**Дніпровський національний університет**

**імені ОЛЕСЯ Гончара**

Факультет прикладної математики

Кафедра математичного забезпечення ЕОМ

**КУРСОВА РОБОТА**

з дисципліни «Бази даних»

на тему: «Розроблення бази даних та сайту для збору інформації, обліку та аналізу пасажирських перевезень»

Виконавець

студент групи ПЗ-21-2

спеціальність

121 Інженерія програмного забезпечення

Євсенєв Михайло Дмитрович

Керівник

Мащенко Леонід Володимирович

Кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Леонід Мащенко

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Світлана КЛИМЕНКО

(підпис)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Марина СИДОРОВА \_

(підпис)

Дніпро – 2024

**ЗМІСТ**

[ПОСТАНОВКА ЗАДАЧІ 3](#_Toc166232056)

[1. ПРОЄКТУВАННЯ СИСТЕМИ БАЗ ДАНИХ 4](#_Toc166232057)

[1.1. ОБСТЕЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ 4](#_Toc166232058)

[1.2. ОСНОВНІ ЗАВДАННЯ, ЩО ТРЕБА РОЗВ'ЯЗАТИ 5](#_Toc166232059)

[1.3. ОСНОВНІ ЗАПИТИ, ЗВІТИ ЗА РІВНЯМИ ДОСТУПУ КОРИСТУВАЧІВ 5](#_Toc166232060)

[1.4. НОРМАЛІЗАЦІЯ ВІДНОШЕНЬ ТА ЇЇ ОБГРУНТУВАННЯ 6](#_Toc166232061)

[1.5. ПОБУДОВА ГРАФ-СХЕМИ ІНФОЛОГІЧНОЇ МОДЕЛІ ТА ПЕРЕВІРКА ЇЇ КОРЕКТНОСТІ 7](#_Toc166232062)

[2. РЕАЛІЗАЦІЯ СИСТЕМИ БАЗ ДАНИХ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 10](#_Toc166232063)

[2.1. ОБҐРУНТУВАННЯ ВИБОРУ СУБД ТА ЗАСОБІВ РОЗРОБЛЕННЯ ЗАСТОСУНКУ 10](#_Toc166232064)

[2.2. ОПИС ТА СТРУКТУРНА СХЕМА ПРОГРАМИ 11](#_Toc166232065)

[2.3. ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧІВ 13](#_Toc166232066)

[2.4. ДІАГРАМА КЛАСІВ 16](#_Toc166232067)

[2.5. ПРИЗНАЧЕННЯ ОСНОВНИХ МЕТОДІВ, ПРОЦЕДУР, ФУНКЦІЙ, ОБРОБНИКІВ ПОДІЙ 17](#_Toc166232068)

[2.6. РЕАЛІЗАЦІЯ СКЛАДНИХ ЗАПИТІВ НА ВИБІРКУ ТА ЗВІТІВ ДЛЯ КОРИСТУВАЧІВ РІЗНИХ РІВНІВ 21](#_Toc166232069)

[3. БЕЗПЕКА ТА СЕРВІСНІ ФУНКЦІЇ СИСТЕМИ 23](#_Toc166232070)

[3.1. РЕЄСТРАЦІЯ ТА АВТЕНТИФІКАЦІЯ КОРИСТУВАЧІВ З РІЗНИМ ДОСТУПОМ 23](#_Toc166232071)

[3.2. ВЕДЕННЯ ЖУРНАЛУ РОБОТИ КОРИСТУВАЧІВ СИСТЕМИ ТА ЙОГО ОБРОБКА 23](#_Toc166232072)

[3.3. АРХІВУВАННЯ, ВІДНОВЛЕННЯ ТА ІНШІ ФУНКЦІЇ З СЕРВІСУ ТА НАЛАШТУВАННЯ 24](#_Toc166232073)

[4. ІНСТРУКЦІЇ ПО ВИКОРИСТАННЮ ТА СУПРОВОДЖЕННЮ КОРИСТУВАЧАМ ЗА РІВНЕМ ДОСТУПУ 28](#_Toc166232074)

[4.1. АНАЛІЗ ТА ОБГРУНТУВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ 28](#_Toc166232075)

[5. АНАЛІЗ ТА ОБГРУНТУВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ 34](#_Toc166232076)

[ВИСНОВКИ 42](#_Toc166232077)

[СПИСОК ЛІТЕРАТУРИ 43](#_Toc166232078)

[ДОДАТКИ 44](#_Toc166232079)

ПОСТАНОВКА ЗАДАЧІ

Розробити базу даних засобами обраної СУБД та програмний за стосунок (або сайт), що забезпечують введення, перегляд, редагування даних та виконання усіх запитів користувачів у відповідності до виявлених завдань за інформаційними потоками у предметній області.

Загальні вимоги до бази даних та роботи:

1. Провести всеосяжне обстеження предметної області;

2. Виявити основні інформаційні потоки та завдання користувачів, які необхідно розв’язати;

3. Визначити вид, правила формування і типи основних даних для вхідних та вихідних документів;

4. Виявити аномалії при роботі з об’єктами предметної області при виконанні основних операцій над даними та провести нормалізацію відношень;

5. Виконати побудову інфологічної моделі предметної області та перевірити її коректність;

6. Для відповідної інфологічної моделі виконати реалізацію в одній з реляційних СУБД;

7. Розробити індивідуальні форми (та сторінки сайту) для користувачів різних рівнів, що забезпечують дружній інтерфейс, орієнтований на предметну область. Ці форми мають надавати можливість навігації по таблиці та вирішення задач користувачів.

8. Врахувати можливість архівації та відновлення таблиць бази даних або всієї бази даних; створення копії бази даних для перенесення на інше робоче місце; можливість перегляду адміністратором журналу дій користувачів за відповідними запитами (фільтрами).

9. Створити запити різноманітних типів та звіти для основних запитів.

Завдяки запитам на вибірку різної складності забезпечити надання необхідної інформації користувачам різних рівнів

10. Забезпечити контроль даних при введенні інформації по типу, діапазону, значенню та цілісність даних;

11. Виконати тестування та налагодження програмного продукту;

12. Розробити інструкції для користувачів різних рівнів;

13. Підготувати звіт з Курсової роботи.

Предметна область курсової роботи: «Санаторно-курортне обсуговування»

1. ПРОЄКТУВАННЯ СИСТЕМИ БАЗ ДАНИХ
   1. ОБСТЕЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Провівши обстеження предметної області «Санаторно-курортне обсуговування», взявши до уваги усі дрібниці та деталі які були помічені під час обстеження предметної області я зміг ідентифікувати наступні об’єкти:

* Клієнти (гості):

1. Ім'я
2. Прізвище
3. Дата народження
4. Адреса
5. Номер телефону
6. Електронна пошта

* Послуги:

1. Назва послуги
2. Опис
3. Ціна
4. Тривалість
5. Категорія послуги (наприклад, масаж, косметологія, фітнес)

* Медичні процедури:

1. Назва процедури
2. Опис
3. Тривалість
4. Ціна
5. Лікар, який виконує процедуру
6. Медичні показники (якщо применимо)

* Персонал :

1. Id клієнта
2. Заробітна плата
3. Ім'я
4. Прізвище

* Номери (місця проживання):

1. Номер кімнати
2. Тип номеру (одномісний, двомісний, люкс тощо)
3. Вартість проживання за ніч
4. Опис
5. Фото

* Розклади медичних процедур:

1. Час початку
2. Час закінчення
3. Id Лікаря, який проводить процедури
4. Id процедури

* Розклади харчування:

1. Часи прийому їжі (сніданок, обід, вечеря)
2. День

* Фінансові транзакції:

1. Дата транзакції
2. Сума транзакції
3. Тип транзакції (оплата за проживання, оплата за послуги, повернення коштів)
4. Деталі транзакції (номер кімнати, назва послуги тощо)
5. Id клієнта
   1. ОСНОВНІ ЗАВДАННЯ, ЩО ТРЕБА РОЗВ'ЯЗАТИ

До основних завдань, що треба розв’язати в цій темі можна віднести наступні:

1. Створення акаунту для користувача – при першому відвіданні сайту користувач зможе зареєструвати аккаунт ввівши в відповідні поля ім’я користувача, електрону пошту та пароль і якщо в системі не зареєстрована така пошта та пароль створиться аккаунт для користувача.

Забезпечити захист сайту – при спробі не аутентифікованого користувача зайти на сайт його поверне на реєстрацію, за спробою зайти в панель адміна, не адміна просто не допустить до сторінки.

1. Пошук санаторних послуг відповідно до заданих параметрів: Користувачі повинні мати можливість шукати та бронювати послуги санаторію, задаючи такі параметри, як дати перебування, вид лікування, тип розміщення тощо. Платформа повинна забезпечувати зручний інтерфейс для фільтрації та сортування доступних варіантів.
2. Управління даними на сайті: Адміністратори повинні мати можливість керувати даними на платформі, включаючи оновлення інформації про послуги, ціни, розклад лікування, а також перегляд та обробку заявок від користувачів. Всі зміни мають проходити відповідну валідацію для забезпечення коректності даних.

Щоб система відповідала бізнес-потребам та очікуванням замовників та користувачів, ці завдання, які є ключовими компонентами її функціональності, потребують ретельної реалізації. Окрім того, при їх реалізації важливо враховувати аспекти безпеки, продуктивності та зручності використання.

* 1. ОСНОВНІ ЗАПИТИ, ЗВІТИ ЗА РІВНЯМИ ДОСТУПУ КОРИСТУВАЧІВ

1. Планування розкладу процедур і послуг: Адміністратор має можливість створювати та коригувати розклад санаторних процедур, включаючи дату, час, та доступні послуги. Система не повинна допускати конфліктів у розкладі.

Рівень доступу: Адміністратор

1. Пошук доступних номерів: Користувач може шукати доступні номери в санаторії, вказуючи параметри, такі як дата заїзду, тривалість перебування, тип номеру та наявність додаткових послуг.

Рівень доступу: Користувач

1. Розрахунок вартості послуг: При пошуку санаторних послуг система автоматично розраховує вартість з урахуванням тривалості перебування, типу номеру та обраних процедур.

Рівень доступу: Користувач

1. Авторизація користувачів: Кожен користувач повинен ввести ім'я користувача та пароль для доступу до системи. Після перевірки система надає доступ або виводить відповідне повідомлення про помилку.

Рівень доступу: Усі користувачі

1. Управління номерами та процедурами: Адміністратор може додавати або видаляти номери, а також коригувати список доступних процедур у системі.

Рівень доступу: Адміністратор

* 1. НОРМАЛІЗАЦІЯ ВІДНОШЕНЬ ТА ЇЇ ОБГРУНТУВАННЯ

В курсовій роботі вимагалось провести нормалізацію включно до третьої нормальної форми для мінімізації надмірності даних та забезпечення цілісності даних. Завдяки навичкам здобутим під час семестру.

Перевіривши об’єкти які я виділив під час обстеження предметної області «Санаторно-курортне обсуговування» я переконався, що вони вже знаходяться у третій нормальній формі і усі неповні функціональні залежності так само як і транзитивні залежності були усунуті. Щоб переконатися в цьому як приклад можна розглянути 1 об’єкт:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | name | description | price | duration | category | photo |
| 1 | Name | Description | 200 | 60 | SPA | url |

Як можна побачити на цьому прикладі в моїй таблиці присутній первинний ключ який є простим, тобто складається з єдиного атрибуту, і усі об’єкти мають до нього повну функціональну залежність від якого залежать усі поля таблиці тому ця таблиця знаходиться у другій нормальній формі. Щоб сказати що таблиця знаходиться у 3 нормальній формі повинні виконуються такі умови:

* таблиця знаходиться у другій нормальній формі.
* жоден неключовий атрибут таблиці не знаходиться в транзитивній функціональній залежності від потенційного ключа таблиці
  1. ПОБУДОВА ГРАФ-СХЕМИ ІНФОЛОГІЧНОЇ МОДЕЛІ ТА ПЕРЕВІРКА ЇЇ КОРЕКТНОСТІ

Згідно з наданою схемою бази даних, можна побудувати наступну граф-схему (дивіться рисунок 1.1).

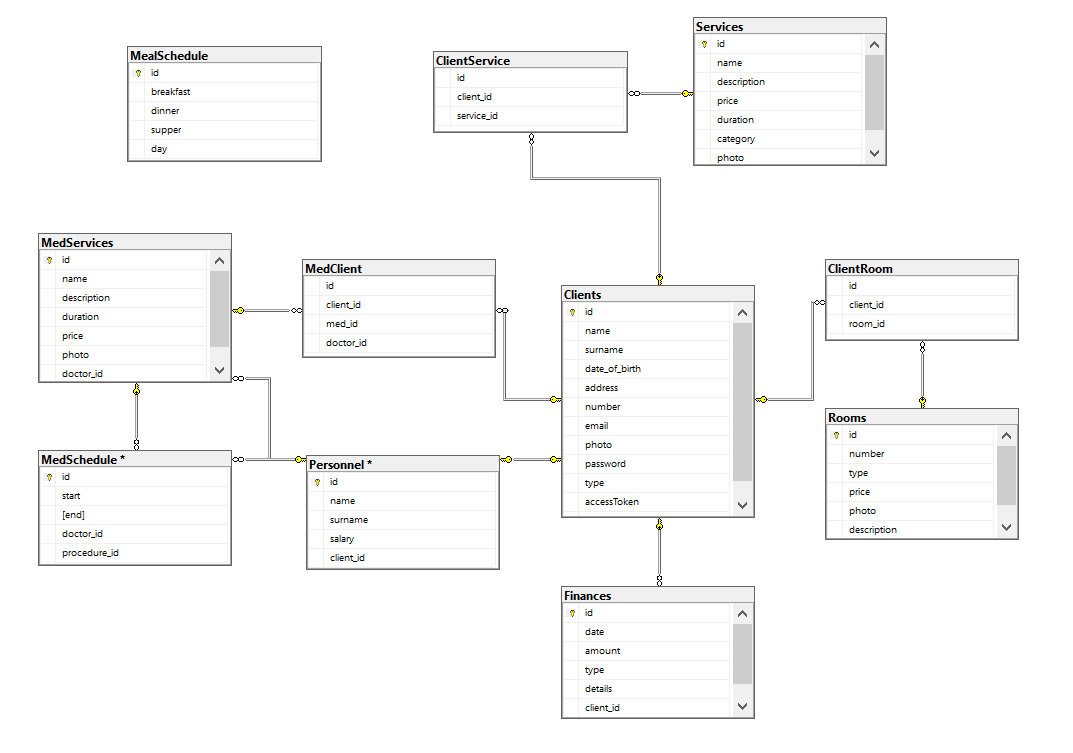


Рисунок 1.1 – Граф схема інфологічної моделі

Вершини:

* **Clients**: Представляє таблицю Clients з її атрибутами.
* **Finances**: Представляє таблицю Finances з її атрибутами.
* **Rooms**: Представляє таблицю Rooms з її атрибутами.
* **ClientRoom**: Представляє таблицю ClientRoom з її атрибутами.
* **Personnel**: Представляє таблицю Personnel з її атрибутами.
* **MedClient**: Представляє таблицю MedClient з її атрибутами.
* **MedServices**: Представляє таблицю MedServices з її атрибутами.
* **MedSchedule**: Представляє таблицю MedSchedule з її атрибутами.
* **MealSchedule**: Представляє таблицю MealSchedule з її атрибутами.
* **Services**: Представляє таблицю Services з її атрибутами.
* **ClientService**: Представляє таблицю ClientService з її атрибутами.

Ребра:

* **Clients ↔ Finances**: Один клієнт може мати декілька фінансових операцій. Зв'язок між цими таблицями є один-до-багатьох .
* **Clients ↔ ClientRoom ↔ Rooms**: Клієнт може зняти в декількох номерах за різні періоди, а один номер може бути зайнятий декількома клієнтами протягом різного часу. Зв'язок реалізується через таблицю ClientRoom.
* **Clients ↔ MedClient↔ MedServices**: Один клієнт може записатися на декілька процедур. Зв'язок між таблицями Clients та MedClient є один-до-багатьох, а MedClient та MedServices багато до одного.
* **MedSchedule ↔ MedServices ↔ Personnel**: У одного розкладу одна медпроцедура та один доктор, але і у доктора и процедурі може бути багато розкладів.
* **Personnel ↔ Clients**: За кожним працівником закріплений їх клієнтський аккаунт.
* **Clients ↔ ClientService ↔ Services**: Клієнт може скористатися кількома послугами, а одна послуга може надаватися декільком клієнтам. Зв'язок через таблицю ClientService.

Дивлячись на граф-схему можна побачити шо вона коректна. Зв'язки між таблицями мають сенс і відповідають реальності. Множинність зв'язків також відповідає очікуваному. Атрибути описують об'єкти відповідним чином і не дублюються. Граф-схема не допускає аномалій даних.Але давайте перевіримо її на прикладі пари запитів:

* Запит на отримання заброньованих кімнат клієнта(дивіться рисунок 1.2).
* Запит на отримання даних про доктора закріпленного до мед послуги у розкладі(дивіться рисунок 1.3).

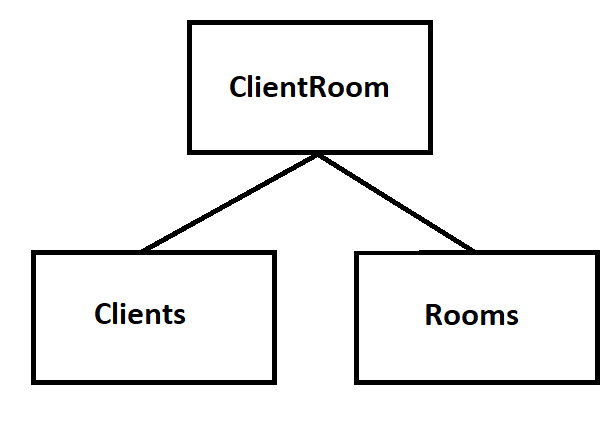


Рисунок 1.2 – Запит на отримання заброньованих кімнат клієнта

На рисунку 1.2 показано низку таблиць, які потрібно пройти, щоб отримати потрібну інформацію про заброньовані кімнати. Розглянемо детально цю діаграму:

1. Запит на пошук кімнат асоційованих з клієнтом : початкова точка, де користувач запитує інформацію кімнати.
2. Таблиця ClientRoom: перша таблиця, яка містить id кімнати та клієнта.
3. Таблиця Clients: друга таблиця, яка містить детальну інформацію про клієнта на якого орендована кімната.
4. Таблиця Rooms: третя таблиця і остання, яка видає кімнату.

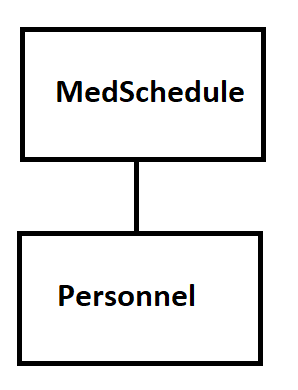


Рисунок 1.3 – Запит на отримання даних про доктора закріпленого до мед послуги у розкладі

На рисунку 1.3 показано низку таблиць, які потрібно пройти, щоб отримати потрібну інформацію про доктора закріпленого до мед послуги у розкладі. Розглянемо детально цю діаграму:

1. Запит на розклад : початкова точка, де користувач запитує інформацію про розклад.
2. Таблиця MedSchedule: перша таблиця, яка містить основну інформацію про розклад
3. Таблиця MedServices: друга таблиця, яка містить детальну інформацію доктора який проводить ці процедури.

РЕАЛІЗАЦІЯ СИСТЕМИ БАЗ ДАНИХ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

* 1. ОБҐРУНТУВАННЯ ВИБОРУ СУБД ТА ЗАСОБІВ РОЗРОБЛЕННЯ ЗАСТОСУНКУ

**Вибір MS SQL Server для СУБД:**

1. **Надійність та стабільність**: MS SQL Server відомий своєю надійністю та здатністю обробляти великі обсяги даних. Він забезпечує високу продуктивність, особливо для веб-застосунків з високими вимогами до транзакцій.
2. **Безпека**: Вбудовані механізми безпеки, такі як шифрування даних, керування ролями користувачів та аудит, роблять MS SQL Server чудовим вибором для забезпечення захисту даних.
3. **Масштабованість**: MS SQL Server легко масштабується як вертикально (збільшення обсягу ресурсів сервера), так і горизонтально (додавання нових серверів), що дозволяє адаптувати базу даних до потреб зростаючого бізнесу.
4. **Інтеграція з іншими продуктами Microsoft**: Оскільки MS SQL Server є продуктом Microsoft, він легко інтегрується з іншими продуктами цієї екосистеми, що може бути корисним для підприємств, які вже використовують рішення Microsoft.

**Вибір Express.js та React для розробки:**

1. **Express.js**:
   * **Легкість та простота використання**: Express.js - це мінімалістичний фреймворк для Node.js, який надає розробникам простий у використанні набір інструментів для створення серверних застосунків та API. Він забезпечує швидку та ефективну розробку веб-сервісів.
   * **Гнучкість**: Express.js дозволяє легко налаштовувати маршрути, обробляти запити та інтегрувати сторонні бібліотеки, що робить його універсальним вибором для розробки серверної частини.
2. **React**:
   * **Декларативний підхід**: React дозволяє розробникам створювати динамічні та інтерактивні інтерфейси користувача за допомогою компонетного підходу, що значно спрощує управління станом застосунку та його структурою.
   * **Висока продуктивність**: Завдяки віртуальному DOM, React оптимізує процес оновлення інтерфейсу користувача, що підвищує продуктивність застосунку.

Таким чином, поєднання MS SQL Server для управління даними та Express.js з React для розробки серверної та клієнтської частин відповідно забезпечує створення надійного, безпечного та масштабованого веб-застосунку, здатного обробляти значні обсяги даних та надавати зручний інтерфейс для користувачів.

* 1. ОБҐРУНТУВАННЯ ВИБОРУ СУБД ТА ЗАСОБІВ РОЗРОБЛЕННЯ ЗАСТОСУНКУ

В курсовій роботі я розділив інтерфейс користувачів на два типи: для звичайних користувачів та для адміністратора. спочатку розглянемо сторінку аутентифікації(дивіться рисунки 2.4-2.5).

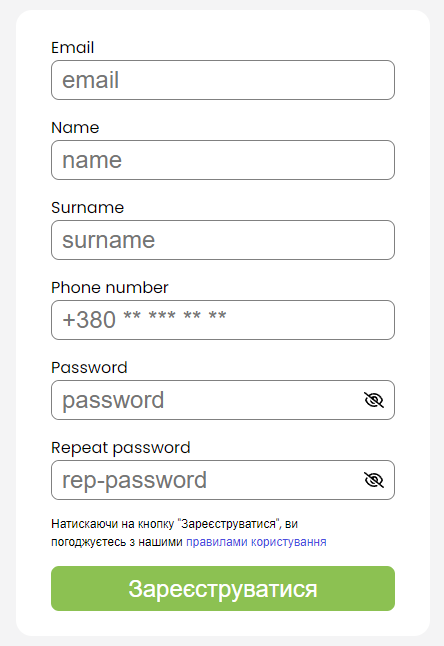


Рисунок 2.4 – Форма реєстрації

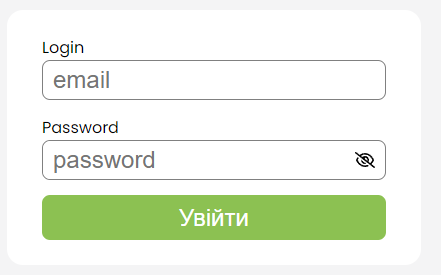


Рисунок 2.5 – Форма автентифікації для клієнта

Після вдалої аутентифікації користувач потрапляє на головну сторінку. Якщо це адмін, то на цій сторінці він може перейти на сторінку адміністратора за допомогою навігації(дивіться рисунки 2.6, 2.7, 2.8, 2.9).



Рисунок 2.6 – Навігація для неаунтефікованих користувачів

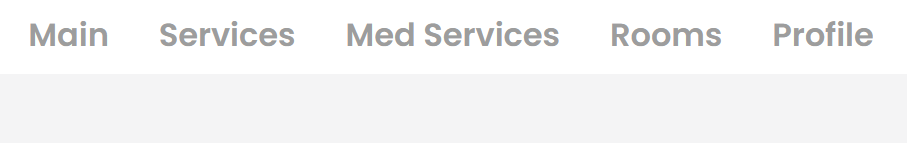


Рисунок 2.7 – Навігація для аунтефікованих користувачів



Рисунок 2.8 – Навігація для адміна

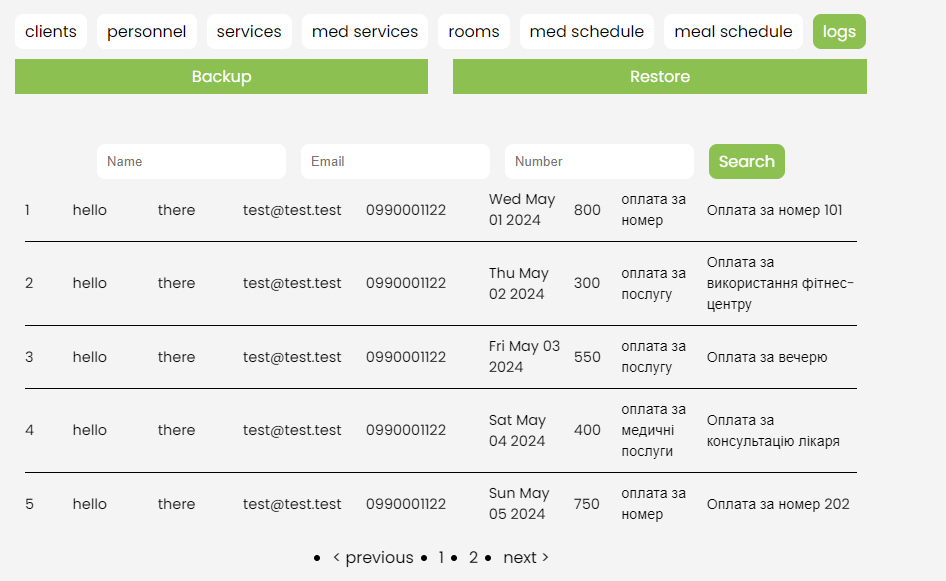


Рисунок 2.9 – Сторінка адміна

Користувач може зайти в свій профіль щоб оновити дані або/та подивитися на свої замовлені послуги. (дивіться рисунки 2.10, 2.11, 2.12)

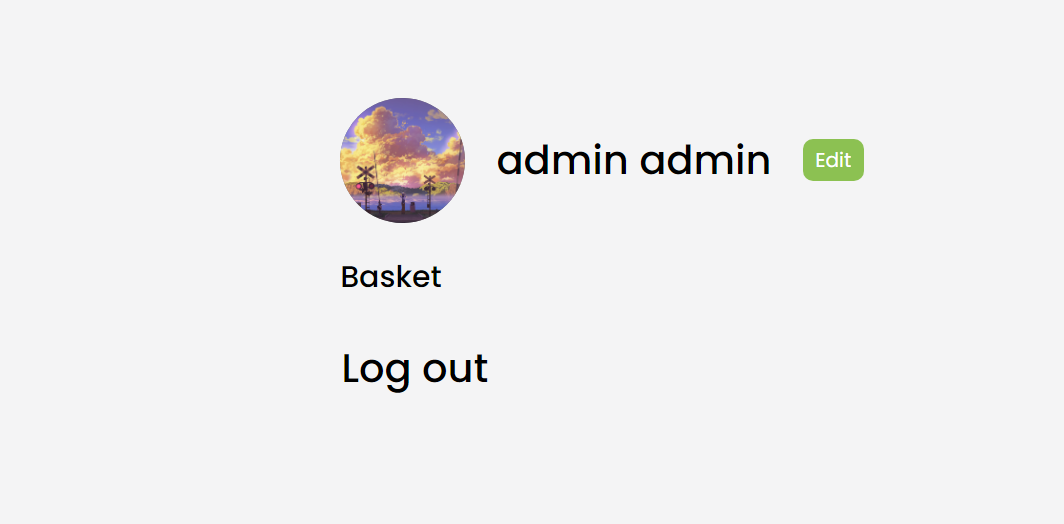


Рисунок 2.10 – Особистий профіль

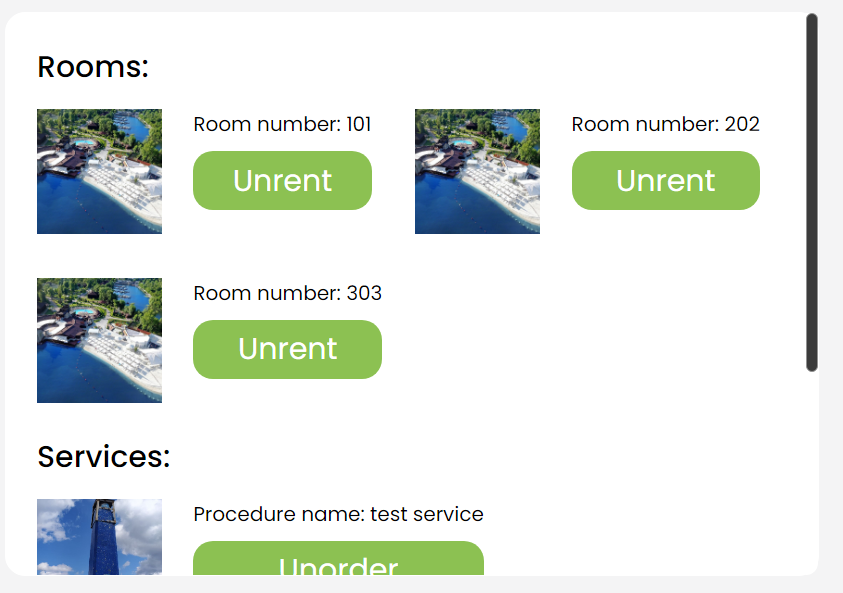


Рисунок 2.11– Замовлені послуги

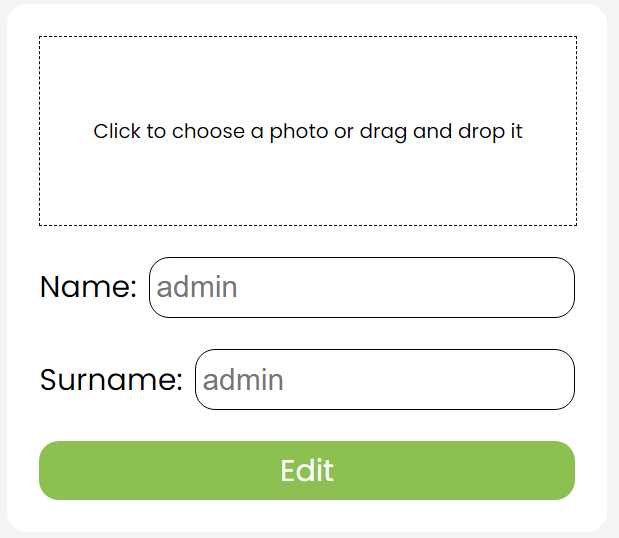


Рисунок 2.12 – Форма оновлення даних

Також можна зайти на сторінки Кімнат, Послуг та Мед послуг. Всі вони виглядають за одним шаблоном(дивіться рисунок 2.13)

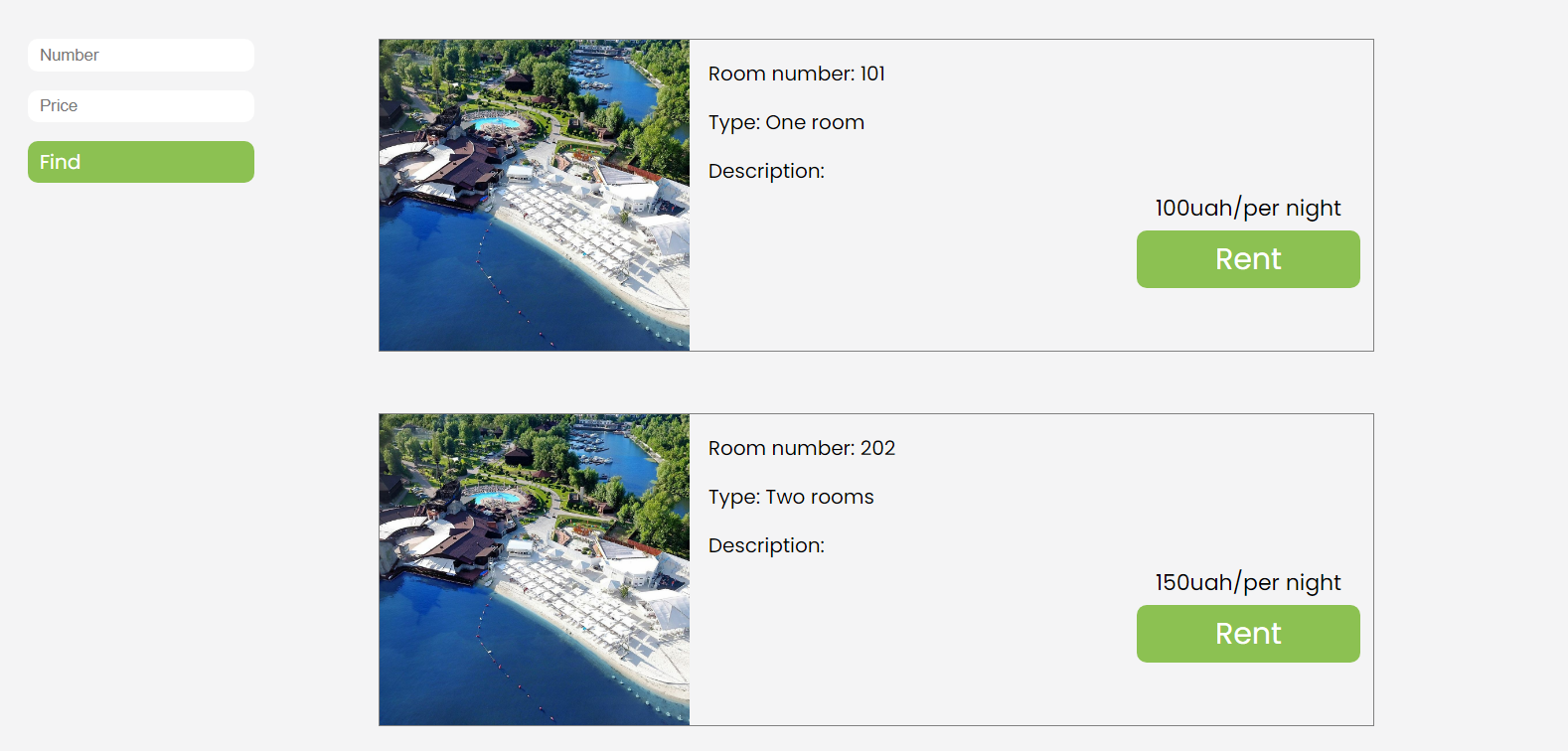


Рисунок 2.12 – Вигляд сторінки кімнат

* 1. ПРИЗНАЧЕННЯ ОСНОВНИХ МЕТОДІВ, ПРОЦЕДУР, ФУНКЦІЙ, ОБРОБНИКІВ ПОДІЙ

Першим ділом було встановлено зв’язок з базою даних(дивіться рисунок 2.13)

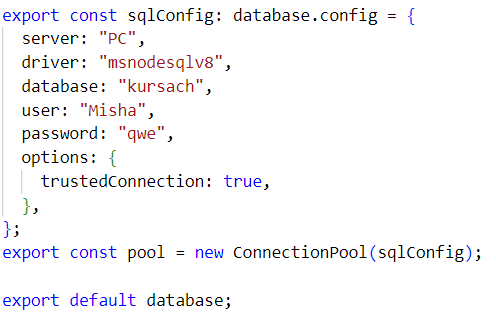


Рисунок 2.13 – Встановлення зв’язку з базою даних

Для кожної сутності були створені CRUD методи (дивіться рисунки 2.14, 2.15)

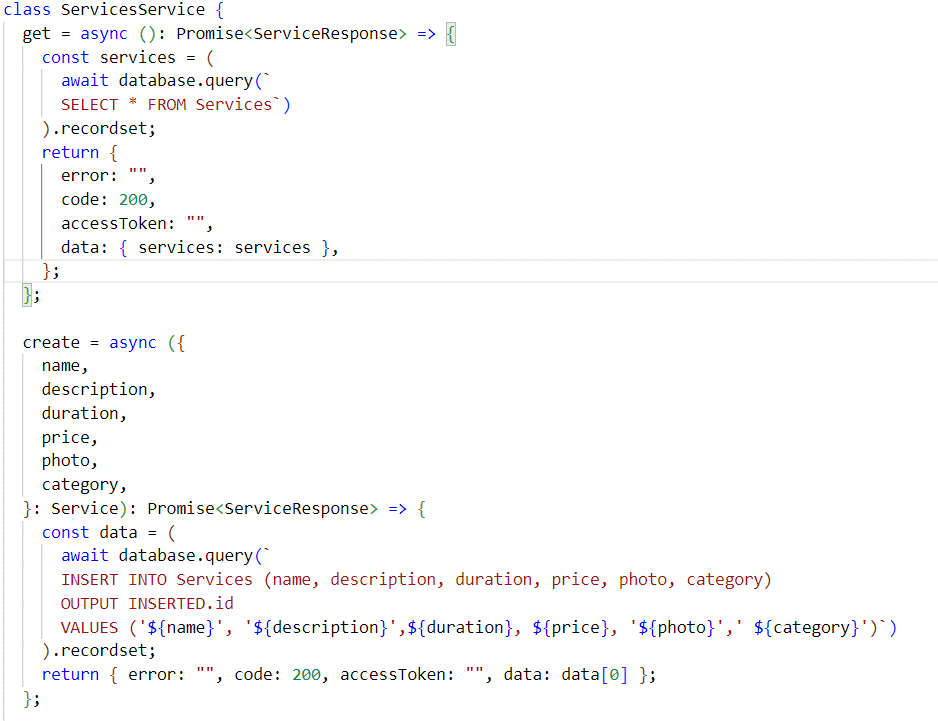


Рисунок 2.14 – CRUD операції

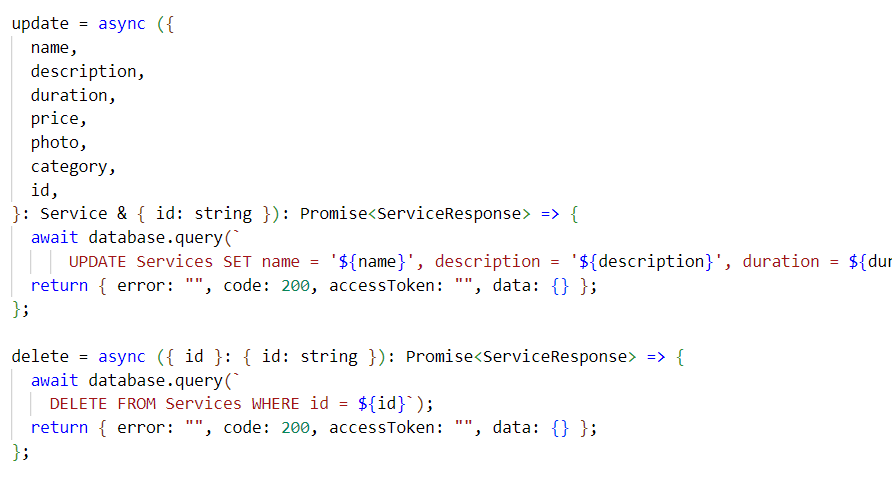


Рисунок 2.15 – CRUD операції

Кожна операція використовує метод query нашої бази даних. Завдяки цим методам забезпечується повний контроль на даними які зберігаються в таблицях

На стороні клієнта були визначені маршрути(дивіться рисунки 2.16, 2.17)

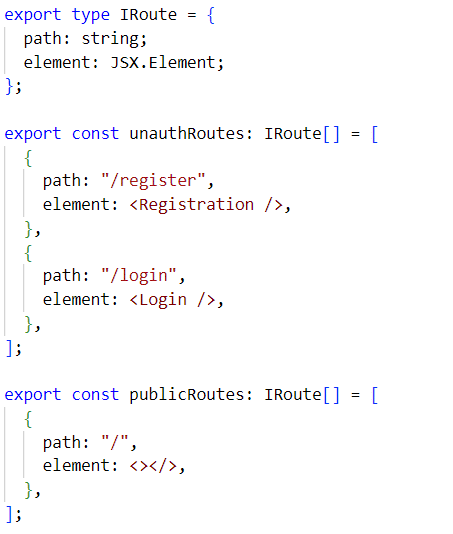


Рисунок 2.16 – Маршрути для сторінок



Рисунок 2.17 – Маршрути для сторінок

Також були створені засоби аунтефікації та авторизації користувачів(дивіться рисунки 2.18, 2.19)

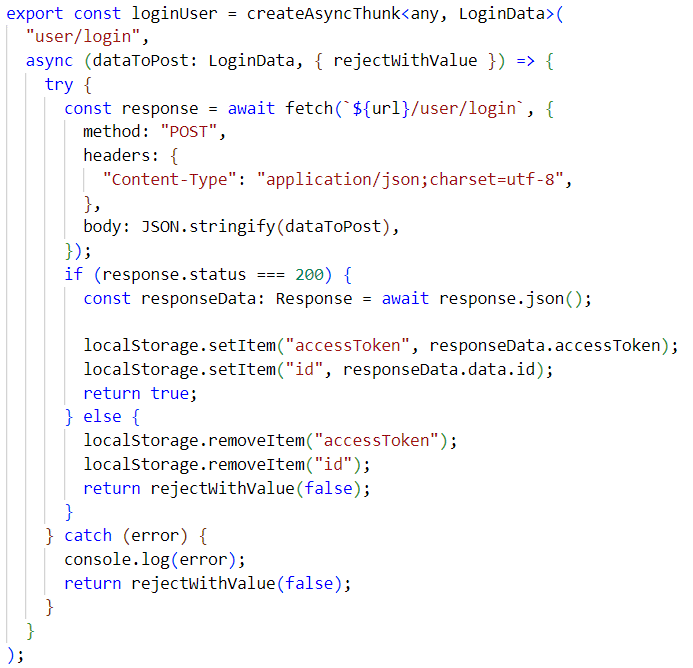


Рисунок 2.18 – Авторизація

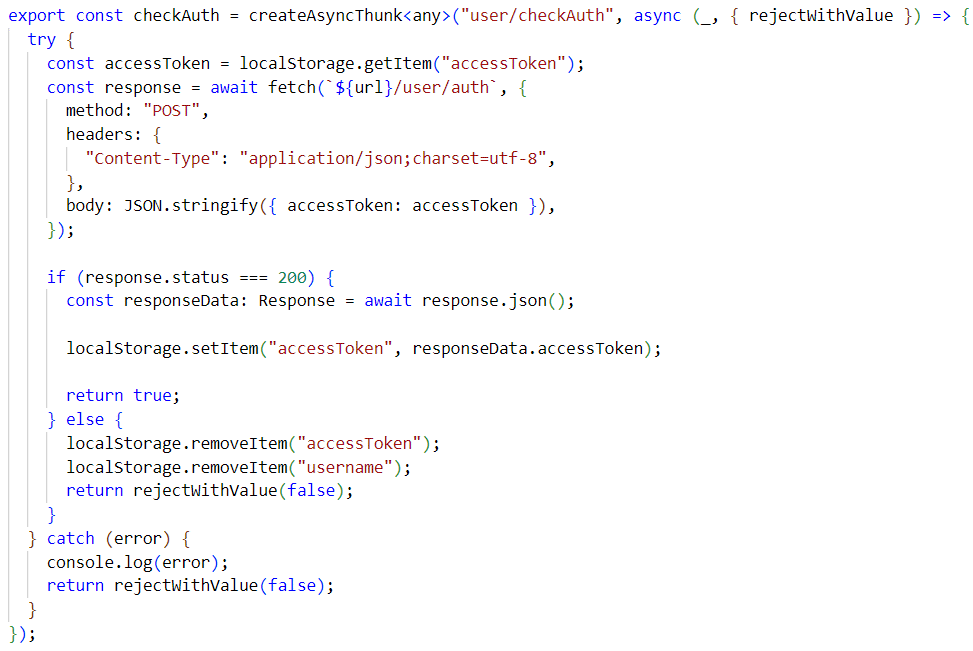


Рисунок 2.19 – Аунтефікація

1. БЕЗПЕКА ТА СЕРВІСНІ ФУНКЦІЇ СИСТЕМИ
   1. РЕЄСТРАЦІЯ ТА АВТЕНТИФІКАЦІЯ КОРИСТУВАЧІВ З РІЗНИМ ДОСТУПОМ

Під час реєстрації можливо зареєструватися лише як користувач. Нового адміністратора безпосередньо може зареєструвати лише вже існуючий адміністратори.

Також у системі для забезпечення захищеності використовується JWT Token і відповідно лише користувачі з цим токеном можуть виконувати будь-які дії на сайті, ці токени надсилаються зі сторони сервісу під час аутентифікації.

* 1. ВЕДЕННЯ ЖУРНАЛУ РОБОТИ КОРИСТУВАЧІВ СИСТЕМИ ТА ЙОГО ОБРОБКА

З метою відстеження дій користувачів системи була створена спеціальна таблиця журнал роботи користувачів в базі даних. Ця таблиця містить всю інформацію про активність користувачів (дивіться рисунок 3.1).

Для зручності на сторінці адміністратора було розроблено додаткову вкладку «Logs» на якій адміністратор може переглядати усі журнали подій, а також для спрощеного пошуку за певними критеріями були створені фільтри(дивіться рисунок 3.2).

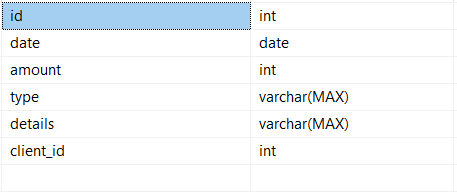


Рисунок 3.1 – Таблиця журнал роботи користувачів

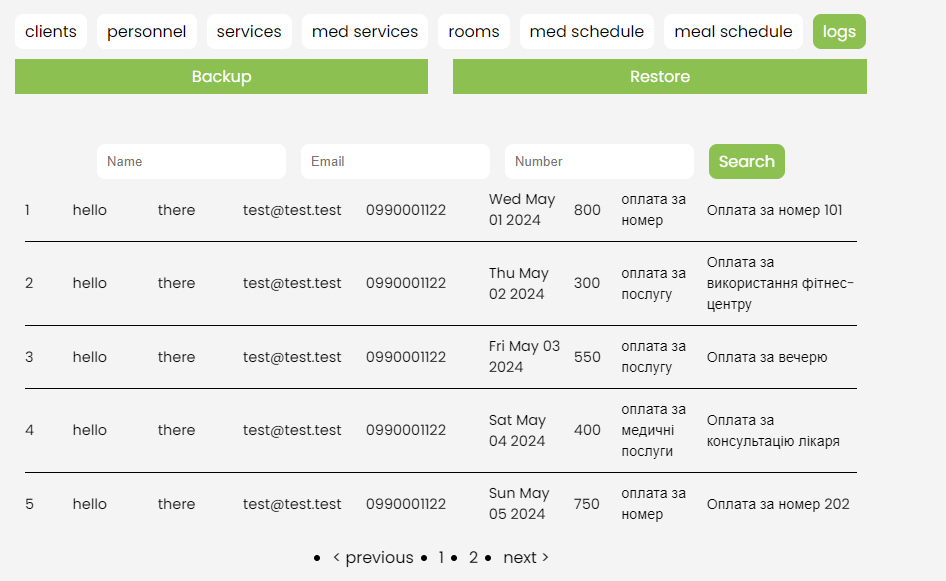


Рисунок 3.2 – Демонстрація журналу роботи користувачів

* 1. АРХІВУВАННЯ, ВІДНОВЛЕННЯ ТА ІНШІ ФУНКЦІЇ З СЕРВІСУ ТА НАЛАШТУВАННЯ

Щоб забезпечити функціонал для створення резервної копії бази даних та відновлення бази даних відповідно до наданої копії було створено сервіс DB, в цьому класі існують наступні методи:

* створити резервну копію бази даних (дивіться рисунок 3.3).
* відновити базу даних(дивіться рисунок 3.4).

Зі сторони сайту було створено вкладку бекап на які адміністратори може натиснувши кнопку зберегти резервну копію бази даних собі на комп'ютер або натиснувши кнопку завантажити зі свого комп'ютера потрібний файл та завантажити його а потім відправити на сервер(дивіться рисунок 3.5).

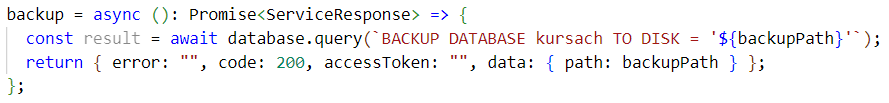


Рисунок 3.3 –Створити резервну копію бази даних

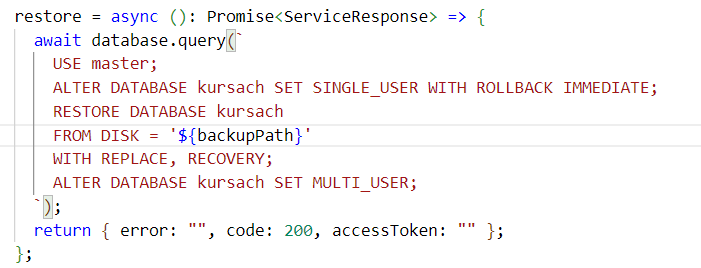


Рисунок 3.4 – Відновити базу даних

 Рисунок 3.5 – Кнопки Backup та Restore

Після натискання на кнопку Backup, на комп’ютер користувача завантажується файл, за допомогою якого можна відновити базу даних.

Після натискання на кнопку Restore, вам треба буде обрати файл для завантаження, після цього файл відправляється на сервер, створює його і відновлює базу даних.

1. ІНСТРУКЦІЇ ПО ВИКОРИСТАННЮ ТА СУПРОВОДЖЕННЮ КОРИСТУВАЧАМ ЗА РІВНЕМ ДОСТУПУ
   1. АНАЛІЗ ТА ОБГРУНТУВАННЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ

Інструкція для користувача як придбати квиток:

1. Спочатку користувача зустрічає форма реєстрації або авторизації(дивіться рисунок 4.1).

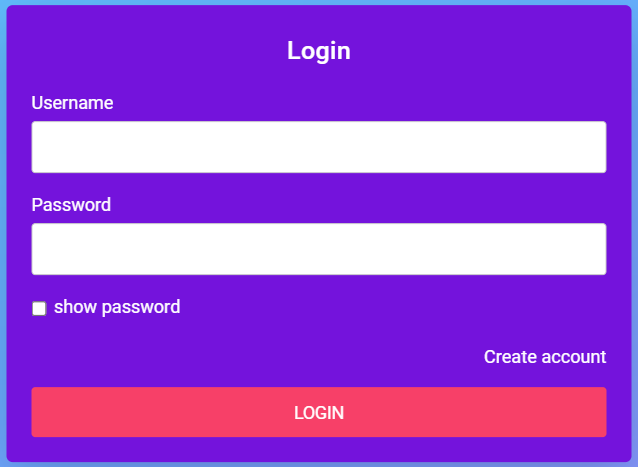


Рисунок 4.1 – Форма реєстрації

Якщо в клієнт вже є акаунт він вводить потрібні дані та натискає Login і якщо все гаразд то потрапляє на головну сторінку(дивіться рисунок 4.2).

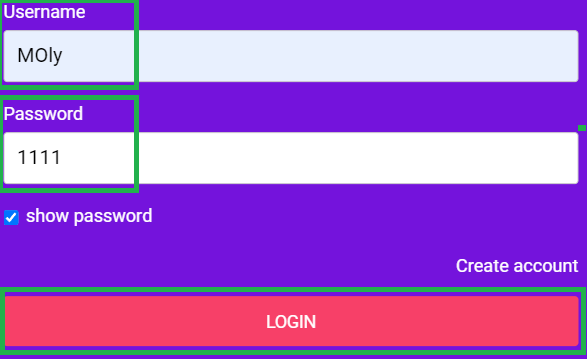


Рисунок 4.2 – Форма реєстрації

Щоб знайти потрібний квиток на головній сторінці користувач вводить звідки він буде їхати, куди він буде їхати та дату свого відправлення після чого натискає кнопку знайти квитки(дивіться рисунок 4.3).

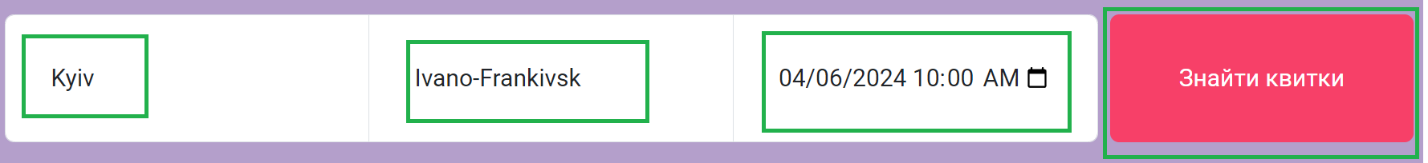


Рисунок 4.3 – Форма реєстрації

Якщо відповідно до запиту користувача знаходяться квитки вони відображаються нижче з датою відправлення, датою прибуття, з місця відправлення, місцем прибуття та ціною яка залежить від комфорту вагону. Коли користувач вирішує в якому вагоні він хоче подорожувати Він нажимає на кнопку Select(дивіться рисунок 4.4).

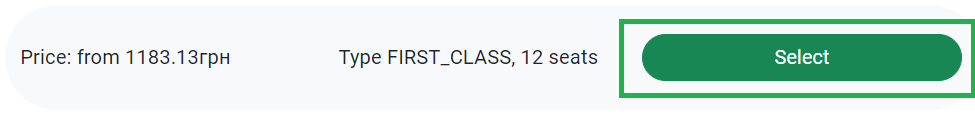
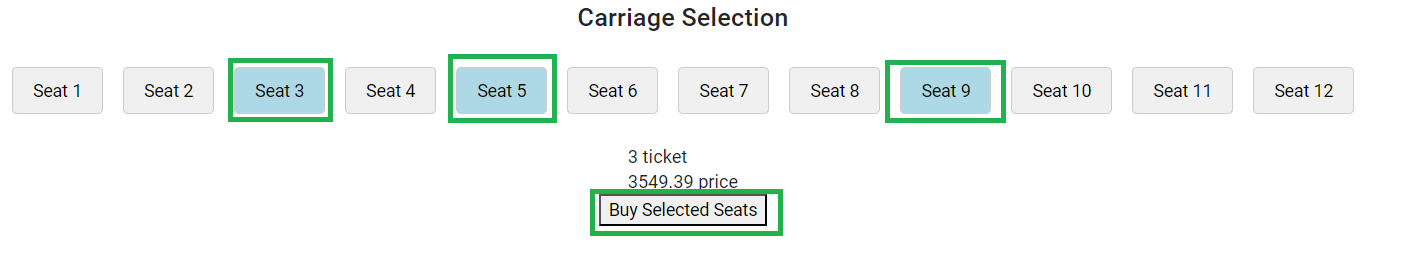
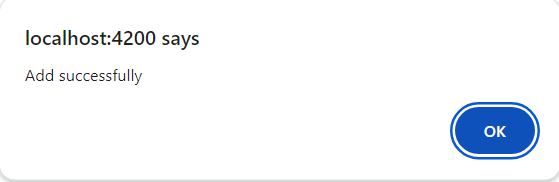


Рисунок 4.4 – Форма реєстрації

після натискання користувача переносить на сторінку де він може вибрати місця які йому до вподоби та натискає кнопку придбати вибрані квитки (дивіться рисунок 4.5).

 Рисунок 4.5 – Форма реєстрації

Після чого в нього в браузері з'являється відповідний повідомлення з написом успішно додано(дивіться рисунок 4.6).

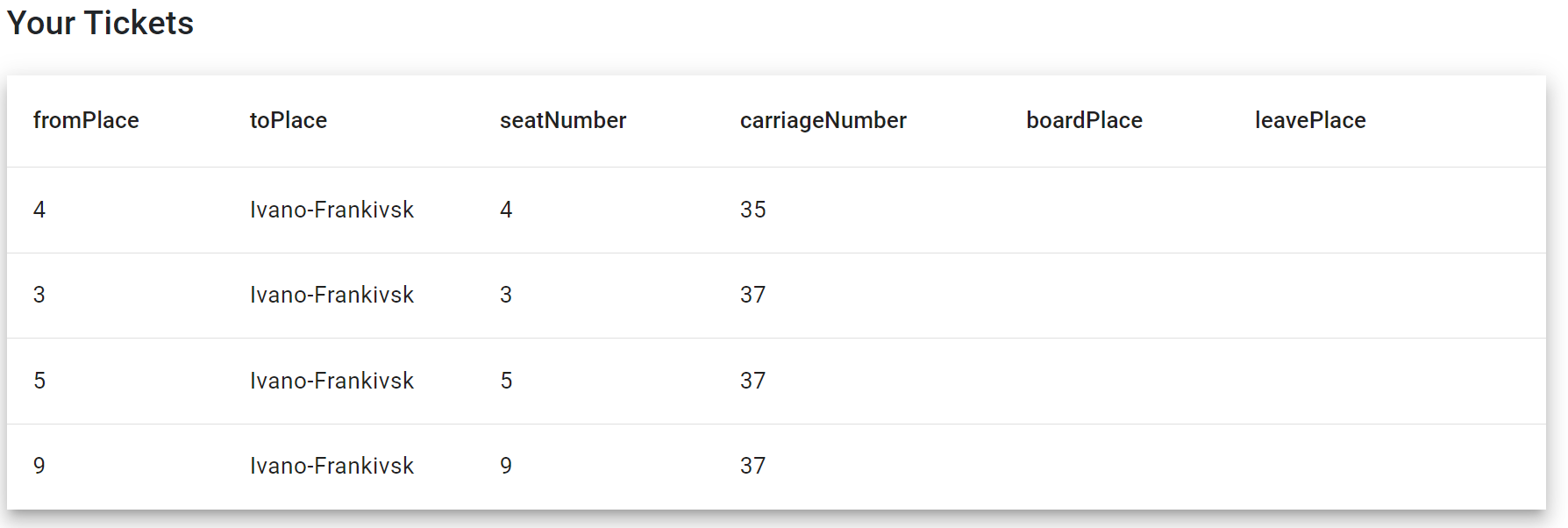
 Рисунок 4.6 – Інформаційне повідомлення

Інструкція для користувача як подивитися придбані квитки:

1. Користувач натисне на своє ім'я у заголовка навігації(header) (дивіться рисунок 4.7).

 Рисунок 4.7 – Інформаційне повідомлення

Після натискання він перейде у свій особистий кабінет де зможе побачити придбані їм квитки(дивіться рисунок 4.8).

 Рисунок 4.8 – Інформаційне повідомлення

Інструкція для адміністратора як завантажити бекап файл:

1. Адміністратор реєструється в систему

В меню обирає опцію Back up(дивіться рисунок 4.9).

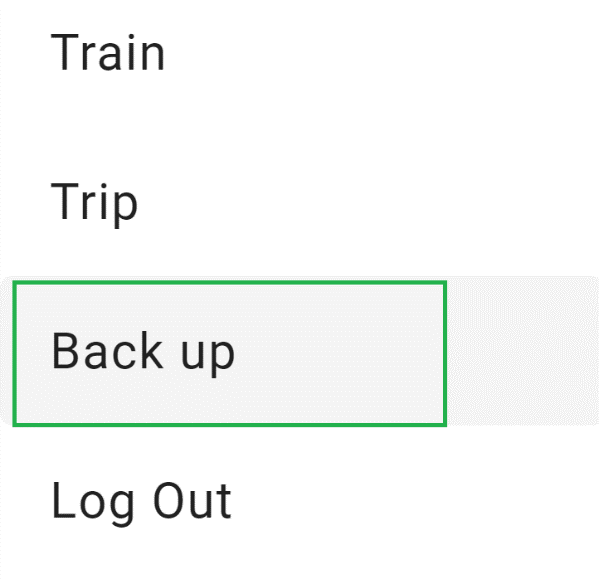


Рисунок 4.9 – Опція Back up

З перелічених опцій не натискає на кнопку зробити бекап після чого бекап файл завантажиться йому на комп'ютер(дивіться рисунок 4.10).

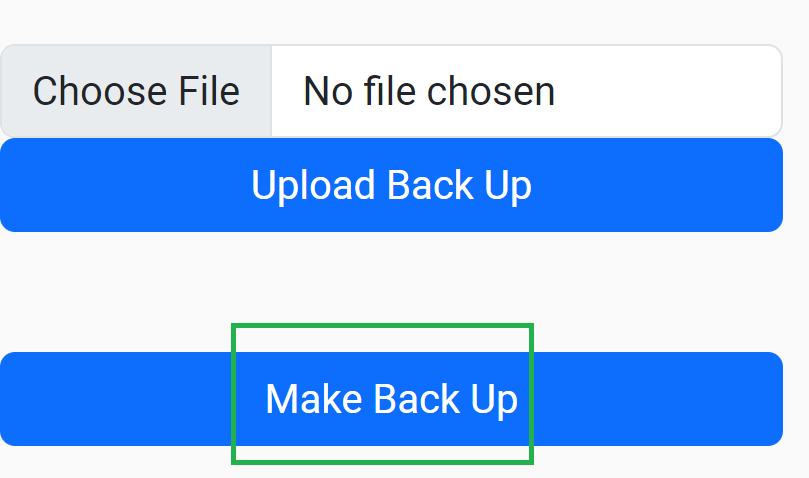


Рисунок 4.10 – Кнопка make back up

Інструкція для адміністратора як змінити поле таблиці:

адміністратор повинен перейти на вкладку, що його цікавить після натискання йому відобразиться таблиця і на полі яке він хоче змінити він повинен натиснути кнопку Edit(дивіться рисунок 4.11).

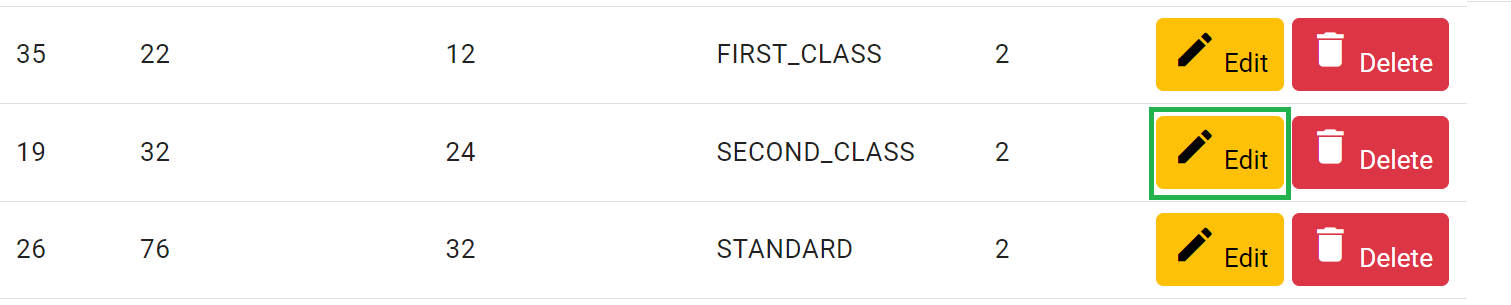
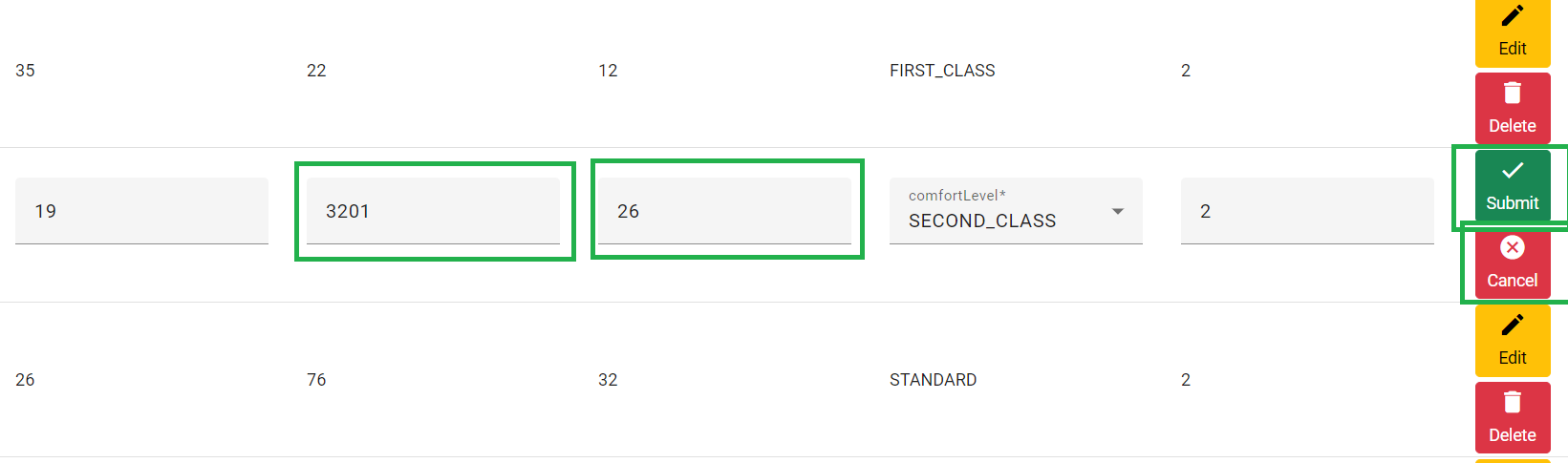


Рисунок 4.11 – Кнопка Edit

Після зміни даних натиснути або кнопку підтвердити щоб зберегти ці змінні або кнопку відмінити щоб повернутися до старих даних(дивіться рисунок 4.12).

 Рисунок 4.12 – Оновлення даних

при натисканні підтвердити адміністратор отримує відповідне повідомлення що оновлення пройшло успішно також таблиця оновиться(дивіться рисунок 4.13).

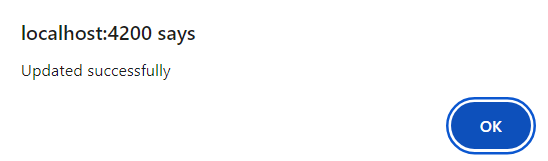


Рисунок 4.13 – Повідомлення про успішне оновлення

Інструкція для адміністратора як додати новий запис в таблицю:

Адміністратор повинен перейти на на вкладку, що його цікавить наприклад подорож(дивіться рисунок 4.14).

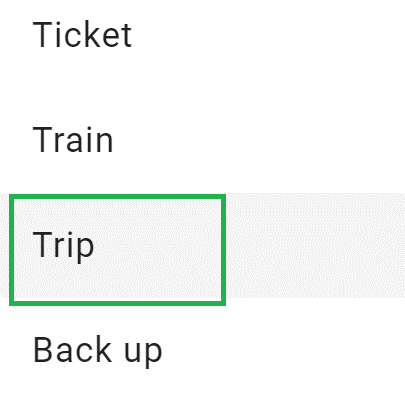


Рисунок 4.14 – Вкладка подорож

Потім адміністратор повинен вести всі необхідні дані та натиснути кнопку додати(дивіться рисунок 4.15).

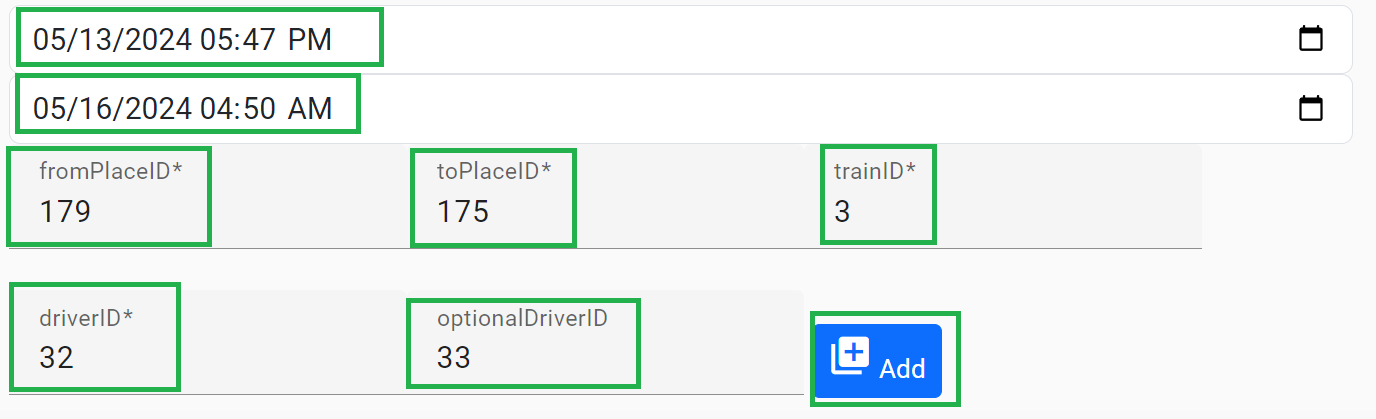


Рисунок 4.15 – Додавання нової нового запису до таблиці

після чого якщо немає помилок отримає повідомлення про успішне додавання(дивіться рисунок 4.16).

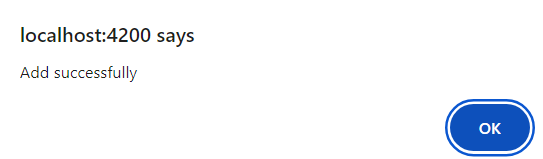


Рисунок 4.15 – Додавання нової нового запису до таблиці