

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА**  
**ФРАНКА**

**ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**Кафедра Цифрової економіки та бізнес-аналітики**

**КУРСОВА РОБОТА**

з навчальної дисципліни

Проектування та адміністрування БД та СД

на тему:

**Інформаційна система прийому і обслуговування клієнтів готелю**

спеціальність: 051 «Економіка»

спеціалізація: «Інформаційні технології в бізнесі»

освітній ступінь: бакалавр

**Науковий керівник:**

К.ф.м. н. доцент Депутат Б. Я

\_\_\_\_\_ “\_\_” \_\_\_\_\_ 20\_\_ р.

**Виконавець:**

студент групи УФЕ-31с

Малецький О. М.

\_\_\_\_\_ “\_\_” \_\_\_\_\_ 20\_\_ р.

**Загальна кількість балів** \_\_\_\_\_

**ЛЬВІВ 2022**

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ .....	5
1.1 Постановка завдання.....	5
1.2 Розробка моделі варіантів використання веб-сайту .....	6
1.3 Аналіз ринку конкурентів .....	8
1.4 Business Model Canvas .....	10
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ .....	13
2.1 Опис моделі даних .....	13
2.2 Нормалізація відношень .....	17
2.3 Визначення типів даних .....	19
2.4 Обмеження цілісності даних.....	22
2.5 Реалізація SQL-скрипту.....	24
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ.....	29
3.1 Структура веб-сайту та макет сторінок .....	29
3.2 Програмування серверної та клієнтської частини.....	32
ВИСНОВКИ.....	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	35
ДОДАТКИ.....	37

## ВСТУП

**Актуальність теми дослідження.** На сьогоднішній день інформаційні технології відіграють неабияку роль в сучасному світі. Вони впливають на наше життя у всіх процесах: при оплаті проїзду в трамваї, листаючи фото і соцмережах та багато іншого. При такому стрімкому розвитку інформаційні системи стають все більше і більше популярними та важливими. Інформаційна система – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. За допомогою інформаційних систем ми можемо легко отримувати доступ до інформації, використовувати її для полегшення життя та економії часу. Зараз кожна бізнес модель вимагає розробки інформаційної системи для кращого та зручного адміністрування та управління бізнес-процесами. Суспільство настільки сильно пристосувалося до різноманітних інформаційних систем, що вже стає важко уявити, яке б наше життя було у разі їх відсутності.

У випадку розробки інформаційної системи прийому і обслуговування клієнтів готелю було розроблено бізнес-модель, у якій на перше місце ставились потреби користувача. Така інформаційна система є необхідною і майже кожен власний готелю ставить собі за мету розробити таку систему задля покращення роботи, просування бренду та отримання більшої вигоди. Отже, можна з впевненістю сказати, що розробка інформаційної системи прийому і обслуговування клієнтів готелю є актуальною і допоможе гостям легко та зручно отримати будь-яку необхідну інформацію та забронювати бажаний номер.

**Мета та завдання дослідження.** Метою даної курсової роботи є вивчення теоретичних навичок по системах адміністрування БД та розробка інформаційної системи з використанням веб-сайту.

Для досягнення мети роботи, потрібно виконати наступні завдання:

1. Проаналізувати варіанти використання веб-сайту, розробивши Use Case Diagram;
2. Спроектувати базу даних, створити ER-діаграму;

3. Окреслити сутності бази даних та її зв'язки;
4. Розробити структуру веб-сайт та зробити його макет;
5. Запрограмувати серверну та клієнтську частину;

**Об'єкт дослідження.** Особливість розробки інформаційної системи для прийому та обслуговування клієнтів готелю за допомогою системи управління базами даних Microsoft SQL Server.

**Предмет дослідження.** Застосування різноманітних технологій та теоритичних аспектів для створення бази даних з веб-сайтом для прийому та обслуговування клієнтів готелю.

**Практичне значення отриманих результатів.** Під час виконання курсової роботи було розроблену інформаційній систему прийому і обслуговування клієнтів готелю використовуючи спроектовану базу даних та веб сайт.

**Використане програмне забезпечення.** Під час розробки використовувалось середовище розробки Visual Studio Code, платформи Node.js, мова розмітки HTML, CSS, та мова програмування JavaScript. Для проектування бази даних було використано Microsoft SQL Server Management Studio.

**Структура роботи.** Курсова робота складається з вступу, трьох розділів, а саме «Аналіз вимог», «Розробка бази даних» та «Розробка веб-додатку», а також висновків, списку використаних джерел та додатів. Загальний обсяг роботи – 67 сторінок.

## РОЗДІЛ 1. АНАЛІЗ ВИМОГ

### 1.1 Постановка завдання

Інформаційні технології мають величезний вплив на формування ведення бізнесу. Більше того, за допомогою інформаційних систем здійснюється різноманітні процеси: введення інформації, її опрацювання, зберігання опрацьованої інформації, виведення інформації, призначеної для користувача, відправка та отримання інформації мережею.

На сьогоднішній день неможливо уявити ефективне функціонування будь-якого готельного бізнесу без використання програмного забезпечення, що дає можливість автоматизувати деякі бізнес-процеси. Використання сучасних інформаційних технологій в розробці інформаційної системи прийому та обслуговування клієнтів готелю – це не просто створення красивої обгортки, але і створення конкуренції.

Для розуміння, що ставиться перед нами в якості завдання потрібно спершу провести аналіз з метою розуміння бізнес-логіки та бізнес потреб інформаційної системи. На даному етапі, в нас є один готель під назвою «Lumia», який розташований у місті Львів, який займається прийомом та обслуговуванням клієнтів. Готель не є великим, він містить всього 75 номерів різних категорій: три типи стандартних, напівлюкс та люкс.

Розробка інформаційної системи прийому та обслуговування клієнтів готелю передбачає створення бази даних, яка буде містити всю необхідну інформацію про гостей, номери готелю та інші необхідні дані. Наступний крок передбачає створення веб-додатку для користувача для використання та взаємодії з інформацією з бази даних.

Для роботи інформаційної системи потрібні наступні дані: інформація про гостя, його персональні дані та бажаний номер в готелі для замовлення, інформація про номер, її ціна та опис, вільні дати для проживання. Інформаційна система повинна взаємодіяти з базою даних через сервер, виконувати запити.

## 1.2 Розробка моделі варіантів використання веб-сайту

Інформаційна система використовується для взаємодії користувача, а саме гостя, з сайтом готелю. Наявність веб-сайту прийому та обслуговування клієнтів готелю - це величезна кількість переваг, адже на веб-сайті можна переглянути всю інформацію про готель, варіанти номерів, які є доступні в готелі, гість має можливість забронювати той чи інший номер. Також наявність веб-сайту допомагає збільшити популярність готелю, залучити нових гостей. Більше того, веб-сайт допомагає підвищити імідж бренду у суспільстві, адже наявність веб-сайт в Інтернеті означає, що Ваша компанія йде у ногу з інноваціями та відповідає всім вимогам сучасного суспільства.

На сьогоднішній день інформаційна система є на етапі розробки та має один варіант використання, а саме використання в ролі гостя. Гість має можливість зайти на сайт, переглянути інформацію про готель, номери, контактну інформацію, також можна забронювати певний номер. Однак є можливість доопрацювання моделі та розробки варіанту використання в ролі менеджера, який буде мати можливість підтвердження чи скасування бронювання певного гостя, редагувати інформацію про номери.

Задля кращого розуміння варіантів використання інформаційної системи використовується Use Case Diagram (діаграма предентів або діаграма варіантів використання). Ця діаграма корисна тим, що на ній зображено відношення між акторами та прецедентами (варіантами використання) в системі. Варіанти використання допоміжні тим, що з їхньою допомогою можна легко побачити функціональні можливості кожного актора в інформаційній системі.

Розробляючи Use Case Diagram було використано Visual paradigm online. Варіанти використання мають різноманітні види відносин. Визначення зв'язку є дуже важливою складовою при розробці, цим займається бізнес-аналітик проекту. При розробці Use Case Diagram є наступні види відношень між акторами та прецедентами:

1. Асоціації (association relationship);

2. Включення (include relationship);
3. Розширення (англ. extend relationship);
4. Узагальнення (англ. generalization relationship).

Use Case Diagram наведена на рисунку 1.1.

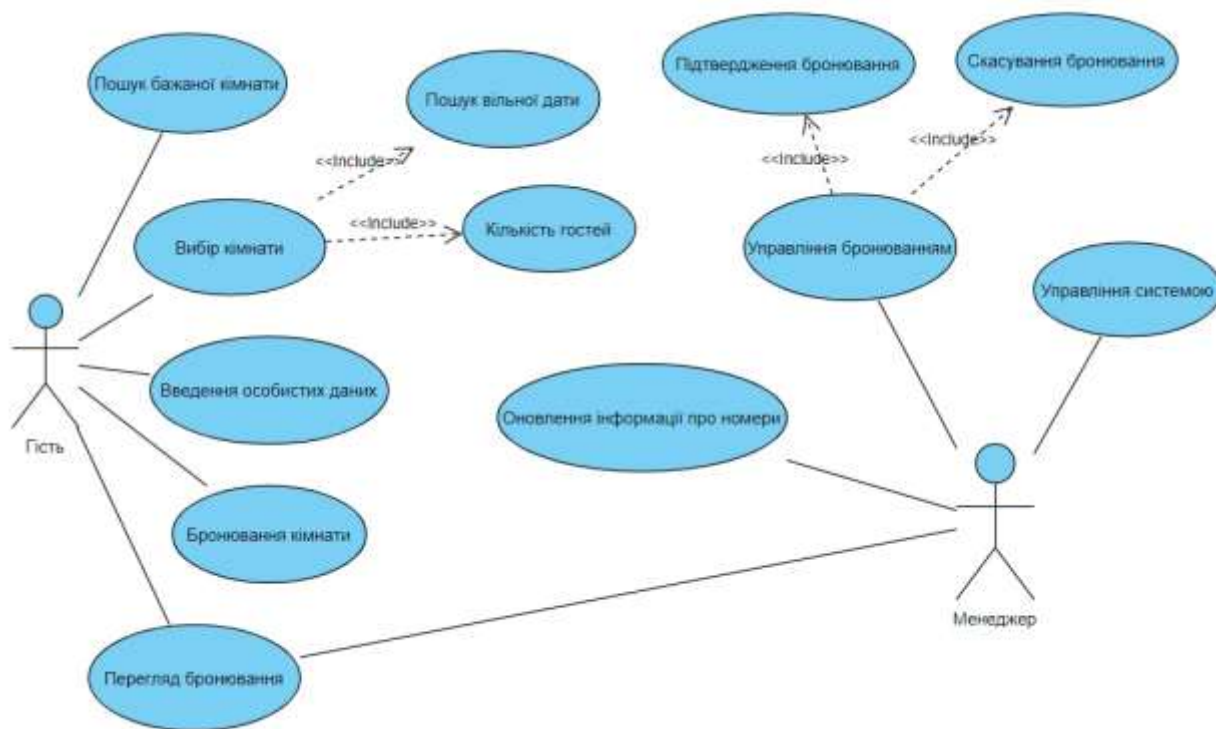


Рис. 1.1 Use Case Diagram (діаграма варіантів використання)

Розглянувши діаграму прецедентів можна побачити, що в ній описано варіанти використання для двох акторів:

1. Гість – головний користувач інформаційної системи
  - Пошук бажаної кімнати – гість має можливість переглянути номери, та порівняти їх між собою;
  - Вибір кімнати – гість може обрати ту кімнату, який підійде по його пошуковим критеріях. Цей use case включає в себе:
    - Пошук вільної дати;
    - Вибір кількості гостей;

- Введення особистих даних – після обирання тої чи іншої кімнати, гість повинен ввести свою особисту інформацію (ім'я, прізвище, номер телефону, електронна пошта);
  - Бронювання кімнати – ввівши всю необхідну інформацію клієнт повинен забронювати кімнату натиснувши кнопку «підтвердити»;
  - Перегляд бронювання – гість має можливість переглянути бронювання.
2. Менеджер – займається адмініструванням інформаційної системи та підтверджує бронювання
- Оновлює інформацію про номери – менеджер займається редагуванням номерів, змінює опис, ціну і іншу необхідну інформацію;
  - Управління системою – менеджер має можливість переглядати різноманітну інформацію:
  - Управління бронюванням – менеджер має можливість управляти бронюванням. Цей use case включає в себе:
    - Підтвердження бронювання;
    - Скасування бронювання.

### **1.3 Аналіз ринку конкурентів**

SWOT-аналіз – це в першу чергу, метод стратегічного планування, що дозволяє визначити сильні і слабкі сторони бізнесу, розрахувати ризики і майбутнє процвітання за рахунок всебічного підходу. Цей аналіз передбачає поділ факторів впливу на 4 групи:

1. S – strengths: сильні сторони проекту. Наприклад, конкурентна ціна, унікальний продукт, кращий на ринку сервіс, бездоганна репутація, інноваційні технології і т.д .;
2. W – weaknesses: слабкі сторони. Наприклад, висока собівартість продукту, вузька цільова аудиторія та інше.
3. O – opportunities: можливості. Наприклад, конкурент закриває всі магазини у вашому місті і всі потенційні покупці звертаються до вас;



4. Т – threats: загрози. Наприклад, у конкурентів вищі рекламні бюджети, товар легший у використанні і т.д. Важливо відзначити, що перші два фактори SWOT-аналізу в маркетингу відносяться до внутрішніх, а останні два – до зовнішніх. Зовнішні елементи відносяться до політичної ситуації, суспільства, економіки і т.д.

Перевагою цього аналізу є те, що цей аналіз універсальний. Незалежно від того який бізнес ви ведете, які послуги надаєте, який продукт робите – він вам підійде. SWOT-аналіз може провести власник, менеджер, директор. Не потрібно проводити складні обчислення – важливо адекватно і чітко оцінювати всі сторони. SWOT-аналіз враховує і внутрішні і зовнішні чинники, дає можливість охопити стан справ в цілому, а не з одного боку.

Якщо говорити про проект інформаційної системи прийому та обслуговування клієнтів готелю, то за допомогою SWOT-аналізу було проведено аналіз одного з конкурентів.

Таблиця 1.1  
SWOT-аналіз готелю «Львів»

<p><b>Strengths</b></p> <ol style="list-style-type: none"> <li>1. Відомий бренд, якому довіряють</li> <li>2. Наявність ресторану</li> <li>3. Є SPA та інші сервіси для збільшення прибутку</li> <li>4. Один з перших готелів в місті, вигідне розташування.</li> </ol>	<p><b>Weaknesses</b></p> <ol style="list-style-type: none"> <li>1. Застарілий дизайн сайту, що неприпустимо для готелю такого рівня.</li> <li>2. Не зовсім трендовий інтер'єр номерів</li> <li>3. Висока вартість номерів</li> <li>4. Недостатня кваліфікація персоналу.</li> </ol>
<p><b>Opportunity</b></p> <ol style="list-style-type: none"> <li>1. Оновлення техніки в номерах</li> <li>2. Розширення цільової аудиторії та залучення нових клієнтів</li> <li>3. Розширення асортименту SPA-послуг та інших сервісів.</li> <li>4. Збільшення доходів населення – зростання кількості клієнтів.</li> </ol>	<p><b>Threats</b></p> <ol style="list-style-type: none"> <li>1. Нестабільна ситуація у зв'язку з пандемією.</li> <li>2. Зміна вподобань постійних клієнтів</li> <li>3. Зростання конкуренції – все частіше відкриваються більш дешеві та сучасні альтернативи.</li> <li>4. Збільшення витрат на надання послуг (комунальні послуги).</li> </ol>

4Р-аналіз – це аналіз який включає в себе 4 основні елементи: product, price, promotion, place. Елемент «Product» відповідає на питання «Що необхідно ринку або цільовій аудиторії?», елемент «Price» допомагає визначити вартість продажу

товару і оцінити рівень рентабельності продажів, елемент «Place» допомагає вибудувати правильну модель дистрибуції (або доставки товару до кінцевого споживача), а елемент «Promotion» відповідає на питання «Яким способом інформація про товар компанії чи послугу буде поширюватися на ринку?».

Якщо говорити про проект інформаційної системи прийому та обслуговування клієнтів готелю, то за допомогою 4P-аналізу було проведено аналіз одного з конкурентів.

Таблиця 1.2

## 4P-Аналіз «Євроготель»

<b>Product</b> <ol style="list-style-type: none"> <li>1. Готельно-ресторанні послуги</li> <li>2. Бронювання номерів для проживання</li> <li>3. Надання конференц-залів</li> <li>4. Можливість проведення бенкетів в ресторані.</li> </ol>	<b>Price</b> <ol style="list-style-type: none"> <li>1. Достатньо високі ціни, орієнтовані на преміум клієнтів.</li> <li>2. У ціну включені багато послуг і сервісів, таких як мінібар, безкоштовний Wi-Fi і т.д.</li> <li>3. При можливості можна замовити сніданок за додаткову плату.</li> <li>4. Знижки для постійних клієнтів.</li> </ol>
<b>Place</b> <ol style="list-style-type: none"> <li>1. Один готель у місті Львів, на вулиці Тершаковців 6А, близько до центру міста.</li> <li>2. Паркувальні місця для власного транспорту.</li> </ol>	<b>Promotion</b> <ol style="list-style-type: none"> <li>1. Сторінки у соцмережах.</li> <li>2. Співпраця з медійними особами.</li> <li>3. Присутність в готельних агрегаторах, таких як Booking.</li> </ol>

Якщо ж уважно вивчити ці два аналізи, можна з легкістю дізнатись, які практики є корисні при розробці інформаційної системи та її бізнес-логіки, а що не варто використовувати. Отже, SWOT та 4P-аналіз є надзвичайно корисними при розробці бізнес-моделі.

### 1.4 Business Model Canvas

Business Model Canvas – це візуальне уявлення бізнес-моделі, що виділяє всі ключові стратегічні фактори. Іншими словами, це загальний, цілісний і повний огляд роботи компанії, клієнтів, потоків доходів тощо. Окрім надання загального огляду бізнес-моделі, ці модель дозволяє компаніям візуалізувати та

аналізувати свою стратегію. Це включає оновлення моделі в міру розвитку компанії, наприклад, зміни на ринку, нові потоки або розширення.

Business Model Canvas надає центральне, загальне джерело знань. Це шаблон, який визначає як кожен розділ взаємодіє з іншими. Наприклад, розуміння ціннісної пропозиції, цільового клієнта та каналів, через які вони залучаються, потрібно аналізувати разом, а не лише в окремих вакуумах.

Крім того, канва бізнес-моделі може використовуватися організаціями для планування, оцінки або виконання нових моделей взагалі. Таким чином, модель підкреслює ключові моменти і гарантує, що жодні життєво важливі фактори не будуть забуті.

Business Model Canvas інформаційної системи прийому та обслуговування клієнтів готелю (рис. 1.2):

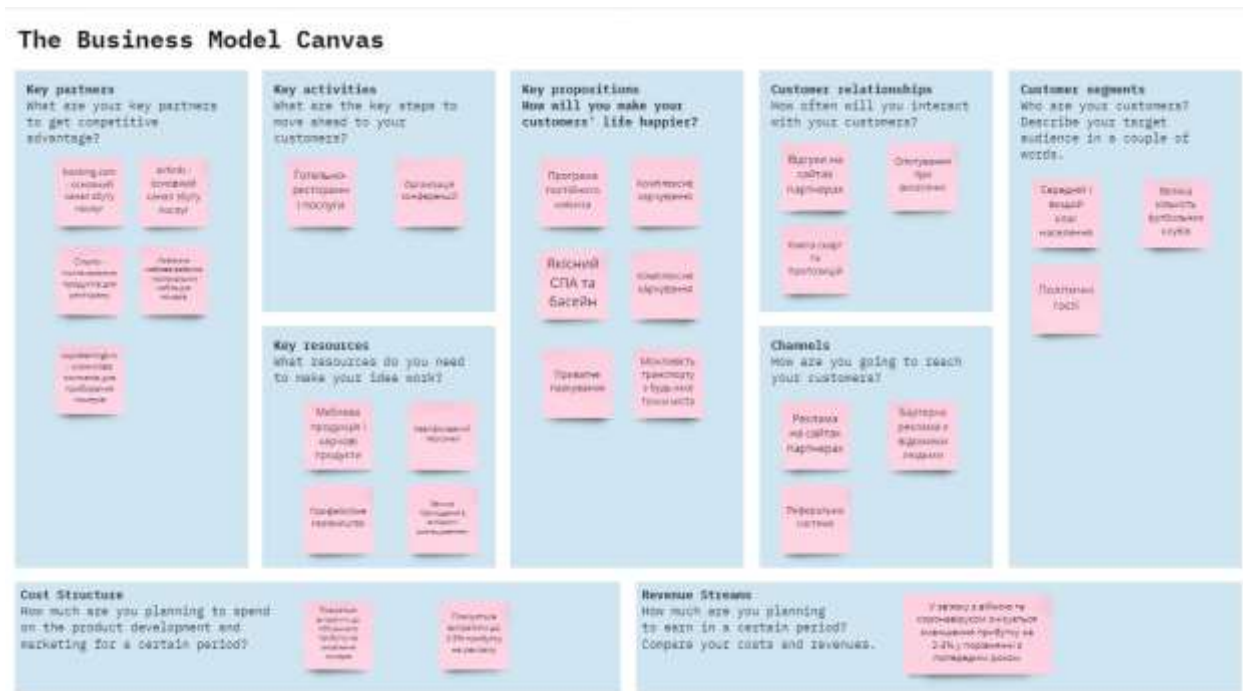


Рис. 1.2 Business Model Canvas

Business Model Canvas показує 9 секцій, кожна з яких описує щось конкретне, а саме:

1. Ключові партнери (Key partners);
2. Ключові діяльності (Key activities);
3. Ключові пропозиції (Key propositions);

4. Відносини з клієнтами (Customer relationships);
5. Канали (Channels);
6. Клієнтські сегменти (Customer segments);
7. Витратна структура (Cost Structure);
8. Структура витрат (Revenue Streams).

## РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

### 2.1 Опис моделі даних

База даних — це певний набір даних, які пов'язані між собою спільною ознакою або властивістю, та впорядковані. Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках — це допомагає працювати з інформацією простіше і швидше.

Бази даних для сайтів дають змогу зберігати інформацію, що виглядає як зв'язані між собою таблиці. Саме в БД зберігаються вся необхідна та корисна інформація для функціонування сайту.

Щоб створити запит до бази даних часто використовують Structured Query Language. SQL дає змогу додавати, редагувати та видаляти інформацію, що міститься у таблицях.

Найбільш популярною системою управління є MySQL, однак при розробці БД до нашої інформаційної системи ми використали Microsoft SQL Server.

Модель даних – це основа бази даних. Модель даних – фіксована система понять і правил для представлення даних структури, стану і динаміки проблемної області в базі даних. У різний час послідовне застосування одержували ієрархічна, мережна і реляційна моделі даних.

Існують наступні види моделей даних:

1. Ієрархічна модель. Ця модель будується у вигляді ієрархічної деревоподібної структури, у якій для кожного головного об'єкта існує кілька підлеглих, а для кожного підлеглого об'єкта може бути тільки один головний. Концептуальна схема ієрархічної моделі являє собою сукупність типів записів, пов'язаних типами зв'язків в одному чи кількох

деревами. Усі типи зв'язків цієї моделі належать до виду «один до декількох» і зображуються у вигляді стрілок.

2. Мережна модель. У мережній моделі один і той же об'єкт може одночасно виступати як у ролі головного, так і підлеглого елемента. Це означає, що кожний об'єкт може брати участь у довільній кількості зв'язків. Зв'язок у цьому випадку може встановлюватися явно, коли значення деяких полів є посилання на дані, що містяться в іншому файлі. Прикладом мережної структури БД може бути структура автобусних маршрутів (із будь-якого населеного пункту існують маршрути в інші).
3. Реляційна модель. У реляційній моделі дані й взаємозв'язки між ними подаються за допомогою прямокутних таблиць. Рядки в реляційній базі даних називають записами, а стовпці — полями. Модель двовимірної таблиці дозволяє звертатися до даних як по рядках, так і по стовпцях, що є значною перевагою. Порядок розміщення рядків і стовпців у таблиці довільний; таблиця такого типу називається відношенням. Оскільки реляційна структура концептуально проста, вона дозволяє реалізовувати невеликі і прості (і тому легкі для створення) бази даних, навіть персональні, сама можливість реалізації яких ніколи навіть і не розглядалася в системах з ієрархічною чи мережною моделлю.

Практично всі існуючі на сьогоднішній день комерційні бази даних і програмні продукти для їх створення використовують реляційну модель даних.

При розробці інформаційної системи використовувалась реляційна модель даних. Для візуального відображення структури створено ER-діаграми (рис. 2.1).



Рис. 2.1 Діаграма сутностей та зв'язків

На рис. 2.1 можна побачити, що дана модель складається з 9 таблиць: room, roomCleaners, cleaners, guest, reservation, manager, guestKitchen, kitchen, amenity.

Room таблиця, яка містить інформацію про номери готелю, полями якої є:

1. id – ідентифікаційний код номера;
2. room\_type – тип номера;
3. description – опис номера;
4. price – ціна номера;
5. max-person – максимальна кількість людей.

RoomCleaners проміжна таблиця, яка містить інформацію про ідентифікаційні коди номерів та покоївок, містить наступні поля:

1. id – ідентифікаційний код;
2. room\_id – ідентифікаційний код номера;
3. cleaners\_id – ідентифікаційний код покоївки;

Cleaners таблиця, яка містить інформацію про покоївок, полями якої є:

1. id – ідентифікаційний код покоївки;
2. firstname – ім'я покоївки;
3. lastname – прізвище покоївки;
4. phone – телефон покоївки;
5. salary\_for\_one\_room – зарплата покоївки за одну прибрану кімнату;

Guest таблиця, яка містить інформацію про гостей готелю, містить поля:

1. id – ідентифікаційний код гостя;
2. firstname – ім'я гостя;
3. lastname – прізвище гостя;
4. phone – телефон гостя;
5. email – електронна пошта гостя;
6. discount – знижка клієнта (якщо присутня).

Reservation таблиця, яка містить інформацію про резервації номерів, містить поля:

1. id – ідентифікаційний код резервації;
2. room\_id – ідентифікаційний код номера;
3. guest\_id – ідентифікаційний код гостя;
4. number\_of\_guest – кількість гостей;
5. check\_in\_date – дата заїзду;
6. check\_out\_date – дата виїзду;
7. manager\_id – ідентифікаційний код менеджера;
8. data\_of\_reservation – дата резервації.

Manager таблиця, яка містить інформацію про менеджерів готелю, містить поля:

1. id – ідентифікаційний код менеджера;
2. firstname – ім'я менеджера;
3. lastname – прізвище менеджера;
4. phone – телефон менеджера;
5. email – електронна пошта менеджера;



6. salary – зарплата менеджера.

GuestKitchen проміжна таблиця, яка містить інформацію про ідентифікаційні коди гостей та страв, містить наступні поля:

1. id – ідентифікаційний код;
2. guest\_id – ідентифікаційний код гостя;
3. kitchen\_id – ідентифікаційний код страви.

Kitchen таблиця, яка містить інформацію страви, містить наступні поля:

1. id – ідентифікаційний код страви;
2. dish\_name – назва страви;
3. dish\_price – ціна страви.

Amenity таблиця, яка містить інформацію про додаткові послуги, містить наступні поля:

1. id – ідентифікаційний код типу додаткової послуги;
2. feed\_type – тип харчування;
3. is\_spa – наявність SPA;
4. amenity\_sum – ціна додаткової послуги.

## 2.2 Нормалізація відношень

Нормалізація – це процес приведення структури реляційних відносин до форми, яка має кращі властивості при включенні, зміні і видаленні даних. Остаточна мета нормалізації зводиться до отримання такого проекту БД, в якому кожен факт з'являється лише в одному місці, тобто виключена надмірність інформації.

Крім завдання більш ефективного використання пам'яті нормалізація дозволяє знизити загрозу порушення цілісності БД через появу в ній внутрішніх суперечностей.

Вводиться поняття нормальних форм (НФ) відносин, кожної з яких відповідає певний набір обмежень. Відношення знаходиться в даній нормальній

формі, якщо задовольняє зазначеним обмеженням. Кожна наступна НФ включає в себе вимоги всіх попередніх, тобто є більш суворим обмеженням.

У реляційних баз даних є кілька нормальних форм (НФ):

1. Перша нормальна форма (1НФ). Першим кроком нормалізації є приведення відношення до першої нормальної форми. Відношення в 1НФ повинно відповідати таким вимогам:

- усі атрибути відношення повинні бути унікальними, тобто не допускається їхнього дублювання, а також атомарними, тобто неподільними;
- усі рядки таблиці повинні мати однакову структуру;
- імена стовпців повинні бути різними, а значення однорідними (однакового типу і формату);
- порядок рядків у таблиці не істотний.

Отже, будь-яка нормалізована таблиця буде знаходитись в першій нормальній формі (1НФ).

2. Друга нормальна форма (2НФ). Відношення знаходиться у другій нормальній формі тоді і тільки тоді, коли відношення перебуває в 1НФ і всі його неключові атрибути функціонально повно залежать від первинного ключа. Неключовий атрибут – це атрибут, що не входить до складу первинного ключа.

Перевага 2НФ – це зручність внесення змін у базу даних. 2НФ повністю виключає можливість виникнення протиріччя даних, а також економить пам'ять.

3. Третя нормальна форма (3НФ). Відношення знаходиться у третій нормальній формі тоді і тільки тоді, коли воно перебуває у 2НФ і всі його неключові атрибути взаємно функціонально незалежні.

Переваги 3НФ – це те, що виключається надлишкове дублювання інформації. Тому можна зробити ще такий висновок: відношення знаходиться в 3НФ, якщо зміна значення будь-якого його атрибута (крім

тих, що входять у первинний ключ) не призведе до необхідності зміни значень інших полів.

4. Нормальна форма Бойса – Кодда. Нормальна форма Бойса–Кодда – це підсилена ЗНФ, у якій вивчаються залежності ключових атрибутів від неключових. Відношення перебуває в НФБК, якщо воно перебуває в ЗНФ і в ньому відсутні залежності ключових атрибутів від неключових атрибутів. Але відношення в ЗНФ не завжди можна привести до нормальної форми Бойса–Кодда, не втративши залежності між його атрибутами.
5. Четверта нормальна форма (4НФ). Якщо відношення має багатозначні залежності між атрибутами, то виконують його декомпозицію й отримують 4НФ.
6. П'ята нормальна форма (5НФ). Відношення знаходиться в п'ятій нормальній формі тоді і тільки тоді, коли воно знаходиться в 4НФ і кожна нетривіальна залежність визначається її потенційним ключем.

### **2.3 Визначення типів даних**

При створенні таблиць в Microsoft SQL Server Management потрібно спершу визначити типи даних для стовпців. В SQL Server в кожного стовпця є певний тип даних. Тип даних – це атрибут, який визначає, які дані можуть зберігатися з об'єкти: цілі числа, символи, дата і час і т.д

Основні типи даних, які використовувались при розробці:

1. INT – від -2 147 483 648 до 2 147 483 647;
2. VARCHAR – до 8000 символів;
3. DATE – від 0001-01-01 до 9999-12-31 (PPPP-ММ-ДД);

Типи даних, які використовувались подані в таблицях:

Таблиця 2.1

Склад таблиці «room»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	room_type	VARCHAR(50)
3	description	VARCHAR(250)
4	price	INT
5	max_person	INT

В таблиці 2.2 можна побачити, що «roomCleaners» містить 3 атрибути.

Таблиця 2.2

Склад таблиці «roomCleaners»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	room_id	INT
3	cleaners_id	INT

Нижче подана таблиця демонструє, що таблиця «cleaners» містить інформацію про покоївок і має 5 атрибутів.

Таблиця 2.3

Склад таблиці «cleaners»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	firstname	VARCHAR(50)
3	lastname	VARCHAR(50)
4	phone	INT
5	salary_for_one_room	INT

Таблиця 2.4 описує таблицю з баз даних «guest», це одна з ключових таблиць в якій фігурують два типи даних.

Таблиця 2.4

Склад таблиці «guest»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	firstname	VARCHAR(50)
3	lastname	VARCHAR(50)
4	phone	INT

5	email	VARCHAR(100)
6	discount	INT

Таблиця 2.5 показує атрибути та типи даних ключової таблиці «reservation».

Таблиця 2.5

Склад таблиці «reservation»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	room_id	INT
3	guest_id	INT
4	amenity_id	INT
5	number_of_guest	INT
6	check_in_date	DATE
7	check_in_date	DATE
8	manager_id	INT
9	date_of_reservation	DATE

Нижче подана таблиця 2.6 інформує про типи даних таблиці «manager».

Таблиця 2.6

Склад таблиці «manager»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	firstname	VARCHAR(50)
3	lastname	VARCHAR(50)
4	phone	INT
5	email	VARCHAR(50)
6	salary	INT

3 типи даних та атрибути наведені у таблиці 2.7.

Таблиця 2.7

Склад таблиці «guestKitchen»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	guest_id	INT
3	kitchen_id	INT

Таблиці 2.8 показує атрибути та типи даних таблиці «kitchen».

Таблиця 2.8

Склад таблиці «kitchen»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	dish_name	VARCHAR(50)
3	dish_price	INT

Склад таблиці «amenity» наведений у таблиці 2.9.

Таблиця 2.9

Склад таблиці «amenity»

№	Ім'я атрибуту	Тип даних
1	id	INT
2	feed_type	VARCHAR(100)
3	is_spa	VARCHAR(5)
4	amenity_sum	INT

## 2.4 Обмеження цілісності даних

Цілісність даних означає, що в базі даних встановлено і коректно підтримуються взаємозв'язки між записами різних таблиць при завантаженні, додаванні і видаленні записів у зв'язаних таблицях, а також при зміні значень ключових полів.

Цілісність бази даних – це відповідність інформації, що знаходиться в базі даних, її внутрішній логіці, структурі і всім явно заданим правилам.

В SQL є такі типи цілісності даних:

1. PRIMARY KEY - Первинний ключ. Використовується для ідентифікації рядків в таблиці, однак є певні особливості:
  - В одній таблиці може бути тільки один первинний ключ;
  - Кожне значення первинного ключа має бути унікальним;
  - Значення первинного ключа не може бути.
2. FOREIGN KEY - Зовнішній ключ. Використовується для того, щоб зв'язати дві таблиці.

3. NOT NULL/NULL – Заборона або дозвіл на введення в поле NULL-значень.
4. UNIQUE - Контроль на унікальність атрибутів.
5. CHECK - Контроль на допустимість атрибутів.

Під час розробки даної бази даних було використано певну кількість primary і foreign keys для зв'язування таблиць:

1. cleaners.id (PK) = roomCleaners.cleaners\_id (FK);
2. room.id (PK) = roomCleaners.room\_id (FK)
3. room.id (PK) = reservation.room\_id (FK);
4. guest.id (PK) = reservation.guest\_id (FK);
5. manager.id (PK) = reservation.manager\_id (FK);
6. guest.id (PK) = guestKitchen.guest\_id (FK);
7. kitchen.id (PK) = guestKitchen.kitchen\_id (FK);
8. amenity.id (PK) = reservation.amenity\_id (FK);

Таблиці з використанням РК та FK подані на рис. 2.2.

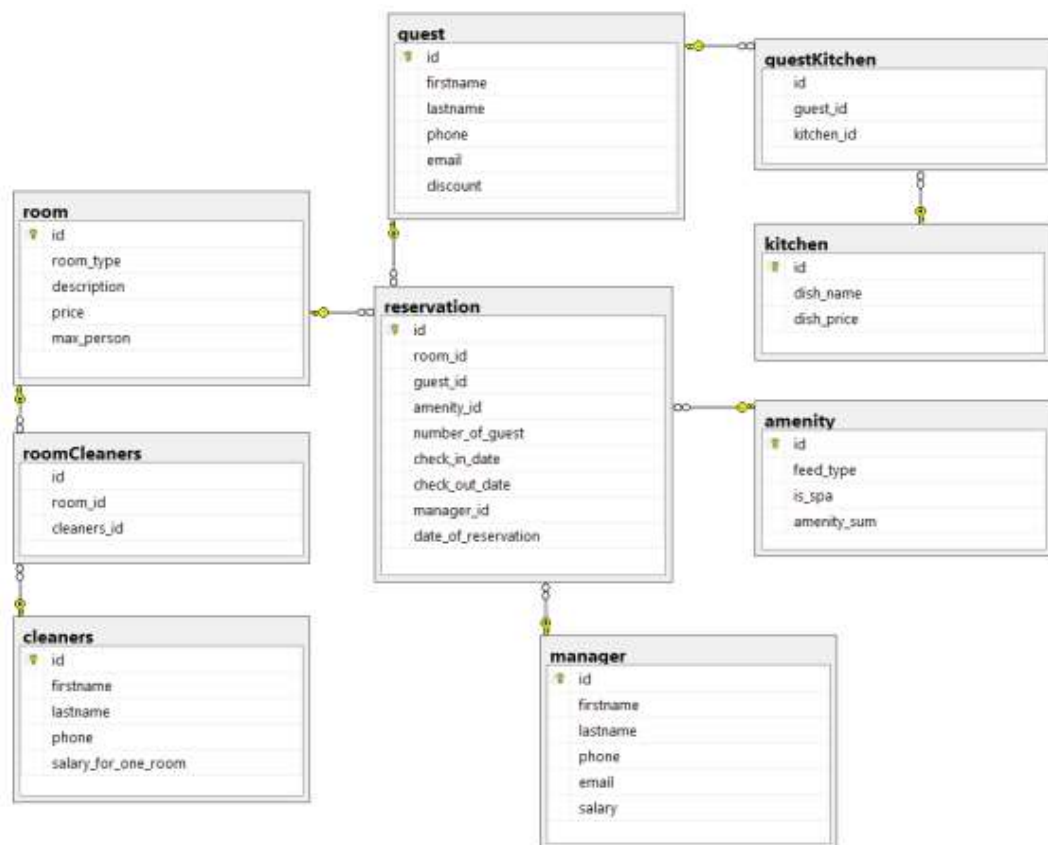


Рис. 2.2 Діаграма сутностей та зв'язків з використанням РК та FK

## 2.5 Реалізація SQL-скрипту

Під час розробки бази даних було реалізовано такі запити:

Лістинг 2.5.1

### 1. Який номер зайнятий був зарезервований в заданий період

```
SELECT room.id AS [Номер], room_type AS [Тип номера] from room JOIN
reservation on room.id = reservation.room_id
where date_of_reservation BETWEEN '2022-06-01' AND '2022-06-30'
ORDER BY room.id;
```

### 2. Який номер є зайнятий в певний період (рис. 2.3).

```
SELECT room.id, room_type from room JOIN reservation on room.id =
reservation.room_id
where check_in_date >= '2022-06-01' AND check_out_date <= '2022-06-
03'
ORDER BY id;
```

	Номер	Тип номеру
1	12	Стандарт одномісний
2	69	Люкс
3	70	Люкс

Рис. 2.3 Виконання запиту «Який номер є зайнятий в певний період»

### 3. Який номер певної категорії є вільний в певний період (рис. 2.4)

```
SELECT room.id AS [Номер кімнати], room_type AS [Тип кімнати] from
room LEFT JOIN reservation ON reservation.room_id = room.id where
room.id not in (SELECT room_id from room JOIN reservation ON
reservation.room_id = room.id where check_in_date >= '2022-06-07'
and check_out_date <= '2022-06-20') and room.room_type = 'Люкс';
```

	Номер кімнати	Тип кімна...
1	71	Люкс
2	73	Люкс
3	75	Люкс

Рис. 2.4 Виконання запиту «Який номер певної категорії є вільний в певний період»

### 4. Тривалість проживання кожного гостя

```
SELECT reservation.id AS [Номер бронювання], guest.firstname AS
[Ім'я], guest.lastname AS [Прізвище], datediff(DAY, check_in_date,
check_out_date) AS [Кількість ночей] from reservation
JOIN guest ON guest_id = guest.id;
```



## 5. Загальна сума для гостя за номер за проживання (рис. 2.5)

```
SELECT guest.firstname AS [Ім'я], guest.lastname AS [Прізвище],
room.id AS [Номер кімнати],
room.price AS [Ціна за одну добу], check_in_date AS [Дата заселення],
check_out_date AS [Дата виселення],
datediff(DAY, check_in_date, check_out_date) AS [Кількість ночей],
amenity.feed_type [Тип харчування], amenity.is_spa[Наявність SPA],
amenity.amenity_sum AS [Ціна за додаткові послуги за добу],
(datediff(DAY, check_in_date, check_out_date)*room.price+datediff(DAY, check_in_date, check_out_date)*amenity.amenity_sum) AS [Сума до оплати]
FROM guest
JOIN reservation ON guest_id = guest.id
JOIN amenity ON amenity_id = amenity.id
JOIN room ON room_id = room.id
```

	Ім'я	Прізвище	Номер кімнати	Ціна за одну добу	Дата заселення	Дата виселення	Кількість ночей	Тип харчування	Наявність SPA	Ціна за додаткові послуги за добу	Сума до оплати
1	Роман	Ковален	10	750	2022-05-29	2022-06-02	4	Без харчування	Ні	0	3000
2	Юлія	Савен	10	750	2022-06-10	2022-06-12	2	Дієтичне харчування	Так	450	2400
3	Анна	Консарецька	20	950	2022-05-31	2022-06-02	2	Без харчування	Так	100	2100
4	Ірина	Кисилівська	70	2490	2022-05-30	2022-06-15	16	Трьохразове харчування	Так	650	50240
5	Олег	Челен	72	2490	2022-05-14	2022-06-20	6	Одноразове харчування	Так	250	16440
6	Діана	Бойчук	1	750	2022-06-26	2022-07-02	6	Без харчування	Так	100	5100
7	Богдан	Сливка	45	1230	2022-05-27	2022-05-30	3	Без харчування	Ні	0	3690
8	Олег	Торієв	15	750	2022-05-28	2022-06-09	33	Дієтичне харчування	Ні	350	36300
9	Любов	Каспрівська	11	750	2022-06-27	2022-05-30	3	Без харчування	Ні	0	2250
10	Россошан	Лісова	12	750	2022-06-01	2022-06-03	2	Дієтичне харчування	Так	450	2400
11	Оксана	Вільчинська	35	950	2022-06-27	2022-06-30	3	Трьохразове харчування	Так	650	4800
12	Марина	Давидів	3	750	2022-05-31	2022-06-06	6	Одноразове харчування	Так	250	6000
13	Наталія	Степанів	74	2490	2022-06-09	2022-06-10	1	Без харчування	Так	100	2590
14	Сергій	Толор	66	1980	2022-05-30	2022-06-30	31	Трьохразове харчування	Ні	550	78430
15	Олег	Прейме	61	1980	2022-06-13	2022-06-16	3	Трьохразове харчування	Так	650	7980
16	Богдан	Васильєв	68	1980	2022-06-24	2022-06-28	4	Одноразове харчування	Ні	150	8520
17	Альберт	Кузнецов	19	950	2022-06-18	2022-06-26	8	Трьохразове харчування	Так	650	12800
18	Наталія	Подольська	33	950	2022-07-04	2022-07-07	3	Без харчування	Ні	0	2850
19	Ірина	Хрущ	39	1230	2022-06-22	2022-06-30	8	Без харчування	Ні	0	9840

Рис. 2.5 Виконання запиту «Загальна сума для гостя за номер за проживання»

## 6. Який гість яку страву замовив

```
SELECT guest.lastname AS [Прізвище], kitchen.dish_name,
SUM(kitchen.dish_price) AS [Сума за страви] FROM guest
JOIN guestKitchen ON guest.id = guestKitchen.guest_id
JOIN kitchen ON guestKitchen.kitchen_id = kitchen.id
GROUP BY guest.lastname, kitchen.dish_name
ORDER BY guest.lastname
```

## 7. Загальна сума за страви для кожного гостя

```
SELECT guest.lastname AS [Прізвище], SUM(kitchen.dish_price) AS
[Сума за страви] FROM guest
JOIN guestKitchen ON guest.id = guestKitchen.guest_id
JOIN kitchen ON guestKitchen.kitchen_id = kitchen.id
GROUP BY guest.lastname;
```

## 8. Загальна кількість номерів по типах

```
SELECT room_type, COUNT(description) AS [Кількість номерів], price
FROM room
GROUP BY room_type, price;
```

## 9. Яку кімнату прибирала покоївка за типом (рис. 2.6)

```
SELECT cleaners.lastname, room.id, room.room_type FROM cleaners
JOIN roomCleaners ON cleaners.id = roomCleaners.cleaners_id
JOIN room ON room.id = roomCleaners.room_id
JOIN reservation ON reservation.room_id = room.id;
```

	lastname	id	room_type
1	Халаня	1	Стандарт одномісний
2	Халаня	3	Стандарт одномісний
3	Халаня	10	Стандарт одномісний
4	Халаня	10	Стандарт одномісний
5	Халаня	11	Стандарт одномісний
6	Димора	12	Стандарт одномісний
7	Димора	15	Стандарт одномісний
8	Димора	19	Стандарт двомісний
9	Чиж	25	Стандарт двомісний
10	Чиж	33	Стандарт двомісний
11	Петрохицька	35	Стандарт двомісний
12	Петрохицька	39	Стандарт сімейний
13	Покалюк	45	Стандарт сімейний
14	Канаєва	61	Напівлюкс
15	Рогатинка	66	Напівлюкс
16	Рогатинка	68	Напівлюкс
17	Рогатинка	69	Люкс
18	Рогатинка	69	Люкс
19	Рогатинка	70	Люкс
20	Рогатинка	70	Люкс
21	Рогатинка	72	Люкс
22	Рогатинка	74	Люкс

Рис. 2.6 Виконання запиту «Яку кімнату прибирала покоївка за типом

## 10. Заробіток покоївок за прибрані номери

```
SELECT cleaners.lastname, COUNT(room.id) AS [К-сть прибраних номерів],
salary_for_one_room [Ціна за одне прибирання],
(COUNT(room.id)*salary_for_one_room) AS [Заробіток] FROM cleaners
JOIN roomCleaners ON cleaners.id = roomCleaners.cleaners_id
JOIN room ON room.id = roomCleaners.room_id
JOIN reservation ON reservation.room_id = room.id
WHERE check_in_date >= '2022-06-01' AND check_out_date <= '2022-06-30'
GROUP BY cleaners.lastname, salary_for_one_room;
```

## 11. Зарплата менеджера за місяць

```
SELECT manager.lastname AS [Прізвище], COUNT(room_id) AS [К-сть номерів],
(SUM(room.price*0.05))+manager.salary AS [Зарплата] FROM reservation
JOIN manager ON manager.id = manager_id
```

```
JOIN room ON room_id = room.id
GROUP BY manager.lastname, manager.salary
```

## 12. Найпопулярніший тип номеру за певний період (рис. 2.7)

```
SELECT room.room_type AS [Тип кімнати], COUNT(reservation.id) AS
[Номери по популярності] FROM room
JOIN reservation on reservation.room_id = room.id
GROUP BY room.room_type
ORDER BY COUNT(reservation.id) DESC;
```

	Тип кімнати	Номери по популярн...
1	Стандарт одномісний	7
2	Люкс	6
3	Стандарт двомісний	4
4	Напівлюкс	3
5	Стандарт сімейний	2

Рис. 2.7 Виконання запиту «Найпопулярніший тип номеру за певний період»

## 13. Найпопулярніша страва

```
SELECT kitchen.dish_name AS [Назва страви],
COUNT(guestKitchen.kitchen_id) AS [Страви по популярності] FROM
kitchen
JOIN guestKitchen on guestKitchen.kitchen_id = kitchen.id
GROUP BY kitchen.dish_name
ORDER BY COUNT(guestKitchen.kitchen_id) DESC;
```

## 14. Найпопулярніше amenity

```
SELECT amenity.id AS [Тип послуги], amenity.feed_type AS [Додаткові
послуги], amenity.is_spa AS [Додаткові послуги],
COUNT(reservation.amenity_id) AS [Послуги по популярності] FROM
amenity
JOIN reservation on reservation.amenity_id = amenity.id
GROUP BY amenity.id, amenity.feed_type, amenity.is_spa
ORDER BY COUNT(reservation.amenity_id) DESC;
```

Приклад використання запиту в коді є в лістингу 2.5.2.

```
const getAvailableRooms = async ({ room_type, start_date, end_date
}) => {
  try {
    const result = await sql.query(
      `SELECT TOP 1 room.id from room LEFT JOIN reservation
ON reservation.room_id = room.id where room.id not in(
```

```

        SELECT room_id from room JOIN reservation ON
reservation.room_id = room.id where check_in_date >= '${start_date}'
and check_out_date <= '${end_date}')) and room.room_type =
'${room_type}';`
    );
    if (!result.recordset || !result.recordset.length) return
null;
    return result.recordset;
} catch (error) {
    console.error("getAvailableRooms error:", error);
    return error;
}
};

```

## РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ

### 3.1 Структура веб-сайту та макет сторінок

В якості середовища розробки веб-додатку було використано Visual Studio Code. Це середовище було обрано для написання коду HTML, CSS, JavaScript. За допомогою порта сайт розміщується на комп'ютері та відкривається через браузер.

Реалізація веб-сайту почалась з шаблону сайту. Для шаблону використовувалась мова розмітки HTML5 та стилі CSS. Також використовувався фреймворк Bootstrap.

Структура веб-сайту – це те, як організований сайт, і як зв'язані між собою сторінки сайту. Логічна структура сайту допомагає відвідувачам легко переміщуватись по ньому і знаходити потрібну інформацію. Надійна структура веб-сайту є дуже важливою при розробці сайту і чим більший є сайт, тим важливішою є структура.

Сайт прийому та обслуговування клієнтів готелю є схожим на більшість сайтів готелю. Сайт складається з двох сторінок. Перша сторінка – це домашня сторінка, де є:

1. Шапка сайту, яка містить назву компанії зліва, та меню справа (рис. 3.1).



Рис. 3.1 Шапка сайту

2. Банер сайту, який містить слова привітання (рис. 3.2).

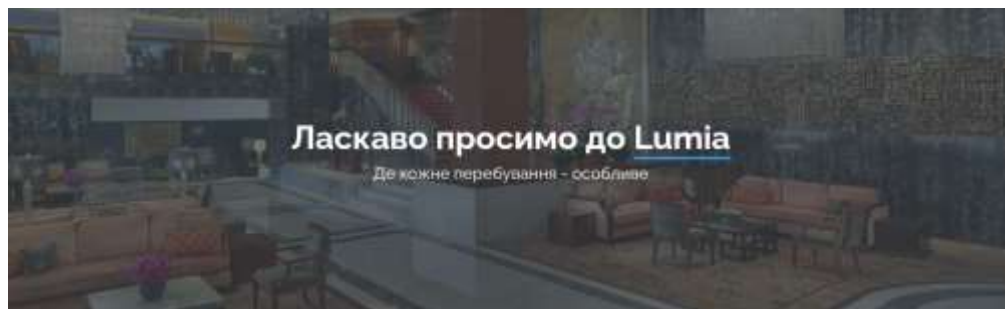


Рис. 3.2 Банер сайту

3. Секція з номерами з можливістю забронювати номер того чи іншого типу, натиснувши кнопку забронювати (рис. 3.3).

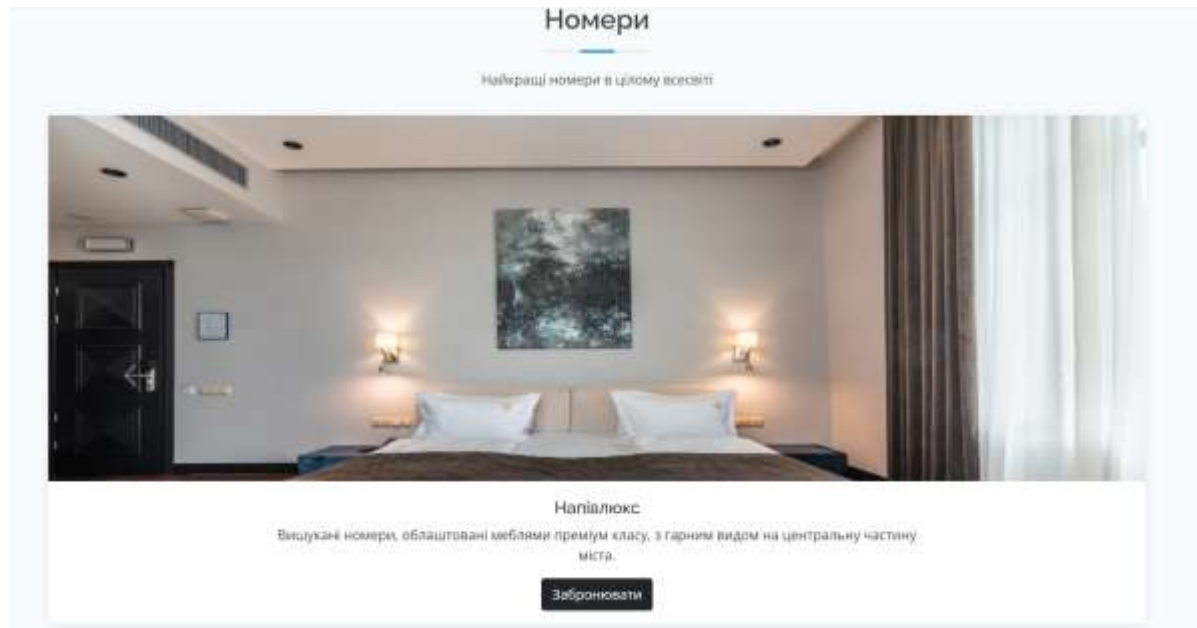


Рис. 3.3 Секція з номерами

4. Секція з інформацією про нас (рис. 3.4).



Рис. 3.4 Секція з інформацією про нас

5. Секція з контактами (рис.3.5).

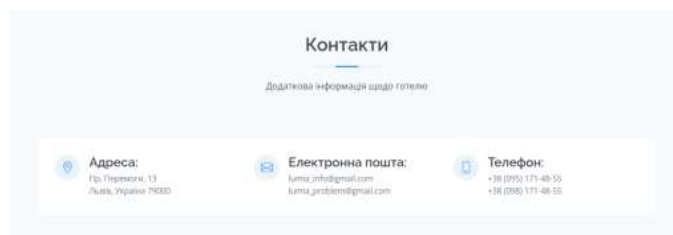


Рис 3.5 Секція з контактами

## 6. Футер (рис. 3.5).

**Lumia**

Пр. Перемоги, 13

Львів 79000

Україна

Телефон: +38 (095) 171-48-55

Електронна пошта: lumia\_info@gmail.com

Рис. 3.5 Футер

Друга сторінка сайту, яка відкривається після натискання кнопки «забронювати» – де є:

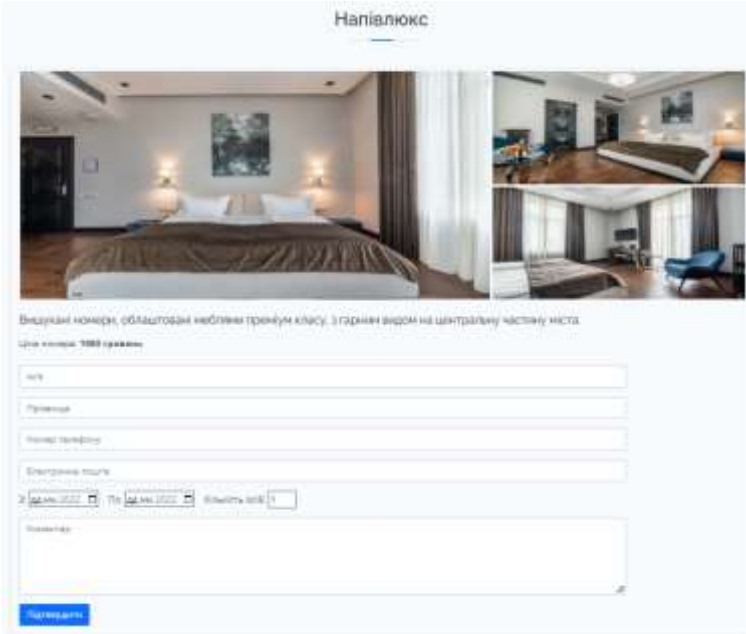
1. Шапка сайту, яка містить назву компанії зліва, та меню справа (рис. 3.6).

LUMIA

Контакти

Рис. 3.6 Шапка сайту другої сторінки

2. Секція з інформацією про обраний тип номеру, полями для особистої інформації (рис. 3.7).



The screenshot shows the 'Напіалюкс' website. At the top, there's a header with the company name 'LUMIA' on the left and 'Контакти' on the right. Below the header, there's a large image of a hotel room with a bed. To the right of the main image are two smaller images showing different views of the room. Below the images, there's a text description: 'Видатні номери, обладнані меблями преміум класу. З гарним видом на центральну частину міста. Ціна номеру 1000 гривень'. Below the text, there's a form with several input fields: 'Ім'я', 'Прізвище', 'Номер телефону', 'Електронна пошта', and a date range selector with 'З' and 'По' labels. There's also a 'Кількість осіб' field. At the bottom of the form is a blue button labeled 'Забронювати'.

Рис. 3.7 Секція з інформацією про обраний тип номеру, полями для особистої інформації

### 3. Секція з контактами (рис. 3.8).

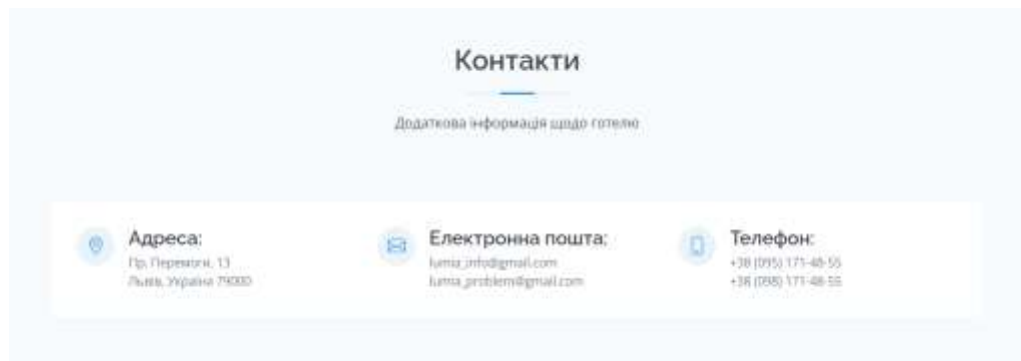


Рис 3.8 Секція з контактами

### 4. Футер (рис. 3.9).



Рис 3.9 Футер

## 3.2 Програмування серверної та клієнтської частини

Програмування серверної та клієнтської частини створене за допомогою платформи Node.js. Дана платформа допомагає запускати JavaScript на комп'ютері.

Сервер під'єднаний до бази даних, програмна реалізація під'єднання знаходиться в лістингу А.17. Для під'єднання серверу до бази даних нам потрібні параметри з'єднання (connection params), а саме:

1. Server
2. User,
3. Database,



4. Password,
5. Port. Ці параметри зберігаються в окремому файлі, код якого знаходиться в лістингу A.19

Сервер має список записів які містяться в окремому файлі (лістинг A.18). Коли клієнт викликає певний запит, тоді сервер, який під'єднаний до бази даних, витягує потрібну нам інформацію і відправляє її на клієнт.

Клієнтська частина виконана так, що всі кімнати, які відображаються на сайті генеруються інтерактивно, відповідно до того, що міститься в базі. Однак фотографії номерів готелю знаходять в файлі `rooms_images_paths` (лістинг A.21). Інші частини програмування клієнтської та серверної частини наведені в лістингах A.10-A.21.

## ВИСНОВКИ

Під час виконання курсової роботи було досліджено варіанти використання HTML, CSS, JavaScript та SQL. Для закріплення набутих знань у процесі виконання було розроблено інформаційну систему прийому та обслуговування клієнтів готелю, яка складається з бази даних, яка міститиме всю необхідну інформацію, та веб-сайт, який дозволяє клієнтам отримати бажану інформацію та забронювати той чи інший номер в готелі.

У процесі виконання було розроблено Use Case Diagram, яка показує варіанти використання веб-сайту двома акторами: менеджером та гостем готелю. У діаграмі показано відношення між акторами та прецедентами (варіантами використання) в системі. Варіанти використання корисні тим, що з їхньою допомогою можна легко побачити функціональні можливості кожного актора в інформаційній системі.

Також під час роботи було спроектовано та розроблено базу даних, її таблиці, поля та записи. Також необхідно зазначити, що ER – діаграма є обов'язковою частиною розробки бази даних, адже вона чітко показує 9 спроектованих таблиць, первинні та вторинні ключі, типи зв'язків між таблицями.

Програмування серверної та клієнтської частини створене за допомогою платформи Node.js, дана платформа допомагає запускати JavaScript на сервері. Програмна реалізація розпочалася з розробки макету сайту за допомогою HTML та CSS. Після того було реалізовано клієнтська та серверна частина, за допомогою вищезгаданої платформи та підключено базу даних.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Learn SQL [Електронний ресурс]. – Режим доступу: <https://www.codecademy.com/learn/learn-sql>
2. Front-End Engineer [Електронний ресурс]. – Режим доступу: <https://www.codecademy.com/learn/paths/front-end-engineer-career-path>
3. Структура Web-додатку [Електронний ресурс]. – Режим доступу: [https://stud.com.ua/97612/informatika/struktura\\_dodatku](https://stud.com.ua/97612/informatika/struktura_dodatku)
4. Типи даних (Transact-SQL) [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>
5. Нормалізація реляційних баз даних [Електронний ресурс]. – Режим доступу: [https://stud.com.ua/93792/informatika/normalizatsiya\\_relyatsiynih\\_danih#google\\_vignette](https://stud.com.ua/93792/informatika/normalizatsiya_relyatsiynih_danih#google_vignette)
6. Нормалізація відношень баз даних [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5454386/page:8/>
7. Bootstrap Templates and Themes [Електронний ресурс]. – Режим доступу: <https://startbootstrap.com/themes>
8. Модель даних: особливості, класифікація та опис [Електронний ресурс]. – Режим доступу: <https://presa.com.ua/navchannia/modeli-danikh-osoblivosti-klasifikatsiya-ta-opis.html#modeli-system-danykh-klasyfikatsiia>
9. Інформаційні системи [Електронний ресурс]. – Режим доступу: [https://pidru4niki.com/1222090547713/informatika/informatsiyni\\_sistemi](https://pidru4niki.com/1222090547713/informatika/informatsiyni_sistemi)
10. Stackoverflow [Електронний ресурс]. – Режим доступу: <https://stackoverflow.com>
11. Node.js [Електронний ресурс]. – Режим доступу: <https://nodejs.org/uk/>
12. JavaScript Учебные материалы [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/ru/docs/Web/JavaScript>

13. Бази даних [Електронний ресурс]. – Режим доступу:  
<https://core.ac.uk/download/pdf/33757091.pdf>
14. Visual Paradigm Online [Електронний ресурс]. – Режим доступу:  
<https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=UseCaseDiagram&width=11&height=8.5&unit=inch>
15. Обслуговування гостей службою прийому та розміщення [Електронний ресурс]. – Режим доступу:  
[https://pidru4niki.com/10560412/turizm/obslugovuvannya\\_gostey\\_sluzhboyu\\_priyomu\\_rozmischennya](https://pidru4niki.com/10560412/turizm/obslugovuvannya_gostey_sluzhboyu_priyomu_rozmischennya)
16. Моделі баз даних [Електронний ресурс]. – Режим доступу:  
[https://uk.wikipedia.org/wiki/Моделі\\_баз\\_даних](https://uk.wikipedia.org/wiki/Моделі_баз_даних)
17. IMESTAMPDIFF() function [Електронний ресурс]. – Режим доступу:  
<https://www.w3resource.com/mysql/date-and-time-functions/mysql-timestampdiff-function.php>
18. Visual Studio Code [Електронний ресурс]. – Режим доступу:  
<https://code.visualstudio.com>
19. SQL Server 2019 [Електронний ресурс]. – Режим доступу:  
<https://www.microsoft.com/ru-ru/sql-server/sql-server-2019?SilentAuth=1>
20. SQL Tutorial [Електронний ресурс]. – Режим доступу:  
<https://www.w3schools.com/sql/>

## ДОДАТКИ

### Додаток А

#### Лістинг А.1 Наповнення таблиці «manager».

```
INSERT INTO manager (id, firstname, lastname, phone, email) VALUES
(1, 'Андрій', 'Васько', '+380630279241', 'andriy.vasko@gmail.com');
INSERT INTO manager (id, firstname, lastname, phone, email) VALUES
(2, 'Богдан', 'Балух', '+380937611773', 'bogdan.balukh@gmail.com');
INSERT INTO manager (id, firstname, lastname, phone, email) VALUES
(3, 'Аліна', 'Бабій', '+380508565365', 'alina.babii@gmail.com');
INSERT INTO manager (id, firstname, lastname, phone, email) VALUES
(4, 'Владислав', 'Баюрчак',
'+380500335511', 'vlad.baiurchak@gmail.com');
INSERT INTO manager (id, firstname, lastname, phone, email) VALUES
(5, 'Анна', 'Шматюк', '+380970555311', 'ann.shmatiuk@gmail.com');
```

#### Лістинг А.2 Наповнення таблиці «room».

```
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (1, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (2, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (3, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (4, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (5, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (6, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (7, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
```

```

VALUES (8, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (9, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (10, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (11, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (12, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (13, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (14, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (15, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (16, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (17, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (18, 'Стандарт одномісний', 'Найоптимальніший варіант для
подорожуючих, які шукають зручне, затишне місце проживання в
самому центрі міста за невисокою ціною.', 750, 1);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (19, 'Стандарт двомісний', 'На Вас чекають надзвичайно
вишукані кімнати, облаштовані сучасними італійськими меблями',
950, 2);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (20, 'Стандарт двомісний', 'На Вас чекають надзвичайно
вишукані кімнати, облаштовані сучасними італійськими меблями',
950, 2);
INSERT INTO room (id, room_type, description, price, max_person)

```

[illegible]

```

VALUES (34, 'Стандарт двомісний', 'На Вас чекають надзвичайно
вишукані кімнати, облаштовані сучасними італійськими меблями',
950, 2);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (35, 'Стандарт двомісний', 'На Вас чекають надзвичайно
вишукані кімнати, облаштовані сучасними італійськими меблями',
950, 2);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (36, 'Стандарт двомісний', 'На Вас чекають надзвичайно
вишукані кімнати, облаштовані сучасними італійськими меблями',
950, 2);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (37, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (38, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (39, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (40, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (41, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (42, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (43, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (44, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (45, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (46, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)

```



```

VALUES (47, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (48, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (49, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (50, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (51, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (52, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (53, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (54, 'Стандарт сімейний', 'Надзвичайно елегантні та
просторі номери стануть чудовим вибором для усіх охочих відпочити
у теплому колі своїх рідних та близьких', 1230, 3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (55, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (56, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (57, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (58, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (59, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)

```

```

VALUES (60, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (61, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (62, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (63, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (64, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (65, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (66, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (67, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (68, 'Напівлюкс', 'Вишукані номери, облаштовані меблями
преміум класу, з гарним видом на центральну частину міста.', 1980,
3);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (69, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (70, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (71, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
INSERT INTO room (id, room_type, description, price, max_person)

```

```
VALUES (72, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (73, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (74, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
INSERT INTO room (id, room_type, description, price, max_person)
VALUES (75, 'Люкс', 'Двокімнатні розкішні номери для справжніх
цінителів гармонійного поєднання сучасних вигод та елегантних
елементів лекору з дотриманням усіх європейських стандартів',
2490, 4);
```

### Лістинг А.3 Наповнення таблиці «cleaners».

```
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (1, 'Галина', 'Халанія',
'+380979874561', 75);
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (2, 'Антоніна', 'Димора',
'+380506698741', 80);
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (3, 'Марина', 'Чиж', '+380631414555',
70);
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (4, 'Дарина', 'Петрожицька',
'+380630228810', 85);
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (5, 'Марія', 'Покалюк',
'+380731315489', 72);
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (6, 'Аліна', 'Канаєва',
'+380506894159', 65);
INSERT INTO cleaners (id, firstname, lastname, phone,
salary_for_one_room) VALUES (7, 'Віталіна', 'Рогатинка',
'+380674849159', 78);
```

### Лістинг А.4 Наповнення таблиці «amenity».

```
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(1, 'Без харчування', 'Ні', 0);
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(2, 'Без харчування', 'Так', 100);
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(3, 'Одноразове харчування', 'Так', 250);
```

```

INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(4, 'Одноразове харчування', 'Hi', 150);
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(5, 'Дворазове харчування', 'Так', 450);
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(6, 'Дворазове харчування', 'Hi', 350);
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(7, 'Трьохразове харчування', 'Так', 650);
INSERT INTO amenity (id, feed_type, is_spa, amenity_sum) VALUES
(8, 'Трьохразове харчування', 'Hi', 550);

```

#### Лістинг А.5 Наповнення таблиці «roomCleaners».

```

INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (1, 1,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (2, 2,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (3, 3,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (4, 4,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (5, 5,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (6, 6,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (7 ,7,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (8, 8,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (9, 9,
1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (10,
10, 1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (11,
11, 1);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (12,
12, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (13,
13, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (14,
14, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (15,
15, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (16,
16, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (17,
17, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (18,
18, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (19,
19, 2);

```

```
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (20,
20, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (21,
21, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (22,
22, 2);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (23,
23, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (24,
24, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (25,
25, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (26,
26, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (27,
27, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (28,
28, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (29,
29, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (30,
30, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (31,
31, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (32,
32, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (33,
33, 3);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (34,
34, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (35,
35, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (36,
36, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (37,
37, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (38,
38, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (39,
39, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (40,
40, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (41,
41, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (42,
42, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (43,
43, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (44,
44, 4);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (45,
45, 5);
```

```

INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (46,
46, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (47,
47, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (48,
48, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (49,
49, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (50,
50, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (51,
51, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (52,
52, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (53,
53, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (54,
54, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (55,
55, 5);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (56,
56, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (57,
57, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (58,
58, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (59,
59, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (60,
60, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (61,
61, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (62,
62, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (63,
63, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (64,
64, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (65,
65, 6);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (66,
66, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (67,
67, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (68,
68, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (69,
69, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (70,
70, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (71,
71, 7);

```

```

INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (72,
72, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (73,
73, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (74,
74, 7);
INSERT INTO roomCleaners (id, room_id, cleaners_id) VALUES (75,
75, 7);

```

#### Лістинг А.6 Наповнення таблиці «kitchen».

```

INSERT INTO kitchen(id, dish_name,dish_price) VALUES (1, 'Салат з
червоною квасолею, куркою та авокадо', 135)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (2, 'Салат з
лососем та авокадо', 165)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (3, 'Біф -
яловичина з тушкованими овочами', 130)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (4, 'Салат з
печінкою та хамоном', 135)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (5, 'Салат
креветками та куркою', 145)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (6, 'Салат з
вугрем', 145)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (7, 'Салат
ростбіф', 130)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (8, 'Борщ
"Карпати" з салом', 65)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (9, 'Бульйон
з перепілки з локшиною та яйцем', 60)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (10, 'Юшка
грибна з локшию', 60)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (11,
'Солянка', 70)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (12, 'Ніжний
лосось під ікорним соусом', 190)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (13, 'Банош з
бринзою', 85)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (14, 'Філе
міньйон', 250)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (15,
'Вареники з картоплею та сметаною', 100)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (16, 'Деруни
з білими грибами', 115)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (17, 'Філе
сома в кисло-солодкому соусі', 165)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (18,
'Телятина з білими грибами', 230)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (19, 'Реберця
у пивному соусі', 190)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (20, 'Рулєт
курячий в беконі з фісташками', 260)

```

```

INSERT INTO kitchen(id, dish_name,dish_price) VALUES (21, 'Короп
смажений', 48)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (22, 'Лаваш
сулугуні з куркою та помідорами', 95)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (23, 'Шашлик
зі свинного ошийка', 585)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (24, 'Короп
гриль',120 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (25, 'Крокети
з сирним соусом', 60)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (26, 'Овочі
гриль',75 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (27,
'Картопля печена з часником', 40)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (28, 'Пюре
картопляне',50 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (29, 'Френч
фрайз', 50)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (30,
'Tipamicy',65 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (31,
'Штрудель яблучний', 65)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (32, 'Фондан
де чоколате', 70)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (33,
'Семіфредо з праліне', 60)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (34, 'Джелато
на вибір', 69)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (35, 'Узвар',
22)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (36, 'Пепсі',
29)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (37,
'Боржомі', 48)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (38,
'Лимонад', 24)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (39, 'Фреш
апельсиновий',60 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (40, 'Фреш
морквяний', 47)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (41, 'Фреш
грейпфрутовий', 50)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (42, 'Фреш
яблучний',46 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (43,
'Пістретто',32 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (44,
'Американо',32 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (45, 'Лате',
42)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (46,
'Еспресо',32 )

```



```

INSERT INTO kitchen(id, dish_name,dish_price) VALUES (47,
'Капучино', 42)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (48, 'Какао',
35)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (49,
'Молоко', 7)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (50, 'Чорний
чай',32 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (51,
'Фруктовий чай', 32)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (52, 'Зелений
чай', 32)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (53,
'Трав'яний чай',32 )
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (54, 'Апероль
шпріц', 95)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (55,
'Мохіто',80)
INSERT INTO kitchen(id, dish_name,dish_price) VALUES (56, 'Мілк
шейк з сиропом', 47);

```

#### Лістинг А.7 Наповнення таблиці «guestKitchen».

```

INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (1, 1,
1);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (2, 1,
5);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (3, 2,
10);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (4, 2,
15);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (5, 3,
20);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (6, 3,
2);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (7, 3,
6);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (8, 4,
11);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (9, 5,
16);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (10, 5,
21);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (11,
31, 3);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (12,
31, 7);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (13,
31, 12);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (14,
25, 17);

```

```

INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (15,
13, 22);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (16,
13, 4);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (17,
19, 8);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (18,
19, 13);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (19,
19, 18);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (20,
40, 23);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (21,
40, 50);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (22,
14, 5);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (23,
14, 10);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (24,
14, 15);
INSERT INTO guestKitchen (id, guest_id, kitchen_id) VALUES (25,
14, 20);

```

#### Лістинг А.8 Наповнення таблиці «guest».

```

INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (1,
'Tіна', 'Кароль', '+380630256987', 'tina.karol@gmail.com', 5);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (2, 'Юлія', 'Саніна', '+380975567320', 'sanina@gmail.com', 3);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (3, 'Богдан', 'Сливка', '+380936312157', 'slyvka@gmail.com',
0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (4,
'Альберт', 'Кузьменко', '+380670555310', 'albert24@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (5, 'Анна', 'Комісаренко', '', 'komisarenko0921@gmail.com',
0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (6,
'Василь', 'Голобородько', '+380687005088', 'vasyl.goloborodko@gmail.c
om', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (7, 'Ірина', 'Пенц', '+380950338565', 'iryna777@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (8, 'Микола', 'Степанюк', '+380687512541', '@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (9,
'Роксолана', 'Кремінець', '+380681245789', 'roksa@gmail.com', 0);

```

```

INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (10,
'Роксолана','Лісова','+380675544879','mavka@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (11,
'Назарій','Мураль','+380685647895','mural85@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (12,
'Олена','Чигін','+380675413258','olenka12@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (13,
'Оксана','Вільчинська','+3806870055412','vilchynska73@gmail.com',4
);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (14, 'Олег','Чигін','+380689845125','chygin@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (15, 'Олег','Оприск','+380631245862','oprysk@gmail.com',
0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (16,
'Богдан','Васильків','+3806850020123','bohdan.vasylkiv1@gmail.com'
,0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (17,
'Марина','Дзюнька','+380502014578','dziunka@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (18,
'Аліна','Александрович','+380632548789','aleksandrovyh13@gmail.co
m',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (19,
'Діана','Бойчук','+3806302145782','boichuk@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (20,
'Сергій','Топорь','+380632548785','toporyk@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (21,
'Валентин','Наливайко','+380932356478','nalyvaiko@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (22,
'Тетяна','Клюка','+380970512654','tetianka452@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (23,
'Ельвіра','Кукурузка','+380500338566','elyaaa12@gmail.com',5);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (24,
'Віталій','Малецький','+380502145632','maletskyi0403@gmail.com',0)
;
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (25,
'Роман','Кохман','+3806302136547','kokman@gmail.com',0);

```

```

INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (26,
'Галина','Ткач','+380689521245','tkach888@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (27,
'Оксана','Залужна','+380501465555','oksanka111@gmail.com',3);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (28,
'Петро','Лесько','+380500556512','lesko14@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (29,
'Василина','Кукурік','+380687415263','kukurik999000@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (30,
'Лілія','Бова','+380501122333','bovalilia@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (31,
'Олег','Торпрук','+380505050111','oleh1065@gmail.com',1);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (32,
'Анна','Старух','+380687005011','starukh@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (33,
'Наталя','Подольчак','+380501112536','podolchak1010@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (34,
'Тарас','Матвіїв','+380501515478','tarasyk888@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (35, 'Святозар','Мацех','+380632589632','
matsekh@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (36,
'Василь','Марашко','+380985544666','marashko11@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (37,
'Діана','Ліщук','+380632154987','lishchukdd@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (38,
'Павло','Манік','+380978899545','manik555@gmail.com',7);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (39,
'Ігор','Павлик','+380500338465','ihor1111@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (40,
'Інна','Кісиличко','+380501235647','kisylychko@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (41,
'Любов','Каспрівська','+380631236987','lubov1@gmail.com',0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (42,
'Любов','Соболь','+380684568974','sobol000@gmail.com',10);

```

```

INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (43,
'Mарія', 'Пенцятко', '+380505011556', 'masha431@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (44, 'Ірина', 'Хрущ', '+380633322487', 'ira452@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (45, 'Мирослава', 'Бапа', '+380', 'bapa88@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (46, 'Олег', 'Прийма', '+380985553100', '@gmail.com', 3);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (47,
'Андріана', 'Марусич', '+380974859126', 'marusych82@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (48,
'Олена', 'Крюк', '+380981210111', 'olena111@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (49,
'Наталя', 'Степишин', '+380685554477', 'stepnat11111@gmail.com', 0);
INSERT INTO guest(id, firstname, lastname, phone, email, discount)
VALUES (50,
'Ірина', 'Бегун', '+380630215641', 'begun89@gmail.com', 0);

```

#### Лістинг А.9 Наповнення таблиці «reservation».

```

INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (1, 10, 25, 1, 1, '2022-05-29', '2022-06-02', 2, '2022-05-
19');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (2, 10, 2, 5, 1, '2022-06-10', '2022-06-12', 5, '2022-05-
21');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (3, 25, 5, 2, 2, '2022-05-31', '2022-06-02', 1, '2022-05-
17');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (4, 70, 40, 7, 3, '2022-05-30', '2022-06-15', 2, '2022-05-
19');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (5, 72, 14, 3, 4, '2022-06-14', '2022-06-20', 3, '2022-05-
25');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)

```

```

VALUES (6, 1, 19, 2, 1, '2022-06-26', '2022-07-02', 2, '2022-05-
24');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (7, 45, 3, 1, 1, '2022-05-27', '2022-05-30', 5, '2022-05-
20');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (8, 15, 31, 6, 1, '2022-05-28', '2022-06-30', 3, '2022-05-
26');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (9, 11, 41, 1, 1, '2022-05-27', '2022-05-30', 5, '2022-05-
20');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (10, 12, 10, 5, 1, '2022-06-01', '2022-06-03', 5, '2022-05-
18');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (11, 35, 13, 7, 1, '2022-06-27', '2022-06-30', 4, '2022-05-
23');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (12, 3, 17, 3, 1, '2022-05-31', '2022-06-06', 1, '2022-05-
28');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (13, 75, 50, 4, 4, '2022-06-15', '2022-06-19', 5, '2022-06-
01');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (14, 74, 49, 2, 3, '2022-06-09', '2022-06-10', 2, '2022-06-
01');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (15, 66, 20, 8, 1, '2022-05-30', '2022-06-30', 1, '2022-06-
02');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (16, 61, 46, 7, 1, '2022-06-13', '2022-06-05', 5, '2022-06-
20');

```

```

INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (17, 68, 16, 4, 2, '2022-06-24', '2022-06-28', 4, '2022-06-
30');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (18, 19, 4, 7, 2, '2022-06-18', '2022-06-26', 4, '2022-06-
01');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (19, 33, 33, 1, 2, '2022-07-04', '2022-07-07', 5, '2022-06-
11');
INSERT INTO reservation (id, room_id, guest_id, amenity_id,
number_of_guest, check_in_date, check_out_date, manager_id,
date_of_reservation)
VALUES (20, 39, 44, 1, 1, '2022-06-22', '2022-06-30', 3, '2022-06-
14');

```

### Лістинг А.10 Файл Api.js

```

import mockDB from "../../rooms_images_paths.json" assert { type:
"json" };

const server = "http://localhost:3000";

export const fetchRooms = async () => {
  const result = await (await
fetch(`${server}/api/rooms`)).json();
  if (!result.data) return null;

  const rooms = result.data.map((room) => {
    const roomItem = mockDB.find((el) => el.type ===
room.room_type);

    return {
      ...room,
      image_paths: roomItem?.image_paths || [],
    };
  });

  return rooms;
};

export const bookRoom = async (data) => {
  const result = await (
    await fetch(`${server}/api/rooms/book`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json; charset=utf-8",

```

```

        },
        body: JSON.stringify(data),
    })
    ).json();

    return result;
};

```

### Лістинг A.11 Файл booking.js

```

import { select, createEl, getQueryParams, showAlert } from
"./utils.js";
import { fetchRooms, bookRoom } from "./api.js";

(function () {
    "use strict";
    /**
     * Easy on scroll event listener
     */

    const getRoomData = (rooms, type) =>
        rooms.find((el) => el.room_type === type) || null;

    window.addEventListener("load", async () => {
        const { type } = getQueryParams();

        const rooms = await fetchRooms();

        const room = getRoomData(rooms, type);

        if (!room) return;

        select(".room_type_title").textContent = room.room_type;
        if (room.count === 0) {
            select(".room_type_description").textContent = "Номер
не доступний";
        }
        select(".full_description").textContent =
room.description;
        select(".price").innerHTML = `Ціна номера:
<b>${room.price} гривень</b>`;

        const images = room.image_paths.map((path, i) => {
            const el = createEl("img");
            el.classList.add(
                "room-image",
                "d-block",
                `${i === 0 ? "main_image" : "secondary_image"}`
            );
            el.src = path;
            return el;
        });
    });

```



```

select(".images_container").append(images[0]);
select(".secondary_images").append(...images.slice(1));

// Form
// Form
// Form
select(".book_form").onsubmit = async (e) => {
    e.preventDefault();
    if (room.count === 0) return null;
    const firstname = select(".user_firstname").value;
    const lastname = select(".user_lastname").value;
    const email = select(".user_email").value;
    const start_date = select("#start_date").value;
    const end_date = select("#end_date").value;
    // const comment = select(".user_comment").value;
    const phone = select(".user_phone").value;
    const guests = select("#guests").value;

    const user = { firstname, lastname, email, phone };
    const reservation = {
        start_date,
        end_date,
        guests,
        room_type: type,
    };

    const result = await bookRoom({ user, reservation });

    if (result.error) showAlert(result.error, "danger");
    else if (result.data) showAlert("Кімнату
заброньовано!", "success");
    };
    // end form
});

window.addEventListener("load", () => {});
})();

```

### Лістинг А.12 Файл main.js

```

import { on, select } from "./utils.js";
import { fetchRooms } from "./api.js";

(function () {
    "use strict";
    /**
     * Easy on scroll event listener
     */
    const onscroll = (el, listener) => {
        el.addEventListener("scroll", listener);
    };

    /**

```

```

    * Navbar links active state on scroll
    */
let navbarlinks = select("#navbar .scrollto", true);
const navbarlinksActive = () => {
    let position = window.scrollY + 200;
    navbarlinks.forEach((navbarlink) => {
        if (!navbarlink.hash) return;
        let section = select(navbarlink.hash);
        if (!section) return;
        if (
            position >= section.offsetTop &&
            position <= section.offsetTop +
section.offsetHeight
        ) {
            navbarlink.classList.add("active");
        } else {
            navbarlink.classList.remove("active");
        }
    });
};
window.addEventListener("load", navbarlinksActive);
onscroll(document, navbarlinksActive);

/**
 * Scrolls to an element with header offset
 */
const scrollTo = (el) => {
    let header = select("#header");
    let offset = header.offsetHeight;

    let elementPos = select(el).offsetTop;
    window.scrollTo({
        top: elementPos - offset,
        behavior: "smooth",
    });
};

/**
 * Toggle .header-scrolled class to #header when page is
scrolled
 */
let selectHeader = select("#header");
if (selectHeader) {
    const headerScrolled = () => {
        if (window.scrollY > 100) {
            selectHeader.classList.add("header-scrolled");
        } else {
            selectHeader.classList.remove("header-scrolled");
        }
    };
    window.addEventListener("load", headerScrolled);
    onscroll(document, headerScrolled);
}

```

```

}

/**
 * Back to top button
 */
let backtotop = select(".back-to-top");
if (backtotop) {
  const toggleBacktotop = () => {
    if (window.scrollY > 100) {
      backtotop.classList.add("active");
    } else {
      backtotop.classList.remove("active");
    }
  };
  window.addEventListener("load", toggleBacktotop);
  onscroll(document, toggleBacktotop);
}

/**
 * Scroll with offset on links with a class name .scrollto
 */
on(
  "click",
  ".scrollto",
  function (e) {
    if (select(this.hash)) {
      e.preventDefault();

      let navbar = select("#navbar");
      if (navbar.classList.contains("navbar-mobile")) {
        navbar.classList.remove("navbar-mobile");
        let navbarToggle = select(".mobile-nav-
toggle");

        navbarToggle.classList.toggle("bi-list");
        navbarToggle.classList.toggle("bi-x");
      }
      scrollTo(this.hash);
    }
  },
  true
);

const onBookClick = (event) => {
  const type = event.target.dataset.type;
  console.log(type);
  window.location.href = `book_room.html?type=${type}`;
};

const createRooms = (rooms) => {
  const container = select(".carousel-inner");
  let html = container.innerHTML;
  rooms.forEach((room, i) => {

```

```

    html += `
    <div class="carousel-item ${i === 0 ? "active" : ""}">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>${room.room_type}</h5>
        <p>${room.description}</p>
        <button type="button" data-
type="${room.room_type}" class="btn btn-dark
book_btn">Забронювати</button>
      </div>
    </div>`;
  });

  container.innerHTML = html;
  on("click", ".book_btn", onBookClick, true);
};

const generateRooms = async () => {
  const rooms = await fetchRooms();
  createRooms(rooms);
};

window.addEventListener("load", generateRooms);
})();

```

### Лістинг А.13 Файл utils.js

```

// This is just file with 'helper' functions

/**
 * selects element from DOM
 */
export const select = (el, all = false) => {
  el = el.trim();
  if (all) {
    return [...document.querySelectorAll(el)];
  } else {
    return document.querySelector(el);
  }
};

/**
 * Easy event listener function
 */
export const on = (type, el, listener, all = false) => {
  let selectEl = select(el, all);
  if (selectEl) {
    if (all) {
      selectEl.forEach((e) => e.addEventListener(type,
listener));
    }
  }
};

```

```

        } else {
            selectEl.addEventListener(type, listener);
        }
    }
};

export const createEl = (type) => document.createElement(type);

export const getQueryParams = () =>
    new Proxy(new URLSearchParams(window.location.search), {
        get: (searchParams, prop) => searchParams.get(prop),
    });

export const showAlert = (text, type) => {
    const html = `
    <div class="alert alert-${type}" role="alert">
        ${text}
    </div>`;
    const container = select(".error-container");
    container.style.display = "block";
    select(".error-container").innerHTML = html;
    container.onclick = () => (container.style.display = "none");

    setTimeout(() => {
        container.style.display = "none";
    }, 7000);
};

```

### Лістинг А.14 Частина файлу book\_room.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta content="width=device-width, initial-scale=1.0"
name="viewport">

    <title>Lumia</title>
    <meta content="" name="description">
    <meta content="" name="keywords">

    <!-- Favicons -->
    <link href="assets/img/logo.jpg" rel="icon">
    <link href="assets/img/apple-touch-icon.png" rel="apple-touch-
icon">

    <!-- Google Fonts -->
    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,4
00,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,6
00i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i"
rel="stylesheet">

```

```

    <!-- Vendor CSS Files -->
    <link href="assets/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
    <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">
    <!-- Template Main CSS File -->
    <link href="assets/css/style.css" rel="stylesheet">
    <link href="assets/css/book_room.css" rel="stylesheet">
</head>

<body>

    <!-- ===== Header ===== -->
    <header id="header" class="fixed-top d-flex align-items-center">
        <div class="container d-flex align-items-center">

            <div class="logo me-auto">
                <h1><a href="index.html">Lumia</a></h1>
            </div>

            <nav id="navbar" class="navbar order-last order-lg-0">
                <ul>
                    <li><a class="nav-link scrollto"
href="#contact">Контакти</a></li>
                </ul>
            </nav><!-- .navbar -->

        </div>
    </header><!-- End Header -->

    <main id="main">
        <!-- ===== Rooms Section ===== -->
        <section class="rooms-section section-bg">
            <div class="container">
                <div class="section-title">
                    <h2 class="room_type_title"></h2>
                </div>

                <div class="rooms-container">
                    <div class="container">
                        <div class="images_container">
                            <div class="secondary_images"></div>
                        </div>
                        <h5 class="full_description"></h5>
                        <p class="price"></p>
                    </div>
                </div>
            </div>
        </section>
    </main>

```

## Лістинг А.16 Частина файлу index.html

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0"
name="viewport">

  <title>Lumia</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/logo.jpg" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-
icon">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,4
00,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,6
00i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i"
rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
  <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="assets/css/style.css" rel="stylesheet">
</head>

```

### Лістинг А.17 Файл index.js (Папка db)

```

const sql = require("mssql");

const connectionParams = {
  server: process.env.DB_HOST,
  user: process.env.DB_USER,
  database: process.env.DB_DATABASE,
  password: process.env.DB_PASSWORD,
  port: process.env.DB_PORT,
  pool: { idleTimeoutMillis: 30000 },
  options: {
    encrypt: false,
    enableArithAbort: false,
    instanceName: 'SQLEXPRESS'
  },
};

const createDBConnection = async () => {
  try {

```

```

        // FOR MySQL
        // const connection =
mysql.createConnection(connectionParams);
        // return connection.promise();

        const pool = await sql.connect(connectionParams);
        console.error("Connection to DB - success");
        return pool;
    } catch (error) {
        console.error("Connection to DB failed error:", error);
    }
    return null;
};

// // create the connection to database
const connection = createDBConnection();

module.exports = connection;

```

### Лістинг А.18 Частина файлу room.js

```

const sql = require("mssql");

const buildResponse = (response) => {
    const result = { error: null, data: null, code: null };

    if (response instanceof Error) {
        result.error = response.message;
    } else result.data = response;

    return result;
};

const bookRoom = async ({ reservation, user }) => {
    try {
        const availableRooms = await
getAvailableRooms(reservation);

        if (!availableRooms || !availableRooms.length)
            throw new Error("Не вдалось знайти вільну кімнату");

        const userId = await insertGuest(user);
        if (!userId) throw new Error("Помилка при створенні
користувача");

        const roomId = availableRooms[0].id;

        const result = await sql.query(
            `insert into reservation (room_id, guest_id,
number_of_guest, check_in_date, check_out_date,
date_of_reservation) values

```



```

        ('${roomId}', '${userId}', '${reservation.guests}',
    '${
        reservation.start_date
        }', '${reservation.end_date}', '${new
Date().toISOString()}');
        SELECT SCOPE_IDENTITY() AS id;`
    );

    if (!result.recordset[0].id) throw new Error("Internal
server error");
    return buildResponse(result.recordset[0].id);
  } catch (error) {
    console.error("bookRoom error:", error);
    return buildResponse(error);
  }
};

```

### Лістинг А.19 Файл env

```

PORT=3000
DB_HOST="WIN-MJSQ4L6F5TO"
DB_USER="hotels_login"
DB_DATABASE=hotel
DB_PASSWORD="123456"
DB_PORT=1434

```

### Лістинг А.20 index.js (серверна частина).

```

// THIS IS Server
const express = require("express"); // express the same as server
const cors = require("cors");

require("dotenv").config();
require("./db");

const {
  getRooms,
  bookRoom,
  getAvailableRooms,
  getRoomsCount,
} = require("./services/room");

const app = express();
const PORT = process.env.PORT || 3000;

app.use(express.json());
app.use(cors());

app.get("/api/rooms", async (request, response) => {
  const rooms = await getRooms();
  return response.send({ data: rooms });
});

```

```

});

app.get("/api/rooms/available", async (request, response) => {
  const rooms = await getAvailableRooms();
  return response.send({ data: rooms });
});

app.post("/api/rooms/book", async (request, response) => {
  const reservationData = request.body;
  // simple validation
  if (!reservationData.reservation || !reservationData.user) {
    return response.status(400).send({ error: "Invalid data"
  });
  }
  const result = await bookRoom(reservationData);
  response.send(result);
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

```

#### Лістинг А.21 Файл rooms\_images\_paths

```

[
  {
    "type": "Стандарт одномісний",
    "image_paths": [
      "assets/img/rooms/standart/1.jpg",
      "assets/img/rooms/standart/22.jpg",
      "assets/img/rooms/standart/3.jpg"
    ]
  },
  {
    "type": "Стандарт двомісний",
    "image_paths": [
      "assets/img/rooms/standart_double/double1.jpg",
      "assets/img/rooms/standart_double/double2.jpg",
      "assets/img/rooms/standart_double/double3.jpg"
    ]
  },
  {
    "type": "Стандарт сімейний",
    "image_paths": [
      "assets/img/rooms/standart_family/family1.jpg",
      "assets/img/rooms/standart_family/family2.jpg",
      "assets/img/rooms/standart_family/family3.jpg"
    ]
  },
  {
    "type": "Напівлюкс",
    "image_paths": [
      "assets/img/rooms/halfLux/halfLux1.jpg",

```

```
        "assets/img/rooms/halfLux/halfLux2.jpg",
        "assets/img/rooms/halfLux/halfLux3.jpg"
    ]
},
{
    "type": "Люк",
    "image_paths": [
        "assets/img/rooms/lux/1.jpg",
        "assets/img/rooms/lux/2.jpg",
        "assets/img/rooms/lux/3.jpg"
    ]
}
]
```