

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ

«До захисту допущена»

Зав. кафедри прикладної математики

д.т.н. професор Турбал Ю.В.

«___»_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка сайту для оренди транспортних засобів»

Виконав: Менчинський Антон Романович

студент навчально-наукового інституту автоматки, кібернетики та
обчислювальної техніки

група КН-41

Керівник: старший викладач Герус Володимир Андрійович

Рівне-2023

Національний університет водного господарства та природокористування

НН інститут **автоматики, кібернетики та обчислювальної техніки**

Кафедра **комп'ютерних наук та прикладної математики**

Освітньо-кваліфікаційний рівень **бакалавр**

Галузь знань 12 «Інформаційні технології»

Спеціальність 122 «Комп'ютерні науки»

Спеціалізація _____

ЗАТВЕРДЖУЮ

Завідувач кафедри

 “ ” _____ 20__ року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Менчинському Антону Романовичу

1. Тема роботи РОЗРОБКА САЙТУ ДЛЯ ОРЕНДИ ТРАНСПОРТНИХ ЗАСОБІВ
керівник роботи старший викладач Герус В.А.
затверджені наказом вищого навчального закладу від «19» квітня 2023 року С
№-449
2. Термін подання роботи студентом 21.06.2023
3. Вихідні дані до роботи технології, що необхідні для розробки сайту для оренди транспортних засобів.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ. РОЗДІЛ 2. ПРОЕКТУВАННЯ САЙТУ. РОЗДІЛ 3. РОЗРОБКА САЙТУ.

5. Перелік графічного матеріалу (з точним зазначення обов'язкових креслень) мультимедійна презентація.

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|----------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Розділ 1 | Старший викладач Герус В.А. | 01.05.2023 | 10.05.2023 |
| Розділ 2 | Старший викладач Герус В.А. | 11.05.2023 | 23.05.2023 |
| Розділ 3 | Старший викладач Герус В.А. | 25.05.2023 | 20.06.2023 |

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|---------------------------------------|-------------------------------|----------|
| 1 | Вивчення літератури | 01.11.2022 – 11.01.2023 | виконав |
| 2 | Формування задачі | 10.03.2023 – 17.03.2023 | виконав |
| 3 | Вибір стеку технологій | 18.03.2023 – 20.03.2023 | виконав |
| 4 | Аналіз даних | 22.03.2023 – 15.03.2023 | виконав |
| 5 | Розробка структури сайту | 25.03.2023 – 10.04.2023 | виконав |
| 6 | Розробка інтерфейсу | 12.04.2023 – 10.05.2023 | виконав |
| 7 | Тестування програми | 11.05.2023 – 15.05.2023 | виконав |
| 8 | Написання звіту до дипломної роботи | 01.05.2023 – 20.06.2023 | виконав |
| 9 | Підготовка мультимедійної презентації | 20.06.2023 – 22.06.2023 | виконав |
| 10 | Підготовка до виступу | 20.06.2023 – 25.06.2023 | виконав |

Студент

_____ Менчинський А.Р.
(підпис) (прізвище та ініціали)

Керівник роботи

_____ Герус В.А.
(підпис) (прізвище та ініціали)

ЗМІСТ

| | |
|---|----|
| РЕФЕРАТ | 6 |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ..... | 7 |
| ВСТУП..... | 8 |
| РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ | 11 |
| 1.1 Загальні відомості про оренду транспортних засобів..... | 11 |
| 1.2 Практичне застосування предметної області..... | 11 |
| 1.3 Оцінка переваг та недоліків найпоширеніших реалізацій | 12 |
| 1.3.1 BLS..... | 13 |
| 1.3.2 Ukr-prokat | 14 |
| 1.3.3 OLX..... | 16 |
| 1.4 Порівняння характеристик | 17 |
| 1.5 Постановка задачі..... | 20 |
| 1.6 Висновки першого розділу..... | 21 |
| РОЗДІЛ 2 ПРОЕКТУВАННЯ САЙТУ | 22 |
| 2.1 Огляд технологій | 22 |
| 2.1.1 React..... | 22 |
| 2.1.2 Node.js та Express | 23 |
| 2.1.3 MongoDB | 25 |
| 2.2 Опис функціоналу системи | 27 |
| 2.2.1 Функціонал неавторизованого користувача..... | 28 |
| 2.2.2 Функціонал для авторизованого користувача..... | 29 |
| 2.2.3 Функціонал для постачальника | 29 |
| 2.2.3 Функціонал для адміністратора системи..... | 30 |
| 2.3 Формування вимог з використанням діаграми прецедентів..... | 30 |
| 2.4 Проектування архітектури сайту | 36 |
| 2.5 Архітектура бекенд частини | 38 |
| 2.6. Висновки другого розділу | 39 |
| РОЗДІЛ 3 | 41 |
| РОЗРОБКА САЙТУ | 41 |
| 3.1 Розробка структури бази даних | 41 |
| 3.2 Реалізація програми | 43 |

| | |
|----------------------------------|----|
| 3.3. Огляд інтерфейсу | 50 |
| ВИСНОВКИ..... | 59 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 60 |

РЕФЕРАТ

Кваліфікаційна робота: 60 с., 32 малюнка, 1 таблиця, 12 джерел.

Мета роботи: розробка функціонального сайту для оренди транспортних засобів, який забезпечить зручність оренди для користувачів. Завдання роботи включають вивчення потреб та вимог користувачів, проектування та розробку веб-додатку, тестування та впровадження розробленого рішення.

Об'єкт досліджування – розробка сайту для оренди транспортних засобів.

Методи дослідження – аналіз літературних джерел, дослідження ринку, проектування архітектури, проектування бази даних, реалізація сайту.

Результатом дипломної роботи є сайт для оренди транспортних засобів. Сайт використовує сучасні технології розробки, такі як Node.js, Express, React, та дотримується архітектури MVC. Вона забезпечує зручний інтерфейс для користувачів, де вони можуть переглядати доступні транспортні засоби, здійснювати бронювання і відстежувати статуси своїх орендних операцій. Результат роботи включає детальну документацію, включаючи опис вимог та архітектуру системи. Розроблений сайт надає можливість користувачу легко орендувати транспортний засіб, а постачальнику публікувати власні транспортні засоби.

Ключові слова: NODE.JS, REACT, MONGODB, EXPRESS, API, ОРЕНДА ТРАНСПОРТУ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

| | | |
|----------|---|---|
| React | — | Бібліотека JavaScript для розробки інтерфейсу користувача. |
| Node.js | — | Виконавче середовище JavaScript, що дозволяє виконувати код на сервері. |
| Express | — | Фреймворк Node.js для створення веб-додатків та API. |
| MongoDB | — | Документ-орієнтована система керування базами даних, яка використовує JSON-подібні документи. |
| API | — | Інтерфейс програмування додатків, який дозволяє взаємодіяти з іншими програмами або сервісами. |
| DB | — | Скорочення від "database" (база даних). |
| JWT | — | JSON Web Token, стандарт відкритих токенів для передачі безпечної інформації у вигляді JSON. |
| ТЗ | — | Скорочення від «транспортний засіб». |
| Фронтенд | — | Клієнтська частина веб-додатка, зазвичай відповідає за відображення і взаємодію з користувачем. |
| Бекенд | — | Серверна частина веб-додатка, зазвичай відповідає за обробку запитів та бізнес-логіку. |

ВСТУП

Актуальність теми. У сучасному світі, який перебуває в стані постійної зміни та руху, мобільність стає необхідним аспектом життя. Необхідність у швидкому та зручному доступі до транспортних засобів постає в різних ситуаціях, зокрема, у зв'язку з війнами, конфліктами та кризовими ситуаціями. Водночас, оренда транспорту є економічно вигідним рішенням у багатьох випадках та може бути єдиним доступним варіантом для людей.

У ці часи доступ до транспорту стає першочерговою проблемою. Інфраструктура, яка колись полегшувала щоденні поїздки на роботу і торгівлю, часто пошкоджена або зруйнована, що призводить до того, що люди опиняються в глухому куті та ізоляції. Потреба в мобільності посилюється, коли люди шукають притулку, медичної допомоги та можливостей працевлаштування за межами зон конфлікту. Крім того, гуманітарні організації покладаються на ефективні транспортні системи, щоб дістатися до постраждалих районів і доставити предмети першої необхідності.

Оренда транспортних засобів стала життєздатним і вигідним рішенням для людей. Традиційне володіння транспортним засобом може бути нездійсненим або фінансово обтяжливим через зростання витрат та обмеженість ресурсів. Таким чином, оренда транспортного засобу є економічно вигідною альтернативою, яка дозволяє людям отримати доступ до транспорту без обтяжливих зобов'язань, пов'язаних з володінням. Крім того, послуги з оренди транспортних засобів пропонують гнучкі умови, що дозволяє людям адаптувати свої потреби в мобільності відповідно до обставин, що змінюються, і різної тривалості перебування.

Практична новизна цієї роботи полягає в поєднанні актуальної потреби в мобільності та доступі до транспортних засобів з використанням сучасних технологій веб-розробки. Розробка веб-сайту для оренди транспортних засобів вимагає вивчення сучасного стану ринку оренди, аналізу потреб та очікувань

користувачів, розробки ефективної структури та функціоналу додатку, а також створення зручного інтерфейсу для користувачів.

Мета і завдання кваліфікаційної роботи – є розробка функціонального веб-сайту для оренди транспортних засобів, який забезпечить зручність та доступність для користувачів. Завдання роботи включають вивчення потреб та вимог користувачів, проектування та розробку веб-додатку, тестування та впровадження розробленого рішення.

Для досягнення поставленої мети в роботі будуть використовані основні методи та підходи до розробки веб-додатків, такі як аналіз вимог, проектування архітектури, програмування та впровадження. Також будуть використані сучасні технології веб-розробки, зокрема мови програмування, фреймворки та бази даних, щоб забезпечити ефективну та надійну роботу веб-сайту для оренди транспортних засобів.

Особистий внесок випускника – розробка WEB-сайту для надання послуг оренди транспортних засобів з використанням React, Node.js, MongoDB та Javascript.

Враховуючи вищевикладене, було сформульовано конкретні цілі, що передбачають:

- Аналіз та розробку основного функціоналу, який служитиме основою для подальшої розробки майбутнього продукту. Це включає вивчення та оцінку вимог користувачів, а також розробку відповідних функцій та можливостей веб-сайту для оренди транспортних засобів.
- Проектування зрозумілого та простого інтерфейсу, який буде доступним та зрозумілим для користувачів незалежно від їхнього досвіду використання подібних веб-застосунків. Це включає розробку інтуїтивно зрозумілих елементів управління, зручну навігацію та забезпечення зручності взаємодії користувача з платформою.
- Проектування веб-додатку з використанням сучасних технологій та дотримання загальноприйнятих норм написання коду та побудови

структури. Метою є створення гнучкої та розширюваної системи, яка сприятиме подальшому вдосконаленню та розширенню функціоналу платформи.

- Реалізація бази даних, яка забезпечить зберігання та організацію інформації, необхідної для роботи веб-сайту. Це включає проектування структури бази даних, наповнення необхідними значеннями та забезпечення взаємодії з ними з веб-додатку.

Отже, розробка веб-сайту для оренди транспортних засобів є актуальною та перспективною задачею, що поєднує потреби у мобільності та доступі до транспорту. Подальша робота буде спрямована на розробку та впровадження функціонального веб-додатку, який полегшить процес оренди та підвищить зручність для користувачів.

РОЗДІЛ 1

ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальні відомості про оренду транспортних засобів

Індустрія оренди, що охоплює різні сектори, такі як житло, обладнання та транспортні засоби, зазнала значного зростання і популярності в останні роки. Це можна пояснити кількома факторами, серед яких зміна споживчих уподобань, фінансові міркування та перехід до економіки спільного користування. Оренда пропонує людям гнучкість у доступі до активів без довгострокових зобов'язань або фінансового тягаря власності. Зручність та економічна ефективність оренди знайшла відгук у широкого кола споживачів, що призвело до різкого зростання попиту на послуги оренди в різних сферах.

Історичне коріння оренди транспортних засобів можна простежити до початку 20-го століття, коли з'явилися перші компанії з прокату автомобілів. Спочатку ці компанії обслуговували переважно ділових людей та туристів. Однак з розвитком автомобільної індустрії та підвищенням доступності подорожей послуги з прокату транспортних засобів набули популярності серед ширшої клієнтської бази.

Розвиток послуг з прокату транспортних засобів був додатково пришвидшений соціальними змінами, включаючи посилення урбанізації, зростання туризму та збільшення наявного доходу. Ці фактори створили попит на тимчасові транспортні засоби, що призвело до створення великих прокатних компаній, які задовольняли різноманітні потреби клієнтів. Ці фактори створили попит на тимчасові транспортні засоби, що призвело до створення великих прокатних компаній, які задовольняли різноманітні потреби клієнтів.

1.2 Практичне застосування предметної області

Веб-сайт для оренди транспортних засобів надає користувачам цілодобовий доступ до широкого спектру транспортних засобів, що дозволяє їм

переглядати та бронювати оренду в зручний для них час. Така зручність усуває необхідність фізичного відвідування агентств з прокату та спрощує весь процес оренди.

Оренда транспортного засобу через онлайн-платформу пропонує економічні переваги порівняно з власним автомобілем або альтернативними видами транспорту. Гнучкі періоди оренди дозволяють користувачам пристосовувати свої транспортні домовленості до своїх конкретних потреб, що призводить до оптимальної економічної ефективності. Статистика показує, що мешканці міст використовують свої автомобілі всього лише 3-4% від усього часу володіння ними. Окрім того, автомобілі займають значну кількість місця на міських дорогах порівняно з громадським транспортом. Оскільки автомобілі займають так багато простору, доцільніше використовувати їх постійно, а не лише 3% часу. Той час, коли автомобіль стоїть без використання, є неефективним. Застосування концепції спільної економіки для додаткового використання автомобілів має безсумнівні переваги. Зменшення кількості автомобілів у місті сприятиме розгрузці доріг і покращенню екологічної ситуації.

1.3 Оцінка переваг та недоліків найпоширеніших реалізацій

Ринок веб-сайтів з прокату транспортних засобів за останні роки зазнав значного зростання завдяки великому попиту на доступні та зручні транспортні рішення. Цей ринок є динамічним та постійно розвивається, пропонуючи широкий спектр послуг для задоволення потреб клієнтів у оренді транспорту на короткий термін. Він стає все більш популярним як серед приватних осіб, так і серед підприємств, забезпечуючи зручні та ефективні рішення для користування транспортом.

Сайти оренди транспортних засобів надають різноманітний функціонал, який спрощує процес оренди та забезпечує комфорт користувачів. Вони пропонують можливість швидкого та зручного пошуку доступних автомобілів за

різними критеріями, такими як тип транспорту, місце розташування та тривалість оренди. Крім того, вони часто надають можливість здійснити бронювання та оплату онлайн, що забезпечує швидку та безпечну транзакцію для користувачів. Багато сайтів також пропонують додаткові сервіси, такі як страхування автомобіля та цілодобова підтримка клієнтів, що сприяє забезпеченню високої якості обслуговування.

Однак, на ринку сайтів оренди транспортних засобів також існують деякі недоліки, які варто враховувати. Наприклад, деякі сайти можуть мати обмежений асортимент автомобілів, що може звужувати вибір для користувачів та не задовольняти їхні потреби. Також, не всі сайти оренди транспорту гарантують однакову якість автомобілів, що може призвести до незадоволеності користувачів. Крім того, вартість оренди транспорту може варіюватися на різних сайтах, і існує ризик непередбачених додаткових витрат або прихованих платежів, що може вплинути на вибір користувачів. А також більшість сайтів не надають можливість користувачу здавати власний транспорт у платне користування.

Отже, при вході на ринок сайтів оренди транспортних засобів важливо уважно розглянути та вирішити ці недоліки, забезпечуючи широкий вибір автомобілів, високу якість обслуговування та прозорість у вартості оренди.

Перейдемо до огляду існуючих сайтів, що надають можливість орендувати транспортні засоби:

1.3.1 BLS

Сайт компанії BLS - це онлайн-платформа, яка спеціалізується на оренді транспортних засобів в Україні. Цей сайт надає можливість знайти та орендувати різноманітні види автомобілів, включаючи легкові авто, мікроавтобуси та позашляховики.

Основні переваги сайту BLS включають:

- Широкий вибір автомобілів: Сайт пропонує великий асортимент транспортних засобів різних марок, моделей та типів, що дозволяє користувачам знайти оптимальний варіант, відповідний їхнім потребам та бюджету.
- Зручний процес бронювання: BLS.ua надає можливість швидкого та зручного бронювання автомобіля через онлайн-систему. Користувачам потрібно ввести дати оренди, місце отримання та повернення авто, а також особисті дані, щоб забезпечити безпроблемну та ефективну процедуру оренди.
- Конкурентні ціни: Сайт пропонує конкурентні ціни на оренду автомобілів, дозволяючи користувачам знайти вигідний варіант. Крім того, BLS.ua регулярно запускає акції та знижки, що додатково забезпечують економію коштів.

Недоліки сайту BLS.ua:

- Оренда обмежена територією: Деякі автомобілі можуть бути доступні лише в певних місцях або обмежені в межах країни. Це може створювати деякі обмеження для користувачів, які планують подорожувати за кордоном або мають специфічні потреби.
- Додаткові витрати: На деякі послуги та опції, такі як страхування, додаткове обладнання чи послуги доставки автомобіля, можуть бути застосовані додаткові платежі. Користувачі повинні бути уважними при виборі опцій та ознайомлюватися з умовами оренди, щоб уникнути непередбачених витрат.

1.3.2 Ukr-prokat

Сайт Ukr-prokat - це веб-платформа, що спеціалізується на оренді транспортних засобів в Україні. Цей сайт пропонує широкий вибір автомобілів, мотоциклів, вантажних авто та інших транспортних засобів для оренди в різних містах країни.

На сайті Ukr-Prokat.com можна знайти різноманітні марки і моделі автомобілів, включаючи економічні легкові машини, представницькі авто, вантажні фургони, SUV-и та багато іншого. Користувачі мають можливість вибрати транспортний засіб, який найкраще відповідає їхнім потребам та бюджету.

Сайт Ukr-Prokat.com пропонує такі переваги:

- Зручний пошук і бронювання: Користувачам надається зручний пошуковий інструмент, що дозволяє вибрати необхідні параметри транспортного засобу, включаючи місце та дати оренди. Після цього можна здійснити онлайн-бронювання без зайвих зусиль.
- Різноманітність автомобілів: Сайт має широкий асортимент транспортних засобів, що задовольняють різні потреби та вимоги. Вибір доступних моделей допомагає знайти оптимальний варіант для подорожей, бізнесу або особистих потреб.
- Проста процедура оренди: Користувачі можуть швидко та легко здійснити процес оренди без зайвих формальностей. Усі необхідні документи та деталі можна оформити онлайн, що зекономить час і зусилля.
- Гнучкі умови: Ukr-Prokat.com пропонує різні варіанти оренди, включаючи короткострокову і довгострокову оренду. Користувачі можуть вибрати зручний період та умови, що відповідають їхнім потребам.

Незважаючи на переваги, варто враховувати деякі недоліки сайту Ukr-Prokat.com:

- Обмеженість місць оренди: У деяких випадках місця оренди можуть бути обмежені лише деякими регіонами або містами України. Це може створювати деякі обмеження для користувачів, які планують подорожувати за межі обслуговування.

- **Додаткові витрати:** Деякі послуги та додаткові опції, такі як страхування, навігаційна система або дитяче автокрісло, можуть мати додаткову плату. Користувачі повинні уважно ознайомитися з умовами оренди та враховувати можливі додаткові витрати.
- **Неможливість зареєструватись** постачальником ТЗ

1.3.3 OLX

Сайт OLX є відомою платформою для онлайн-оголошень в Україні. Хоча цей сервіс не є спеціалізованим для оренди транспортних засобів, проте він надає таку можливість і користується популярністю у цьому напрямку. Сайт надає можливість користувачам оголошувати про свої транспортні засоби, доступні для оренди, а також переглядати пропозиції інших користувачів.

Основні переваги сайту OLX.ua як платформи для оренди транспортних засобів:

- **Широкий вибір:** OLX.ua забезпечує велику кількість пропозицій щодо оренди транспортних засобів. Користувачі можуть знайти автомобілі різних марок, моделей і характеристик, що відповідають їхнім потребам.
- **Безпосередні контакти:** На OLX.ua можна зв'язатися безпосередньо з власником транспортного засобу, що видається зручним для уточнення умов оренди, деталей та узгодження домовленостей.
- **Можливість переговорів:** OLX.ua надає можливість вести переговори з власником автомобіля щодо умов оренди, цін та інших аспектів. Це дозволяє знайти оптимальні умови, які задовольняють обидві сторони.
- **Гнучкість:** Платформа OLX.ua дозволяє власникам автомобілів самостійно встановлювати умови оренди, такі як тривалість, ціна та додаткові вимоги. Користувачі можуть шукати варіанти, що найкраще відповідають їхнім вимогам та бюджету.

Основні недоліки:

- **Відповідальність:** OLX.ua не несе відповідальності за угоди між орендарями та власниками автомобілів. Користувачам необхідно самостійно перевіряти інформацію про власників та умови оренди для забезпечення безпеки та надійності.
- **Конкуренція:** Завдяки великій кількості пропозицій, на OLX.ua може бути значна конкуренція між орендарями. Користувачам потрібно бути готовими до швидкого реагування та конкурувати за найкращі пропозиції.
- **Обмежена функціональність:** Платформа OLX.ua може мати обмежений функціонал порівняно зі спеціалізованими сайтами для оренди транспортних засобів. Відсутність деяких функцій, таких як фільтри пошуку або онлайн-бронювання, може ускладнити процес узгодження оренди.

1.4 Порівняння характеристик

Розробка веб-сайту для оренди транспортних засобів має на меті вирішити існуючі проблеми в індустрії прокату і забезпечити ефективну та зручну платформу для клієнтів. Для розробки веб-сайту дуже оцінити існуючі веб-сайти з прокату автомобілів і визначити найбільш значущі проблеми, які потребують вирішення. У цьому дослідженні проведено оцінку ринку та аналіз проблем трьох відомих веб-сайтів з прокату автомобілів: BLS.ua, ukr-prokat.com та OLX.ua. Використовуючи систему оцінювання, ми прагнемо виявити сильні та слабкі сторони цих сайтів, визначити пріоритетні напрямки для вдосконалення та забезпечити розробку надійного та орієнтованого на користувача веб-сайту для оренди транспортних засобів.

Система оцінювання, застосована в цьому дослідженні, використовує шкалу від 1 до 4, де кожен бал означає наступне:

- 1) «1» — характеристика має значні недоліки, що перешкоджають користувацькому досвіду та не відповідають основним вимогам.

- 2) «2» — характеристика відповідає мінімальним очікуванням, але має простір для вдосконалення для підвищення задоволеності користувачів.
- 3) «3» — характеристика відповідає очікуванням користувача та забезпечує задовільний досвід.
- 4) «4» — характеристика перевершує очікування користувача, демонструючи видатну продуктивність і забезпечуючи винятковий користувацький досвід.

Перейдемо до оцінки сайтів:

Таблиця 1.1

Аналіз сайтів для оренди транспортних засобів

| Критерії | OLX | Укр-Прокат | BLS | Вага |
|------------------------------|-----|------------|-----|------|
| Інтерфейс користувача | 1 | 3 | 4 | 0,1 |
| Прозорість ціноутворення | 2 | 2 | 3 | 0,2 |
| Можливість бути орендодавцем | 3 | 1 | 1 | 0,1 |
| Фільтрація транспорту | 2 | 3 | 3 | 0,2 |
| Наявність локалізації | 2 | 3 | 3 | 0,1 |
| Інформація про транспорт | 2 | 2 | 2 | 0,15 |
| Зручність орендування | 1 | 4 | 4 | 0,15 |

У наведеній вище таблиці кожній характеристиці присвоєно вагу, яка відображає її відносну важливість.

Вага - це числове значення, що лежить в межах від 0 до 1, присвоєне кожній оцінюваній характеристиці, що відображає її відносну важливість або пріоритет у загальній оцінці. Це дозволяє систематично порівнювати різні аспекти і допомагає визначити їхню значущість у впливі на загальну продуктивність або користувацький досвід.

Оцінки для BLS.ua, ukr-prokat.com та OLX.ua надані на основі їхніх показників за кожною характеристикою. Ці бали дають змогу комплексно

порівняти та оцінити можливості сайтів, а також визначити їхню загальну придатність для користувачів.

Розрахуємо фінальні за допомогою формули $\sum s * w$, де s – оцінка характеристики сайту, а w – її вага:

Таблиця 1.2

Оцінки характеристик сайтів для оренди транспортних засобів

| | OLX | Укр-Прокат | BLS |
|--------|------|------------|-----|
| Оцінки | 1,85 | 2,6 | 2,9 |

Отже, оцінка веб-сайтів з прокату автомобілів, включаючи BLS.ua, ukr-prokat.com та OLX.ua, надала цінну інформацію про їхні сильні та слабкі сторони. За результатами оцінки очевидно, що сайт BLS виявився найкращим, оскільки відповідає найважливішим характеристикам з точки зору користувацького інтерфейсу, прозорості ціноутворення, процесу бронювання та загального користувацького досвіду. З іншого боку, сайт OLX відстає в оцінці, оскільки він є платформою загальних оголошень і не має спеціалізації, розробленої для послуг з оренди автомобілів.

Просуваючись далі в розробці сайту з прокату автомобілів, дуже важливо розглянути критерії оцінки в порядку зменшення їхньої ваги. Віддаючи перевагу характеристикам з більшою вагою, ми можемо зосередитися на тих сферах, які суттєво впливають на користувацький досвід і загальну продуктивність. Такий підхід гарантує, що отриманий веб-сайт не тільки відповідає очікуванням користувачів, але й перевершує їх у таких важливих аспектах, як користувацький інтерфейс, вибір транспортних засобів і прозорість ціноутворення, про що свідчить успіх сайту BLS.

Крім того, процес оцінювання підкреслює важливість створення спеціалізованого веб-сайту, оптимізованого під конкретну задачу прокату автомобілів. Така спеціалізація забезпечує більш індивідуальний і безперешкодний користувацький досвід, сприяючи ефективному вибору

автомобіля, прозорому ціноутворенню і спрощенню процесу бронювання. Враховуючи ці висновки в процесі розробки, метою є створення веб-сайту з прокату автомобілів, який не тільки перевершить своїх конкурентів, але й забезпечить клієнтам найкращий та найприємніший досвід прокату автомобілів.

Таким чином, оцінка служить цінним орієнтиром для розробки надійного і орієнтованого на користувача веб-сайту з прокату автомобілів.

1.5 Постановка задачі

Метою кваліфікаційної роботи бакалавра є розробка сайту для оренди транспортних засобів. Сайт допоможе користувачу обрати автомобіль, ознайомитись з цінами, а також забронювати вибраний автомобіль для оренди. Також сайт дасть можливість розміщувати свої авто для оренди.

Згідно вище описаних переваг та недоліків основні завдання проекту включають:

1. Сайт повинен бути розроблений у вигляді веб-додатку, який буде складатись з бекенд та фронтенд частин.
2. Система повинна мати базу даних в якій буде зберігатись вся необхідна інформація для коректної роботи сайту.
3. Користувач повинен мати змогу реєструватись.
4. Користувач має змогу вибрати на сайті авто, обирати локацію, постачальника та характеристик авто для зручного пошуку необхідного транспортного засобу.
5. Орендодавець може реєструватись та додавати свої авто для оренди.
6. Орендар та орендодавець повинні отримувати повідомлення на пошту та сповіщення на сайті.
7. Адміністратор повинен мати змогу додавати нових постачальників.
8. Орендодавець має змогу редагувати свої авто, обирати чи створювати локації, а також керувати бронюванням.

1.6 Висновки першого розділу

У першій частині роботи були розглянуті основні критерії, за якими можна порівнювати сайти для оренди транспортних засобів з точки зору розробки. Було встановлено, що такі критерії, як швидкість завантаження, безпека, мобільна сумісність, зручність інтерфейсу та використання технологій, є важливими для забезпечення успішного та задоволеного користування сайтом.

Детальний аналіз кожного з цих критеріїв дав можливість визначити сильні та слабкі сторони різних сайтів. Виявлено, що найкращі сайти відрізняються зручними та інтуїтивно зрозумілими інтерфейсами для користувачів, а використані технології відповідають сучасним стандартам розробки веб-сайтів.

Загалом, використання цих критеріїв дозволяє провести об'єктивне порівняння між різними сайтами для оренди транспортних засобів з точки зору їх розробки. Це допомагає вибрати найбільш оптимальну та високоякісну платформу для користувачів, яка забезпечує зручність, надійність та задоволення від використання.

РОЗДІЛ 2

ПРОЕКТУВАННЯ САЙТУ

2.1 Огляд технологій

2.1.1 React

React, широко розповсюджена бібліотека JavaScript, яка здобула значне визнання у сфері веб-розробки завдяки своїй ефективності у створенні користувацьких інтерфейсів, гнучкістю та можливістю багаторазового використання. Завдяки глибокому розумінню основоположних принципів React та притаманних йому можливостей, розробники можуть вправно використовувати весь його потенціал, тим самим дозволяючи створювати стійкі та високоефективні фронтенд-додатки.

React відомий своїм декларативним синтаксисом, який дозволяє розробникам описувати бажаний стан інтерфейсу користувача. Замість того, щоб вручну маніпулювати з DOM деревом, розробники визначають компоненти та їх властивості, а React реалізує ефективне оновлення фактичного DOM. Такий декларативний підхід спрощує процес розробки, роблячи код більш лаконічним і зрозумілим.

Однією з визначальних особливостей React є його компонентна архітектура. Компоненти - це будівельні блоки React-додатків, що інкапсулюють логіку, структуру та стилізацію певного елемента інтерфейсу користувача. Ці компоненти можна легко використовувати повторно, компонувати та вкладати один в одного, що дозволяє розробникам створювати складні та модульні ієрархії інтерфейсу. Можливість багаторазового використання React-компонентів підвищує зручність супроводу коду, заохочує модульність коду та сприяє ефективній розробці.

Існує кілька вагомих причин для вибору React як найкращої технології для розробки фронтенду. По-перше, React сприяє повторному використанню коду, дозволяючи розробникам створювати бібліотеки багаторазових компонентів, які можна використовувати в різних проектах. Таке багаторазове використання

зменшує дублювання коду, підвищує продуктивність та забезпечує узгодженість патернів інтерфейсу користувача у всьому додатку.

По-друге, концепція віртуального DOM та ефективний алгоритм узгодження React забезпечує чудову продуктивність нашого додатку. React підтримує легке віртуальне представлення фактичного DOM і оновлює лише необхідні частини, коли стан додатку змінюється. Такий підхід мінімізує непотрібний повторний рендеринг, значно покращуючи швидкість відображення сторінки та загальну продуктивність додатку.

Крім того, React має активну спільноту, яка сприяла створенню багатой екосистеми сторонніх бібліотек та інструментів. Розробники можуть використовувати ці бібліотеки для покращення функціональності своїх React-додатків. React Router, наприклад, спрощує реалізацію маршрутизації в React-додатку. Наявність таких інструментів дозволяє розробникам з легкістю створювати багатофункціональні додатки.

React став революційною технологією у фронтенд-розробці, пропонуючи декларативний та компонентний підхід до побудови користувацьких інтерфейсів. Його ключові особливості, такі як декларативний синтаксис та багаторазове використання компонентів, сприяють модульності коду, зручності супроводу та ефективності. Концепція віртуального DOM та ефективний алгоритм узгодження в React сприяють його продуктивності. Широка екосистема сторонніх бібліотек та інструментів ще більше розширює можливості React, надаючи нам додаткові функціональні можливості та рішення для поширених проблем.

2.1.2 Node.js та Express

Node.js - це середовище виконання JavaScript, побудоване на JavaScript-движку V8 від Chrome. Воно дозволяє розробникам запускати код JavaScript за межами браузера, забезпечуючи розробку на стороні сервера. Node.js використовує модель вводу/виводу, керовану не блокуючими подіями, що

робить його високоефективним і придатним для обробки великих обсягів одночасних запитів.

Express, швидкий і незалежний веб-фреймворк для Node.js, що спрощує створення надійних і масштабованих веб-додатків. Він надає мінімалістичний, але потужний набір функцій для маршрутизації, інтеграції проміжного програмного забезпечення та обробки запитів. Express сприяє модульності коду, дозволяючи розробникам розбивати складні додатки на менші, керовані модулі. Гнучкість Express дозволяє розробникам налаштовувати та розширювати його функціональність відповідно до вимог проекту.

Вибір Node.js та Express для цього проекту був зумовлений кількома факторами. По-перше, асинхронна, неблокуюча природа Node.js робить його високопродуктивним, дозволяючи ефективно обробляти паралельні запити та масштабувати. Архітектура Node.js, керована подіями, дозволяє оптимально використовувати ресурси, що призводить до підвищення продуктивності та швидкості реагування.

Крім того, Node.js та Express сприяють оптимізації процесу розробки. JavaScript, будучи основною мовою як для фронтенд, так і для бекенд-розробки, дозволяє повторно використовувати код і скорочує час розробки додатку. Цей уніфікований мовний підхід дозволяє безперешкодно працювати з різними рівнями додатку, що призводить до пришвидшення циклів розробки.

Крім того, розгалужена екосистема та активна спільнота, що оточує Node.js та Express, надають безліч бібліотек, інструментів та ресурсів. Ці ресурси сприяють швидкості розробки, якості коду та доступності найкращих практик. Зріла та надійна природа екосистеми забезпечує довгострокову підтримку та полегшує розробку масштабованих та підтримуваних додатків.

Отже, Node.js та Express є потужними технологіями для бекенд-розробки, пропонуючи ефективність, масштабованість та універсальність. Подійно-керований, неблокуючий характер Node.js у поєднанні з простотою та гнучкістю Express надає нам надійний інструментарій для створення високопродуктивних веб-додатків. Рішення обрати Node.js та Express для цього проекту ґрунтувалося

на їхніх перевагах у продуктивності, спрощеному процесі розробки, а також широкій підтримці екосистеми та спільноти. А також уніфікована мова JavaScript у поєднанні з великою екосистемою забезпечує безперебійну розробку та доступ до широкого спектру інструментів і ресурсів. Node.js та Express.

2.1.3 MongoDB

MongoDB - це документно-орієнтована база даних NoSQL, яка зберігає дані у JSON-подібному форматі, відомому як BSON (Binary JSON). Її гнучка структура дозволяє зберігати документи динамічно і без схем, пропонуючи високий ступінь адаптивності до структур даних. Архітектура MongoDB розроблена для горизонтального масштабування, що дозволяє безперешкодно розподіляти дані між декількома серверами та забезпечує високу доступність і відмовостійкість.

Однією з основних причин вибору MongoDB для нашого проекту є її здатність працювати з неструктурованими даними, що швидко змінюються. На відміну від традиційних реляційних баз даних, документно-орієнтована модель MongoDB дозволяє гнучко проектувати схеми, пристосовуючись до змін у структурі даних без необхідності дорогої міграції. Це робить MongoDB сприятливим вибором для проектів, що мають справу з динамічними та мінливими вимогами до даних.

Крім того, MongoDB пропонує високу продуктивність і масштабованість. Її здатність розподіляти дані між декількома вузлами дозволяє здійснювати горизонтальне масштабування, що забезпечує плавне розширення зі зростанням обсягу даних. Функція автоматичного шардингу MongoDB забезпечує ефективний розподіл даних, що призводить до підвищення продуктивності та пропускної здатності запитів. Крім того, вбудований механізм реплікації MongoDB забезпечує відмовостійкість і високу доступність, гарантуючи довговічність даних і мінімальний час простою.

Ще однією перевагою MongoDB є широкі можливості для створення запитів. Вона підтримує багату мову запитів, яка включає такі потужні функції, як конвеєри агрегації, повнотекстовий пошук та геопросторові запити. Ці розширені можливості запитів дозволяють розробникам виконувати складні операції аналізу та пошуку даних, що робить MongoDB придатною для додатків зі складними вимогами до обробки даних.

У порівнянні з традиційними реляційними базами даних, MongoDB має кілька явних переваг. По-перше, її гнучка схема усуває необхідність попереднього проектування схеми, адаптуватись до мінливих бізнес-вимог. Така гнучкість особливо корисна у швидкоплинних середовищах розробки, де швидке створення прототипів і часті зміни є звичним явищем.

Крім того, MongoDB має переваги масштабованості та продуктивності. Завдяки можливості горизонтального масштабування шляхом додавання нових вузлів до кластера, MongoDB може без проблем обробляти великі обсяги даних і високі навантаження на трафік. Розподілена природа MongoDB забезпечує плавне масштабування без втрати продуктивності, що робить її придатною для додатків, які передбачають швидке зростання і збільшення вимог до даних.

Крім того, документно-орієнтована модель даних MongoDB добре узгоджується з сучасними парадигмами розробки додатків, такими як мікросервіси та гнучка розробка. Здатність зберігати складні та вкладені структури даних в одному документі спрощує доступ до даних та їх пошук, зменшуючи потребу в складних об'єднаннях та підвищуючи загальну продуктивність додатків.

Отже, MongoDB це провідна технологія NoSQL баз даних, пропонує потужне і гнучке рішення для ефективного і масштабованого управління даними. Документно-орієнтований підхід, гнучкий дизайн схем, висока продуктивність і надійна масштабованість роблять її привабливим вибором для сучасних проєктів. Рішення обрати MongoDB для нашого проєкту ґрунтувалося на її здатності працювати з неструктурованими даними, що швидко змінюються,

високій продуктивності та масштабованості, а також широких можливостях створення запитів.

У порівнянні з традиційними реляційними базами даних, гнучка схема, горизонтальна масштабованість і документно-орієнтована модель даних MongoDB мають значні переваги. Здатність адаптуватися до мінливих структур даних, можливості безперешкодного масштабування та всеосяжна мова запитів роблять її добре пристосованою для наших потреб розробки додатка.

2.2 Опис функціоналу системи

Сайт для оренди транспортних засобів буде працювати наступним чином: неавторизований користувач заходить на наш сайт. Користувач може обрати мову сайту, українську або англійську. Далі він може обрати локацію, з якої він хоче орендувати транспорт, а також дату початку та кінця оренди. Також користувач може зареєструватися в системі, вказавши необхідні дані такі як: електронна пошта, повне ім'я, телефон, дату народження та пароль. Після реєстрації на пошту користувача прийде повідомлення з посиланням для підтвердження. Після реєстрації та входу користувачу будуть доступні: Домашня сторінка, на якій вибір локації та дати оренди, а також сторінка із замовленнями, на якій будуть відображатись всі замовлення цього користувача. Після вибору локації та дати оренди, користувачу буде виведений список автомобілів у цій локації. На цій сторінці також доступні основні фільтри, що допомагають обрати: постачальника, тип двигуна, тип коробки переключення передач, обмеження по кілометражу та по ціні. Також поруч з фільтрами буде відображений список авто, що підходять під вибрані користувачем параметри. У картка з авто міститиме: фото транспортного засобу, його назву, характеристики, а також ціну за день та за весь вибраний період. Також у картці присутній список з цінами на додаткові послуги, так як: додатковий водій, страхування. У кінці картки є кнопка, що відповідає за вибір відповідного авто. Після того як ми натиснули цю кнопку, у нас з'явиться вікно з підтвердженням, вибором

додаткових опцій а також вибором способи оплати. У користувача також є можливість змінювати інформацію у своєму профілі, а також змінювати пароль.

У постачальника транспортних засобів після входу стає доступна панель керування, в якій він може бачити всі бронювання, їхні статуси, а також фільтри. Крім цього постачальник може редагувати кожне бронювання та змінювати його статус. Після кожної зміни статусу, користувачу відправлятиметься повідомлення на електронну скриньку про зміну статусу його бронювання. У постачальника також є доступ до списку постачальників, проте редагувати він може лише інформацію про себе. Він також має змогу створювати нові локації на англійській та українській мові. Також постачальник має доступ до списку всіх користувачів та постачальників, проте не має можливості змінювати інформацію про них. І наостанок постачальник має доступ до сторінки з транспортом. На цій сторінці схожий функціонал як і для звичайного користувача з тими відмінностями, що постачальник може лише переглядати автомобілі інших постачальників, редагувати власні автомобілі, та додавати нові авто. На сторінці додавання авто, постачальник транспорту може вказати всю необхідну інформацію, таку як: назва, фото, мінімальний вік орендаря, точки видачі авто, ціна за день, розмір задатку, тип двигуна, тип коробки перемикачів, кількість місць, кількість дверей, наявність кондиціонера, обмеження по пробігу, плата при скасуванні, ціни за страхування. Також після створення авто всі ці дані можна змінити, а також зробити авто недоступним для оренди.

Адміністратор системи для входу повинен ввести свої дані. Він має такі ж функціонал як і постачальник з тією відмінністю, що адміністратор може змінювати будь які транспортні засоби, створювати нових постачальників, редагувати всі бронювання. При створенні нового постачальника необхідно вказати повну назву та електронну пошту. Також можна додати логотип, вказати локацію, телефон та опис.

Отже підсумуємо:

2.2.1 Функціонал неавторизованого користувача

1. Вибір мови сайту (українська або англійська).
2. Вибір локації для оренди транспортного засобу.
3. Вказання дати початку та кінця оренди.
4. Реєстрація в системі з введенням необхідних даних (електронна пошта, повне ім'я, телефон, дата народження та пароль).
5. Підтвердження реєстрації через електронну пошту.
6. Доступ до домашньої сторінки з вибором локації та дати оренди.
7. Оренда транспортного засобу для водія

2.2.2 Функціонал для авторизованого користувача

1. Доступ до домашньої сторінки з вибором локації та дати оренди.
2. Сторінка замовлень, де відображаються всі замовлення користувача.
3. Відображення списку автомобілів у вибраній локації з основними фільтрами для вибору.
4. Відображення карток авто з фото, назвою, характеристиками та ціною за день та весь вибраний період.
5. Список цін на додаткові послуги, такі як додатковий водій та страхування.
6. Можливість вибору автомобіля та підтвердження замовлення.
7. Вікно з підтвердженням, вибором додаткових опцій та способу оплати.
8. Зміна інформації у профілі користувача.
9. Зміна пароля.

2.2.3 Функціонал для постачальника

1. Доступ до панелі керування після входу.
2. Відображення всіх бронювань з їх статусами та фільтрами.
3. Редагування бронювань та зміна їх статусу.

4. Відправлення повідомлень на електронну пошту користувача про зміну статусу бронювання.
5. Доступ до списку постачальників та можливість редагування власної інформації.
6. Створення нових локацій на англійській та українській мовах.
7. Доступ до списку всіх користувачів та постачальників без можливості змінювати їхню інформацію.
8. Доступ до сторінки з транспортом з можливістю перегляду автомобілів інших постачальників, редагування власних автомобілів та додавання нових авто.

2.2.3 Функціонал для адміністратора системи

1. Вхід в систему.
2. Зміна будь-яких транспортних засобів.
3. Створення нових постачальників.
4. Редагування будь-яких бронювань.
5. Редагування інформації про будь-якого постачальника

2.3 Формування вимог з використанням діаграми прецедентів

Діаграма прецедентів – це візуальне представлення, яке відображає взаємодію між системними акторами (користувачами або зовнішніми системами) і системою, що розглядається. Вона ілюструє різні варіанти використання або функціональні можливості системи, представляючи їх у чіткій і стислій формі. Діаграми варіантів використання відображають поведінку системи з точки зору користувача, демонструючи різні дії, які можуть бути виконані, і взаємозв'язки між ними.

Генерація вимог за допомогою діаграми прецедентів зосереджена на фіксації та документуванні функціональних вимог до системи з точки зору її

користувачів. Вона фокусується на розумінні учасників, їхніх цілей та взаємодії, необхідної для досягнення цих цілей. Діаграма слугує планом для проектування системи, забезпечуючи візуальне представлення, яке полегшує проектування та розробку застосунку.

Процес створення діаграми прецедентів передбачає збір інформації від зацікавлених сторін, включаючи кінцевих користувачів. Цей підхід допомагає виявити і зафіксувати основні вимоги, які в іншому випадку могли б залишитися поза увагою. Він сприяє активній участі і гарантує, що система відповідає потребам і очікуванням її цільових користувачів.

Діаграми прецедентів допомагають визначити межі і сферу застосування системи, визначаючи учасників і їхні відповідні цілі. Вони надають загальний огляд функціональних можливостей системи, що дозволяє проектним групам визначати пріоритети вимог, ефективно розподіляти ресурси та керувати обмеженнями проекту. Вони також встановлюють простежуваний зв'язок між вимогами користувачів і функціональністю системи. А також дозволяють нам відстежувати і контролювати хід реалізації, гарантуючи, що всі визначені варіанти використання враховані і виконані в процесі розробки.

Враховуючи всі попередньо наведені вимоги та функції, необхідно побудувати діаграму прецедентів, для виявлення функціональних можливостей системи, а також поведінку системи з точки зору користувача. Акторами у наступних діаграмах будуть: неавторизований користувач, авторизований користувач, постачальник, адміністратор системи.

На рисунку 2.1 зображено функціональні можливості неавторизованого користувача з нашим сайтом.



Рис. 2.1. Діаграма прецедентів взаємодії неавторизованого користувача і сайту

На рисунку 2.2 зображено функціональні можливості авторизованого користувача з сайтом. Для цього типу користувача функціональних можливостей стає більше, адже нам це дозволяє прив'язати всі введені користувачем дані до його профілю. А на рисунках 2.3-2.4 зображені способи використання сайту постачальником та адміністратором системи відповідно.

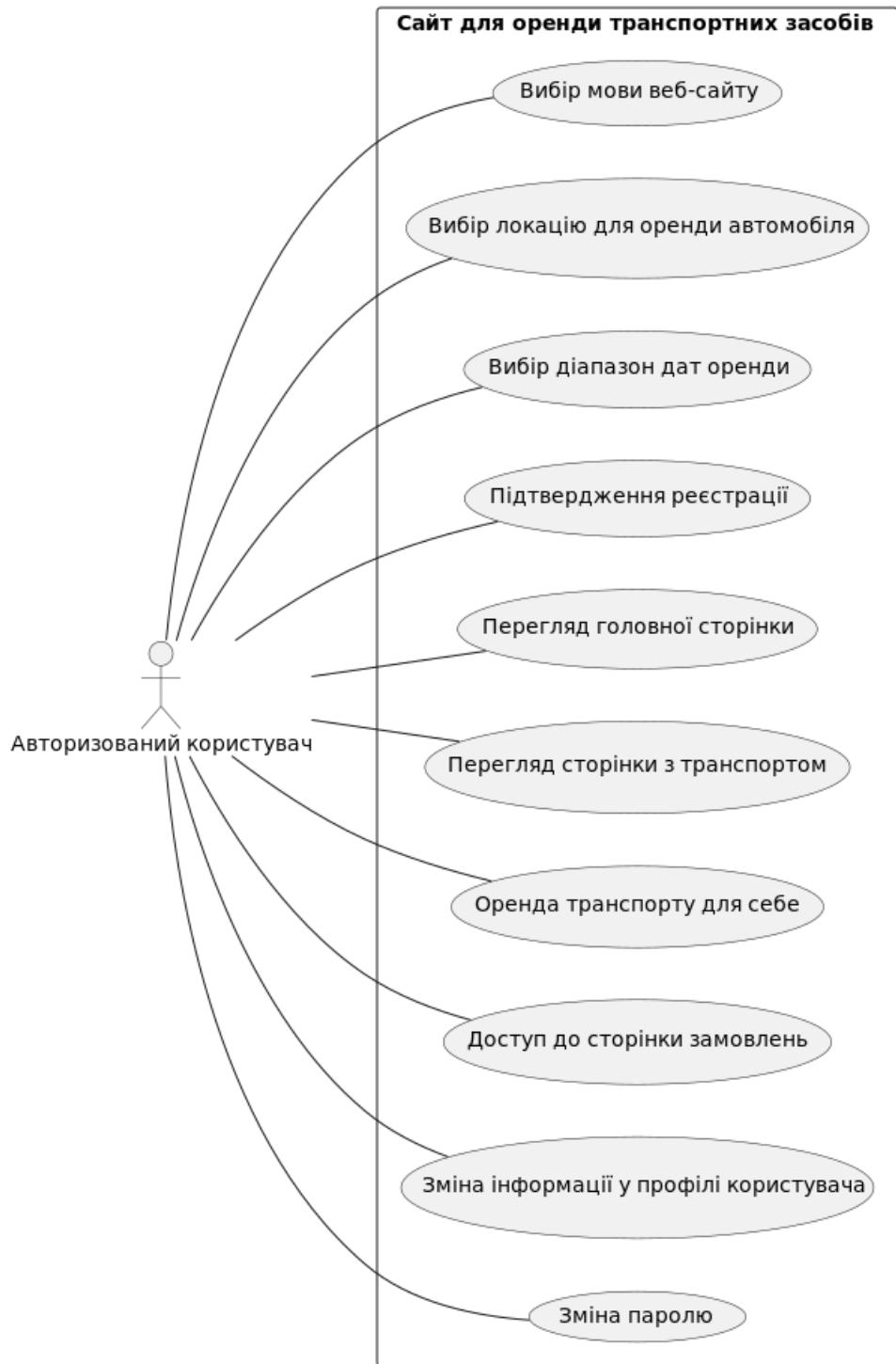


Рис. 2.2. Діаграма прецедентів взаємодії авторизованого користувача і сайту

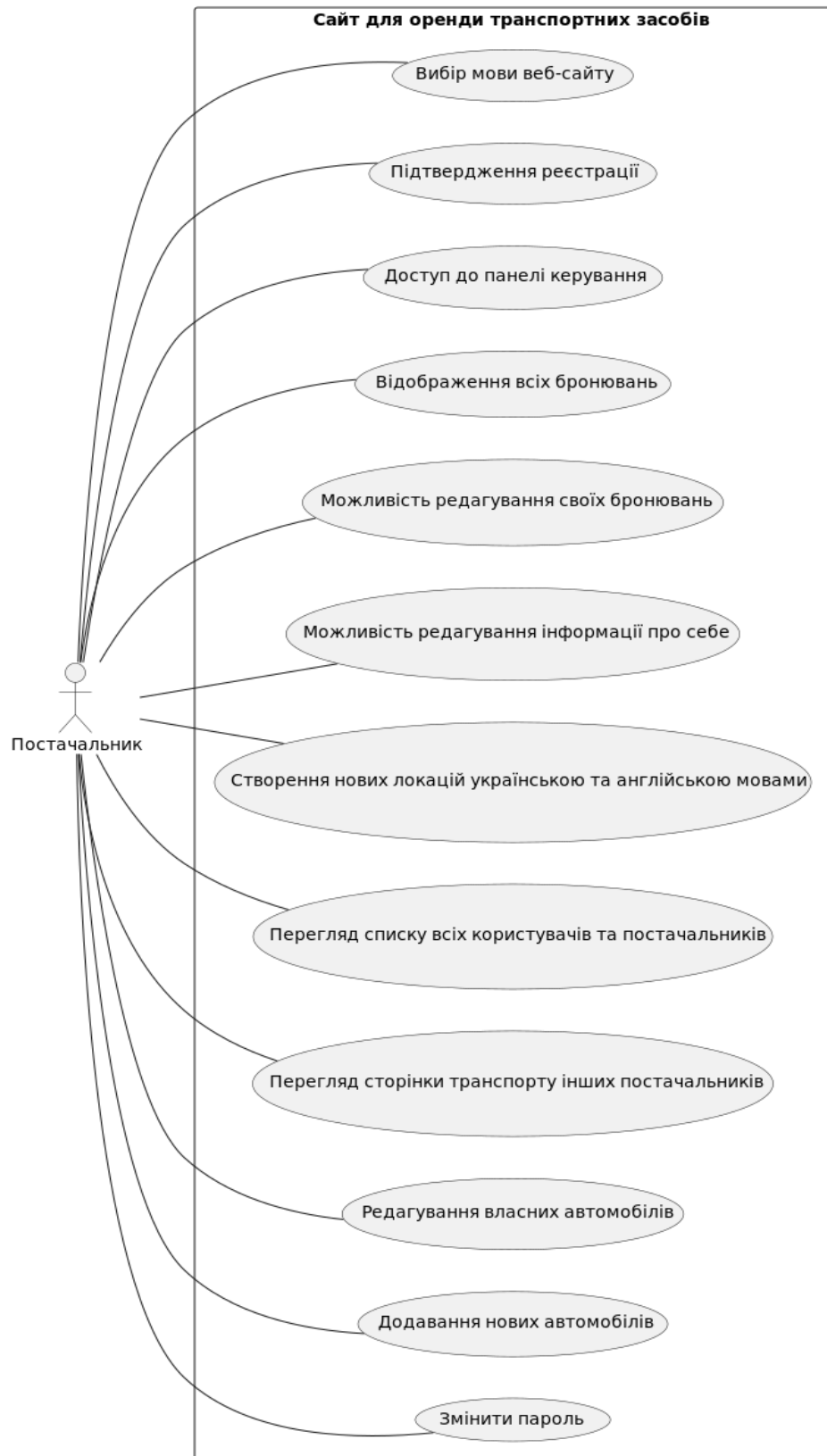


Рис. 2.3. Діаграма прецедентів взаємодії постачальника і сайту

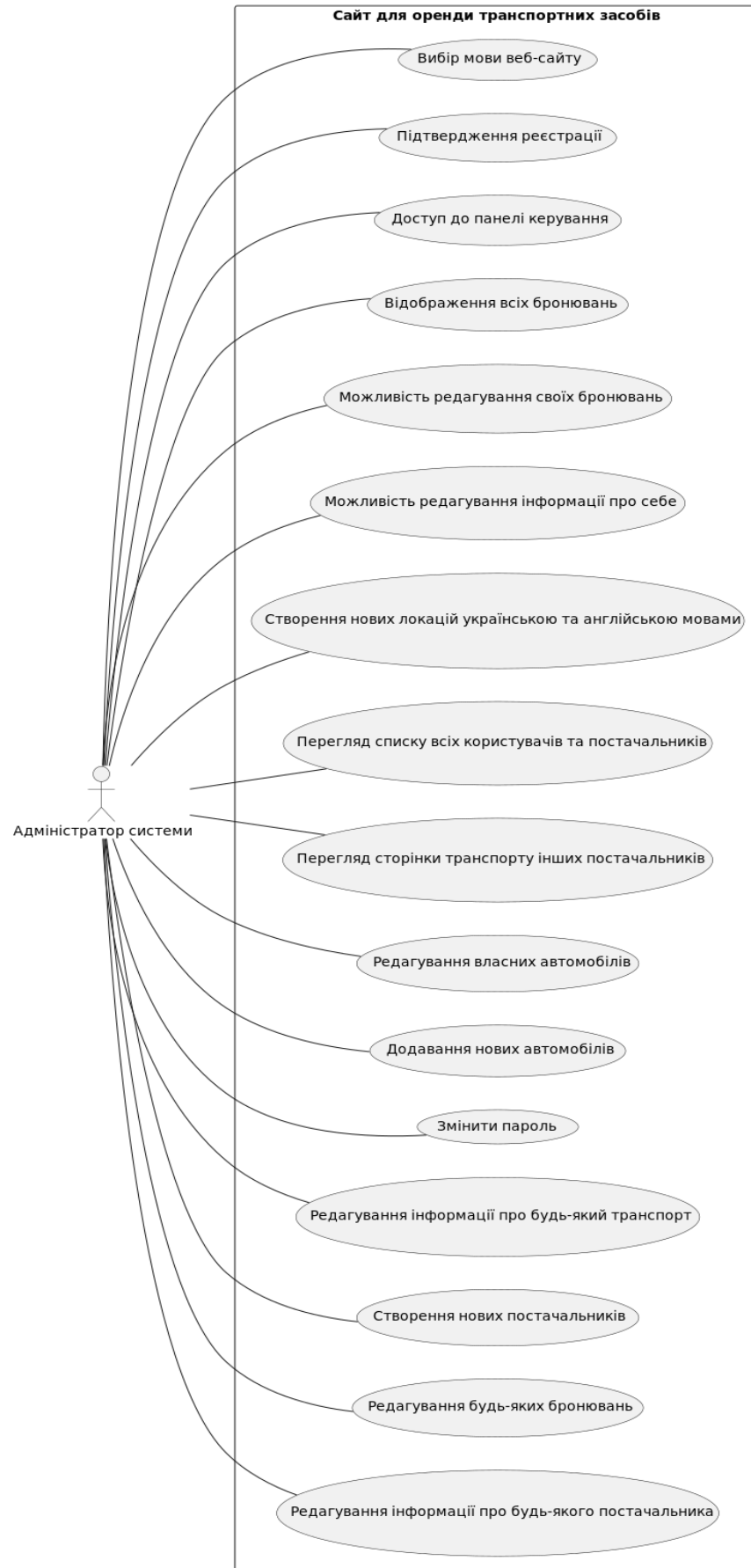


Рис. 2.4. Діаграма прецедентів взаємодії адміністратора системи і сайту

2.4 Проектування архітектури сайту

Архітектура сайту для оренди транспортних засобів відіграє ключову роль у забезпеченні його функціональності, масштабованості та ефективності.

Фронтенд частина для клієнта формує користувацький компонент додатку для оренди транспортних засобів. Вона складається з візуально привабливого та інтуїтивно зрозумілого інтерфейсу, який дозволяє користувачам шукати, переглядати та бронювати транспорт. Цей інтерфейсний компонент використовує сучасні веб-технології, такі як HTML, CSS та JavaScript, та з використанням React фреймворку, для забезпечення динамічного та інтерактивного користувацького інтерфейсу. Користувацький інтерфейс дозволяє клієнтам переглядати деталі автомобіля, обирати дати оренди та продовжувати процес бронювання.

Інтерфейс адміністратора та постачальника задовольняє специфічні потреби та ролі цих типів користувачів у додатку для прокату транспортних засобів. Інтерфейс адміністратора надає функції для управління обліковими записами користувачів, списком транспортних засобів та ціноутворенням. Він пропонує інтуїтивно зрозумілу інформаційну панель з доступом до різних адміністративних завдань і функцій візуалізації даних для моніторингу та аналізу продуктивності системи. Інтерфейс постачальника, з іншого боку, полегшує управління списками транспортних засобів, їх доступністю та оновленнями. Він дозволяє постачальникам додавати нові транспортні засоби, оновлювати дані про оренду та відстежувати статус своїх транспортних засобів у системі.

Бекенд API слугує основою додатку для оренди транспортних засобів, керуючи логікою та бізнес-процесами. Він відповідає за обробку запитів користувачів, виконання необхідних перевірок, взаємодію з базою даних і координацію потоку даних між фронтенд частинами і базою даних. Внутрішній API розроблений з використанням серверної технології, такої як Node.js, яка пропонує керований подіями, неблокуючий ввід/вивід, що робить його ідеальним для створення масштабованих та ефективних веб-додатків. API

використовує архітектурний стиль REST (Representational State Transfer - передача репрезентативного стану) для забезпечення стандартизованого протоколу зв'язку між фронтендом і сервером, що не залежить від стану. Це забезпечує безперешкодну інтеграцію з різними клієнтськими додатками та дозволяє легко масштабувати та обслуговувати їх.

Обране рішення з використанням фронтенду для звичайних користувачів, адміністраторів та постачальників, разом із внутрішнім API та базою даних MongoDB, має декілька переваг. По-перше, воно забезпечує безперебійну роботу користувачів, пропонуючи адаптовані інтерфейси для різних типів користувачів, що дозволяє їм ефективно виконувати свої завдання. По-друге, модульний підхід до розробки дозволяє окремим командам одночасно працювати над зовнішнім і внутрішнім компонентами, що сприяє пришвидшенню циклів розробки та полегшенню обслуговування. Нарешті, використання MongoDB в якості бази даних забезпечує гнучкість і масштабованість, необхідну для управління різноманітними даними, що постійно змінюються, пов'язаними з додатком для оренди транспортних засобів.

На рисунку 2.5. зображена діаграма архітектури сайту для оренди транспортних засобів.

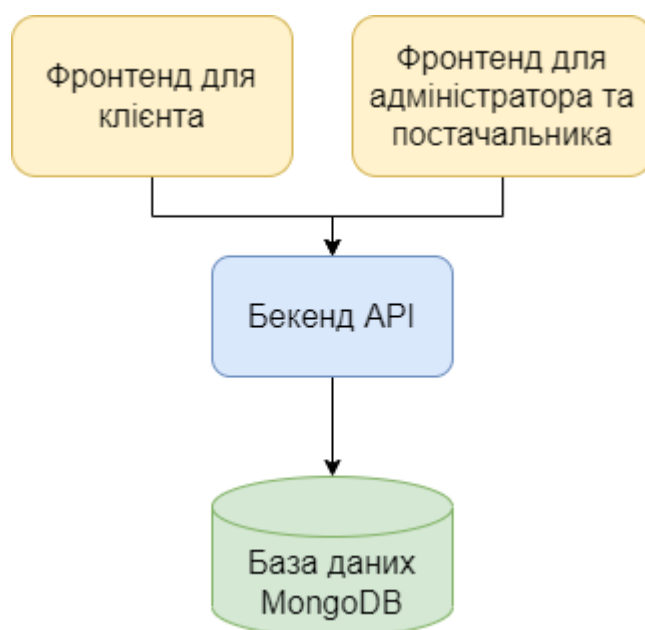


Рис. 2.5. Діаграма архітектурної структури сайту

2.5 Архітектура бекенд частини

Проектування і реалізація добре структурованої архітектури бекенда є ключовими у розробці ефективних і масштабованих веб-додатків. Архітектурний патерн MVC отримав значне визнання завдяки своїй здатності покращувати організацію коду, розподіл завдань і зручність супроводу. На рисунку 2.6. зображена діаграма MVC шаблону.



Рис. 2.6. Діаграма MVC патерну.

Архітектура MVC розділяє додаток на три взаємопов'язані компоненти: модель (Model), представлення (View) і контролер (Controller). Модель представляє дані та бізнес-логіку, інкапсулюючи операції з базою даних, маніпуляції з даними та валідацію. Представлення відповідає за рендеринг користувацького інтерфейсу і представлення даних користувачам. Нарешті, Контролер діє як посередник, отримуючи і обробляючи запити користувачів, оновлюючи Модель і даючи вказівки Представленню відобразити відповідну інформацію.

Використання архітектури MVC для бекенд частини виправдано декількома факторами. По-перше, MVC сприяє розділенню завдань, дозволяючи нам ізолювати різні аспекти програми і зосередитися на конкретних функціональних можливостях. Такий поділ покращує організацію коду, зручність обслуговування та повторне використання, зменшуючи заплутаність коду та полегшуючи виявлення та вирішення проблем.

По-друге, MVC пропонує модульність, що дозволяє незалежну розробку та тестування компонентів моделі, представлення та контролера. Ця модульність покращує повторне використання коду, оскільки кожен компонент може бути використаний у різних частинах програми, мінімізуючи дублювання коду та підвищуючи ефективність розробки.

Крім того, архітектура MVC добре узгоджується з асинхронною природою Node.js, що робить її придатною для ефективної обробки паралельних запитів. Node.js використовує модель вводу/виводу, керовану неблокуючими подіями, а компонент Контролер слугує центральним координатором для управління потоком подій та даних. Це дозволяє розробляти високомасштабовані та швидко реагуючі внутрішні API, здатні обробляти високі навантаження трафіку.

Використання архітектури MVC у бекенд API Node.js має кілька переваг. По-перше, розділення завдань полегшує підтримку коду, оскільки модифікації або вдосконалення одного компонента не впливають на інші. Таке розділення також покращує читабельність коду, полегшуючи розуміння та налагодження різних функціональних можливостей додатку.

По-друге, модульність MVC сприяє повторному використанню коду, скорочуючи час і зусилля на розробку. Моделі, представлення та контролери можуть бути розроблені незалежно і повторно використані в різних частинах програми, що підвищує ефективність та узгодженість розробки.

Крім того, MVC сприяє тестуванню, дозволяючи тестувати кожен компонент ізольовано. Це гарантує, що зміни, внесені в одну область, не вплинуть ненавмисно на інші частини програми, що призводить до підвищення надійності та якості.

2.6. Висновки другого розділу

У другому розділі "Проектування сайту" було розглянуто ключові аспекти процесу проектування додатку для оренди транспортних засобів. В цьому розділі

було детально розглянуто вимоги та функціонал, які повинні бути реалізовані в системі.

Основна мета цього розділу полягала в проектуванні оптимальної структури та архітектури додатку, що забезпечує зручність використання для користувачів, адміністраторів та постачальників. Розробка фронтенд частини для різних типів користувачів, а також бекенд апі частини та бази даних MongoDB були обґрунтовані вибором технологій та архітектурних рішень.

Проектування сайту вимагало ретельного врахування функціональних та нефункціональних вимог, а також дослідження інтерфейсів та взаємодії між користувачами та системою. Було враховано особливості вибору локалізації, мови програмування, а також технологій, які допоможуть досягти бажаної продуктивності та ефективності.

У результаті проектування сайту була створена загальна архітектура та архітектура бекенд частини, які забезпечують високий рівень функціональності, зручності використання та масштабованість системи. Вибір такої архітектури був обґрунтований відповідно до вимог проекту та досліджень відповідних технологій.

Завершуючи другий розділ, можна зазначити, що успішне проектування сайту є важливим етапом у розробці додатку для оренди транспортних засобів. Врахування всіх вимог та забезпечення оптимальної архітектури є гарантією ефективною реалізації функціональності та задоволення потреб користувачів.

РОЗДІЛ 3

РОЗРОБКА САЙТУ

3.1 Розробка структури бази даних

Розробка веб-сайту з прокату автомобілів вимагає добре продуманої та ефективної структури бази даних для зберігання і управління різними сутностями та їхніми зв'язками. Для роботи з MongoDB базой на нашому сервері будемо використовувати Mongoose.

Mongoose - це популярна бібліотека об'єктного моделювання даних (ODM) для Node.js та MongoDB. Вона забезпечує простий спосіб взаємодії з MongoDB шляхом перетворення даних в базі даних в об'єкти JavaScript і навпаки. Mongoose дозволяє розробникам визначати схеми, які є планами структури даних, що зберігаються в MongoDB, і надає потужні функції для запитів, валідації та проміжного програмного забезпечення.

Буде розглянуто схему MongoDB для кількох ключових сутностей, а саме:

- Користувач
- Додатковий водій
- Бронювання
- Автомобіль
- Місцезнаходження
- Значення місцезнаходження(для локалізації)
- Сповіщення
- Лічильник сповіщень
- Повідомлення пуш сповіщення
- Токен

На рисунку 3.1. зображена діаграма бази даних з усіма вищепереліченими сутностями та зв'язками між ними:

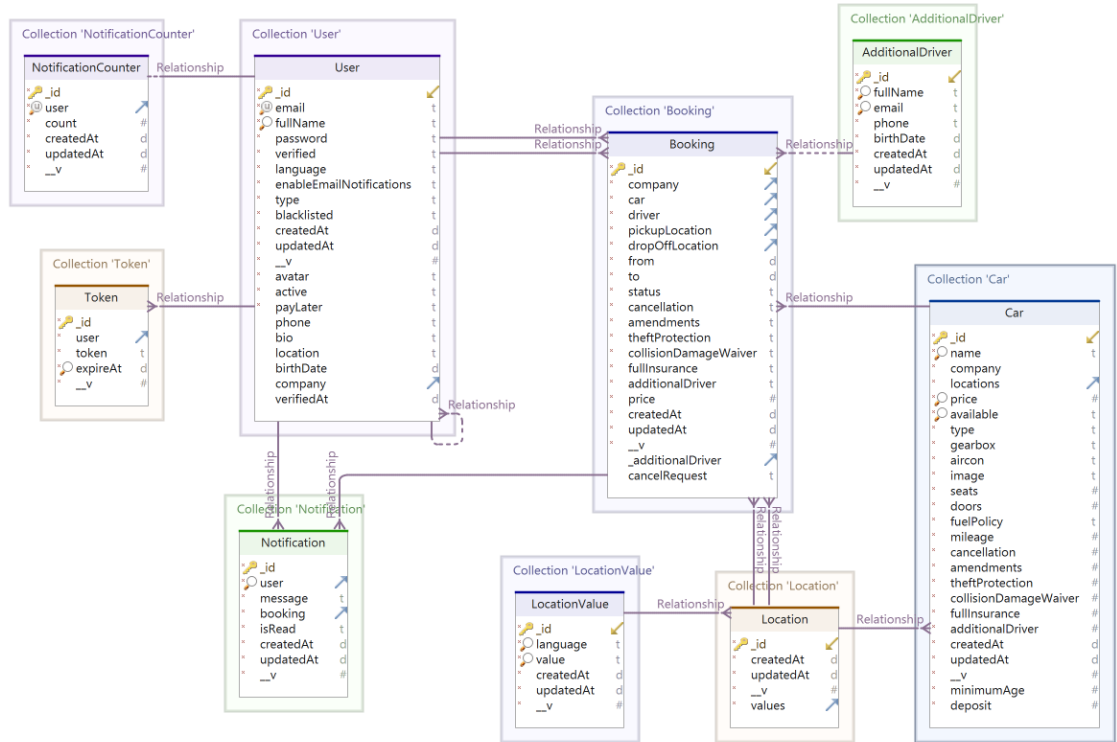


Рис. 3.1. Діаграма зв'язку сутностей бази даних

У наступному прикладі коду ми описуємо схему та її поля, а також валідацію для цих полів. Схожим чином створюються й інші сутності в базі даних.

```
const Schema = mongoose.Schema;

const locationSchema = new Schema(
  {
    values: {
      type: [Schema.Types.ObjectId],
      ref: "LocationValue",
      validate: (value) => Array.isArray(value) && value.length > 1
    }
  },
  {
    timestamps: true,
    strict: true,
    collection: "Location"
  }
);
```

Рис. 3.2. Опис в коді сутності бази

3.2 Реалізація програми

Для коректного налаштування проекту, а також передачу необхідних параметрів ми використовуємо .env файли. Файли .env (оточення) використовуються для зберігання параметрів конфігурації та налаштувань у програмах. Це текстові файли, які містять пари ключ-значення, де кожен рядок представляє окремий параметр. Ці знаходяться в кореневому каталозі бекенд та фронтенд частин і містять конфіденційну інформацію, таку як паролі, ключі API, URL-адреси баз даних та інші параметри, які можуть змінюватися залежно від середовища. Наприклад, середовище розробки, середовище тестування, локальне середовище.

Основна мета використання файлів .env - відокремити конфіденційні дані та конфігурацію від вихідного коду програми. Це забезпечує безпеку та зручне керування конфігурацією, оскільки файли .env можна легко налаштувати для кожного середовища окремо, не змінюючи код програми. На рисунку 3.3. зображений приклад значень файлу:

```
JWT_SECRET = SECRET;  
JWT_EXPIRE_AT = 86400;
```

Рис. 3.3. Значення .env файлу

Під час виконання програми значення з .env-файлів можна завантажити в середовище виконання (наприклад, прочитати в змінні середовища), щоб їх можна було використовувати в коді програми. Приклад зчитування значення зображений на рисунку 3.4.

```
const JWT_SECRET = process.env.BC_JWT_SECRET;  
const JWT_EXPIRE_AT = parseInt(process.env.BC_JWT_EXPIRE_AT);
```

Рис. 3.4. Зчитування значень з .env файлу та запис у змінні

Використання файлів .env спрощує налаштування та управління конфігурацією, дозволяючи легко змінювати значення безпосередньо у файлі

.env, не впливаючи на код програми. Це полегшує розгортання, обслуговування та дозволяє зберігати конфіденційну інформацію в безпечному місці, що є важливим аспектом безпеки додатків.

Розділ автентифікації на сайті відповідає за надання безпечного доступу користувачів і захист конфіденційної інформації. Він включає в себе такі процедури, як хешування пароля під час реєстрації та видачу веб-токенів JSON (JWT) для цілей автентифікації.

Під час процесу реєстрації, коли користувач створює обліковий запис, його пароль надійно хешується перед тим, як зберігається в базі даних. Хешування пароля - це одностороння криптографічна функція, яка перетворює пароль у вигляді звичайного тексту на незворотний рядок символів. Цей процес гарантує, що навіть якщо база даних буде скомпрометована, оригінальні паролі неможливо буде легко отримати. Для цього використовується алгоритми хешування bcrypt, завдяки його високим захисним властивостям. На рисунку 3.5. зображено програмну реалізацію хешування при реєстрації користувача.

```
const salt = await bcrypt.genSalt(10);
const password = req.body.password;
const passwordHash = await bcrypt.hash(password, salt);
user.password = passwordHash;

user.active = true;
user.verified = true;
user
  .save()
  .then(() => {
    return res.sendStatus(200);
  })
  .catch((err) => {
    console.error(strings.DB_ERROR, err);
    return res.status(400).send(strings.DB_ERROR + err);
  });
```

Рис. 3.5. Хешування паролю при реєстрації користувача, та збереження в базу даних.

Коли користувач успішно входить на сайт, застосовується механізм аутентифікації за допомогою JWT. JWT - це компактний, URL-безпечний формат токенів, який безпечно передає інформацію між сервером та користувачем. Після успішної аутентифікації генерується токен JWT, який видається користувачеві. Цей токен містить закодовану інформацію, таку як ідентифікатор або роль користувача, і підписується секретним ключем, відомим лише серверу. Цей підпис забезпечує цілісність токена і запобігає його підробці.

Токен JWT включається в заголовок авторизації(x-access-token) наступних запитів, які робить користувач. Це надає аутентифікацію без збереження стану, оскільки сервер може перевірити цілісність токена і витягти з нього ідентифікатор користувача без необхідності запитувати базу даних при кожному запиті. Час закінчення терміну дії токена можна встановити, щоб забезпечити дійсність сеансу і підвищити безпеку.

Для перевірки вхідних токенів JWT сервер перевіряє підпис токена за допомогою секретного ключа. Якщо підпис дійсний, сервер може довіряти вмісту токена і надавати відповідному користувачеві доступ до захищених ресурсів. Якщо токен недійсний або термін його дії закінчився, сервер відмовляє в доступі і вимагає від користувача повторної автентифікації.

У контексті веб-розробки проміжне програмне забезпечення – це компонент або функція, яка знаходиться між клієнтським і серверним компонентами програми. Він перехоплює вхідні запити і виконує певні дії або перевірки, перш ніж передати запит наступному компоненту в конвеєрі. Для перевірки токена створюється проміжна функція(middleware), що перевіряє токен на актуальність та на правильність підпису, якщо токен не коректний то перевірка не пропускає далі. Реалізація такої функції зображена на рисунку 3.6.

```

const verifyToken = (req, res, next) => {
  const token = req.headers["x-access-token"];

  if (!token) {
    return res.status(403).send({ message: "No token provided!" });
  }

  jwt.verify(token, JWT_SECRET, (err, decoded) => {
    if (err) {
      console.log(err);
      return res.status(401).send({ message: "Unauthorized!" });
    }

    req.userId = decoded.id;
    next();
  });
};

```

Рис. 3.6. Проміжна функція для перевірки JWT токена.

Для пошуку та фільтрації транспорту по вибраних параметрах у нас використовується контролер `getCars`. У тіло запита передаються всі параметри, по яким необхідно відфільтрувати транспорт. Для фільтрації використовується об'єкт `$match`, у якому описуються основні параметри та логічні оператори, для фільтрації. На рисунку 3.7. зображений об'єкт для передачі фільтрів:

```

const $match = {
  $and: [
    { name: { $regex: keyword, $options: options } },
    { company: { $in: companies } }
  ]
};

if (fuel) {
  $match.$and.push({ type: { $in: fuel } });
}

if (gearbox) {
  $match.$and.push({ gearbox: { $in: gearbox } });
}

```

Рис. 3.7 об'єкт фільтрації транспорту

Перевагою такого підходу є те що ми можемо наповнювати наш об'єкт різними умовами, по яким будемо відфільтровувати авто. Також ми можемо додавати обмеження в залежності від умов. Ідея такого фільтрування в тому, що ми не фільтруємо всю колекцію авто на серверній частині, адже їх може бути багато, а вивантаження всіх даних з бази може займати багато часу. На серверній частині ми лише формуємо умови, по яким потрібно відфільтрувати нашу колекцію, а далі передаємо цей об'єкт базі даних, де вона використовуючи оптимальні алгоритми відсортує та поверне нам очікуваний результат.

Для структурування та розбиття коду на логічні модулі у нас використовується маршрутизація запитів. Маршрутизація - це процес визначення способу обробки вхідного HTTP-запиту на основі його URL-адреси та HTTP-методу. Маршрутизація зіставляє певні URL-адреси з відповідними контролерами, які повинні бути виконані для обробки запиту. Він служить мостом між запитом клієнта і відповідним кодом, який згенерує відповідь. Маршрути визначаються у коді програми. Приклад на рисунку 3.8.

```
const routes = express.Router();

routes.route(routeNames.validate).post(authJwt.verifyToken, companyController.validate);
routes.route(routeNames.update).put(authJwt.verifyToken, companyController.update);
routes
  .route(routeNames.delete)
  .delete(authJwt.verifyToken, companyController.deleteCompany);
routes
  .route(routeNames.getCompany)
  .get(authJwt.verifyToken, companyController.getCompany);
routes
  .route(routeNames.getCompanies)
  .get(authJwt.verifyToken, companyController.getCompanies);
routes.route(routeNames.getAllCompanies).get(companyController.getAllCompanies);
```

Рис. 3.8. Реєстрація маршрутів для обробки запитів по компаніям.

Кожен маршрут складається з шаблону URL-адреси (або шаблону маршруту) і відповідним методом контролера. Шаблон URL-адреси може містити заповнювачі для динамічних сегментів URL-адреси, таких як параметри або ідентифікатори ресурсів. Шаблон URL-адрес зображений на рисунку 3.9.

```

export default {
  validate: "/api/validate-company",
  update: "/api/update-company",
  delete: "/api/delete-company/:id",
  getCompany: "/api/company/:id",
  getCompanies: "/api/companies/:page/:size",
  getAllCompanies: "/api/all-companies"
};

```

Рис. 3.9. Шаблон URL-адреси

Наш сервіс використовує мережу доставки контенту(CDN - Content Delivery Network) для зберігання та ефективної доставки зображень нашим користувачам. Основна мета використання CDN це оптимізація доставки статичних ресурсів користувачам. Також за допомогою CDN ми можемо масштабувати наш сайт, виділяючи сервери зі статичним контентом(такі як картинки) ближче до локації користувача, тим самим збільшуючи швидкість доставки, а також ми не навантажуюмо наш API сервіс. Процес зберігання складається з наступних кроків:

- Налаштування співставлення папки сервера та шляху в посиланні
- Завантаження та зберігання зображень у виділену директорію
- Інтеграція фронтенд сервісів з CDN сервером.
- Отримання зображень.

На рисунку 3.10. Зображено директорії для збереження фото, а на рисунку 3.11. отримання зображення зі CDN серверу:

```

CDN_USERS = /var/www/cdn/ridebooker/users
CDN_TEMP_USERS = /var/www/cdn/ridebooker/temp/users
CDN_CARS = /var/www/cdn/ridebooker/cars
CDN_TEMP_CARS = /var/www/cdn/ridebooker/temp/cars

```

Рис. 3.10. Директорії для збереження фото транспорту та користувачів


```

<MaterialAvatar
  src={Helper.joinURL(Env.CDN_USERS, user.avatar)}
  className="avatar"
/>

```

Рис. 3.11. Відображення картинки профілю користувача передаючи посилання на картинку.

Для адаптації сайту до конкретних культурних, мовних і регіональних уподобань користувачів з різних цільових ринках на сайті присутня локалізація. Вона включає в себе переклад і кастомізацію контенту, елементів інтерфейсу користувача та інших аспектів, щоб зробити веб-сайт більш релевантним і зручним для користувачів у різних регіонах. Локалізація необхідна для охоплення глобальної аудиторії та забезпечення інклюзивного користувацького досвіду. Вона виходить за рамки простого перекладу мови і охоплює різні елементи, такі як формати дати і часу, валютні символи та культурні норми. Хоча наш сайт розробляється для українського ринку, проте для його актуальності у сфері туризму нам необхідно додати англійську мову. Щоб реалізувати локалізацію, для відображення інтерфейсу ми використовуємо спеціальні об'єкти, що передаються у розмітку для відображення контенту. На рисунку 3.12. зображений такий об'єкт.

```

export const strings = new LocalizedStrings({
  uk: {
    ACTIVATE_HEADING: "Активация облікового запису",
    TOKEN_EXPIRED: "Термін дії посилання для активации облікового запису минув.",
    ACTIVATE: "Активувати"
  },
  en: {
    ACTIVATE_HEADING: "Account Activation",
    TOKEN_EXPIRED: "Your account activation link expired.",
    ACTIVATE: "Activate"
  }
});

```

Рис. 3.12. Об'єкт локалізації інтерфейсу

Як ми бачимо відбувається створення об'єктів для різних локалізацій з однаковими ключами та локалізованими значеннями. Такий підхід нам дозволяє не лише локалізувати сайт, а й змінювати текст не втручаючись у розмітку сторінок, що пришвидшує розробку та виправлення помилок.

3.3. Огляд інтерфейсу

Робота користувача з сайтом починається з головної сторінки. На ній користувач може обрати локації з якої він хоче забрати авто та куди його доставить, а також дати, на які йому це авто потрібне. На рисунку 3.13. зображена головна сторінка.

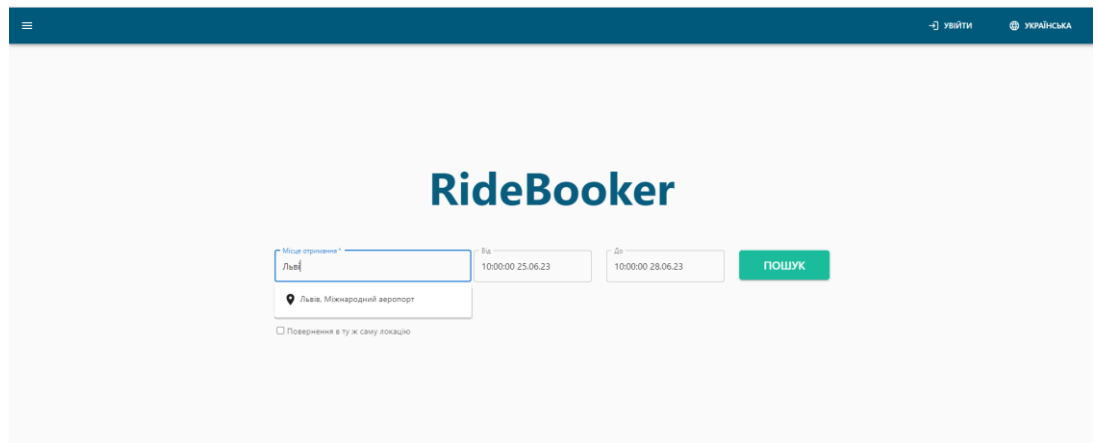


Рис. 3.13. Головна сторінка

Також на ній користувач може обрати локалізацію сайту рисунок 3.14, а також увійти рисунок 3.15.

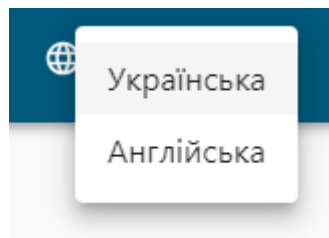
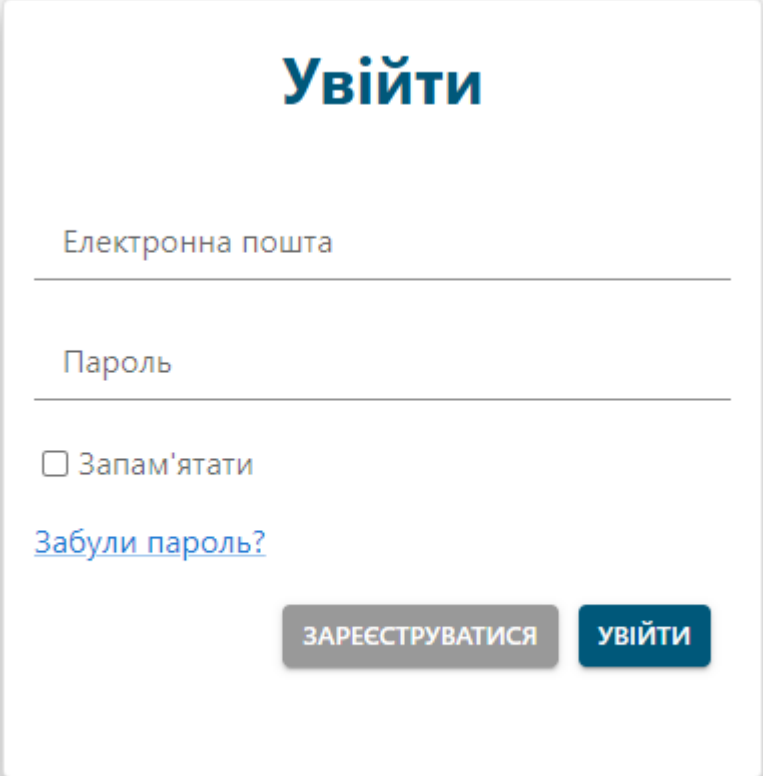


Рис. 3.14. Список вибору локалізації



Увійти

Електронна пошта

Пароль

Запам'ятати

[Забули пароль?](#)

ЗАРЕЄСТРУВАТИСЯ **УВІЙТИ**

Рис. 3.15. Вікно для входу в систему

Також якщо у користувача немає акаунту, він може його створити натиснувши кнопку «ЗАРЕЄСТРУВАТИСЬ» рисунок 3.16.

Зареєструватися

📅

[Я прочитав та погоджуюся з Умовами використання.](#)

ЗАРЕЄСТРУВАТИСЯ
СКАСУВАТИ

Рис. 3.16. Вікно для реєстрації

Після того як ми виберемо локації та натиснемо кнопку пошук, нас переведе на сторінку вибору авто. На ній відображені всі автомобілі що знаходяться у відповідних локаціях, кнопки для вибору авто, характеристики авто, а також фільтрація по характеристиках та постачальниках.

The screenshot shows a car rental website interface. On the left, there are search filters for pickup location (Lviv International Airport), pickup time (10:00:00 on 25.06.23), and drop-off time (10:00:00 on 28.06.23). Below these are checkboxes for 'Повернення в ту ж саму локацію' and a list of rental companies (Alamo, Europcar, Keddy, Payless, United, Dollar, Hertz, Menchynski, Thrifty) with checkboxes. A 'ПОШУК' button is visible. The main area displays two car listings:

- Fiat 500**: Price for 3 days is 1080 грн (360 грн per day). Features include: 5 seats, manual transmission, 4 doors, 2 doors, and a sunroof. It lists 'Пробіг: без обмежень', 'Політика палива: Повнакомплектційне паливо', 'Скасування: Включено', and 'Злини: Включено'. Insurance options include 'Страховання від ушкоджень при зіткненні: 100 грн/день', 'Захист від крадіжки: 150 грн/день', 'Повне страхування: 200 грн/день', and 'Додатковий водій: 220 грн/день'. A 'ВИБРАТИ ЦЕЙ АВТОМОБІЛЬ' button is present.
- Fiat Panda**: Price for 3 days is 870 грн (290 грн per day). Features include: 5 seats, manual transmission, 4 doors, and 4 doors. It lists 'Пробіг: без обмежень' and 'Політика палива: Повнакомплектційне паливо'.

Рис. 3.17. Вікно вибору авто

Після перегляду всього наявного транспорту ми можемо обрати будь-який та нас переведе на сторінку бронювання рисунок 3.18 та 3.19. У цьому вікні ми можемо обрати додаткові опції, а також побачити деталі нашого бронювання.

Забронювати Зараз

🚗 Варіанти бронювання

- Скасування** Включено
- Зміни** Включено
- Страхування від ушкоджень при зіткненні** + 300 грн (100 грн/день)
- Захист від крадіжки** + 450 грн (150 грн/день)
- Повне страхування** + 600 грн (200 грн/день)
- Додатковий водій** + 660 грн (220 грн/день)

🚗 Деталі вашого бронювання


| | |
|------------------------|--|
| Дні | 3 дні (10:00:00 25.06.23 - 10:00:00 28.06.23) |
| Місце отримання | Львів, Міжнародний аеропорт |
| Місце доставки | Львів, Міжнародний аеропорт |
| Автомобіль | Fiat 500 (360 грн/день) |
| Постачальник |  Unifed |
| ВАРТІСТЬ | 1080 грн |

Рис. 3.18. Вікно бронювання

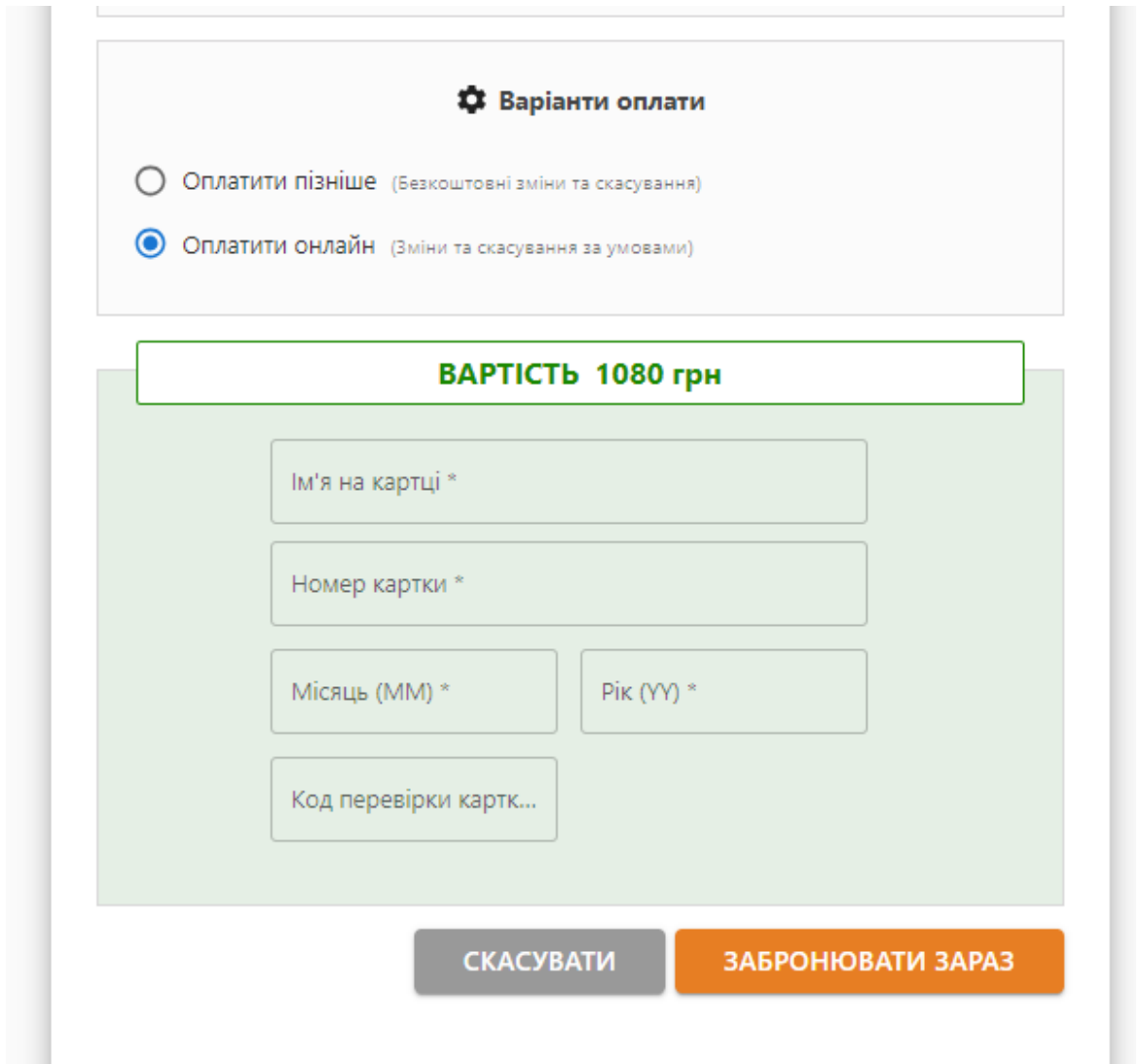
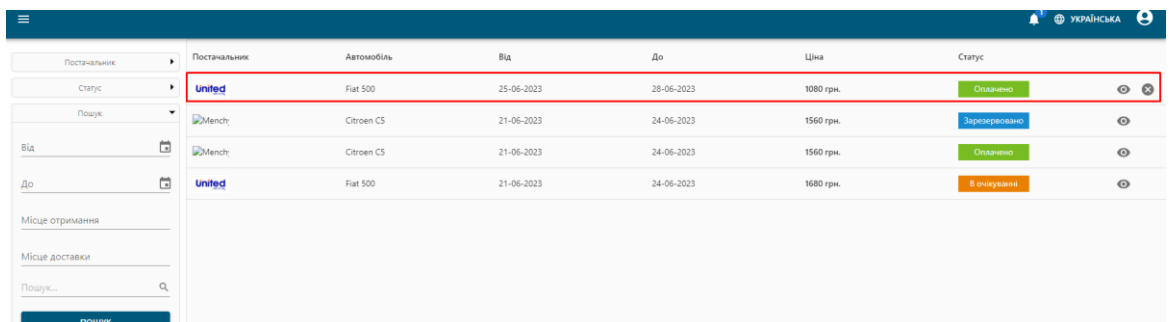


Рис. 3.19. Способи оплати нашого бронювання.

Після вибору всіх опцій, способу оплати та введення інформації, необхідної для оплати інформації, ми можемо натиснути кнопку бронювання.

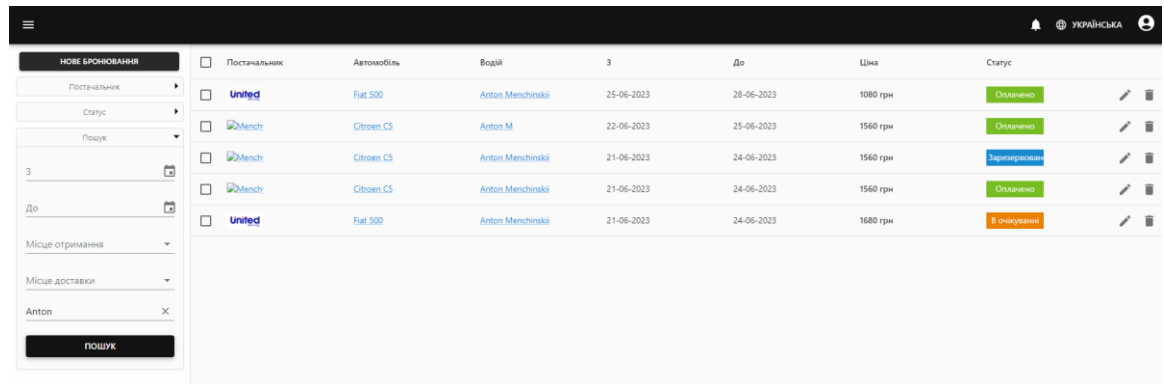
Далі нам прийде інформація про наше бронювання на електронну пошту, а також ми можемо переглянути наші бронювання в окремому вікні:



| Постачальник | Автомобіль | Від | До | Ціна | Статус |
|--------------|------------|------------|------------|-----------|---------------|
| united | Fiat 500 | 25-06-2023 | 28-06-2023 | 1080 грн. | Оплачено |
| Mench | Citroen C5 | 21-06-2023 | 24-06-2023 | 1560 грн. | Зарезервовано |
| Mench | Citroen C5 | 21-06-2023 | 24-06-2023 | 1560 грн. | Оплачено |
| united | Fiat 500 | 21-06-2023 | 24-06-2023 | 1680 грн. | В очікуванні |

Рис. 3.20. Вікно бронювань.

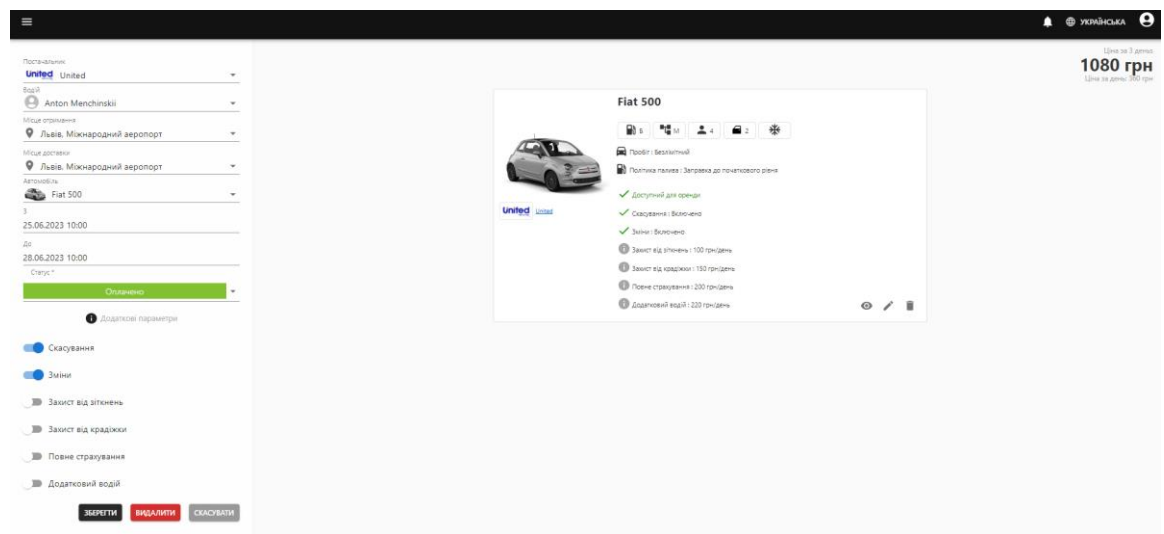
Також у нашій системі є окрема фронтенд частина для адміністраторів та постачальників. Після входу в неї ми бачимо панель управління, де у нас відображаються всі бронювання до яких у нас є доступ.



| Постачальник | Автомобіль | Водій | З | До | Ціна | Статус |
|--------------|------------|------------------|------------|------------|----------|---------------|
| United | Fiat 500 | Anton Menshinski | 25-06-2023 | 28-06-2023 | 1080 грн | Оплачено |
| Mench | Citroen CS | Anton M | 22-06-2023 | 25-06-2023 | 1560 грн | Оплачено |
| Mench | Citroen CS | Anton Menshinski | 21-06-2023 | 24-06-2023 | 1560 грн | Зарезервовано |
| Mench | Citroen CS | Anton Menshinski | 21-06-2023 | 24-06-2023 | 1560 грн | Оплачено |
| United | Fiat 500 | Anton Menshinski | 21-06-2023 | 24-06-2023 | 1680 грн | В очікуванні |

Рис. 3.21. Вікно управління бронюваннями

Після того як ми ввійдемо в режим редагування ми побачимо всі поля бронювання, які можемо змінити та інформацію про авто, яку ми також можемо відредагувати.



Постачальник: United
 Водій: Anton Menshinski
 Місце отримання: Львів, Міжнародний аеропорт
 Місце доставки: Львів, Міжнародний аеропорт
 Автомобіль: Fiat 500
 З: 25.06.2023 10:00
 До: 28.06.2023 10:00
 Статус: Оплачено

1080 грн
 Ціна за день, 100 грн

Fiat 500
 Провід і безпечний
 Постачка палива - Заправка до початкового рівня
 Доступний для оренди
 Скасування: Включено
 Залиш: Включено
 Заст. від зіткнень: 100 гривень
 Заст. від крадіжок: 150 гривень
 Прем. страхування: 200 гривень
 Додатковий водій: 200 гривень

Додаткові параметри:
 Скасування
 Зміна
 Заст. від зіткнень
 Заст. від крадіжок
 Повне страхування
 Додатковий водій

ЗАБЕРТИТИ ВИДАЛИТИ СКАСУВАТИ

Рис. 3.22. Вікно редагування бронювань

Також ми можемо редагувати та створювати: локації(рисунок 3.23), транспорт(рисунок 3.24), користувачів(рисунок 3.25) та постачальників(рисунок 3.26).

Оновлення Розташування

Українська *


Львів, Міжнародний аеропорт

English *

Lviv, International Airport

ЗБЕРЕГТИ
СКАСУВАТИ

Рис. 3.23. Оновлення розташувань транспорту.



Рекомендований розмір зображення: 300x200

Назва *

BMW X1

Постачальник

Europcar **Europcar**

Мінімальний вік *

21

Місце видачі

Львів, Міжнародний аеропорт

Ціна (грн/день) *

780

Задаток при отриманні (грн) *

9500

Доступний для оренди

Тип двигуна *

Дизельне

Коробка передач *

Автоматична

Місяць *

5

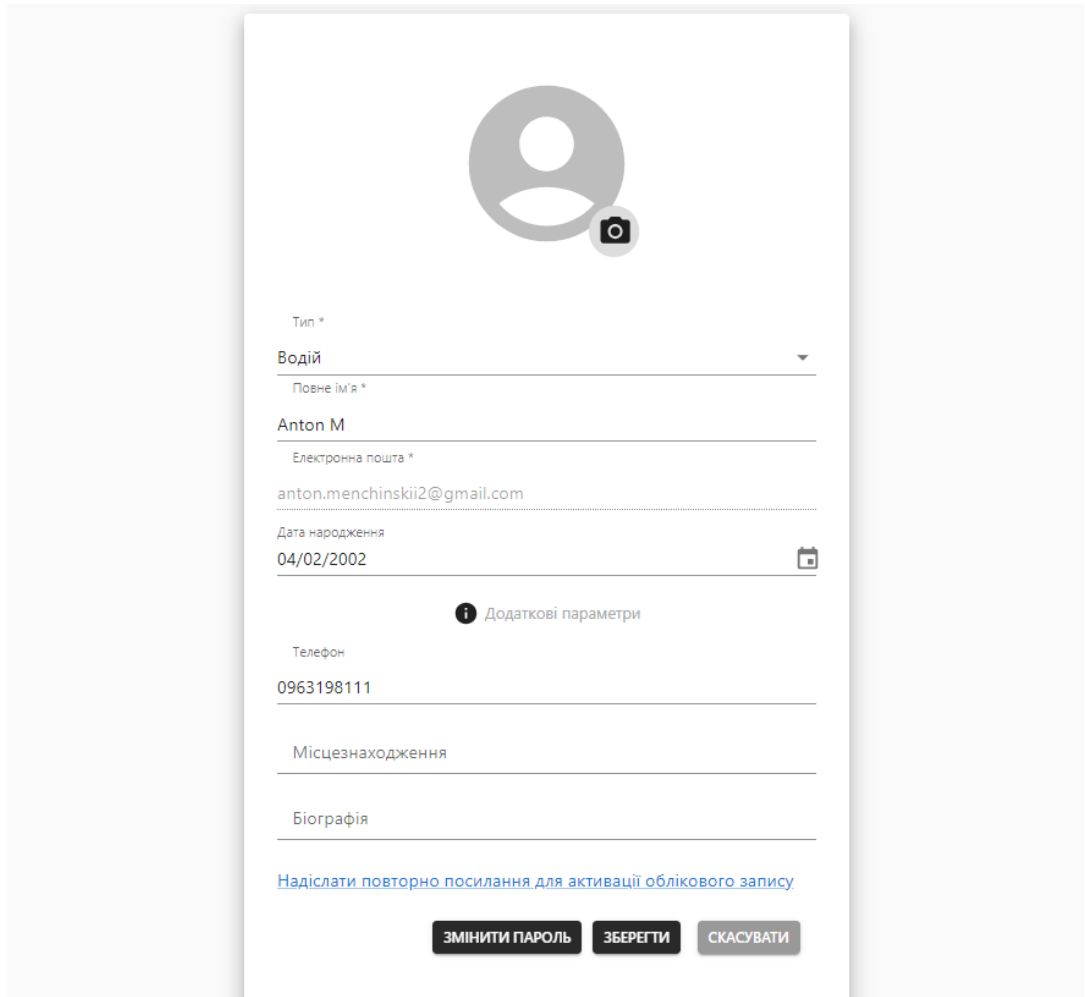
Двері *

4

Політика палива *

Безкоштовний бак

Рис. 3.24. Оновлення інформації про транспорт.



The image shows a user profile update form. At the top, there is a large grey circular placeholder for a profile picture with a camera icon in the bottom right corner. Below this, the form contains several input fields and a dropdown menu. The 'Тип' (Type) dropdown is set to 'Водій' (Driver). The 'Повне ім'я' (Full name) field contains 'Anton M'. The 'Електронна пошта' (Email) field contains 'anton.menchinskii2@gmail.com'. The 'Дата народження' (Date of birth) field contains '04/02/2002' and has a calendar icon. Below these fields is a section for 'Додаткові параметри' (Additional parameters) with an information icon. The 'Телефон' (Phone) field contains '0963198111'. There are also empty fields for 'Місцезнаходження' (Location) and 'Біографія' (Biography). At the bottom, there is a blue link: 'Надіслати повторно посилання для активації облікового запису' (Resend activation link). Three buttons are at the very bottom: 'ЗМІНИТИ ПАРОЛЬ' (Change password), 'ЗБЕРЕГТИ' (Save), and 'СКАСУВАТИ' (Cancel).

Тип *

Водій

Повне ім'я *

Anton M

Електронна пошта *

anton.menchinskii2@gmail.com

Дата народження

04/02/2002

Додаткові параметри

Телефон

0963198111

Місцезнаходження

Біографія

[Надіслати повторно посилання для активації облікового запису](#)

ЗМІНИТИ ПАРОЛЬ ЗБЕРЕГТИ СКАСУВАТИ

Рис. 3.25. Оновлення інформації про користувачів.

Menchyri

Рекомендований розмір зображення: 60x30

Повне ім'я *

TransportCompany

Електронна пошта *

menchynskyi_ak19@nuwm.edu.ua

Авторизуйте платіж пізніше

Додаткові параметри

Телефон

Місцезнаходження

Біографія

[Надіслати повторно посилання для активації облікового запису](#)

ЗМІНИТИ ПАРОЛЬ ЗБЕРЕГТИ СКАСУВАТИ

Рис. 3.25. Оновлення інформації про постачальників.

ВИСНОВКИ

В ході цієї кваліфікаційної роботи ми успішно спроектували та реалізували сайт для оренди транспортних засобів, надійну структуру бази даних MongoDB, яка дозволяє ефективно зберігати та отримувати дані. Використання схем і моделей забезпечило структурований підхід до визначення сутностей та їхніх взаємозв'язків, гарантуючи цілісність і узгодженість даних. Ця структура бази даних слугує міцною основою для нашого веб-сайту, забезпечуючи безперебійне управління даними та масштабованість. Також заклали надійну архітектуру в основну нашого сайту, що дозволить розширювати функціонал у майбутньому без проблем. Вибір сучасних технологій дозволить цьому сайту не втрачати актуальність.

У порівнянні з іншими відомими результатами, представлена робота, демонструє комплексний підхід до розробки багатофункціонального та зручного для користувачів сайту. Було враховано помилки інших сайтів, стосовно перевантаженості, відсутності локалізації, можливість постачати транспортні засоби. Також архітектура нашого сервісу відрізняється з точки зору продуктивності, масштабованості та глобальної доступності.

З економічної точки зору, здатність веб-сайту спростити процес оренди транспортних засобів може мати позитивний вплив на транспортну галузь. З'єднуючи користувачів з постачальниками та сприяючи ефективним транзакціям, він сприяє зростанню бізнесу та генерує можливості для отримання прибутку для обох залучених сторін.

Загалом, розробка цього веб-сайту для оренди транспортних засобів продемонструвала потенціал технологічного прогресу для підвищення доступності та зручності транспортних послуг. Висновки цього дослідження можуть слугувати основою для подальших досліджень і розробок у цій галузі, стимулюючи інновації та сприяючи розвитку індустрії оренди транспорту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бібліотека React.js офіційний сайт –
<https://uk.legacy.reactjs.org/docs/getting-started.html>
2. Офіційна документація Node.js – <https://nodejs.org/en/docs>
3. Документація Express – <https://expressjs.com/en/guide/>
4. Документація MongoDB – <https://www.mongodb.com/docs/>
5. Цивільний кодекс України / Глава 58 НАЙМ (ОРЕНДА).
[Електронний ресурс] / Законодавство України –
<https://zakon.rada.gov.ua/laws/show/435-15/page13> – Дата звернення:
21.10.2020 р.
6. "Node.js in Action" by Mike Cantelon, Marc Harter, TJ Holowaychuk,
and Nathan Rajlich
7. "React: Up and Running: Building Web Applications" by Stoyan
Stefanov.
8. "JavaScript: The Definitive Guide: Master the World's Most-Used
Programming Language" by Flanagan, D.
9. "MongoDB: The Definitive Guide", Chodorow, K..
10. "Design Patterns: Elements of Reusable Object-Oriented Software" by
Gamma, E., Helm, R., Johnson, R., & Vlissides, J.
11. "Patterns of Enterprise Application Architecture" by Fowler, M.
12. "Writing Effective Use Cases" by Alistair Cockburn