має змогу орендувати заповнивши заявку
SR-03	Модуль відображення новин та оновлень у web-додатку	S	Формує можливість перегляду останніх новин та змін
SR-04	База даних	M	Надає можливість відобразити адміністратору інформацію про автомобілі на сайті.
SR-05	База даних з контентом	M	Відповідає за заповнення web-додатку відповідною інформацією.

Продовження таблиці А.3.

ID	Системні вимоги	Пріоритет	Опис
SR-06	Панель редагування	М	Відповідає за подальше наповнення та редагування сайту адміністратором

Умовні позначення в табл. А.2:

- Must have (M) – вимоги, які повинні бути реалізовані в системі;
- Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;
- Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.3. Інтерфейс інформаційної системи

Для детального представлення майбутнього додатку виконаємо детальний опис кожної сторінки подальшого для прототипування.

2.3.1. Головна сторінка

Сторінка з основною довідкою та можливостями інформаційної системи надання послуг з оренди автомобілів. Дана сторінка є першою на яку потрапляє користувач.

2.3.2. Реєстрація

Сторінка для реєстрації нових користувачів. Користувач повинен ввести певний перелік даних для створення акаунта.

Перелік даних необхідних для реєстрації:

- Ім'я;
- Прізвище;
- Електронна пошта;

- Пароль;
- День народження користувача;
- Стать (чоловік/жінка);

2.3.3. Авторизація

Сторінка для авторизації. Не залежно від типу сторінки, користувач повинен ввести такі дані:

- Електронна пошта;
- Пароль.

2.3.4. Вибір автомобілю

Дана сторінка потрібна для підбору автомобілю. Можна виконувати пошук та фільтрацію даних за певними параметрами.

2.4. Програмні засоби

Розробка інформаційної системи поділяється на розробку візуальної частини, та функціональну частину. Для сучасної front-end розробки використовують: HTML, CSS, JavaScript.

У той же час, для розробки back-end частини популярними технологіями є PHP, MySQL.

3 СКЛАД І ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ САЙТУ

Докладний опис етапів роботи зі створення сайту наведено в табл.А.3.1.

Таблиця А.3.1 – Етапи створення інформаційної системи

№	Зміст робіт	Час роботи (дні)
1	Аналітика	5
2	Створення технічного завдання	3
3	Прототипування	2
4	Розробка макету сайту	5
5	Front-end розробка	10
6	Back-end розробка	10
7	Заповнення бази даних	1
8	Тестування	3
9	Адміністрування сайту	1
Загальний час розробки інформаційної системи		40

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ для розробки інформаційної системи «Інформаційна система оренди автомобілів»

1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

1.1 Розробка концепції проекту

1.1.1. Ідентифікація ідеї проекту.

Головна мета проекту – розробка інформаційної система для оренди автомобілів.

1.1.2. Деталізація мети методом SMART

Можна досягти успіху завдяки SMART техніці постановки цілей, ця техніка формує модель поведінки, виробляє стійкі правила життєвої позиції. Розглянемо детальніше інформацію за кожним з пунктів у таблиці Б.1.1.

Таблиця Б.1.1 – Таблиця опису проекту за SMART методом

Specific (конкретна)	Розробка інформаційної системи, використовуючи яку будуть виконуватися поставлені цілі та задачі, будуть формуватися замовлення.
Measurable (вимірювана)	Прогрес та якість буде вимірюватися в кількості користувачів системи, замовленнях та кількості позитивних відгуків.
Achievable (досяжна)	Данна інформаційна система буде розроблятися з використанням HTML, CSS, JavaScript, Vue.js та PHP, MySQL, Laravel.
Relevant (реалістична)	Для виконання проекту є всі необхідні засоби та знання.
Time-framed (обмежена у часі)	Для проекту визначений дедлайн та побудована Діаграма Ганта для відстеження часу виконання.

1.2 Техніко-економічне дослідження

1.2.1. Дослідження IT-проекту, організації, регіону

Створений програмний продукт орієнтований на сервіси оренди автомобілів. Web-додаток буде розроблений з метою зручного використання послуг споживачем.

1.2.2. Оцінка цінності, життєздатності, економічної ефективності та життєстійкості IT-проекту

Цінність – це особисте сприйняття зацікавленими сторонами можливостей продукту проекту створювати вигоди в соціальному, політичному, економічному та духовному аспектах їх життєдіяльності завдяки унікальним властивостям продукту проекту. Цінність визначається вигодою, яку надає експлуатація продукту проекту конкретному споживачу за умови відповідальності вимогам, які містяться в місії.

Життєздатність проекту — здатність зберігати існування й цінність в мінливих умовах середовища ринку та моди. Проект оцінюється як життєздатний, бо сфера зараз у розквіті й має великі перспективи.

Економічна ефективність — досягнення найбільших результатів за найменших затрат живої та уречевленої праці. Економічна ефективність є конкретною формою дії закону економії часу. Економічна ефективність проекту є високою, обґрунтовується це тим, що над ним працює одна людина і не витрачається велика кількість часу на передачу інформації від одного працівника до іншого.

Можливості програмного продукту є корисними та результативними для використання споживачами з міст усієї України, так як буде можливе збереження і обробка даних, які були внесені до інформаційної системи.

Життєздатність проекту підтверджується наявністю попиту і актуальності даного проекту на даний момент часу.

2 УПРАВЛІННЯ ВИМОГАМИ

2.1 Загальні вимоги до інформаційної системи

2.1.1 Вимоги до структури й функціонування інформаційної системи

Інформаційна система повинна бути реалізована у вигляді Web-додатку, доступного в мережі Інтернет. Web-додаток повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2. Вимоги до персоналу

Для підтримки сайту й експлуатації веб-інтерфейсу системи керування сайтом від персоналу повинні вимагатися загальні навички роботи з персональним комп'ютером і стандартним веб-браузером.

2.1.3. Вимоги до стилістичного оформлення сайту

Інтерфейс сайту повинен забезпечувати інтуїтивно зрозуміле уявлення структури розміщеної на ньому інформації, логічний перехід до розділів і сторінок. Посилання на сторінки повинні бути забезпечені заголовками. При виборі будь-якого з пунктів меню користувачем повинна завантажуватися відповідна йому інформаційна сторінка, а в блоці меню має відкриватися список підрозділів.

2.2 Вимоги до розділів

Web-додаток повинен складатися з наступних розділів:

- Доступні автомобілі;
- Про нас – текстова сторінка;
- Місцезнаходження та оплата – текстова сторінка;
- Вхід/Реєстрація;
- Особистий кабінет(посилання авторизованим користувачам).

3 ОПИСАННЯ ФАЗИ РОЗРОБКИ ІТ-ПРОЕКТУ

3.1 Планування змісту структури робіт ІТ-проекту

WBS проекту (вона ж Work Breakdown Structure або ICP, Ієрархічна Структура Робіт) - це розбиття проекту на конкретні етапи, які повинні бути виконані для досягнення цілей проекту. Важливо розуміти, що в WBS збираються саме результати робіт, а не завдання, які потрібно виконати для отримання цих результатів. WBS є ієрархічною та інкрементною декомпозицією проекту у фазі, кінцеві результати та пакети робіт. Вона є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль. Наприклад, програма, проект чи договір.

Менеджери використовують структуру декомпозиції, щоб структурувати і ділити проекти на легко керовані компоненти. Вони, в свою чергу, поділяються до тих пір, поки вони не призначають конкретного фахівця в команді.

Загальна WBS структура представлена на рисунку 2.1.

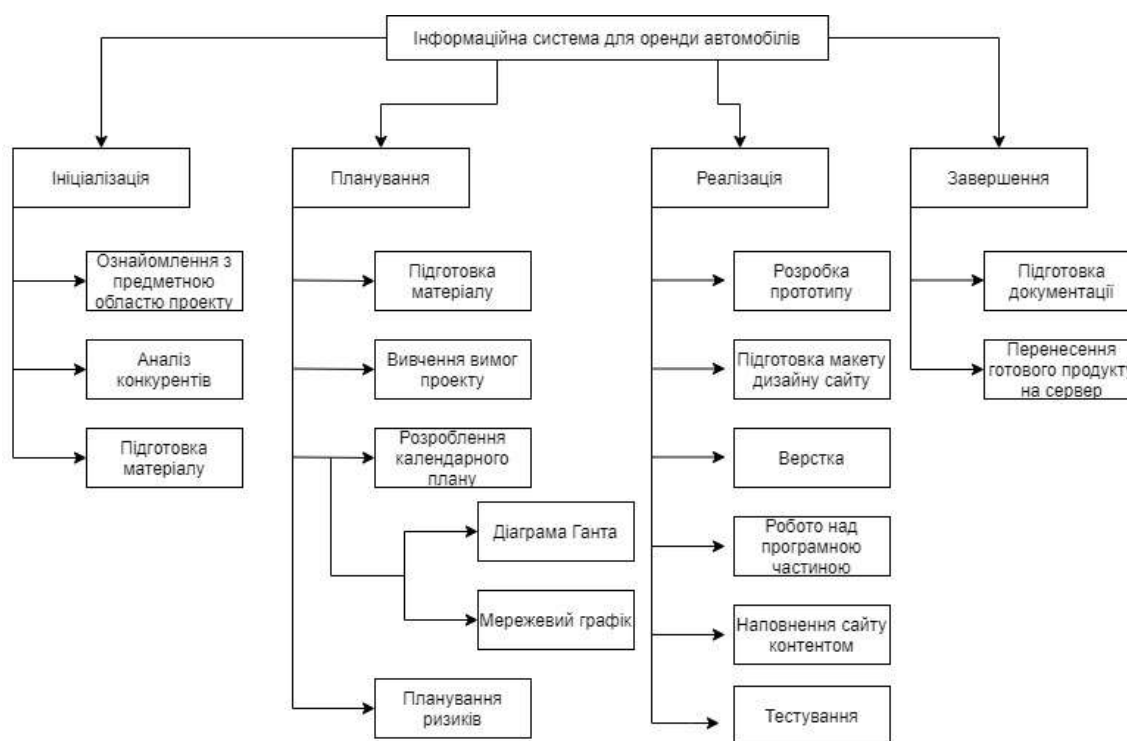


Рисунок Б.2.1 – WBS-структура проекту

3.2. Планування структури організації

Структура розподілу організації або OBS – це ієрархічна модель, що описує встановлені організаційні рамки для планування проектів, управління ресурсами, відстеження часу та витрат, розподілу витрат, звітування про доходи / прибуток та управління роботою.

Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту. На верхньому рівні OBS розташована команда проекту.

Ця структура стосується тільки внутрішньої організаційної структури проекту і не чіпає відносин проектних груп чи учасників з батьківськими організаціями. Будується OBS аналогічно робочій структурі, а саме:

- На першому рівні відображається організаційна структура як єдиний елемент;
- На другому і нижчих рівнях триває поділ структури на основні організаційні елементи.

Загальна OBS структура представлена на рисунку 2.2.

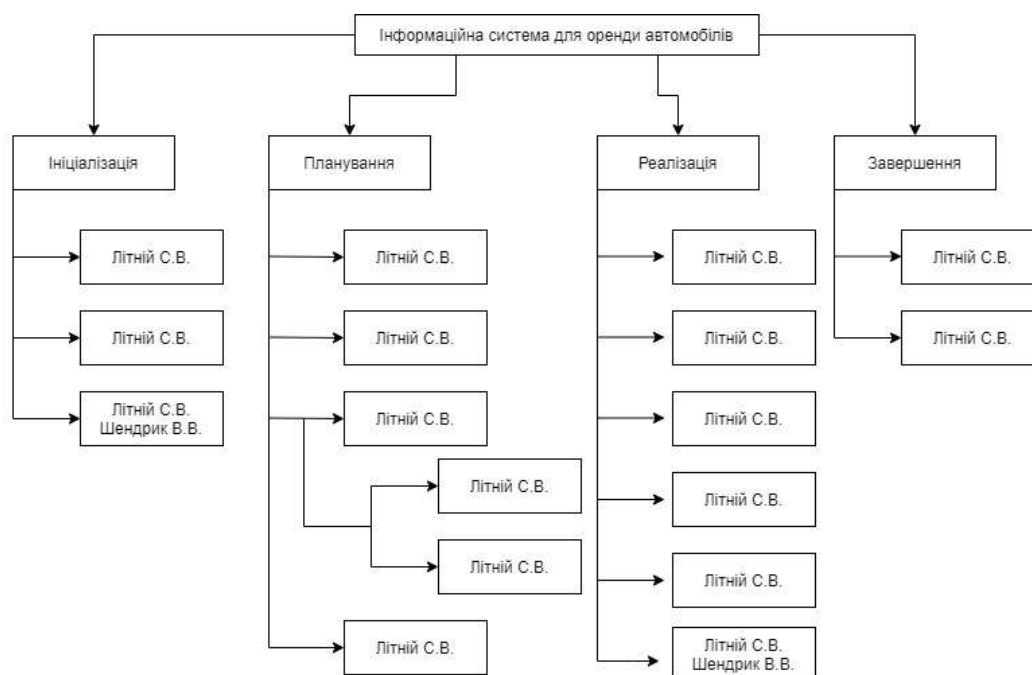


Рисунок Б.2.2 – OBS структура проекту

3.3. Побудова матриці відповідальності

На підставі OBS та WBS структур будується матриця відповідальності проекту. Матриця відповідальності закріплює за кожною елементарною роботою виконавця. Використовується для контролю відповідності розподілу ролей цілям проекту. На верхньому рівні розподіляються ролі та відповідальність по елементах OBS. На нижньому — фази проекту. Матриця відповідальності представлена в таблиці Б.3.1

Таблиця Б.3.1 – Матриця відповідальності

		Літній С.В.	Шендрик В.В.
1	Ініціалізація	+	+
1.1	Ознайомлення з предметної областю проекту	+	-
1.2	Аналіз конкурентів	+	-
1.3	Ідентифікація ідей	+	+
2	Планування	+	-
2.1	Підготовка матеріалу	+	-
2.2	Вивчення вимог проекту	+	-
2.3	Розробка календарного плану	+	-
2.4	Планування ризиків	+	-
3	Реалізація	+	+
3.1	Розробка прототипу	+	-
3.2	Підготовка макету дизайну сайту	+	-
3.3	Верстка	+	-
3.4	Робота над програмною частиною	+	-
3.5	Наповнення сайту контентом	+	-
3.6	Тестування	+	+
4	Завершення	+	-
4.1	Підготовка документації	+	-
4.1	Перенесення сайту на сервер	+	-

3.4. Побудова календарного графіку виконання ІТ-проекту

Діаграма Ганта – це горизонтальна діаграма з тимчасової шкалою, яка використовується для ілюстрації плану робіт за проектом з прив'язкою до часу.

За допомогою діаграм Ганта керівники проектів і менеджери по продукту розбивають проекти на робочі завдання для зручності управління, підтримують порядок в роботі і роблять залежності між завданнями наочними.

Діаграми Ганта дозволяють спростити складові проекту. За допомогою цього засобу можна досить наочно і зручно для узагальнення представити велику кількість даних. Завдяки цій гістограмі велика кількість зацікавлених осіб, команд або їх учасників не стане проблемою для запису завдань, як і часті зміни обсягу роботи. Ще одна перевага використання діаграми Ганта полягає в тому, що вона дає загальне уявлення про проект в цілому, в тому числі про всі контрольних точках і терміни виконання. Діаграму Ганта можна уявити як ефективний засіб раннього попередження.

Розглянемо створену діаграму Ганта до заданої інформаційної системи (рис.Б.3.1-2).

Назва задачі	Протяжність днів
Ознайомлення з предметною областю	2
Аналіз конкурентів	2
Ідентифікація ідей	8
Підготовка матеріалу	10
Вивчення вимог проекту	8
Планування ризиків	4
Діаграма Ганта	7
Мережевий графік	7
Розробка прототипу	3
Підготовка макету дизайну сайту	4
Верстка	6
Робота над програмною частиною	6
Наповнення сайту контентом	5
Тестування	8
Підготовка документації	10
Перенесення сайту на сервер	11

Рисунок Б.3.1 – Діаграма Ганта (інформація)

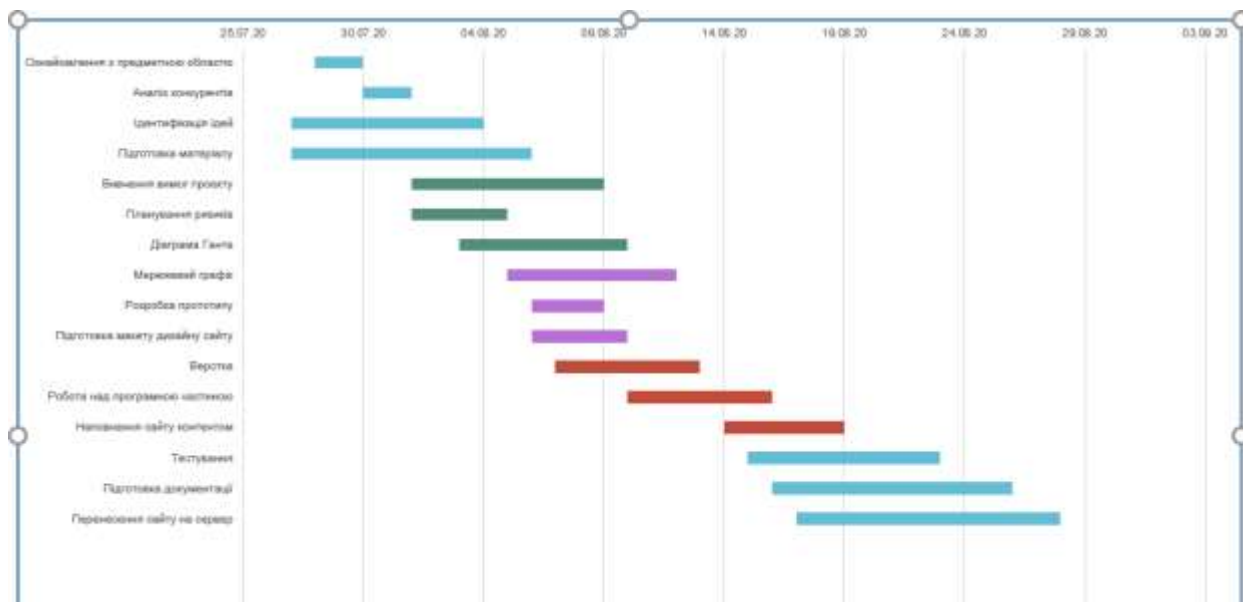


Рисунок Б.3.2 – Діаграма Ганта(графік)

4 УПРАВЛІННЯ РИЗИКАМИ

Ризиком є ймовірна подія, яка у випадку її виникнення може як негативно, так і позитивно вплинути на конкретний проект. Управління ризиком – це процес зміни ризиків та реагування на події під час виконання проекту. При реалізації проекту важливою частиною є моніторинг ризиків. Отже, в даному випадку можна виділити деякі ризики.

Під час процесу аналізу для визначення числових значень ймовірності появи ступеня впливу, застосовувалася методика експертних оцінок. Виходячи цих оцінок можливо знайти ранг ризиків: $R = P * L$, де

- R – ранг ризику;
- P – ймовірність виникнення;
- L – ступінь впливу.

Шкала оцінки ризику може відповідати емпіричній шкалі оцінки ризику:

- 5 балів - критичний ризик (0,81 - 1);
- 4 бали - максимальний ризик (0,61 - 0,8);
- 3 бали - високий ризик (0,41 - 0,6);
- 2 бали - нормальний ризик (0,31 - 0,4); 1 бал - малий ризик (0 - 0,3).

Оцінки ризиків проекту наведено в табл. 4.1.

Таблиця Б.4.1 – Таблиця ризиків

№	Об'єкт ризику	Ризик	P	L	R
1	Час	Зміна пріоритету проекту	0,5	0,3	0,15
2	Якість	Системи, що не відповідають задачам, грубі помилки в алгоритмах процесів, критичні збої системи	0,1	0,4	0,04
3	Бюджет	Поява позапланових робіт по проекту	0,7	0,2	0,14
4	Трудові ресурси та їх кваліфікація	Неможливість участі в запланованих роботах по проекту необхідних співробітників зі сторони замовника та виконавця у зв'язку з відпусткою, відрядженням та ін.	0,1	0,1	0,01
5	Інтеграція	Не вірна інформація, щодо зовнішніх систем, з яких передбачена взаємодія в межах проекту	0,5	0,4	0,2
6	Ринок	Розширення функціональних характеристик програмних продуктів, які використовувалися замовником в межах мети проекту	0,1	0,05	0,005
7	Ринок	Зміна вимог замовника, економічні зміни, посилення конкуренції, втрата позицій на ринку;	0,5	0,1	0,05
8	Бюджет	Зриви планів робіт; невірна стратегія; персонал без потрібної кваліфікації; переоплати за роботу/матеріали; непогоджені частини проекту; невірний кошторис.	0,5	0,2	0,1

ДОДАТОК В.

КОД РЕАЛІЗАЦІЇ WEB-ДОДАТКУ

Index.vue

Код головної сторінки сайту.

```

<template>
  <div>
    <div class="header-bg"></div>
    <b-container>
      <div class="block-title">
        Оренда авто в Україні
        <div class="sub-title">
          У нас ви можете взяти автомобіль в оренду без водія і не відчувати обмежень при пересуванні. Нашу високу планку надання послуг вже змогли оцінити постійні клієнти в Харкові, Києві, Дніпрі, Львові, Одесі та Вінниці. Оренда авто з нашою компанією - це не розкіш, а елемент комфортного побуту. Використовуючи прокат автомобілів в своїх цілях ви можете організувати автомобільний парк для свого бізнесу або взяти машину напрокат для особистого користування.
        </div>
      </div>
    </div>
    <b-row>
      <b-col class="help-item">
        
        <div class="title">
          Тільки у справі
        </div>
        <div class="description">
          Всі відгуки залишені клієнтами, які вже скористалися послугами.
        </div>
      </b-col>
      <b-col class="help-item">
        
        <div class="title">
          Знаємо, що пропонуємо
        </div>
        <div class="description">
          Ми стежимо за рівнем надання послуг і рекомендуємо перевірені нами компанії.
        </div>
      </b-col>
      <b-col class="help-item">
        
        <div class="title">
          Простий онлайн-запис
        </div>
        <div class="description">
          Замовляєте послугу, отримуєте SMS з підтвердженням замовлення, а також SMS-нагадування про майбутній візит.
        </div>
      </b-col>
    </b-row>
  </div class="block-title">

```

```

    ОСТАННІ НОВИНИ
  </div>
  <NewsItem :data="news"></NewsItem>

  <b-row class="auto-items">
    <b-col cols="4" v-for="item in cars" :key="item.id">
      <ServiceItem :data="item"></ServiceItem>
    </b-col>
    <b-col cols="4">
      <div class="go-all">
        <div class="text">Обери найкращий варіант для себе</div>
        <router-link to="/auto">Всі авто</router-link>
      </div>
    </b-col>
  </b-row>
</b-container>
</div>
</template>
<script>
import ServiceItem from "../../components/site/ServiceItem";
import NewsItem from "../../components/site/NewsItem";
export default {
  components: {
    ServiceItem,
    NewsItem
  },
  data() {
    return {
      news: [],
      cars: []
    }
  },
  created() {
    this.getNews();
    this.getCars();
  },
  methods: {
    getNews() {
      axios.get('/api/news', {
        params: {
          limit: 3
        }
      })
      .then((response) => {
        this.news = response.data
      })
    },
    getCars() {
      axios.get('/api/cars', {
        params: {
          limit: 2
        }
      })
      .then((response) => {
        this.cars = response.data
      })
    }
  }
}

```

```

        ))
    },
}
}
</script>
<style lang="css" scoped>
.auto-items {
    margin: 90px 0 50px 0;
}
.auto-items .go-all {
    color: #1F2326;
    font-weight: 600;
    font-size: 38px;
    line-height: 51px;
    position: relative;
    height: 100%;
    background: #fff;
    box-shadow: 0px 3px 6px rgba(75, 81, 91, 0.15), 0px 1px 3px rgba(0, 0, 0, 0.15);
    border-radius: 10px;
}
.auto-items .go-all .text {
    padding: 25px 35px;
}
.auto-items .go-all a {
    width: 100%;
    text-decoration: none;
    display: block;
    padding: 15px 0;
    position: absolute;
    bottom: 0;
    background: #137B4F;
    box-shadow: 0px 3px 6px rgba(75, 81, 91, 0.15), 0px 1px 3px rgba(0, 0, 0, 0.15);
    border-radius: 10px;
    text-align: center;
    font-weight: bold;
    font-size: 24px;
    color: #fff;
    text-transform: uppercase;
}
.help-item {
    text-align: center;
    padding: 0 50px;
    color: #fff;
}
.help-item .title {
    color: #fff;
    font-weight: normal;
    font-size: 20px;
    line-height: 23px;
    margin-bottom: 20px;
    padding: 30px 0 0 0;
}
.help-item .description {
    color: #fff;
    font-weight: 300;
}

```



```

    font-size: 13px;
    line-height: 20px;
  }
  .block-title {
    text-align: center;
    font-size: 31px;
    line-height: 37px;
    color: #fff;
    font-weight: normal;
    padding: 60px 0;
  }
  .block-title .sub-title {
    font-weight: 300;
    font-size: 13px;
    line-height: 20px;
    text-align: center;
    margin-top: 13px;
  }
  .header-bg {
    height: 600px;
    background: url('/img/home_bg.png') no-repeat;
    background-size: cover;
  }
</style>

```

Cars.vue

Код сторінки пошуку авто.

```

<template>
  <div style="background: #137B4F">
    <div class="title-page">
      Автомобілі
    </div>
    <b-container class="wrapper-page">
      <div class="form-search">
        <b-row class="inputs m-0">
          <b-col class="pl-0">
            <select v-model="search.models">
              <option value="" disabled selected>Модель</option>
              <option v-for="item in models" :value="item" :key="item.id">{{ item.title }}</option>
            </select>
          </b-col>
          <b-col>
            <select v-model="search.series_id">
              <option value="" disabled selected>Серія</option>
              <option v-for="item in search.models.series" :value="item.id" :key="item.id">{{ item.title
            }}</option>
            </select>
          </b-col>
          <b-col>
            <select v-model="search.regions">
              <option value="" disabled selected>Область</option>
              <option v-for="item in regions" :value="item.id" :key="item.id">{{ item.title }}</option>
            </select>
          </b-col>
        </b-row>
      </div>
    </b-container>
  </div>
</template>

```

```

    <b-col class="pr-0">
      <button class="button" @click="fetchData">Пошук</button>
    </b-col>
  </b-row>
</div>
<div class="d-flex justify-content-center" v-if="preloader">
  <b-spinner style="width: 4rem; height: 4rem;" label="Large Spinner"></b-spinner>
</div>
<div v-if="!preloader && data.length == 0" class="text-center">
  Нічого не знайдено
</div>
<b-row>
  <b-col cols="4" v-for="(item, index) in data" :key="index">
    <ServiceItem :data="item"></ServiceItem>
  </b-col>
</b-row>
  <b-pagination class="mt-5" v-if="data.length > 0" v-model="currentPage" :total-rows="data.length" :per-
page="prePage" align="center"></b-pagination>
</b-container>
<br><br>
</div>
</template>
<script>
import ServiceItem from "../../components/site/ServiceItem";
export default {
  components: {
    ServiceItem
  },
  data() {
    return {
      preloader: true,
      loading: false,
      data: [],
      models: [],
      regions: [],
      series: [],
      currentPage: 1,
      prePage: 9,
      search: {
        models: "",
        series: "",
        regions: "",
        series_id: ""
      }
    }
  },
  computed: {
    paginatedList() {
      let start = (this.currentPage - 1) * this.prePage;
      let end = start + this.prePage;
      return this.data.slice(start, end);
    }
  },
  created() {
    this.getModels();
  }
}

```

```

    this.getRegions();
    this.fetchData();
  },
  methods: {
    fetchData() {
      axios.get('/api/cars', {
        params: {
          series_id: this.search.series_id,
          regions: this.search.regions,
          models: this.search.models.id
        }
      })
      .then((response) => {
        this.preloader = false;
        this.data = response.data;
      }).catch(() => {
        this.preloader = false;
      });
    },
    getModels() {
      axios.get('/api/models').then(response => {
        this.models = response.data;
      })
    },
    getRegions() {
      axios.get('/api/regions').then(response => {
        this.regions = response.data;
      })
    },
  },
}
}
</script>
<style lang="css" scoped>
  .form-search {
    margin-bottom: 40px;
  }
  .form-search .button {
    width: 100%;
    color: #ffffff;
    font-weight: bold;
    background: #1F2326;
    border: 0;
    padding: 10px 0;
    border-radius: 10px;
  }
  .form-search select {
    background: #fff;
    border-radius: 5px;
    padding: 8px 10px;
    margin-left: 15px;
    width: 100%;
    height: 100%;
  }
  .form-search select * {
    color: #000;
  }

```

```

}
</style>

```

CarView.vue

Код сторінки авто.

```

<template>
<div>
  <div style="background: #137B4F;padding: 30px 0">
    <b-container>
      <b-row>
        <b-col cols="9">
          <div class="info block">
            <b-row>
              <b-col cols="5" style="display:flex; align-items: center; overflow: hidden">
                <img v-if="data.photo.length > 0" :src="data.photo[0].src">
              </b-col>
              <b-col class="description">
                <div class="title">
                  {{ data.series.model.title }} {{ data.series.title }} <div class="filler"></div> <span>{{
data.price }} грн за добу</span>
                </div>
                <ul>
                  <li>Тип кузова <div class="filler"></div> <span>{{ data.body_type.title }}</span></li>
                  <li>Об'єм двигуна <div class="filler"></div> <span>{{ data.engine_volume }}</span></li>
                  <li>Тип трансмісії <div class="filler"></div> <span>{{ data.transmission_type
}}</span></li>
                  <li>Тип палива <div class="filler"></div> <span>{{ data.fuel_type }}</span></li>
                  <li>Кількість місць <div class="filler"></div> <span>{{ data.number_passengers
}}</span></li>
                </ul>
              </b-col>
            </b-row>
          </div>
        </b-col>
        <b-col cols="3">
          <div class="menu">
            <ul>
              <li><router-link :to="'/auto/' + $route.params.id">Інформація</router-link></li>
              <li><router-link :to="'/auto/' + $route.params.id + '/photo'">Фото</router-link></li>
              <li><router-link :to="'/auto/' + $route.params.id + '/reviews'">Відгуки</router-link></li>
              <li><router-link :to="'/auto/' + $route.params.id + '/rent'">Орендувати</router-link></li>
            </ul>
          </div>
        </b-col>
      </b-row>
    </b-container>
  </div>
  <b-container class="wrapper">
    <About
      class="pages about block"
      v-if="$route.name == 'service-view'"
      :data="data"
    ></About>
  </b-container>

```

```

<b-container class="py-5" v-if="$route.name === 'service-view-reviews'">
  <Reviews
    style="margin-top: 20px"
    :data="data.reviews"
  ></Reviews>
</b-container>
<div style="background: #137B4F" v-if="$route.name === 'service-view-reviews'">
  <b-container class="form-comment">
    <div class="title" style="color: #fff">Залишити відгук</div>
    <b-row>
      <b-col cols="6">
        <input type="text" placeholder="Прізвище та ім'я" v-model="comment.name">
      </b-col>
      <b-col cols="6">
        <input type="text" placeholder="Електронна адреса" v-model="comment.email">
      </b-col>
      <b-col cols="12">
        <textarea cols="30" rows="10" v-model="comment.text"></textarea>
      </b-col>
    </b-row>
    <div class="text-center">
      <button @click="postReview()">Залитишити відгук</button>
    </div>
  </b-container>
</div>
<b-container class="wrapper">
  <Photo
    class="pages block"
    v-if="$route.name === 'service-view-photo'"
    :photo="data.photo"
  ></Photo>
</b-container>
<b-container class="wrapper">
  <Rent
    class="pages block"
    v-if="$route.name === 'service-view-rent'"
  ></Rent>
</b-container>
</div>
</template>
<script>
import Reviews from "../../components/site/service/Reviews";
import Rent from "../../components/site/service/Rent";
import Photo from "../../components/site/service/Photo";
import About from "../../components/site/service/About";
export default {
  components: {
    Reviews,
    Rent,
    Photo,
    About
  },
  data() {
    return {
      data: {

```

```

    photo: [],
    price: "",
    engine_volume: "",
    transmission_type: "",
    fuel_type: "",
    number_passengers: "",
    body_type: {
      title: ""
    },
    series: {
      title: "",
      model: {
        title: ""
      }
    },
    comment: {
      name: "",
      email: "",
      text: ""
    }
  },
  created() {
    this.fetchData();
  },
  methods: {
    fetchData() {
      axios.get('/api/car/'+this.$route.params.id)
        .then((response) => {
          this.data = Object.assign(this.data, response.data);
        })
    },
    postReview() {
      axios.post('/api/review/'+this.$route.params.id, this.comment)
        .then(() => {
          this.fetchData();
          this.comment.name = "";
          this.comment.email = "";
          this.comment.text = "";
        })
    }
  }
}
</script>
<style lang="css" scoped>
  .about {
    color: #000;
    padding: 40px;
  }
  .menu {
    height: 100%;
    display: flex;
    align-items: stretch;
    justify-content: center;

```

```

    flex-direction: column;
}
.menu ul {
  display: flex;
  padding: 0;
  margin: 0;
  flex-wrap: nowrap;
  flex-direction: column;
  align-content: center;
  justify-content: center;
  align-items: center;
}
.menu ul li {
  list-style: none;
  width: 100%;
  background: #FFFFFF;
  box-shadow: 0px 2.06299px 4.12598px rgba(75, 81, 91, 0.15), 0px 0.687664px 2.06299px rgba(0, 0, 0, 0.15);
  border-radius: 7px;
  padding: 12px 0;
  text-align: center;
  margin: 10px 0;
}
.menu ul li a {
  text-transform: uppercase;
  color: #1F2326;
  font-weight: bold;
  font-size: 12px;
}
.menu ul li:hover, .menu ul li.active {
  background: #1F2326;
}
.menu ul li:hover a, .menu ul li.active a {
  color: #fff;
  text-decoration: none;
}
.block {
  background: #FFFFFF;
  box-shadow: 0px 3px 6px rgba(75, 81, 91, 0.15), 0px 1px 3px rgba(0, 0, 0, 0.15);
  border-radius: 10px;
}
.info .description {
  padding: 20px 30px;
}
.info ul {
  margin: 0;
  padding: 0;
  color: #373B53;
}
.info ul li {
  list-style: none;
  border-bottom: 1px solid #5a5a5a;
  padding: 5px 0;
  display: flex;
  font-size: 18px;
}

```

```
    margin-bottom: 7px;
  }
.info img {
  margin: 0 auto;
  height: 210px;
}
.title {
  color: #373B53;
  font-weight: 500;
  font-size: 30px;
  line-height: 44px;
  margin-bottom: 20px;
  display: flex;
}
.title span {
  font-size: 25px;
  color: #137B4F;
  font-weight: 400;
}
.info .description {
  color: #A6ACBE;
  font-weight: 500;
  font-size: 14px;
  line-height: 21px;
  padding: 20px 50px;
}
.pages {
  margin-top: 20px;
  padding: 50px;
}
.form-comment {
  padding: 30px 20px;
}
.form-comment .title {
  font-weight: bold;
  font-size: 30px;
  margin-bottom: 40px;
}
input {
  outline: none;
  border: 2px solid #EBEAED;
  box-sizing: border-box;
  border-radius: 100px;
  width: 100%;
  font-weight: bold;
  font-size: 18px;
  color: #000;
  padding: 14px 24px;
  margin-bottom: 30px;
}
textarea {
  padding: 14px 24px;
  outline: none;
  width: 100%;
  background: #fff;
```



```

border: 2px solid #EBEAED;
box-sizing: border-box;
border-radius: 10px;
margin-bottom: 30px;
}
button {
padding: 16px 45px;
color: #FFF;
background: linear-gradient(180deg, #30CE95 0%, #A1D85A 100%);
font-weight: 600;
font-size: 15px;
border: 0;
border-radius: 50px;
}
</style>

```

Calendar.vue

Код сторінки календаря.

```

<template>
<div style="background: #137B4F;">
  <b-container>
    <div class="title-page">
      Календар
    </div>
    <b-row>
      <b-col cols="9">
        <div class="calendar-block">
          <table id="calendar">
            <thead>
              <tr>
                <td><b-icon icon="chevron-left" scale="1.5" style="cursor: pointer" @click="nextMonth()"></b-
icon></td>
                <td colspan="5"></td>
                <td><b-icon icon="chevron-right" scale="1.5" style="cursor: pointer" @click="prewMonth()"></b-
icon></td>
              </tr>
              <tr>
                <td style="padding: 70px 0 40px 0">Пн</td>
                <td style="padding: 70px 0 40px 0">Вт</td>
                <td style="padding: 70px 0 40px 0">Ср</td>
                <td style="padding: 70px 0 40px 0">Чт</td>
                <td style="padding: 70px 0 40px 0">Пт</td>
                <td style="padding: 70px 0 40px 0">Сб</td>
                <td style="padding: 70px 0 40px 0">Дд</td>
              </tr>
            </thead>
            <tbody>
            </tbody>
          </table>
        </div>
      </b-col>
      <b-col cols="3" class="form-search">
        <select v-model="search.models">
          <option value="" disabled selected>Модель</option>

```

```

    <option v-for="item in models" :value="item" :key="item.id">{{ item.title }}</option>
  </select>
  <select v-model="search.series_id">
    <option value="" disabled selected>Серія</option>
    <option v-for="item in search.models.series" :value="item.id" :key="item.id">{{ item.title }}</option>
  </select>
  <button class="button" @click="getDays()">Пошук</button>
</b-col>
</b-row>
</b-container>
<br><br>
</div>
</template>
<script>
export default {
  data() {
    return {
      models: [],
      days: [],
      search: {
        models: "",
        series_id: ""
      }
    }
  },
  created() {
    this.getModels();
    this.getDays();
  },
  methods: {
    nextMonth() {
      this.Calendar("calendar", document.querySelector('#calendar thead td:nth-child(2)').dataset.year,
parseFloat(document.querySelector('#calendar thead td:nth-child(2)').dataset.month)-1);
    },
    prewMonth() {
      this.Calendar("calendar", document.querySelector('#calendar thead td:nth-child(2)').dataset.year,
parseFloat(document.querySelector('#calendar thead td:nth-child(2)').dataset.month)+1);
    },
    getModels() {
      axios.get('/api/models').then(response => {
        this.models = response.data;
      })
    },
    getDays() {
      axios.get('/api/order-days', {
        params: {
          series_id: this.search.series_id,
          models: this.search.models.id
        }
      }).then(response => {
        this.days = response.data;
        this.Calendar("calendar", new Date().getFullYear(), new Date().getMonth());
      })
    },
    Calendar(id, year, month) {

```



```

margin-bottom: 15px;
width: 100%;
height: 60px;
}
.form-search .button {
width: 100%;
color: #ffffff;
font-weight: bold;
background: #1F2326;
border: 0;
padding: 10px 0;
border-radius: 5px;
height: 60px;
}
.form-search select * {
color: #000;
}
}
</style>

```

CarController.php

Контролер для авто.

```

<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Models;
use App\Models\Series;
use App\Models\BodyType;
use App\Models\Cars;
use App\Models\CityHasCars;
use App\Models\CarsPhoto;
use App\Models\Reviews;

class CarController extends Controller
{
protected $publicStorage = 'uploads/cars/';

function getCars(Request $request) {
$model = Cars::with('bodyType', 'series.model', 'reviews', 'photo', 'regions');
if(isset($request->models) && $request->models != "") {
$model->whereHas('series.model', function($q) use ($request) {
$q->where('id', $request->models);
});
}
if(isset($request->series_id) && $request->series_id != "") {
$model->whereHas('series', function($q) use ($request) {
$q->where('id', $request->series_id);
});
}
if(isset($request->regions) && $request->regions != "") {
$model->whereHas('regions', function($q) use ($request) {
$q->where('region_id', $request->regions);
}
}
}
}

```

```

    });
  }
  if(isset($request->limit)) {
    $model->limit($request->limit);
  }
  $data = $model->get();
  return response()->json($data);
}

```

```

function getCarOne($id) {
  $data = Cars::with('bodyType', 'series.model', 'reviews', 'photo', 'regions.region')->find($id);
  return response()->json($data);
}

```

```

function postCar(Request $request) {
  $model = new Cars();
  $data = $request->all();
  $response = $model->create($data);
  if($request['regions']) {
    foreach (json_decode($request['regions']) as $key => $value) {
      $model = new CityHasCars();
      $model->create([
        'cars_id' => $response['id'],
        'region_id' => $value
      ]);
    }
  }
  $uploadedFiles = $request->images;
  foreach ($uploadedFiles as $key => $file) {
    $foto = new CarsPhoto;
    $image = str_replace('data:image/jpeg;base64,', '', $file);
    $image = str_replace(' ', '+', $image);
    $imageName = $this->publicStorage . uniqid() . '.' . 'jpg';
    \File::put(public_path(). $imageName, base64_decode($image));
    $foto->create([
      "cars_id" => $response['id'],
      "src" => $imageName
    ]);
  }
  return response('ok', 200);
}

```

```

function putCar(Request $request, $id) {
  $model = Cars::find($id);
  $data = $request->all();
  $model->update($data);
  if($request['regions']) {
    CityHasCars::where('cars_id', $id)->delete();
    foreach (json_decode($request['regions']) as $key => $value) {
      $model = new CityHasCars();
      $model->create([
        'cars_id' => $id,
        'region_id' => $value
      ]);
    }
  }
}

```

```

    }
    $uploadedFiles = $request->images;
    if(isset($uploadedFiles)) {
        foreach ($uploadedFiles as $key => $file) {
            $foto = new CarsPhoto;
            $image = str_replace('data:image/jpeg;base64,', '', $file);
            $image = str_replace(' ', '+', $image);
            $imageName = $this->publicStorage . uniqid() . '.' . 'jpg';
            \File::put(public_path(). $imageName, base64_decode($image));
            $foto->create([
                "cars_id" => $id,
                "src" => $imageName
            ]);
        }
    }
    return response('ok', 200);
}

function getSeries() {
    $data = Series::get();
    return response()->json($data);
}

function getModels() {
    $data = Models::with('series')->get();
    return response()->json($data);
}

function getBodyType() {
    $data = BodyType::get();
    return response()->json($data);
}

function postReview(Request $request, $car_id) {
    $model = new Reviews();
    $data = $request->all();
    $data['cars_id'] = $car_id;
    $model->create($data);
    return response('ok', 200);
}
}

```

OrderController.php

Контролер для замовлень.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Orders;
use App\Models\DayHasOrder;

class OrderController extends Controller

```

```

{
function getOrderDays(Request $request) {
    $model = DayHasOrder::whereHas('order', function($q) {
        $q->where('status', 0);
    });
    if(isset($request->models) && $request->models != "") {
        $model->whereHas('car.series.model', function($q) use ($request) {
            $q->where('id', $request->models);
        });
    }
    if(isset($request->series_id) && $request->series_id != "") {
        $model->whereHas('car.series', function($q) use ($request) {
            $q->where('id', $request->series_id);
        });
    }
    $data = $model->get();
    return response()->json($data);
}

function getOrders() {
    $data = Orders::with('car.series.model')->get();
    return response()->json($data);
}

// postOrder
function postOrder(Request $request, $id) {
    $model = new Orders();
    $data = $request->all();
    $data['cars_id'] = $id;
    $response = $model->create($data);
    for ($i = 0; $i < $request->days; $i++) {
        $model = new DayHasOrder();
        $date = date('Y-m-d', strtotime($request->start_date . "+" . $i . " days"));
        $model->create([
            "day" => $date,
            "cars_id" => $id,
            "orders_id" => $response['id']
        ]);
    }
    return response('ok', 200);
}

// updateOrder
function updateOrder($id) {
    Orders::find($id)->update([
        "status" => 1
    ]);
    return response('ok', 200);
}

// delOrder
function delOrder($id) {
    Orders::find($id)->delete();
    return response('ok', 200);
}
}

```

NewsController.php

Контролер новин.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Carbon\Carbon;
use Illuminate\Support\Facades\Auth;

use App\Models\News;
use App\Models\Comments;

class NewsController extends Controller
{
    protected $publicStorage = '/uploads/news/';

    function get(Request $request) {
        $model = new News;
        if(isset($request->limit)) {
            $model->limit($request->limit);
        }
        $data = $model->get();
        return response()->json($data);
    }

    function post(Request $request) {
        $post = new News();
        $data = $request->all();
        if(gettype($request['photo']) == "object") {
            $name = $this->publicStorage . uniqid() . '.' . $request['photo']->getClientOriginalExtension();
            $request['photo']->move(public_path() . $this->publicStorage, $name);
            $data['photo'] = $name;
        }
        $post->create($data);
        return response('ok', 200);
    }

    function update(Request $request, $id) {
        $data = $request->all();
        $model = News::find($id);
        if(gettype($request['photo']) == "object") {
            $name = $this->publicStorage . uniqid() . '.' . $request['photo']->getClientOriginalExtension();
            $request['photo']->move(public_path() . $this->publicStorage, $name);
            $data['photo'] = $name;
        } else {
            $data['photo'] = $model->photo;
        }
        $model->update($data);
        return response('ok', 200);
    }

    function delete($id) {
```



```

    News::find($id)->delete();
    return response('ok', 200);
}

function getId($id) {
    $data = News::with('comments')->find($id);
    return response()->json($data);
}

function postComment(Request $request, $id) {
    $model = new Comments();
    $model->create([
        "news_id" => $id,
        "text" => $request->text,
        "email" => $request->email,
        "name" => $request->name
    ]);
    return response('ok', 200);
}
}

```

Cars.php

Модель авто.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cars extends Model
{
    use HasFactory;

    protected $table = 'cars';

    protected $fillable = [
        'body_type_id',
        'series_id',
        'engine_volume',
        'transmission_type',
        'fuel_type',
        'description',
        'number_passengers',
        'price',
        'status',
        'gas_mileage'
    ];

    function bodyType() {
        return $this->belongsTo('App\Models\BodyType', 'body_type_id');
    }
    function series() {
        return $this->belongsTo('App\Models\Series', 'series_id');
    }
}

```

```

    }
    function reviews() {
        return $this->hasMany('App\Models\Reviews', 'cars_id');
    }
    function photo() {
        return $this->hasMany('App\Models\CarsPhoto', 'cars_id');
    }
    function regions() {
        return $this->hasMany('App\Models\CityHasCars', 'cars_id');
    }
}

```

Orders.php

Модель замовлень.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Orders extends Model
{
    use HasFactory;

    protected $table = 'orders';

    protected $fillable = [
        'name',
        'email',
        'phone',
        'comment',
        'status',
        'cars_id'
    ];

    function car() {
        return $this->belongsTo('App\Models\Cars', 'cars_id');
    }
}

```

Reviews.php

Модель відгуків.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Reviews extends Model
{
    use HasFactory;
}

```

```
protected $table = 'reviews';

protected $fillable = [
    'name',
    'email',
    'text',
    'cars_id'
];

protected $casts = [
    'created_at' => 'datetime:d.m.Y'
];

function car() {
    return $this->belongsTo('App\Models\Cars', 'cars_id');
}
}
```