

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»

на тему:

«База даних управління системою прокату автомобілів»

студента II курсу групи ПЗ-20-4
спеціальності 121 «Інженерія програмного
забезпечення»

Дубинченка Богдана Миколайовича
(прізвище, ім'я та по-батькові)

Керівник ст. викл. кафедри ПЗ
Чижмотря О.В.

Дата захисту: " ____ " _____ 20__ р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

_____	<u>Ольга КОРОТУН</u>
(підпис)	(прізвище та ініціали)
_____	<u>Світлана КРАВЧЕНКО.</u>
(підпис)	(прізвище та ініціали)
_____	<u>Інна СУГОНЯК</u>
(підпис)	(прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп'ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о зав кафедри ІПЗ

Андрій МОРОЗОВ

“ ” 2021 р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ
Дубинченку Богдану Миколайовичу

- Тема роботи: Розробка база даних управління системою прокату автомобілів
керівник роботи: ст. викл. Кафедри ІПЗ Чижмотря О.В.
- Строк подання студентом: “15” липня 2022р.
- Вихідні дані до роботи: Розробити базу даних управління системою прокату автомобілів.
- Зміст розрахунково-пояснювальної записки (перелік питань, що підлягають розробці)
 - Аналіз інформаційних потоків та особливостей предметної області дослідження
 - Проектування бази даних за напрямком курсової роботи
 - Реалізація підсистеми обробки даних за напрямком курсової роботи
 - Адміністрування баз даних
- Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 - Презентація до КР _____
 - Посилання на репозиторій: <https://gitlab.com/2020-2024/ipz-20-4/dubinchenko-bogdan/kursova-car-rental>
- Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Чижмотря О.В., ст. викладач каф. ІПЗ	10.02.2022	10.02.2022

7. Дата видачі завдання “10” лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Примітки
1	Постановка задачі	10.02.21-11.02.21	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	11.02.21-13.02.21	Виконано
3	Формулювання технічного завдання	13.02.21-17.02.21	Виконано
4	Опрацювання літературних джерел	17.02.21-23.02.21	Виконано
5	Проектування структури	23.02.21-25.02.21	Виконано
6	Написання програмного коду	25.02.21-17.07.22	Виконано
7	Відлагодження	17.07.22-22.07.22	Виконано
8	Написання пояснювальної записки	22.07.22 - 25.07.22	Виконано
9	Захист	26.07.22	

Студент

_____ (підпис)

Дубинченко Б.М.

(прізвище та ініціали)

Керівник проекту

_____ (підпис)

Чижмотря О.В.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до курсового проекту на тему «База даних управління системою прокату автомобілів» складається з переліку умовних скорочень, вступу, чотирьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 55 сторінках друкованого тексту. Пояснювальна записка має 8 сторінок додатків. Список використаних джерел містить 12 найменувань і займає 1 сторінку. В роботі наведено 42 рисунків. Загальний обсяг роботи - 64 сторінок.

У першому розділі було проаналізовано інформаційні потреби, визначено предметну область та реалізовано засоби реалізації бази даних.

У другому розділі проведено аналіз інформаційних процесів адміністраторів, спроектовано базу даних для прокату автомобілів та розроблено математичні моделі та алгоритми обробки інформації додатку управління БД.

У третьому розділі було розроблено систему управління прокату авто.

У четвертому розділі було розроблено систему захисту інформації в базі даних.

Висновок містить в собі результати виконаної роботи при створенні сайту на тему “База даних управління системою прокату автомобілів”. У додатку представлений лістинг розробленого програмного продукту.

Ключові слова: NODE.JS, EXPRESS, MONGODB, WEB-САЙТ, БД, КОЛЕКЦІЇ, ОРЕНДА АВТО, REACT, REDUX, CLOUDINARY, ЗАПИТ, АВТОРИЗАЦІЯ

					ДУ «Житомирська політехніка».22.121.07.000 - ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Дубинченко Б.М.			Розробка база даних управління системою прокату автомобілів	Лім.	Арк.	Аркушів	
Перевір.							4	0	
Керівник		Чижмoтря О.В				ФІКТ Гр. ІПЗ-20-4[1]			
Н. контр.									
Зав. каф.									

Зміст

ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	8
1.1 Аналіз інформаційних потреб та визначення предметної області дослідження	8
1.2 Обґрунтування вибору засобів реалізації.....	11
Висновки до першого розділу :	16
РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ ПРОКАТУ АВТОМОБІЛІВ	17
2.1 Аналіз інформаційних процесів.....	17
2.2 Проектування структури бази даних прокату автомобілів	20
2.3 Розробка математичної моделі та алгоритмів обробки даних додатку прокату автомобілів	25
Висновки до 2 розділу :.....	27
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ПРОКАТУ АВТОМОБІЛІВ	28
3.1 Проектування інтерфейсу обробки даних	28
3.2. Реалізація операцій обробки даних в БД	42
3.3. Організація звітності системи.....	47
Висновки до 3 розділу :.....	49
РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ	50
4.1. Розробка заходів захисту інформації в БД.....	50
4.2. Налаштування параметрів роботи MONGODB.....	51
Висновки до 4 розділу :.....	53
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ	56

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних

КР – курсова робота

Node.JS – JavaScript – оточення побудоване на JavaScript-рушієві Chrome V8.

CRUD – create, read, update, delete

СУБД – система управління базою даних

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У цій курсовій роботі буде наведено процес розробки клієнтської та серверної частини додатку для управління системою прокату автомобілів.

Актуальність теми. Послуга автомобіля на прокат досить поширена в наш час. Найчастіше автомобіль на прокат беруть ті, у кого немає особистого транспортного засобу, або ще одна категорія користувачів – туристи, які хочуть пересуватися по чужому місту з комфортом.

Маючи зручну систему обліку, керування прокатом автомобілів можна спростити у декілька разів, бо це дозволяє легко створювати нові замовлення, вести звітність, переглядати наявні автомобілі та їх характеристики.

Тому все більше з'являється потреба у створенні додатків для автоматизації та моніторингу систем прокату автомобілів.

Метою курсової роботи є дослідження особливостей проектування та реалізації баз даних за визначеним темою курсової роботи.

Завданням на курсову роботу є :

- аналіз теоретичних засад проектування та реалізації систем на основі баз даних;
- визначення інформаційних потреб предметної області дослідження;
- аналіз напрямку ризиків інформаційних потоків та їх структури;
- проектування бази даних за визначеною предметною областю;
- розробка математичної та алгоритмічної моделі функціонування системи на основі БД;
- реалізація БД та інтерфейсних засобів інформаційної системи.

Об'єктом дослідження є методи та засоби проектування баз даних за визначеними предметними областями.

Предмет дослідження можливості застосування концепції БД та СУБД для забезпечення інформаційних потреб предметної області.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз інформаційних потреб та визначення предметної області дослідження

База даних (БД) - це організована структура, призначена для зберігання, зміни й обробки взаємопов'язаної інформації, переважно великих обсягів. БД активно використовують у Web-додатках майже у всіх сферах діяльності, найчастіше це інтернет-магазини, системи обліку, інформаційні портали, блоги. Також об'єднання великої кількості даних в єдину базу дає змогу для формування безлічі варіації групування інформації – дані клієнтів, автомобілів, історія оренд. Головною перевагою БД є швидкість внесення та використання потрібної інформації.

Предметна область – застосування конкретної БД для зберігання інформації лише для системи прокату автомобілів та забезпечення можливості виконання різних операцій, такі, як видалення, створення, редагування, вибірка за параметрами.

Наш додаток потребує використання бази даних, адже повинен зберігати чимало різних даних. Потрібно реалізувати додавання нових автомобілів, оформлювати прокат та переглядати всі наявні записи. Оскільки це додаток для ведення обліку локального прокату автомобілів, неавторизований користувач не буде мати доступу. Крім цього потрібно буде передбачити введення системи реєстрації та аутентифікації користувачів. У будь-який час користувач матиме змогу увійти в систему під своїм обліковим записом, або за потреби створити новий, але зі обмеженим доступом, а точніше тільки перегляд наявних автомобілів та категорії.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижевотря О.В				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Для коректного виконання запитів на отримання даних, дані повинні вноситись та змінюватись у БД. За це будуть відповідати деякий перелік операцій:

- Реєстрації користувача: Додаток повинен перевіряти унікальність поштових адресів, що при успішній перевірці даних зберігає їх у колекції користувачів.

- Авторизація користувача: Система перевіряє коректність введених даних, та при успішній перевірці надає доступ до різних даних та функцій, також генерує токени для подальшої більшої безпеки користувача, за це і відповідає дана операція.

- Пошук, вибірка, та сортування користувачів за різними критеріями одночасно: Завдання цієї операції дозволяє користувачам зручно знайти потрібний запис.

- Редагування та видалення користувачів: За потреби адміністратор може редагувати інших користувачів, таким чином він може змінити паспортні дані, або надати роль адміністратора. Також можна видалити записи, за винятком себе.

- Створення категорій для автомобілів: Ця операція надає змогу додати категорію, яку можна застосувати для автомобілів, що зробить пошук більш зручним та розділить записи на групи.

- Сортування та пошук категорії: Потрібна для зручного знаходження потрібно запису

- Редагування та видалення категорій: Адміністратор має можливість змінювати назву та опис категорії, система також перевіряє унікальність назв. Також можливо видаляти зайві записи.

- Перегляд та сортування автомобілів: Для зручності використання всі данні відображаються у таблиці зі основними стовпцями з можливістю сортування записів по декількох параметрах одночасно

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Фільтрація та пошук автомобілів: Щоб знайти найкращий автомобіль за бажаннями орендатора, є можливість фільтрувати за багатьма критеріями, такі як кількість місць, рік виготовлення, рейтинг, статус (“В прокаті”, “На паркінгу”), колір та тип автомобіля, також пошук за полями (“Назва”, “Модель”).

- Перегляд певного автомобіля: Дана операція надає більш детальні подробиці про автомобіль та відображає відгуки інших користувачів, які орендували його раніше.

- Оформлення прокату: Система перевіряє, чи не знаходиться автомобіль в прокаті, якщо ні то з’являється можливість оформлення. При заповненні форми з вибором паспортних даних та кількості днів, у колекцію оренд додається новий запис з обрахування суми до сплати та завдатку.

- Редагування та видалення автомобілів: Дана операція надає можливість адміністратору змінювати інформацію та фото певного автомобіля, за потреби видалити його.

- Створення автомобіля: Операція для додавання нового запису до колекції автомобілів. Адміністратор заповнює форму та додає картинки, а система перевіряє валідність даних.

- Взаємодія зі замовленнями: Якщо статус оренди “В прокаті”, адміністратор має можливість видалити це замовлення або завершити його, таким чином замовлення змінить статус на “Закінчено”.

- Фільтрація, пошук, сортування для замовлень та паспортних даних дуже схожі на попередні операції, відрізняються тільки параметрами.

- Статистика: Для більшості колекцій БД присутні операції для аналізування даних. Наприклад для замовлень є можливість обрати період та групування (по днях, місяцях, роках) в результаті чого отримаємо графік про кількість замовлень та загальну суму. Для користувачів це відображення кількості облікових записів з різними ролями, також графік найкращих працівників з оформлення оренд. Колекція з автомобілями має статистику по кількості оренд.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				10
Змн.	Арк.	№ докум.	Підпис	Дата		

- Резервне копіювання бази даних: Адміністратор має можливість зробити резервну копію БД щоб збереже дані.

- Відновлення бази даних: Операція яка очищує колекції БД, та заповнює даними зі останнього файлу резервного копіювання.

1.2 Обґрунтування вибору засобів реалізації.

Аналіз та вибір СУБД проведемо з урахуванням того, що число клієнтських місць становить від 30 до 500, а доступ до даних має бути максимально ефективним. Обчислювальна техніка працюватиме під керівництвом ОС Windows і Linux. В таблицях наведено порівняльні характеристики СУБД.

Таблиця 1.1

Функціональне порівняння СУБД

Характеристика	Microsoft SQL Server	MongoDB	Postgresql
Адміністративне керування	Добре	Відмінно	Добре
Графічні інструменти	Добре	Відмінно	Відмінно
Простота обслуговування	Відмінно	Відмінно	Відмінно
Механізм даних	Добре	Добре	Добре
Робота с декількома ЦП	Задовільно	Задовільно	Задовільно
Функції з'єднання і вибір індексів	Відмінно	Відмінно	Задовільно
Одночасний доступ декількох користувачів	Відмінно	Відмінно	Відмінно
Обробка даних мультимедіа	Добре	Відмінно	Відмінно
Підключення до Web	Відмінно	Відмінно	Відмінно
Повнотекстовий пошук	Відмінно	Відмінно	Відмінно

Функціональна сумісність	Добре	Добре	Добре
Інтеграція з іншими СУБД	Добре	Добре	Добре
Робота під керівництвом ОС	Задовільно	Добре	Добре
Процедури, що зберігаються та тригери	Відмінно	Добре	Відсутня (нереляційна)
Вбудована мова програмування	Відмінно	Відмінно	Відмінно
Побудова БД	Задовільно	Відмінно	Задовільно
Мова SQL	Відмінно	Відмінно	Відсутня (нереляційна)
Підтримка об'єктно-орієнтованої парадигми	Відмінно	Відмінно	Відмінно
Робота у режимі віддаленого доступу	Відмінно	Відмінно	Відмінно
Розподілена обробка транзакцій	Відмінно	Відмінно	Відсутня (нереляційна)
Дистанційне адміністрування	Задовільно	Відмінно	Задовільно
Засоби аналізу	Відмінно	Відмінно	Відмінно

MongoDB є провідною базою даних документів . Яка побудована на розподіленій масштабованій архітектурі та стала комплексною хмарною платформою для керування та доставки даних у програми. MongoDB обробляє транзакційні, операційні та аналітичні навантаження.

MongoDB зберігає дані як документи у двійковому представленні під назвою BSON (Binary JSON). Поля можуть відрізнятися від документа до

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				12
Змн.	Арк.	№ докум.	Підпис	Дата		

документа; немає необхідності оголошувати системі структуру документів – документи описуються самі. Якщо до документа потрібно додати нове поле, це поле можна створити, не впливаючи на всі інші документи в колекції, не оновлюючи центральний системний каталог, оновлюючи ORM і не переводячи систему в автономний режим. За бажанням перевірку схеми можна використовувати для забезпечення контролю керування даними над кожною колекцією.

PostgreSQL — об'єктно-реляційна система управління базами даних. Є альтернативою як комерційним СУБД (Oracle Database, Microsoft SQL Server,), так і СУБД з відкритим кодом (MySQL SQLite). Ця БД, має певні стратегії для обробки індексування, підвищення паралельності та впровадження оптимізацій і підвищення продуктивності, включаючи розширене індексування, розділення таблиць та інші механізми.

MS SQL Server - система управління базами даних, яка розробляється корпорацією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства.

Адміністративне керування присутнє у всіх поданих БД. Можна створити користувача та надати йому привілеї у формі ролей, для кожного випадку є певні ролі, але за потреби є функції для створення свої ролей з різними правами доступу. Також у кожні БД є гнучка система керування користувачами.

Для MongoDB найкращим графічним інтерфейсом є MongoDB Compass, він дозволяє в інтерактивному режимі запитувати та вивчати дані в колекціях. За

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				13
Змн.	Арк.	№ докум.	Підпис	Дата		

допомогою інструментів візуального редагування спрощуються введення та зміни даних. Розумно інтегрований редактор, який перевіряє схему JSON. Також побудова потужних агрегації дуже проста.

Для спрощення адміністрування на сервері postgresql базовий комплект установки входить такий інструмент як pgAdmin. Він представляє графічний клієнт для роботи з сервером, через який ми у зручному вигляді можемо створювати, видаляти, змінювати бази даних та керувати ними.

Модель даних документа MongoDB природним чином відображається на об'єктах у кодї програми, що полегшує розробникам вивчення та використання. Документи дають можливість представляти ієрархічні зв'язки для зберігання масивів та інших складніших структур. Документи JSON можуть зберігати дані в полях, у вигляді масивів у вигляді вкладених під документів. Таким чином пов'язану інформацію можна зберігати разом для швидкого доступу до запитів за допомогою багатофункціональної та виразної мови запитів MongoDB.

Однією з найпотужніших функцій реляційних баз даних, що полегшує написання програм, є транзакції ACID. Деталі того, як визначаються та реалізуються транзакції ACID.

У реляційній БД дані, моделюватимуться в окремих батьківських і дочірніх таблицях у табличній схемі. Це означає, що оновлення всіх записів одночасно потребуватиме транзакції. У певному сенсі базам даних документів легше реалізовувати транзакції, оскільки вони згруповують дані в документі, а запис і читання документа є операцією для якої не потрібна транзакція з кількома документами. Одне або кілька полів можна записати за одну операцію, включаючи оновлення кількох піддокументів і елементів масиву.

MongoDB гарантує повну ізоляцію під час оновлення документа. Будь-які помилки ініціюватимуть операцію оновлення до відкату, скасовуючи зміни та гарантуючи, що клієнти отримають узгоджений вигляд документа.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				14
Змн.	Арк.	№ докум.	Підпис	Дата		

MongoDB також підтримує транзакції бази даних у багатьох документах, тому фрагменти пов'язаних змін можна зафіксувати або відкотити. Завдяки можливості транзакцій з багатьма документами MongoDB є однією з небагатьох баз даних, яка поєднує в собі гарантії ACID традиційних реляційних баз даних зі швидкістю, гнучкістю та потужністю моделі документів.

У PostgreSQL та інших базах даних SQL необхідно створити схему бази даних і встановити зв'язки даних.

Сильною стороною SQL є його потужна мова запитів із великою системою інструментів. MongoDB не використовує SQL за замовчуванням. Натомість для роботи з документами в MongoDB і видалення даних MongoDB надає власну мову запитів (MQL).

Продуктивність запитів у MongoDB можна прискорити, створивши індекси для полів у документах і під документах. MongoDB дозволяє індексувати будь-які поля документа, включно з тими, які вкладені в масиви та під документи.

MongoDB створено для масштабування. Також дана БД базується на розподіленій архітектурі, яка дозволяє користувачам масштабувати багато екземплярів.

MongoDB реалізував сучасний набір елементів керування та інтеграції кібербезпеки, таке як шифрування на стороні клієнта, що дозволяє здійснювати надсилання даних більш безпечно. PostgreSQL має повний набір функцій безпеки, включаючи багато типів шифрування.

Модель безпеки MS SQL Server — це тісна інтеграція між режимом автентифікації Windows у Windows Server і базою даних. Режим автентифікації Windows працює краще в таких сценаріях: коли є контролер домену, з екземплярами бази даних SQL Server Express.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Для поданих БД притаманні такі риси, як підтримка одночасного маніпулювання даними, підтримка одночасного доступу кільком користувачам, пошук даних за певним значення, яке має зустрічатись в кожному полі.

Найкращі засоби аналізу на мою думку має MongoDB, адже на їх сайті можна переглянути повну інформацію про БД, так як в SQL MS та Postgresql потрібно використовувати скрипти.

Після порівняння вказаних вище СУБД, було вирішено використати MongoDB.

Висновки до першого розділу :

В ході виконання першого розділу було визначено головні операції з БД та навели короткий опис логіки функцій. Також було проаналізовано три СУБД, визначено переваги кожної з них та недоліки. При завершенні розділу було вирішено виконувати курсову роботу за допомогою MongoDB, так як вона має зручний графічний інтерфейс, має не складну документацію, що допоможе виконати поставлені задачі якісно та швидко.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				16
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ ПРОКАТУ АВТОМОБІЛІВ

2.1 Аналіз інформаційних процесів

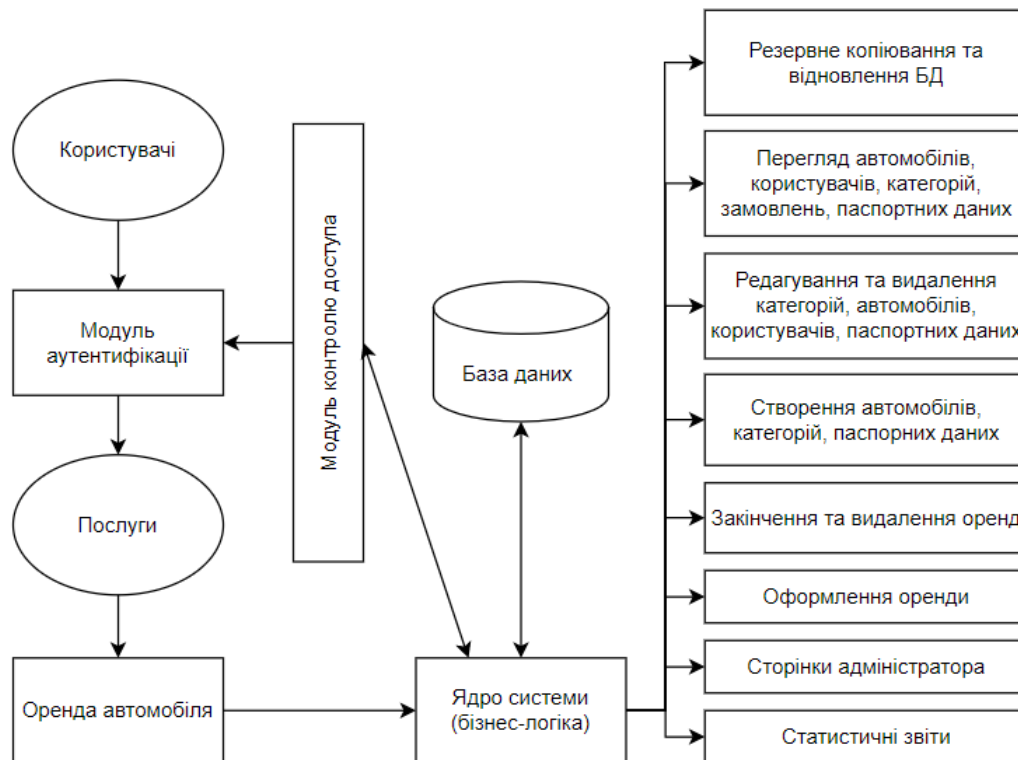


Рис. 2.1. Структура автоматизованої інформаційної системи прокату автомобілів

Аутентифіковані користувачі матимуть змогу переглядати наявні автомобілі та їх категорії, для отримання більшого функціоналу, такого як редагування та видалення записів, оформлення оренд, перегляд паспортних даних та виконання резервного копіювання БД, потрібно мати статус супер користувача.

Роль адміністратора можуть надавати інші користувачі які мають цей статус. Не аутентифіковані користувачі не можуть користуватися додатком.

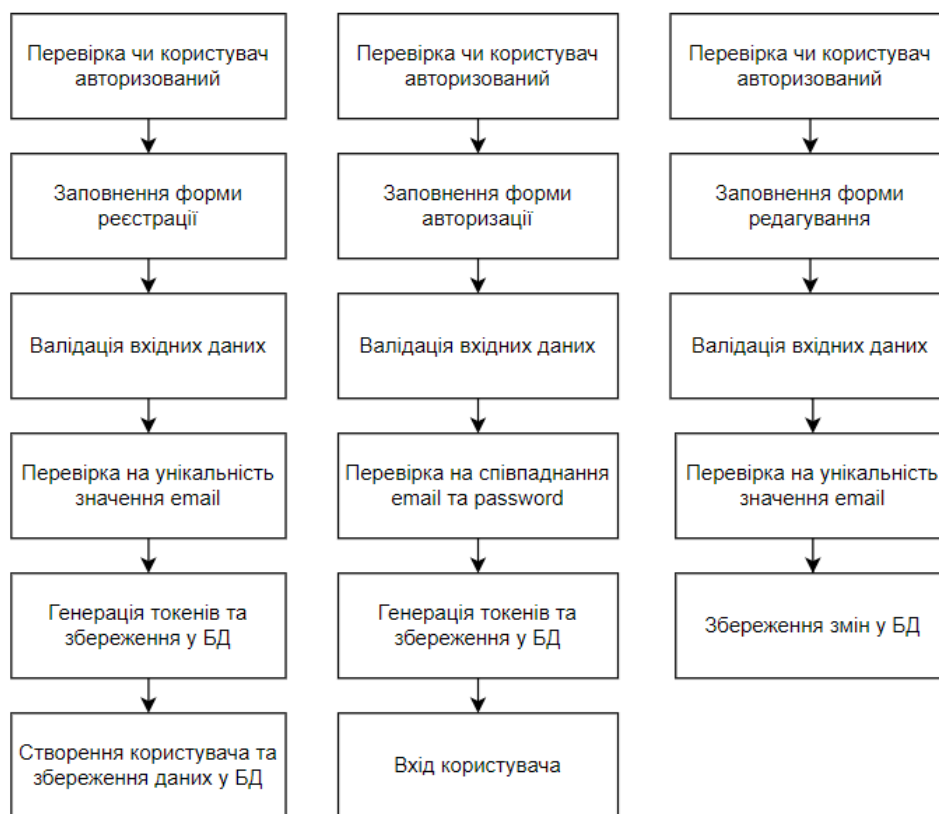


Рис. 2.2. Структурна схема функції систем: реєстрації, авторизації, редагування користувача.



Рис. 2.3. Структурна схема функції систем: створення, редагування, видалення автомобіля та видалення фото.

Для взаємодії з категоріями автомобілів та паспортними даними, структурні схеми функцій ідентичні. На початку перевіряється чи авторизований користувач і чи має він роль адміністратора. Наступний етап валідація даних і при успішному проходженні їх виконується збереження або оновлення запису у певній колекції. Підтвердження

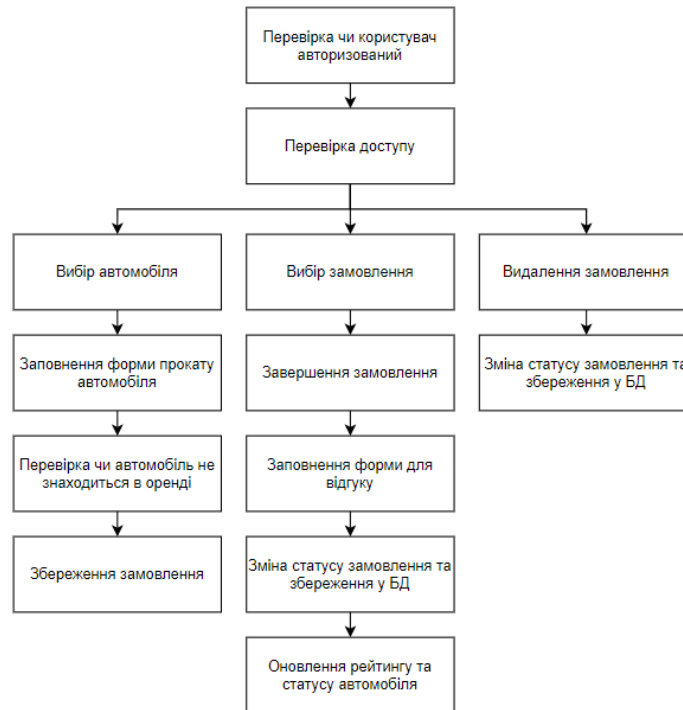


Рис. 2.4. Структурна схема функції систем замовлень, а саме оформлення прокату, завершення прокату, видалення прокату.

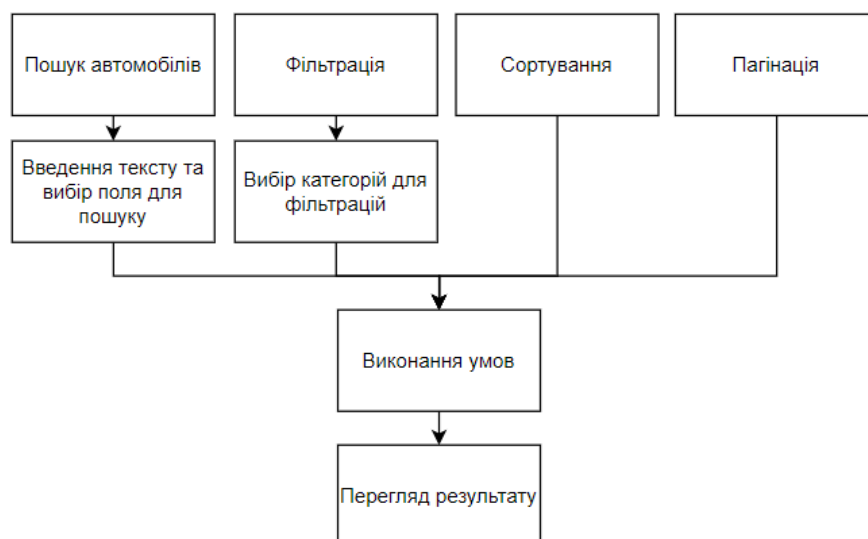


Рис. 2.5. Структурна схема функції систем пошуку, фільтрації, сортування, пагінації



Рис. 2.6. Структурна схема функції систем резервне копіювання та відновлення БД

2.2 Проектування структури бази даних прокату автомобілів

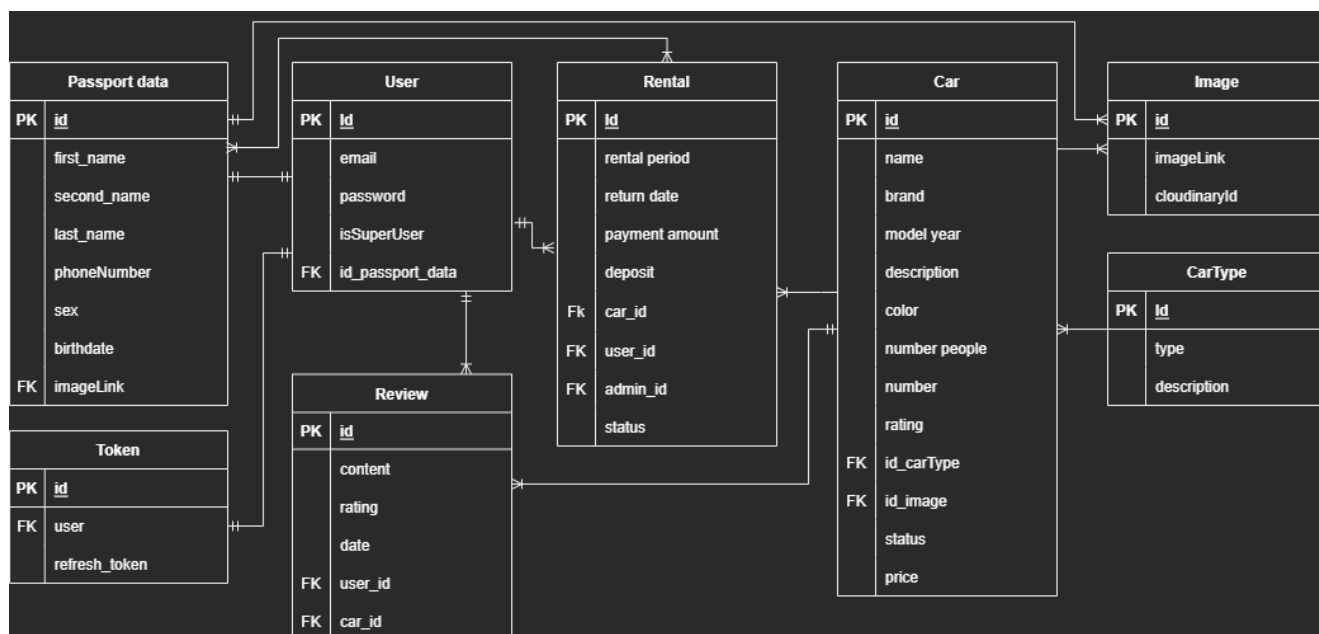


Рис. 2.7. Діаграма „сутність-зв’язок” (логічний рівень) системи
Внаслідок проектування до бази даних включено колекції:

- User
- Car
- Rental
- Image
- CarType

- Review
- Token
- PassportData

Для збереження зареєстрованих користувачів використовується колекція “User”. В поданій колекції зберігається основна інформація про користувача, а саме електрона пошта, хешований пароль, роль та посилання на паспортні дані. Структура таблиці наведена нижче:

Таблиця 2.1

Структура таблиці "User"

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id користувача
email	String	-	-	Електрона пошта користувача
password	String	-	-	Хешований пароль користувача
is_superuser	Boolean	-	-	Роль адміністратора
createdAt	Date	-	-	Дата створення
updatedAt	Date	-	-	Дата останнього оновлення
passportData	ObjectId	-	+	Посилання на паспортні дані

Для збереження автомобілів використовується колекція “Car”. В поданій колекції зберігається інформація про автомобіль. До них належать назва, модель, рік виготовлення, опис, колір, кількість місць автомобіля, рейтинг, фото, ціна оренди за день та автомобільний номер. Структура таблиці наведена нижче:

Таблиця 2.2

Структура таблиці "Car "

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id автомобіля
name	String	-	-	Назва атомобіля
brand	String	-	-	Бренд (Модель) автомобіля
modelYear	Int32	-	-	Рік виготовлення
description	String	-	-	Опис автомобіля

color	String	-	-	Колір автомобіля
numberPeople	Int32	-	-	Кількість місць автомобіля
number	String	-	-	Автомобільний номер
rating	Date	-	-	Рейтинг автомобіля
carImages	Array ObjectId	-	+	Масив посилання на фото автомобіля
carType	ObjectId	-	+	Посилання на тип автомобіля
status	Boolean	-	-	Статус оренди автомобіля (“В прокаті”, “На паркінгу”)
price	Number	-	-	Ціна оренди за один день

Для збереження оренд автомобілів використовується колекція “Rental”. В поданій колекції зберігається основна інформація про замовлення. До них належать період прокату, дата повернення, вартість прокату, завдаток, автомобіль, паспортні дані, користувач, дата створення, дата оновлення. Структура таблиці наведена нижче:

Таблиця 2.3

Структура таблиці " Rental"

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id замовлення
rentalPeriod	string	-	-	Період прокату
returnDate	string	-	-	Дата повернення автомобіля
paymentAmount	Double	-	-	Вартість прокату
deposit	Double	-	-	Завдаток прокату
user	ObjectId	-	+	Посилання на паспортні дані орендатора
car	ObjectId	-	+	Посилання на орендований автомобіль
admin	ObjectId	-	+	Посилання на користувача який оформив прокат

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				22
Змн.	Арк.	№ докум.	Підпис	Дата		

status	Boolean	-	-	Статус оренди(“В оренді”, “Закінчено”)
createdAt	Date	-	-	Дата створення
updatedAt	Date	-	-	Дата останнього оновлення

Для збереження інформації про картинки, які зберігаються у хмарному сховищі, використовується колекція “Image”. В поданій колекції зберігається інформація про картинки які додаються при створенні запису про автомобіль або паспортні дані. До них належать url картинки та id на хмарному сховищі. Структура таблиці наведена нижче:

Таблиця 2.4

Структура таблиці " Image "

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id картинки
imageLink	String	-	-	URL картинки
clouinaryId	String	-	-	Id картинки на хмарному сховищі

Для збереження категорій автомобілів використовується колекція “CarType”. В поданій колекції зберігається загальна інформація про тип авто. До них належать назва та опис. Структура таблиці наведена нижче:

Таблиця 2.5

Структура таблиці " CarType"

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id типу
type	String	-	-	Назва типу
description	String	-	-	Опис типу

Для збереження відгуків про автомобілі використовується колекція “Review”. В поданій колекції зберігається інформація про відгуки. До них належать текст відгуку, оцінка автомобіля, паспортні дані, автомобіль. Структура таблиці наведена нижче:

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижемотря О.В				23
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.6

Структура таблиці " Review"

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id відгуку
content	String	-	-	Текст відгуку
rating	Ins32	-	-	Оцінка автомобіля
user	ObjectId	-	+	Посилання на паспортні дані
car	ObjectId	-	+	Посилання на автомобіль
createdAt	Date	-	-	Дата створення
updatedAt	Date	-	-	Дата останнього оновлення

Для збереження токенів використовується колекція "Token". В поданій колекції зберігається refresh токен та посилання на користувача. До них належать зареєстрований користувач, refresh токен. Структура таблиці наведена нижче:

Таблиця 2.7

Структура таблиці "Token"

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id токена
user	ObjectId	-	+	Посилання на користувача
refreshToken	string	-	-	Refresh токен для поновлення access токена

Для збереження паспортних даних клієнтів використовується колекція "PassportData". В поданій колекції зберігається всі паспортні дані та фото. До них належать Структура таблиці наведена нижче:

Таблиця 2.8

Структура таблиці " PassportData"

Назва	Тип даних	ПК	ЗК	Опис поля
_id	ObjectId	+	-	Id паспортних даних
firstname	String	-	-	Ім'я

secondName	String	-	-	По батькові
lastname	String	-	-	Прізвище
phoneNumber	String	-	-	Номер телефону
sex	String	-	-	Стать
birthdate	Dete	-	-	Дата народження
imageLink	ObjectId	-	+	Посилання на фото документів

2.3 Розробка математичної моделі та алгоритмів обробки даних додатку прокату автомобілів

Для початку користувачу якій тільки зайшов на Web-сайт потрібно зареєструватися, для того щоб його данні було збережено в базу даних. Користувач заповнює форму реєстрації, у якій він заповнює поля електронної пошти, пароль та підтвердження паролю. Клієнтська частина перевірить валідність пошти та паролю, після відправки форми серверна частина перевірить унікальність електронної пошти та відправить відповідь. За наявності вже створеного облікового запису користувач може авторизуватись, сервер перевірить дані та відправить відповідь у вигляді токена.

Після авторизації користувач матиме змогу переглядати данні сортувати, фільтрувати, шукати. Для більшої взаємодії з Web-додатком потрібно мати певні права доступу, які можуть надавати адміністратори.

За наявності потрібних ролей користувач може переглядати користувачів. Також може взаємодіяти з записами наприклад видалення та редагування, для цих дій присутнє окрема колонка в таблиці. Під час редагування ми можемо додати додаткову інформацію, таку як ім'я, прізвище, по батькові та змінити вже наявні дані, або надати роль адміністратора. Після підтвердження форми редагування, дані у БД оновляться та за наявності додаткових даних створиться запис у колекції паспортних даних.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				25
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці автомобілів користувач може знайти наявний автомобіль за потрібними критеріями, також про будь-які автомобілі можна подивитися додаткову інформацію та відгуки. Для кожної колекції притаманний функціонал видалення та редагування. При редагування автомобіля користувач може видалити наявні фото, або додати нові та змінити текстові данні, після збереження змін серверна частина оновить данні, а фото збереже у хмарному сховищі та створить нові записи у колекції картинок.

Для додавання автомобілів можна заповнити форму, яка схожа на редагування. На цій сторінці можна вказати назву, рік виготовлення автомобіля, обрати категорію з колекції БД та ін. Після відправки сервер проведе валідацію даних, та при виникненні помилок повідомить про це клієнтську частину Web-сайту.

Категорії аналогічно з автомобілями можна переглядати, створювати, редагувати, видаляти. Плюсом також є статистика, сервер сортує та групує за певними критеріями та відсилає кількість автомобілів для кожної категорії.

Адміністратор має можливість оформити прокат автомобілю, перейшовши на сторінку потрібного авто та заповнити форму оренди, а саме вказавши людину зі списку паспортних даних та тривалість прокату. Замовлення можна видаляти за критерієм статусу, якщо оренда закінчена користувач немає змоги видалити цей запис, в іншому випадку ця операція буде успішною і автомобіль буде оновлено для наступних оренд. Також прокат можна завершувати натиснувши потрібну кнопку, в такому випадку клієнтська частина запропонує залишити відгук, після чого сервер змінить статус оренди та статус автомобіля, відгук збереже у відповідній колекції та оновить рейтинг орендованого автомобіля.

Паспортні дані за аналогом критерій, можна переглядати, додавати та видаляти фото, редагувати.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				26
Змн.	Арк.	№ докум.	Підпис	Дата		

Для створення резервної копії БД адміністратор має перейти на сторінку з інструментами та натиснути на відповідну кнопку по створенню копії. Далі запит надсилається на сервер, де він виконується, виконуючи відповідні команди для створення копії БД. Після чого користувач побачить оновлену статистику по колекціям.

Для відновлення з резервної копії потрібно натиснути відповідну команду, а сервер очистить БД та заповнить даними з файлу резервного копіювання, після чого аналогічно оновиться статистика.

Висновки до 2 розділу :

У результаті виконання даного розділу було проаналізовано інформаційні процеси, спроектовано структуру БД, описано приблизну логіку функцій для роботи з базою на клієнті та деяку логіку обробки, збереження та генерації даних для БД.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижевотря О.В				27
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОБРОБКИ ДАНИХ ПРОКАТУ АВТОМОБІЛІВ

3.1 Проектування інтерфейсу обробки даних

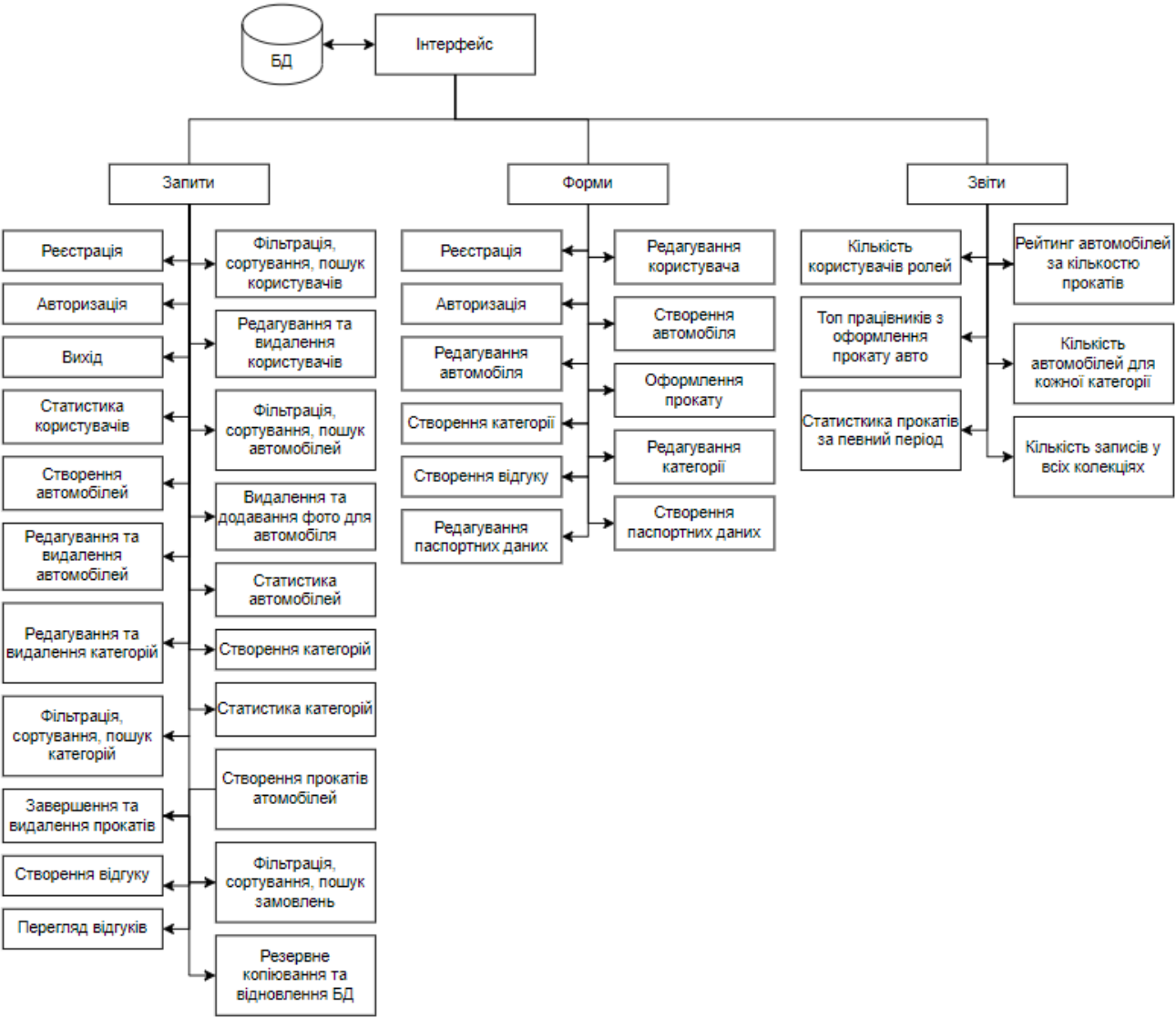


Рис. 3.1. Структура інтерфейсу системи управління прокату автомобілей

При створенні серверної частини додатку було використано Node.js та фрейворк express. Для всіх запитів було розроблено функції для обробки даних та взаємодії з БД. На структурі інтерфейсу системи (Рис. 3.1) було зображено всі можливі варіанти обробки даних. Далі будуть зображені форми та сторінки, що демонструють фінальний результат проектування інтерфейсу Web-сайту.

Для клієнтської частини було використано React, який надає змогу зручно

розробляти інтерфейси користувача. Також великою перевагою є простота та масштабованість. У React є велика спільнота, що надає більшу кількість різновид бібліотек та модулів.

Отож перейдемо до демонстрації роботи системи управління прокату автомобілів.

При першому завантаженні додатку користувачу відображається форма авторизації. При заповненні електронної пошти та паролю клієнтська частина перевіряє валідність даних, а після відправки запиту на під'єднання, відбувається перевірка чи існує даний користувач і чи співпадає пароль. Якщо на сервері виникає помилка, то клієнтська частина повідомить про це користувача.

Рис. 3.2. Форма автентифікації

Рис. 3.3. Помилка при автентифікації

Якщо ми ще не маємо облікового запису в системі, то ми повинні зареєструватись. Форма реєстрації приймає електронну пошту, пароль та підтвердження паролю. Всі поля є обов'язковими, а при заповненні буде

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				29
Змн.	Арк.	№ докум.	Підпис	Дата		

відбуватися валідація даних. Серверна частина буде перевіряти унікальність поштової адреси, та при успішній перевірці буде створено пару токенів.

The registration form is titled "Реєстрація". It contains three input fields: "Email", "Пароль" (Password), and "Підтвердити пароль" (Confirm password). Each field is preceded by an asterisk and its respective label. The "Пароль" and "Підтвердити пароль" fields have an eye icon to toggle visibility. Below the fields is a blue button labeled "Зареєструватися" (Register) and a link labeled "Увійти в обліковий запис" (Log in).

Рис. 3.4. Форма реєстрації

The registration form is shown with validation errors. The "Email" field contains "1234" and has a red border and a red "x" icon, with the error message "Неправильний формат" (Incorrect format) below it. The "Пароль" field contains "Пароль" and has a red border and a red "x" icon, with the error message "Будь ласка, введіть пароль" (Please enter password) below it. The "Підтвердити пароль" field contains "Підтвердити пароль" and has a red border and a red "x" icon, with the error message "Будь ласка, введіть пароль" (Please enter password) below it. The "Зареєструватися" button and the "Увійти в обліковий запис" link are still present.

Рис. 3.4. Валідація форми реєстрації

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				30
Змн.	Арк.	№ докум.	Підпис	Дата		

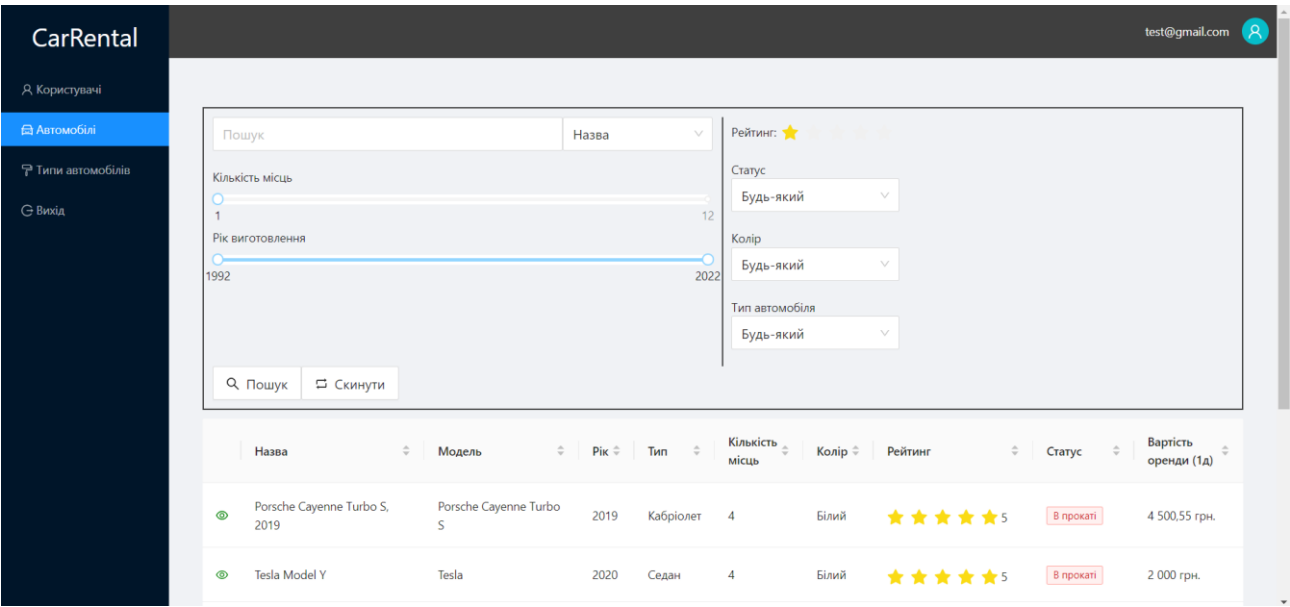


Рис 3.5. Початкове вікно авторизованого користувача

Після того як ми увійшли на сайт під новим обліковим записом, користувач має небагато функціоналу, але має змогу переглядати користувачів, автомобілі та типи автомобілів, також за потреби сортувати та фільтрувати записи. Щоб завершити роботу з додатком присутня опція виходу з облікового запису.

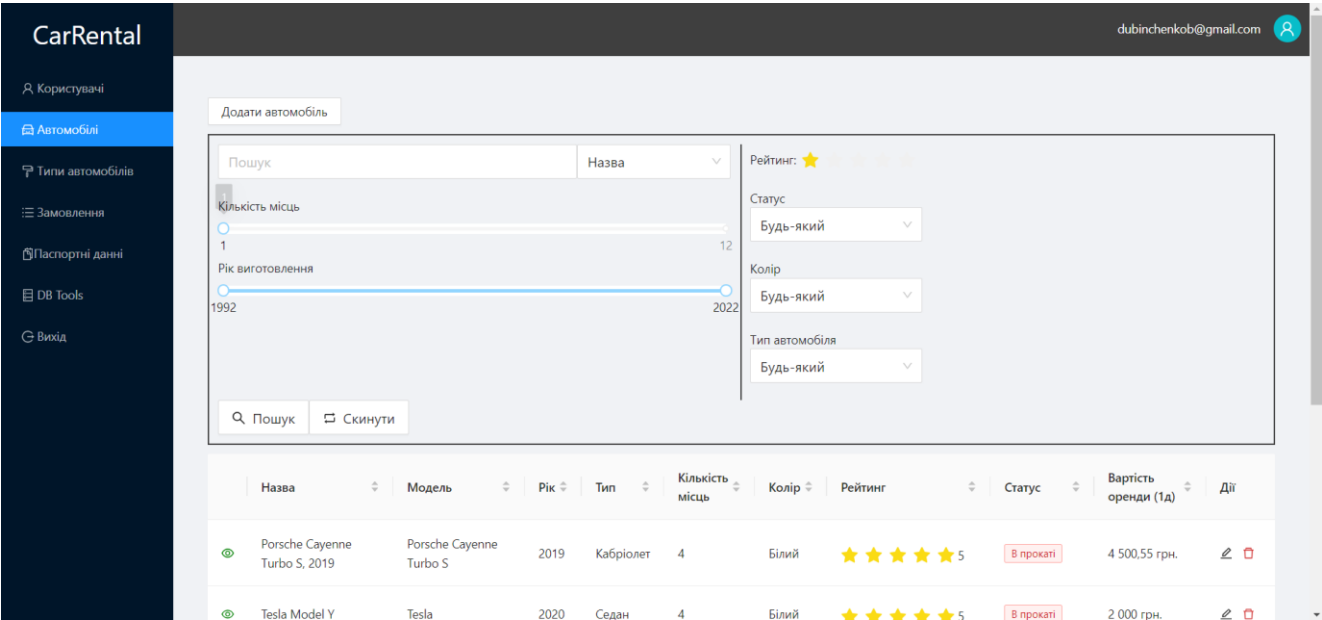


Рис. 3.6. Початкове вікно авторизованого адміністратора

На рисунку 3.6 зображено головну сторінку додатку, як бачимо користувача з роллю адміністратора має набагато більше додаткових функцій та

перегляду даних, а саме повний контроль над всіма даними в додатку та перегляд замовлень та паспортних даних, також присутній функціонал резервного копіювання та відновлення БД.

Далі буде зображено функціонал адміністратора, так як для звичайного користувача функціонал реалізований тільки для перегляду та вибірки даних.

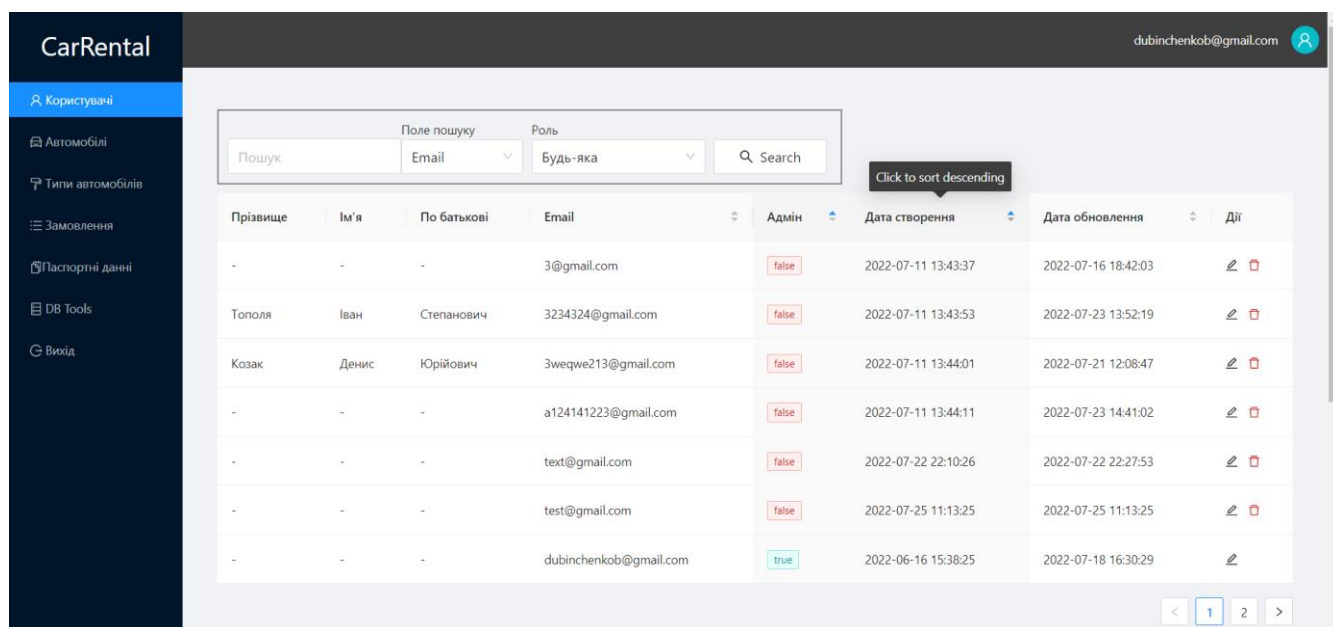


Рис. 3.7. Сторінка користувачів

Для всіх таблиць зі записами притаманний функціонал пошуку за полями, фільтрація за різними критеріями та сортування, великим перевагою є те, що всі види вибірки можна використовувати одночасно, за для зручності використання та знаходження потрібної інформації.

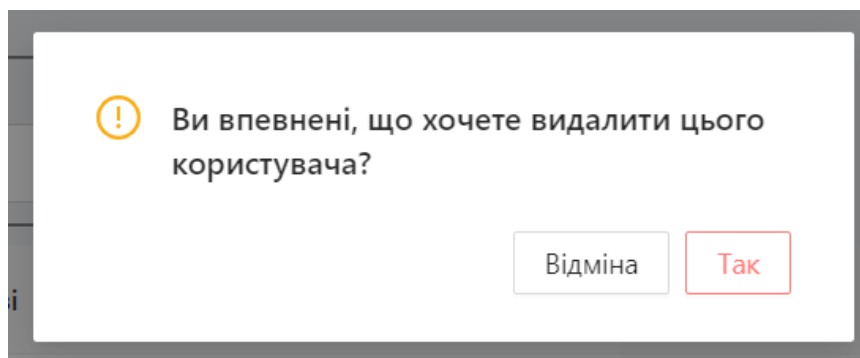


Рис. 3.8. Видалення користувача

За потреби адміністратор має змогу видалити іншого користувача, при натисканні на відповідну кнопку, відобразиться вікно з підтвердження операції. Після чого данні в таблиці оновляться.

Рис. 3.9. Редагування користувача

Статистика користувачів графічно зображує кількість облікових записів для кожної ролі та кількість оформлених зомовлень.

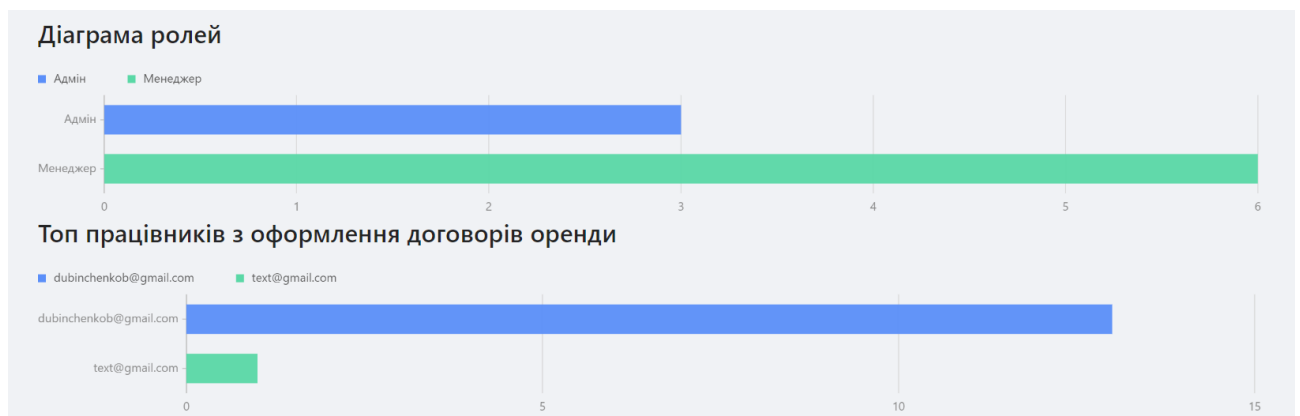


Рис. 3.10. Статистика користувачів

В цьому модальному вікні можна змінити або додати паспортні дані, також надати роль адміністратора. При збереженні сервер перевірить унікальність електронної пошти та збереже зміни.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				33
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувач

Автомобілі

Типи автомобілів

Замовлення

Паспортні дані

DB Tools

Вихід

Додати автомобіль

Пошук

Назва

Рейтинг: ★★★★★

Статус: Будь-який

Колір: Будь-який

Тип автомобіля: Будь-який

Кількість місць: 1

Рік виготовлення: 1992

Пошук

Скинути

Назва	Модель	Рік	Тип	Кількість місць	Колір	Рейтинг	Статус	Вартість оренди (1д)	Дії
Porsche Cayenne Turbo S, 2019	Porsche Cayenne Turbo S	2019	Кабріолет	4	Білий	★★★★★ 5	В прокаті	4 500,55 грн.	
Tesla Model Y	Tesla	2020	Седан	4	Білий	★★★★★ 5	В прокаті	2 000 грн.	
BMW 3 серії, 2021	BMW	2021	Седан	4	Синій	★★★★☆ 3.8	На паркінгу	3 000 грн.	

Рис. 3.11. Сторінка автомобілів

На цій сторінці відображені параметри вибірки, таблиці зі всіма записами та кнопки додавання, перегляду, редагування, видалення. У таблиці відображається загальна інформація, рейтинг та статус автомобіля.

Кількість місць

4

* Автомобільний номер

AT 3245 IB

Фото

Додати фото

Вибрати файли

Файл не вибран

* Тип автомобіля

Кабріолет

Оновити

Рис. 3.12. Сторінка редагування даних автомобіля

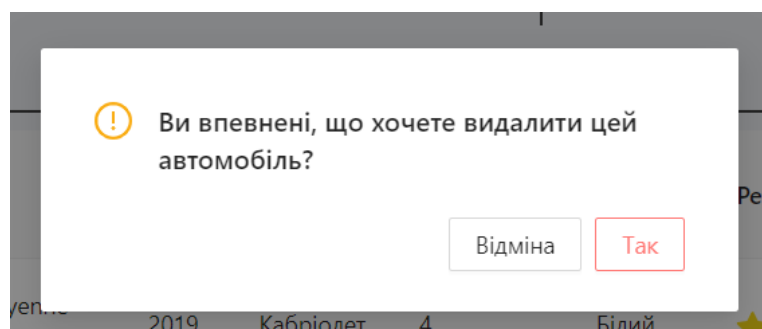


Рис. 3.13. Видалення автомобіля

Рис. 3.14. Додавання автомобіля

На сторінці створення автомобіля вказується різна інформація та додаються фото, при підтвердженні форми данні валідуються і зберігаються у БД. Після чого новий запис відобразиться у таблиці.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				35
Змн.	Арк.	№ докум.	Підпис	Дата		

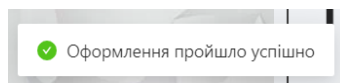


Рис. 3.17. Повідомлення про статус оформлення

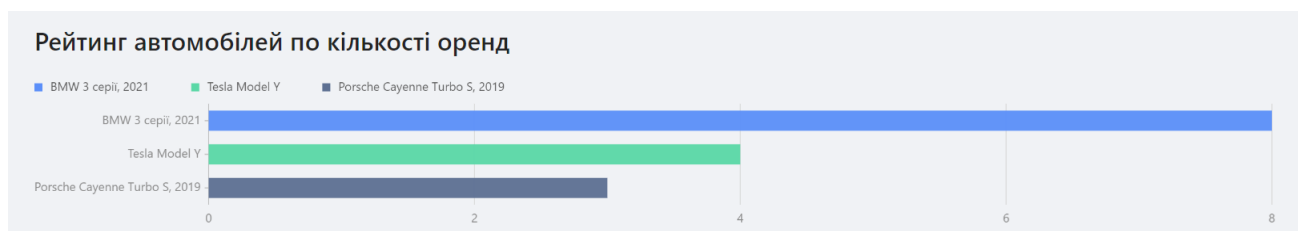


Рис. 3.18. Рейтинг автомобілів

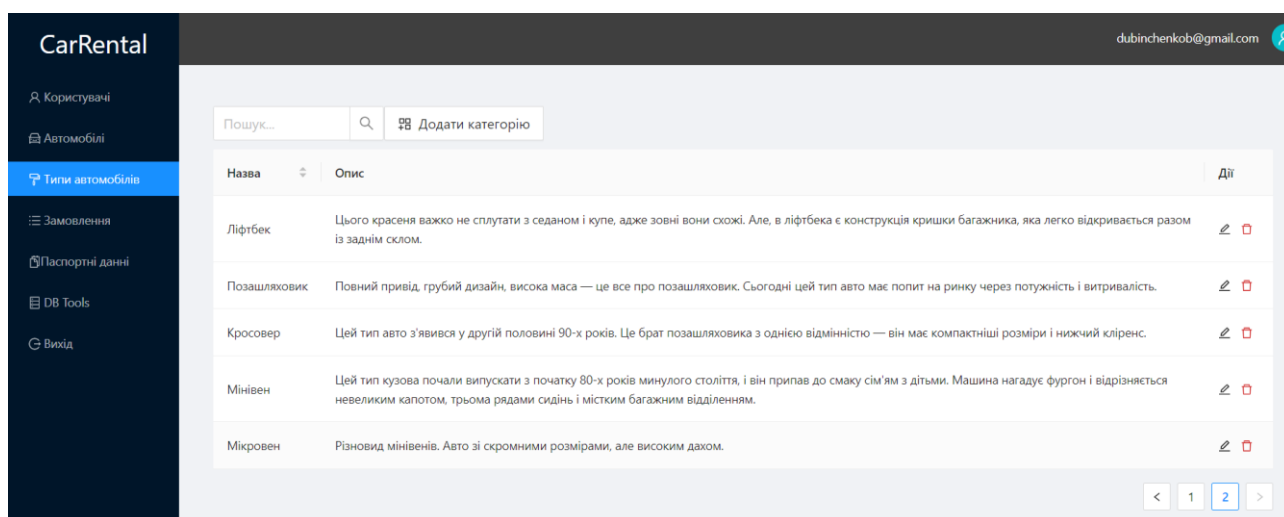


Рис. 3.19. Сторінка типів автомобілів

На даній сторінці можна переглянути всі наявні типи, присутній функціонал редагування, видалення та створення. В подальшому ці записи будуть відображатися у випадających списках при створенні автомобіля, або редагування.

Рис. 3.20. Форма створення категорії

Створення категорії

✖ Даний тип вже існує

* Назва категорії

Ліфтбек

Рис. 3.21. Помилка створення

При спробі адміністратором створити категорію, сервер перевірятиме унікальність назви і при виникненні помилки повідомить про це.

Редагування категорії

Назва категорії

Кросовер

Опис

Цей тип авто з'явився у другій половині 90-х років. Це брат позашляховика з однією відмінністю — він має компактніші розміри і нижчий кліренс.

Відміна Зберегти

Рис. 3.22. Вікно редагування категорії

Це вікно відображає актуальну інформацію повного типу, та надає змогу відредагувати та зберегти зміни.

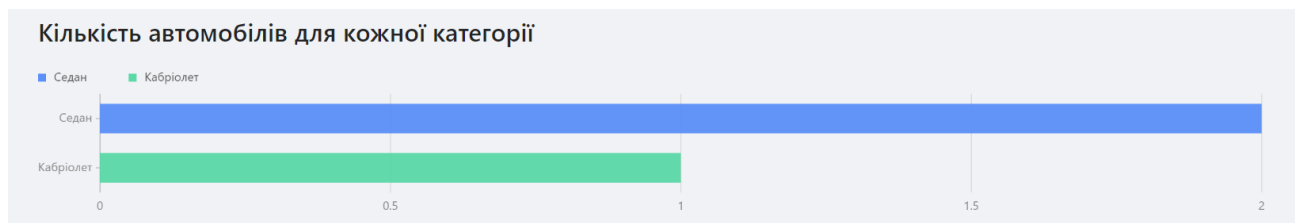


Рис. 3.23. Статистика для категорії

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				38
Змн.	Арк.	№ докум.	Підпис	Дата		

Автомобілі

Типи автомобілів

Замовлення

Паспортні дані

DB Tools

Вихід

Пошук

Назва автомобіля

Дата оренди

Start date → End date

Дата повернення

Start date → End date

Статус оренди

Будь-який

Пошук

Скинути

Назва автомобіля	Авто. номер	Днів оренди	Всього до сплати	Завдаток	Дата повернення	Орендар	Дії	Статус оренди
Porsche Cayenne Turbo S, 2019	AT 3245 IB	5	20 000 грн.	6 667 грн.	2022-07-25 12:13:31	-	✓	В оренді
Tesla Model Y	BM 2143 CI	4	8 000 грн.	2 667 грн.	2022-07-29 12:05:10	Василь Банди	✓	В оренді
BMW 3 серії, 2021	AM 1231 AM	5	15 000 грн.	5 000 грн.	2022-07-24 23:12:25	Анна Шевченко		Закінчено
Tesla Model Y	BM 2143 CI	2	4 000 грн.	1 334 грн.	2022-07-20 16:11:33	Анна Шевченко		Закінчено
Porsche Cayenne Turbo S, 2019	AT 3245 IB	2	8 000 грн.	2 667 грн.	2022-07-21 22:45:33	Іван Тополяк		Закінчено

Рис. 3.24. Сторінка замовленнями

Ви впевнені, що хочете видалити це замовлення?

Відміна

Так

Рис. 3.25. Видалення замовлень

Завершення оренди

Відгук

Загальні враження

★ ★ ★ ★ ★

Відміна

Відправити

Рис. 3.26. Завершення орпенди

Для оренд які не завершені, доступна функція видалення, після видалення статус автомобіля зміниться. Також є можливість завершити оренду та залишити відгук за бажанням, після чого відгук збережеться для відповідного автомобіля і зміниться статус замовлення.

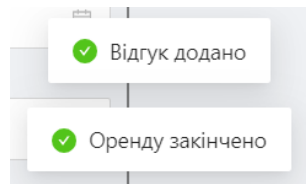


Рис. 3.26. Повідомлення про виконання завершення прокату

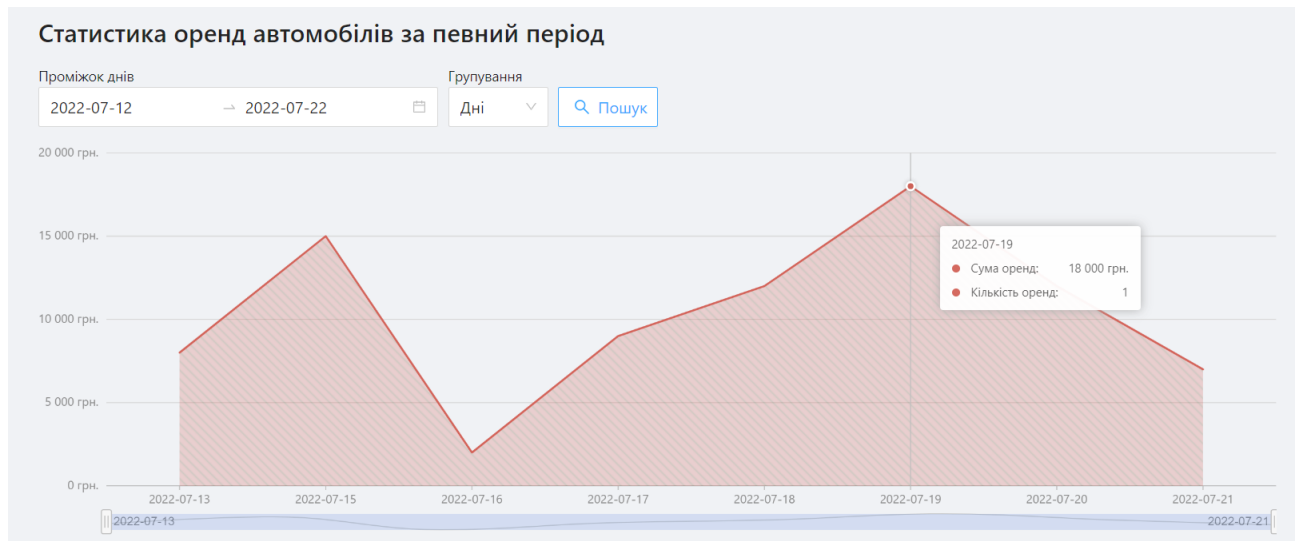


Рис. 3.27. Статистика прокатів

Даний графік відображає загальну суму та кількість оренд. За потреби данні можна групувати по днях, місяцях та роках, також для зручності є поле для проміжку часу.

CarRental dubinchenkob@gmail.com

Додати користувача

Пошук... Прізвище Стать Дата народження

Пошук... Скінути

Прізвище	Ім'я	По батькові	Номер телефону	Стать	Дата народження	Фото паспорту	Дії
Тополь	Іван	Степанович	+380685632743	Чоловіча	1980-08-14		
Козак	Денис	Юрійович	-	-	-		
Шевченко	Анна	Володимирівна	+380684378654	Жіноча	1999-05-19		
Бандера	Василь	Михайлович	+380685456780	Чоловік	2003-06-11		

Рис. 3.28. Сторінка паспортних даних

Редагування паспортних даних

* Ім'я
 ✓

* Прізвище
 ✓

* По батькові
 ✓

* Номер телефону
 ✓

* Стать
 ▼

Дата народження
 📅

Фото

Додати фото

Рис. 3.29. Форма редагування паспортних даних

CarRental

- Користувачі
- Автомобілі
- Типи автомобілів
- Замовлення
- Паспортні дані
- DB Tools**
- Вихід

dubinchekob@gmail.com

Резервне копіювання та відновлення бази даних

Інформація про колекції

Оренда	Користувачі	Картинки	Типи атомобілів	Відгуки
Кількість записів: 15	Кількість записів: 9	Кількість записів: 20	Кількість записів: 13	Кількість записів: 5
Паспортні дані	Автомобілі			
Кількість записів: 5	Кількість записів: 3			

Рис. 3.30. Сторінка інструментів БД

Інформація про колекції

Оренда	Користувачі	Картинки	Типи атомобілів	Відгуки
Кількість записів: 15	Кількість записів: 9	Кількість записів: 20	Кількість записів: 13	Кількість записів: 5
Паспортні дані	Автомобілі			
Кількість записів: 5	Кількість записів: 3			

Рис. 3.31. Статистика по колекціям

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижемотря О.В				41
Змн.	Арк.	№ докум.	Підпис	Дата		

На цій сторінці відображається статистика про колекції, а саме кількість записів для кожної з них. Також є кнопки для резервного копіювання та відновлення БД.

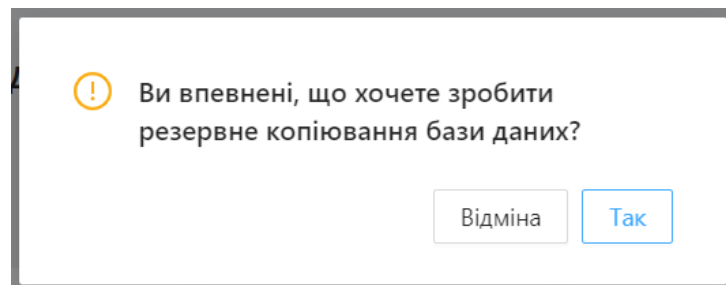


Рис. 3.32. Вікно підтвердження резервного копіювання БД

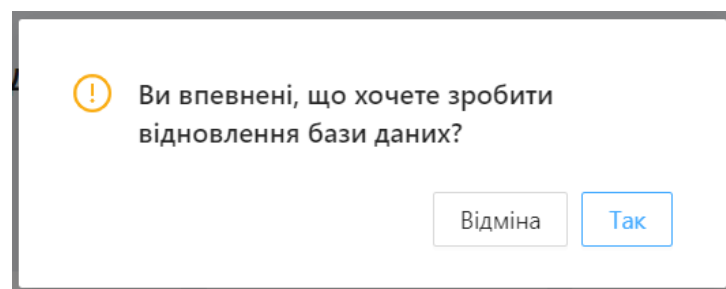


Рис. 3.33. Вікно підтвердження відновлення БД

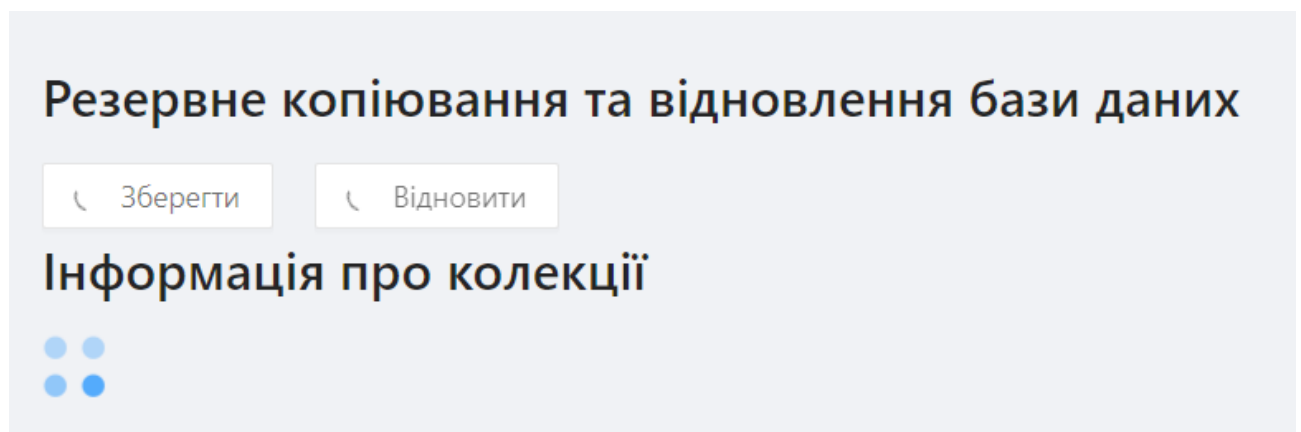


Рис. 3.34. Виконання збереження або оновлення

Після вибору дій над БД, буде здійснено перезавантаження та відображення нових даних.

3.2. Реалізація операцій обробки даних в БД

На старті Web-сайту потрібно пройти автентифікацію або реєстрацію, тому наведемо та розглянемо приклади роботи цих функцій.

Роут та валідації полів автентифікації:

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

router.post('/login',
[
  body('email')
    .isEmail()
    .normalizeEmail()
    .withMessage("Невірний формат email")
    .custom(async (value) => {
      const user = await UserModel.findOne({email:value});
      if (!user) {
        throw new Error('Неправильний email або пароль');
      }
    }),
  body('password')
    .trim()
    .custom(async (value, {req}) => {
      const user = await UserModel.findOne({email: req.body.email});
      const isMatch = await bcrypt.compare(value, user.password);
      if (!isMatch) {
        throw new Error('Неправильний email або пароль');
      }
    }),
],
validationRes,
userControllers.postLogin
);

```

Для більшості запитів до серверу притаманні валідації полів та параметрів. Будь-які помилки перевіряються проміжним програмним забезпеченням та формують відповідь.

Після виклику роута виконується валідація вхідних полів. Для електронної пошти перевіряється правильність структури та наявності цих даних у БД. Пароль перевіряється з записом користувача та при виникненні помилок сервер відправляє відповідь з ними.

Лістинг сервісу автентифікації:

```

const login = async (email, password) => {
  const user = await UserModel.findOne({email});
  const tokens = tokenService.generateTokens(user._id);
  await tokenService.saveToken(user._id, tokens.refreshToken);

  return {...tokens, user};
}

```

Дана функція приймає email та password. Після виклику вона знаходить в БД користувача по електронній пошті, далі генерується нова пара токенів, в подальшому refresh токен зберігаються в окремій колекції, а Access токен записується в httpOnly cookie.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Функція реєстрації схожа на автентифікацію, але відрізняється тим що пароль хешується, а користувач створюється та зберігається у певній колекції.

Лістинг редагування користувача:

```
const editUser = async(id, userData) => {
  const user = await UserModel.findById(id);
  const update = ['email', 'is_superuser'];
  update.forEach((update) => user[update] = userData[update]);
  await user.save();
}
```

В даному методі знаходиться користувач по id, дані якого згодом оновлюємо та зберігаємо.

Розглянемо далі метод створення автомобіля

```
const createCar = async (carData, carImages) => {
  const uploadedResponse = await
cloudinaryService.uploadToCloudinary(carImages,
process.env.UPLOAD_PRESET_FOR_CAR_IMAGES);
  const imagesId = await imageService.saveImagesToDB(uploadedResponse);

  const newCar = new CarModel({
    name: carData.name,
    brand: carData.brand,
    modelYear: carData.modelYear,
    description: carData.description,
    price: carData.price,
    color: carData.color,
    numberPeople: carData.numberPeople,
    number: carData.number,
    carImages: imagesId,

    carType: carData.carType
  })
  await newCar.save();

  return newCar;
}
```

Даний метод приймає дані про автомобіль та його фото. Для початку картинки завантажуються на хмарне сховище і повертають данні про них. Ці дані передаються в метод збереження картинок, що в подальшому зберігаються у відповідній колекції. В кінці створюється новий запис з даними про автомобіль, які зберігаються в БД.

Для оновлення запису реалізований метод за аналогом користувачів, ми отримуємо id, знаходимо потрібний запис та оновлюємо дані.

Розглянемо метод оновлення рейтингу автомобіля:

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const updateCarRating = async (id) => {
  const avgRating = await ReviewModel.aggregate([
    {
      $match: {
        'car': ObjectId(id)
      }
    },
    {
      $group: {
        _id: null,
        avgReviews: {
          $avg: "$rating"
        }
      }
    }
  ]);
  const car = await CarModel.findOne({ _id: id })
  car.rating = avgRating[0].avgReviews.toFixed(1)
  await car.save();
}

```

Цей асинхронний метод приймає id автомобіля. Наступний крок запит до БД, з пошуком по Id та групування з обрахунком середнього значення рейтингу, після отримання даних зберігаємо їх у відповідний запис автомобіля та зберігаємо його.

Далі розглянемо отримання замовлень прокату автомобілів з різними параметрами пошуку, фільтрації, пагінації, сортування.

```

const { limit, skip, sort, filters } = await queryParser(req.query,
RentalModel);

const query = [{
  $lookup: {
    from: 'users',
    localField: 'admin',
    foreignField: '_id',
    as: 'admin'
  }
}, {
  $lookup: {
    from: 'cars',
    localField: 'car',
    foreignField: '_id',
    as: 'car'
  }
}, {
  $lookup: {
    from: 'passportdatas',
    localField: 'user',
    foreignField: '_id',
    as: 'user'
  }
},

```

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    { $sort: {'status': 1} },
    {
      $match: filters
    }
  ]

let count = await RentalModel
  .aggregate(query.concat([
    {
      $count: 'countDocuments'
    }
  ]))
count = count[0]?.countDocuments;
const totalPages = Math.ceil(count / limit);

if (sort) {
  query.push({ $sort: sort });
}
query.push({ $skip: skip });
query.push({ $limit: limit });

const rentals = await RentalModel
  .aggregate(query)

```

За допомогою унікального методу для query парсингу, отримуємо параметри для вибірки записів в колекції прокатів. Далі прописано агрегації для об'єднання дочірних записів. Після цього обраховується кількість документів для відображення загальної кількості сторінок. За наявності параметрів сортування вони додаються до загального запиту до БД. В результаті ми отримуємо великий запит який передаємо у агрегацію моделі замовлень.

Ще для прикладу розглянемо видалення замовлень

```

exports.deleteRental = async (req, res, next) => {
  try {
    const rental = await RentalModel.findByIdAndDelete(req.params.id);
    const rentedCar = await CarModel.findById(rental.car);

    rentedCar.status = false;
    await rentedCar.save();

    res.status(201).json({
      message: "Rental successfully deleted",
    });
  } catch (error) {
    next(error);
  }
}

```

Метод приймає тіло запиту у якому зберігаються параметри, а саме id замовлення. Видаляємо замовлення та отримуємо данні про його. З цих даних ми беремо id автомобіля та знаходимо його в колекції. В цьому записі змінюється статус на false та зберігається в БД.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

```
function backupMongoDB() {
  return new Promise((resolve, reject) => {
    const child = spawn(COMMAND_MONGODUMP, [
      '--uri=${process.env.MONGO_URL}',
      '--archive=${ARCHIVE_PATH}',
      '--gzip'
    ])
    child.on('close', (code, signal) => {
      if (code) console.log('Process exit with code: ', code);
      else if (signal) console.log('Process killed with signal: ', signal)
      else {
        console.log('Backup is successful');
        resolve();
      }
    })
  })
}
```

Для резервного копіювання використовується MongoDB Database Tools, за допомогою пакету `child_process`, викликається команда у якій вказується посилання на БД, шлях збереження файлу та формат. Після роботи цього методу створиться файл з резервною копією.

Дані методи є базовими для системи управління прокату і інші необхідні методи будуть написані зі схожою поведінкою, тільки в іншому контексті.

3.3. Організація звітності системи

У даному підрозділі буде розглянуто лише ті звітності, що були розроблені спеціально для додатку, а саме статистика для користувачів, автомобілів, типів. А для замовлень реалізовано статистику з параметрам, такі як вибір проміжку дат та тип групування.

Отримання статистики кількості користувачів по ролям

```
exports.countUserByRole = async (req, res, next) => {
  try {
    let countUser = await UserModel.aggregate([
      {
        $group: {
          _id: "$is_superuser",
          count: {
            $count: {}
          }
        }
      }
    ])

    countUser = countUser.map((data) => {
      return {...data, _id: data._id ? 'Адмін' : 'Менеджер'}
    })
  }
}
```

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    res.status(200).json({
      countUser: countUser
    })
  } catch (e) {
    next(e);
  }
}

```

Цей асинхронний метод групує користувачів по ролям, що в подальшому формує відповідь з назвами та кількості ролей.

Переглянемо запит для отримання рейтингу працівників з формування договорів оренди:

```

let topUsers = await RentalModel.aggregate([
  $group: {
    _id: "$admin",
    count: { $count: {} }
  }, {
    $sort: { count: -1 }
  }, {
    $lookup: {
      from: 'users',
      localField: '_id',
      foreignField: '_id',
      as: '_id'
    }
  },
])

topUsers.map((data) => {
  return data._id = data._id[0].email
})

```

Для колекції замовлень створюється агрегація у якій записи групуються по полю admin та результати сортуються. До отриманих записів додається інформація про користувача з дочірньої колекції. Відповідь оформлюється у зручний формат.

Отримання рейтингу автомобілів по кількості оренд схоже до рейтингу працівників, але відміна в тому що групування здійснюється по полю автомобілі.

Отримання статистики по історії прокату автомобілів.

```

const { filters } = await queryParser(req.query, RentalModel);

const period = req.query.period || 'day';
format = { 'day': '%Y-%m-%d', 'month': '%Y-%m', 'year': '%Y' };

const rentalsStatistics = await RentalModel
  .aggregate([
    {
      $match: {...filters, 'status': true}
    },
  ],

```



```

    {
      $project: {
        "paymentAmount": "$paymentAmount",
        "createdAt": {
          $dateToString: {
            "format": format[period],
            "date": "$createdAt"
          }
        }
      }
    },
    {
      $group: {
        _id: "$createdAt",
        amount: { $sum: "$paymentAmount" },
        count: { $count: {} },
      }
    },
    {
      $sort: {'_id': 1}
    }
  ]
})

```

З методу query парсингу отримуємо параметри для фільтрації, а саме початкову дату та кінцеву дату, щоб вказати період часу. Отримуємо з запити данні для групування. Далі створюємо агрегації для вибірки, групування та обрахунку. В результаті отримуємо відсортовану статистику.

Висновки до третього розділу :

В ході виконання даного розділу відбулось написання серверного та клієнтського додатку, що надає змогу користувачу зручно взаємодіяти з БД. Було спроектовано та реалізовано інтерфейси системи управління прокату автомобілів. Також були розроблені форми для різних задач зі створення, редагування записів. Додано різноманітні статистичні діаграми для відображення інформації про користувачів, автомобілів, типів та прокату. Крім цього, було створено можливість резервного копіювання БД, та відновлення його з файлу.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				49
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

4.1. Розробка заходів захисту інформації в БД

Отже, розглянемо категорії користувачів системи управління прокату автомобілів: головний адміністратор, звичайні користувачі. Найбільші права доступу до даних має головний адміністратор, що необхідно для організації роботи. Він може переглядати будь-які дані даної системи. Також головний адміністратор повинен мати змогу робити резервне копіювання та відновлювати базу даних. Тому доцільно створити окрему роль "Супер користувач" для забезпечення надання доступу до цих даних. Більш детально перелік об'єктів БД, доступ до яких надано ролі "Супер користувач" наведено в табл. 4.1. На перетині рядків і стовпців зазначено дії, які може виконувати користувач даного типу ролі: 1 – перегляд даних, 2 – редагування, 3 – видалення, 4 – повний доступ.

Таблиця 4.1

	Car	CarType	Image	Passport Data	Rental	Categories	Review	Token	User
Admin	4	4	4	4	4	4	4	4	4

Окрему групу користувачів системи складають авторизовані користувачі та гості Web-сайту. Авторизовані користувачі мають змогу переглядати автомобілі, категорії та користувачів. Гості в свою чергу можуть тільки автентифікуватися або зареєструватися. Детальніше ролевий доступ зображений в таблиці 4.2.

Таблиця 4.2

	Car	CarType	Image	Passport Data	Rental	Categories	Review	Token	User
Authorized user	1	1	0	0	0	0	0	0	1
Guestes	0	0	0	0	0	0	0	0	0

Для контролю доступу використовуються проміжне програмне забезпечення, лістинг наведено нижче

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```
const auth = async (req, res, next) => {
  try {
    const accessToken = req.header('Authorization').replace("Bearer ", "");
    const decoded = tokenService.validateAccessToken(accessToken);
    const user = await UserModel.findOne({_id: decoded._id});

    if (!user) {
      return next(new CustomError.UnauthorizedError());
    }
    req.user = user;

    next();
  } catch (error) {
    return next(new CustomError.UnauthorizedError());
  }
}
```

Цей метод перевіряє користувача на автентифікацію таким чином, що з header отримується access token, перевіряється на валідність. Далі по отриманим даним знаходимо користувача, якщо він відсутній, сервер поверне помилку “Не авторизований”, в іншому випадку з’явиться доступ до потрібних запитів.

```
module.exports = async function (req, res, next) {
  try {
    const accessToken = req.header('Authorization').replace("Bearer ", "");
    const decoded = tokenService.validateAccessToken(accessToken);
    const user = await UserModel.findOne({_id: decoded._id});

    if (!user.is_superuser) {
      next(new CustomError.AccessDeniedError());
    }

    next();
  } catch (errors) {
    next(new CustomError.BadRequestError('Помилка валідації', errors.array()));
  }
}
```

Також одним з проміжним програмним забезпеченням є перевірка доступу користувача. Цей метод є схожим до автентифікації, але перевірка відбувається за полем ролі.

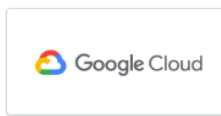
4.2. Налаштування параметрів роботи MONGODB

Для початку налаштування параметрів роботи з MongoDB, потрібно створити новий кластер. При створення обрано хмарний провайдер Amazon Web Services та регіон Франкфурт для швидкої взаємодії з БД.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В				51
Змн.	Арк.	№ докум.	Підпис	Дата		

Cloud Provider & Region

AWS, N. Virginia (us-east-1) ▾



Multi-Cloud, Multi-Region & Workload Isolation (M10+ clusters)

Distribute data across clouds or regions for improved availability and local read performance, or introduce read-only and analytics nodes. [Learn more](#)



Рис. 4.1. Вибір хмарного провайдера

★ Recommended region ⓘ Carbon emission data unavailable ⓘ

NORTH AMERICA

N. Virginia (us-east-1) ★

Ohio (us-east-2) ★

N. California (us-west-1)

Oregon (us-west-2) ★

Montreal (ca-central-1) ★

SOUTH AMERICA

EUROPE

Stockholm (eu-north-1) ★

Ireland (eu-west-1) ★

London (eu-west-2) ★

Paris (eu-west-3) ★

Frankfurt (eu-central-1)

Milan (eu-south-1) ★

AUSTRALIA

Sydney (ap-southeast-2) ★

ASIA

Hong Kong (ap-east-1) ★

Jakarta (ap-southeast-3) ★

Tokyo (ap-northeast-1) ★

Seoul (ap-northeast-2) ★

Рис. 4.2. Вибір регіону

```
const connectionToDB = async () => {
  try {
    await mongoose.connect(process.env.MONGO_URL, {
      useNewUrlParser: true,
      useUnifiedTopology: true
    });
  } catch (e) {
    new Error('Помилка підключення до БД')
  }
}
```

Далі створено метод для підключення БД до серверної частити. В файлу конфігурації вказано url для підключення.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмоторя О.В.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до четвертого розділу :

У ході виконання даного розділу проведено налаштування прав доступу користувача до бази, а саме за допомогою проміжного програмного забезпечення контролю ролей та автентифікації. Також у ході виконання цього розділу було показано, як саме відбулось створення та підключення додатку до БД.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				53
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час написання даної курсової роботи було отримано навички роботи з віддаленими базами даних, створення резервних копій та відновлення БД.

В першому розділі курсової роботи, було проаналізовано три СУБД, визначено переваги кожної з СУБД та обрано одну для реалізації додатку. Також було визначено головні функції програми та приблизну логіку доступу до даних.

В другому розділі було проаналізовано інформаційні процеси, було спроектовано структуру БД, описано приблизну логіку функцій для роботи з базою, розписано деяку логіку обробки, збереження та створення даних для БД. Також було описано як буде відбуватись побудова статистичної інформації про різні колекції.

В третьому розділі відбулось написання повного коду для взаємодії користувача з сервером, а сервера з БД та спроектовано інтерфейси управління системою прокату. Крім цього, було створено можливість зберігати поточний стан бази, та відновлювати його з резервної копії. Також було написано та описано логіку та код виконання звітності системи.

В четвертому розділі проведено налаштування прав доступу користувача до бази, а саме контроль за допомогою ролей. Також у ході виконання цього розділу було показано, як саме відбулось створення, налаштування віддаленої БД та підключення додатку до його.

В результаті виконання курсової роботи отримано систему управління прокату автомобілів, яка повністю відповідає запланованому функціоналу, умовам та вимогам, поставленим на початку проектування.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				54
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. NodeJS - The Complete Guide (MVC, REST APIs, GraphQL, Deno)
[Електронний ресурс] – 2022. Режим доступу:
<https://www.udemy.com/course/nodejs-the-complete-guide/>
2. Express Documentation [Електронний ресурс] - 2020. Режим доступу:
<https://expressjs.com/ru/guide/routing.html>
3. Документація Node.js. [Електронний ресурс] – 2021. Режим доступу:
<https://metanit.com/web/nodejs/1.1.php>
4. Node JS фундаментальний курс от А до Я. Node.js Теорія і практика
[Електронний ресурс] – 2019. Режим доступу:
<https://www.youtube.com/watch?v=243pQXC5Ebs>
5. MongoDB Documentation [Електронний ресурс] - 2020. Режим доступу:
<https://www.mongodb.com/docs/>
6. Mongoose Documentation [Електронний ресурс] - 2018. Режим доступу:
<https://mongoosejs.com/docs/>
7. Cloudinary Documentation [Електронний ресурс] - 2015. Режим доступу:
<https://cloudinary.com/documentation/>
8. Документація по React [Електронний ресурс] – 2020. Режим доступу:
<https://ru.reactjs.org/docs/getting-started.html>
9. React JS фундаментальний курс от А до Я [Електронний ресурс] – 2021.
Режим доступу: <https://www.youtube.com/watch?v=GNrdg3PzpJQ&t=5795s>
- 10.Redux Toolkit Documentation [Електронний ресурс] – 2021. Режим доступу:
<https://redux-toolkit.js.org/introduction/getting-started>
- 11.Ant Design Documentation [Електронний ресурс] – 2021. Режим доступу:
<https://ant.design/docs/react/introduce>
- 12.Stackoverflow [Електронний ресурс] – 2021. Режим доступу:
<https://stackoverflow.com/>

		Дубинченко Б.М.			ДУ «Житомирська політехніка». 22.121.07.000 - ПЗ	Арк.
		Чижемотря О.В				55
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Технічне завдання

1. Загальне положення

1.1 Найменування програмного засобу

Повне найменування програмної системи: “База даних управління системою прокату автомобілів”. Коротка назва програмної системи - "система прокату автомобілів"

1.2 Призначення розробки та область застосування

Програмна система "База даних управління системою прокату автомобілів" призначена для збору, обробки та відображення даних про замовлень, що є в системі. В програмі передбачено використання механізму пошуку по параметрам доступних записів. За результатами обробки даних програма має формувати звітність за певний період.

1.3.Мета

Програмна система "База даних управління системою прокату автомобілів" дозволить підвищити ефективність, повноту та швидкість роботи працівників. Також полегшить ввід даних, їх пошук та сортування.

1.4. Найменування розробника та замовника.

Розробник даного продукту - студент групи ІПЗ-20-4 Дубинченко Богдан (надалі “розробник”).

Замовник програмного продукту – кафедра інженерії програмного забезпечення Житомирського державного університету “Житомирська політехніка”, в межах виконання курсової з дисципліни «Бази даних» (надалі замовник).

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				57
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Підстава для розробки

2.1. Документ на підставі якого ведеться розробка

Робота ведеться на підставі навчального плану за напрямом 6.050201 «Інженерія програмного забезпечення» наказу про закріплення тем курсових робіт за студентами.

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик.

3.1.1. Загальні вимоги

Програмна система має забезпечувати:

- можливість дистанційної роботи з робочих станцій локальної та глобальної мережі підприємства;
- інтерфейс користувача, що не залежить від операційної системи;
- постійний доступ користувачів до БД;
- оптимальне збереження даних (за обсягом та структурою);
- аутентифікацію користувачів та захист інформації від несанкціонованого доступу;
- надійне збереження даних та можливість відновлення даних у випадку непередбачуваних збоїв системи;
- можливість модернізації системи через зміну функціональних потреб користувача або модернізацію обладнання;

3.1.1 Склад виконуваних функцій

Розробити базу даних управління системою прокату автомобілів, що підтримує виконання наступних транзакцій:

1. Реєстрацію користувачів;

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижевотря О.В				58
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Додання автомобілів, типів, замовлень, паспортних даних
3. Оформлення прокату автомобіля;
4. Зміна даних про автомобілі, категорії, паспортні дані.
5. Пошук та відбір замовлень по мультикритеріями (назва автомобіля, автомобільного номеру, дати оренди, дата повернення, статус оренди)
6. Формування звітності щодо :
 - Кількості користувачів по ролям
 - Рейтинг працівників з оформлення договорів оренди
 - Рейтинг автомобілів по кількості оренд
 - Кількість автомобілів для кожної категорії
 - Статистика оренд автомобілів за певний період

3.1.2. Організація вхідних і вихідних даних

Вхідними даними є дані про автомобілі, типи автомобілів, замовлення, відгуки, користувачі, паспортні дані.

Організація вхідних і вихідних даних повинна відповідати інформаційній структурі виконуваних операцій, вхідним та вихідним паперовим документами. Введення оперативних даних повинно виконуватися з використанням діалогових екранних форм, побудованих на основі візуальних компонентів. Введення даних виконується на основі затверджених форм документів: форма на додавання автомобіля, форма на додання категорії, форма з оформлення прокату, форма на додавання відгуку. Будь який звіт повинен мати своє представлення.

3.1.3. Часові характеристики і розмір пам'яті, необхідної для роботи програми.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижевотря О.В				59
Змн.	Арк.	№ докум.	Підпис	Дата		

Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с. Час виконання команд меню не більше 1 с.

Відображення масивів даних за запитами не більше 3 хвилин.

Доступність БД – 90% цілодобово.

Операції з'єднання з БД не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи програми не менше 1Гб.

Дисковий простір, необхідний для збереження програми і файлів даних не більше 300 Мбат для робочої станції та 20 Гбайт.

Інсталяційний пакет програми, що містить у складі БД не повинні перевищувати 100 Мбайт.

3.2. Вимоги до надійності.

3.2.1. Вимоги до надійного функціонування

Програма повинна нормально функціонувати при безперебійній роботі ПК.

Доступність

БД 90% при одночасному доступі 30 користувачів.

При апаратних збоях, відновлення нормальної роботи програми повинне виконуватися після:

- а) апаратні збої сервера - перезавантаження ОС сервера, запуск сервера БД (запуску резервного сервера, використання технологій RAID для збереження даних);

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				60
Змн.	Арк.	№ докум.	Підпис	Дата		

- b) апаратні збої робочої станції – перезавантаження ОС ПК запуск виконуваного файлу програми.
- c) БД повинна повертатись в найближчий несуперечний стан – передбачити точки відновлення.

При збоях програмного забезпечення:

- a) система повинна забезпечувати можливість відновлення даних та фіксацію і «відкат» транзакцій.
- b) в системі має бути реалізована коректна обробка виняткових ситуацій.

3.2.2.Контроль вхідної і вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора:

- Перевірка відповідності даних доменам інформаційних атрибутів;
- Використанням механізму авто заповнення та вибору за переліком для зв'язаних даних;
- Захист від помилок оператора (залипань, випадкових символів тощо).

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування.

Для вихідної інформації передбачити:

- відсутність блокування даних через багато користувацький доступ;
- постійне оновлення даних у відображених на екрані звітних формах.
- повідомлення про використання режиму монопольного доступу супервайзерами БД

3.2.3. Час відновлення після відмови.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезавантаження операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.3. Умови експлуатації і збереження

Програма використовується у багато користувачькому середовищі. Регламенті операції проводити за графіком:

- створення резервних копій даних: 1 раз на тиждень;
- збереження резервної копії журналу транзакцій – 1 раз на тиждень;
- архівування даних – 1 раз на місяць;
- обслуговуючі функції (прибирання “сміття”, де фрагментація тощо) - щоденне;
- оновлення системи ідентифікації та аутентифікації користувачів – 1 раз на місяць та за потребою.

Збереження архівних копій - на визначених носіях. Збереження резервних копій – на визначеному диску без перезапису. Знищення копій за регламентом.

3.4. Вимоги до інформаційної і програмної сумісності.

3.4.1. Вимоги до інформаційних структур на вході і виході

Формат відображення даних має дозволяти імпорт даних в додатки MS Office для редагування та перегляду.

3.4.2. Вимоги до методів рішення і мов програмування

Вибір методів рішення здійснюється розробникам без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижевська О.В.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

3.4.3. Вимоги до системи програмних засобів.

Вимоги до програмного забезпечення сервера:

ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником).

Вимоги до програмного забезпечення робочої станції:

ОС - родина Windows не нижче Windows XP.

Драйвери периферійних пристроїв - введення/виводу визначаються та встановлюються при встановленні ОС в залежності від конфігурації робочої станції.

Пакет додатків MS Office.

Робоча станція є клієнтом мереж Microsoft.

3.5. Вимоги до складу і параметрів технічних засобів.

Вимоги до складу технічних засобів:

1. Сервер:

- сервер у базовій конфігурації із підтримкою RAID;
- з'ємний запам'ятовуючий пристрій для архівування даних;
- принтер для друку;
- засоби для під'єднання до локальної мережі.

2. Робоча станція:

- ПК на базі процесорів Intel, AMD у стандартній комплектації;
- периферійні пристрої друку;
- засоби для під'єднання до локальної мережі. засоби для під'єднання до локальної мережі.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В				63
Змн.	Арк.	№ докум.	Підпис	Дата		

- Pentium III з тактовою частотою 750 МГц;
- оперативна пам'ять - 128Мбайт;
- об'єм дискової пам'яті - 20Гбайт.

4. Вимоги до програмної документації

Програмна документація повинна включати наступні відомості:

1. "Інструкція по інсталяції (встановленню) програми", складається з опису інсталяційного пакету, переліку етапів інсталяції та їх послідовності, опису встановлених програмних компонентів та режимів їх роботи після інсталяції. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться у 4 розділі.

2. "Керівництво користувача" складається з опису послідовності завантаження програми, основних режимів роботи, опису основних екранних форм, переліку виняткових ситуацій та реакції користувача на них, порядку виконання завдань в системі. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 3 розділі курсової роботи.

3. "Керівництво адміністратора баз даних", складається з опису складу таблиць бази даних та правил доступів до них, опису та послідовності робіт по обслуговуванню бази даних (архівування, резервне копіювання, з вказівкою періодичності виконання та засобів, що для цього використовуються), порядку налаштування серверу та клієнтських додатків. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 2,3 та 4 розділах курсової роботи згідно плану.

		Дубинченко Б.М.			ДУ «Житомирська політехніка».22.121.07.000 - ПЗ	Арк.
		Чижмотря О.В.				64
Змн.	Арк.	№ докум.	Підпис	Дата		