НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматики та управління в технічних системах

Тема «Готель: сервіс бронювання номерів в готелі»\_

Курсова робота

З дисципліни «Технології розробки програмного забезпечення – 2. Технології розробки WEB-застосувань на платформі Microsoft.NET»

Керівник Виконавець

Доц. Лебідь Сергій Олександрович Бурачик Костянтин Леонідович

«Допущений до захисту» залікова книжка № ІС – 8107

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ гр. ІС-81

(Особистий підпис керівника) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

« » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020р. (особистий підпис виконавця)

Захищений з оцінкою « »\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_у

(оцінка)

Члени комісії:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(особистий підпис) (розшифровка підпису)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(особистий підпис) (розшифровка підпису)

Київ – 2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

(назва навчального закладу)

Кафедра АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ТА УПРАВЛІННЯ

Дисципліна «Техології розробки програмного забезпечення – 2. Технології розробки WEB-застосувань на платформі Microsoft.NET»

Курс IIІ Група ІС-81 Семестр I\_

**ЗАВДАННЯ**

**на курсову роботу студента**

**Бурачик Костянтин Леонідович**

(прізвище, ім’я, по батькові)

1.Тема\_роботи Готель: сервіс бронювання номерів в готелі

2. Строк здачі студентом закінченої роботи 24 грудня 2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Вихідні дані до роботи:

Мова програмування C#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Дата видачі завдання 19 вересня 2020 року\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№,**  **п/п** | **Назва етапів виконання курсової роботи** | **Строк**  **виконання**  **етапів**  **роботи** | **Підписи або**  **примітки** |
| 1. | Затвердження теми курсової роботи | 1 тиждень |  |
| 2. | Аналіз предметної області та огляд існуючих рішень | 1 тиждень |  |
| 3. | Розробка концептуальної моделі | 1 тиждень |  |
| 4. | Розробка даталогічної моделі та вибір засобів реалізації проекту | 1 тиждень |  |
| 5. | Створення бази даних та розробка застосування | 3 тижня |  |
| 6. | Оформлення пояснювальної записки | 1 тиждень |  |
| 7. | Здача курсової роботи на перевірку | 1 тиждень |  |
| 8. | Захист курсової роботи | 1 тиждень |  |

**Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Бурачик Костянтин Леонідович\_\_

(підпис) (прізвище, ім’я, по батькові)

**Керівник\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Крамар Юлія Михайлівна

(підпис) (прізвище, ім’я, по батькові)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р.

Зміст

[**1** **Постановка задачі** 6](#_Toc60200697)

[**1.1** **Загальні вимоги до програмного забезпечення** 6](#_Toc60200698)

[**1.2** **Функціональні вимоги до програмного забезпечення за варіантом.** 6](#_Toc60200699)

[**2** **Теоретичні відомості** 8](#_Toc60200700)

[**2.1** **Трирівнева архітектура** 8](#_Toc60200701)

[**2.2** **Web API** 8](#_Toc60200702)

[**3** **ЗАГАЛЬНИЙ ОПИС СКЛАДОВИХ ПРОЕКТУ** 9](#_Toc60200703)

[**3.1** **Загальна структура DAL** 9](#_Toc60200704)

[**3.2** **Загальна структура BLL** 9](#_Toc60200705)

[**3.3** **Загальна структура PL** 10](#_Toc60200706)

[**4** **Скріншоти роботи програми** 16](#_Toc60200716)

[Висновки 18](#_Toc60200717)

[Перелік посилань 19](#_Toc60200718)

# **Постановка задачі**

## **Загальні вимоги до програмного забезпечення**

Розробити програмну частину та базовий інтерфейс для демонстрації коректної роботи веб-застосунку за темою «Готель: сервіс бронювання номерів в готелі»:

* В якості типу застосування обрати застосування з web-інтерфейсом на основі технологій asp.net MVC чи web API.
* Спроектувати та реалізувати систему класів, в основу якої покладено логічну структуру даних, для накопичення та обробки даних домену.
* Код програмної системи має складатись не менш як з трьох частин (окремих проектів) відповідно до багатошарової архітектури системи, де шарами архітектури є: шар (рівень) доступу до даних (DAL), шар бізнес логіки (BLL), шар представлення (інтерфейс програмної системи) (PL).

## **Функціональні вимоги до програмного забезпечення за варіантом.**

Повинні бути реалізовані наступні функції:

1. Управління готелями:

1.1. Можливість додавати готель

1.2. Можливість видаляти готель

1.3. Можливість перегляду готелю

1.4. Можливість перегляду всіх готелів: опис і кількість місць

1.5. Добавлення заявки на замовлення певного номера на певний термін

1.6. Можливість видалення заявки на замовлення певного номера

1.7. Можливість замінити текст заявки на замовлення певного номера в готелі

1.8. Можливість переглянути даних заявок на замовлення номерів в готелі за певний термін.

2. Управління клієнтами

2.1. Можливість добавлення клієнтів

2.2. Можливість видалення клієнтів

2.3. Можливість змінення даних про клієнтів

2.4. Можливість переглянути дані про конкретного клієнта

2.5. Можливість переглянути дані про всіх клієнтів

2.6. Можливість відсортувати список по імені

2.7. Можливість відсортувати список по прізвищу

3. Управління замовленнями номерів

3.1. Можливість замовити клієнтом номер в певному готелі.

3.2. Можливість відмінити замовити клієнтом номер в певному готелі.

3.3. 5.3.Можливість переглянути дані конкретного замовлення номера в певному готелі.

3.4. Можливість переглянути дані про кількість заброньованих місць в готелі, і які саме

3.5. Можливість переглянути дані про кількість вільних місць в готелі, і які саме.

3.6. Можливість переглянути дані про вартість послуг замовлення номерів з урахування вартості на один день.

3.7. Можливість переглянути дані про клієнтів, які забронювали номери в готелі.

4. Пошук

4.1. Можливість пошуку по ключовому слову серед готелів

4.2. Можливість пошуку по ключовому слову серед клієнтів

# **Теоретичні відомості**

## **Трирівнева архітектура**

Трирівнева n-layer архітектура – схема логічної організації певної програмної системи, яка складається з трьох частин:

* Presentation layer (рівень представлення) – це той рівень, з яким безпосередньо взаємодіє користувач. Містить компоненти користувацького інтерфейсу та механізм отримання введення від користувача.
* Business layer (рівень бізнес-логіки) – містить набір компонентів, що відповідають за обробку отриманих з рівня представлення даних, реалізує всю логіку застосунку, розрахунки, взаємодіє з базою даних і передає рівню представлення результати обробки.
* Data Access Layer або рівень доступу до даних зазвичай містить моделі усіх даних, що зберігаються в базі, а також класи, з допомогою яких здійснюється взаємодія з бд.

## **Web API**

Web API представляє спосіб побудови програми ASP.NET, який спеціально заточений для роботи в стилі REST (Representation State Transfer або "передача стану представлення"). REST-архітектура передбачає застосування таких методів або типів запитів HTTP для взаємодії з сервером:

* GET
* POST
* PUT
* DELETE

# **ЗАГАЛЬНИЙ ОПИС СКЛАДОВИХ ПРОЕКТУ**

Проект складається з 3 частин: HotelApp.BLL, HotelApp.DAL, HotelApp.PL (Рисунок 3.1 - Складові частини проекту)

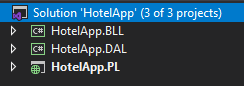


Рисунок 3.1 - Складові частини проекту

## **Загальна структура DAL**

Рівень DAL забезпечує взaємодію з базою даних. Він складається з основних груп класів: сутностей, засобів Entity Framework для налаштування зв’язку та роботи з базою даних, власних обробників виняткових ситуацій, інтерфейсів та репозиторіїв (Рисунок 3.2 - Складові частини DAL).

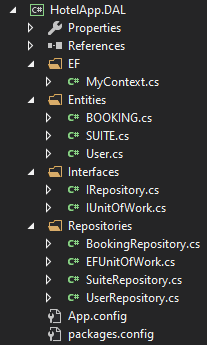


Рисунок 3.2 - Складові частини DAL

## **Загальна структура BLL**

Рівень BLL містить усю бізнес-логіку проекту. Він складається з основних груп класів: сутностей, що відповідають описаним на попередньому рівні, засобів мапінгу сутностей на різних рівнях, інтерфейсів та сервісів (Рисунок 3.3 - Складові частини BLL).

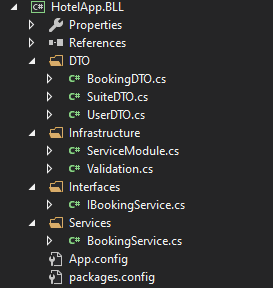


Рисунок 3.3 - Складові частини BLL

## **Загальна структура PL**

PL має класичну структуру WebAPI. Основними компонентами рівня є контролери для обробки даних та зв’язку з рівнем бізнес-логіки, а також представлення для введення користувачем даних та надання йому відповідної до запиту інформації (Рисунок 3.4 - Складові частини PL).

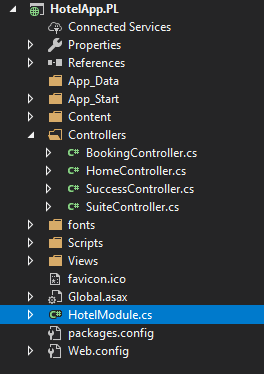


Рисунок 3.4 - Складові частини PL

# **Скріншоти роботи програми**

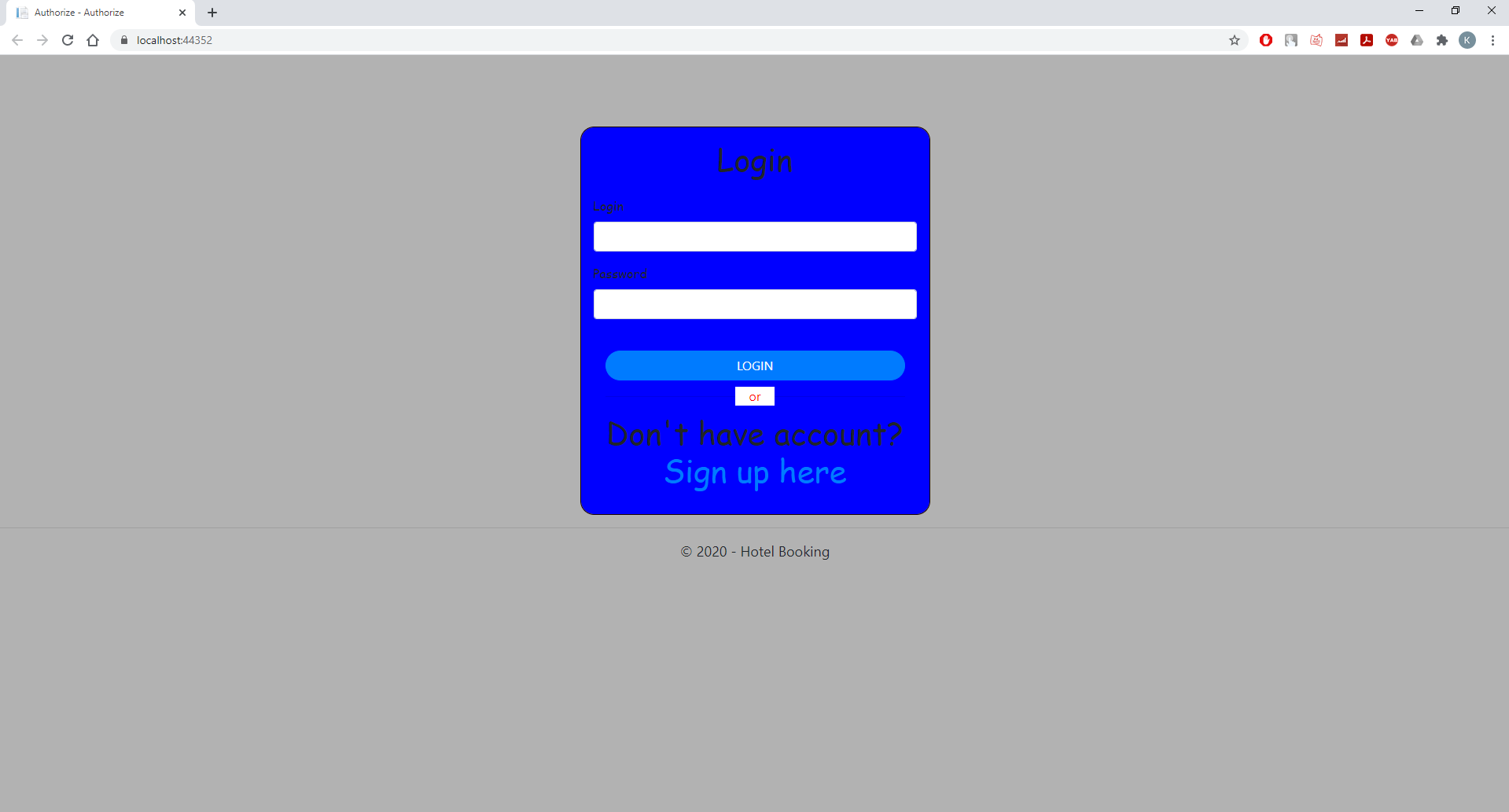


Рисунок 4.1 -Вхід в особистий кабінет

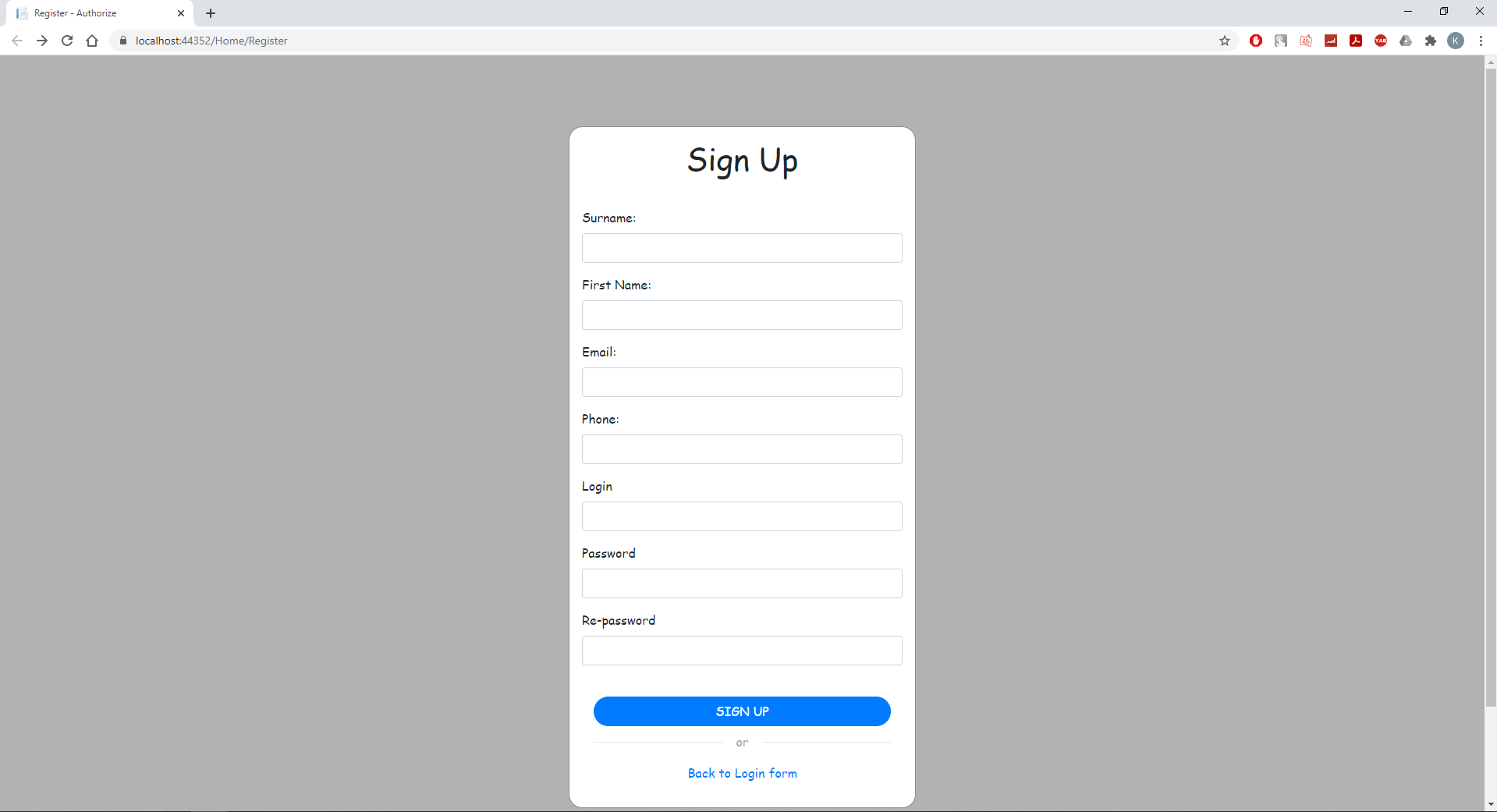


Рисунок 4.2 -Реєстрація нового користувача

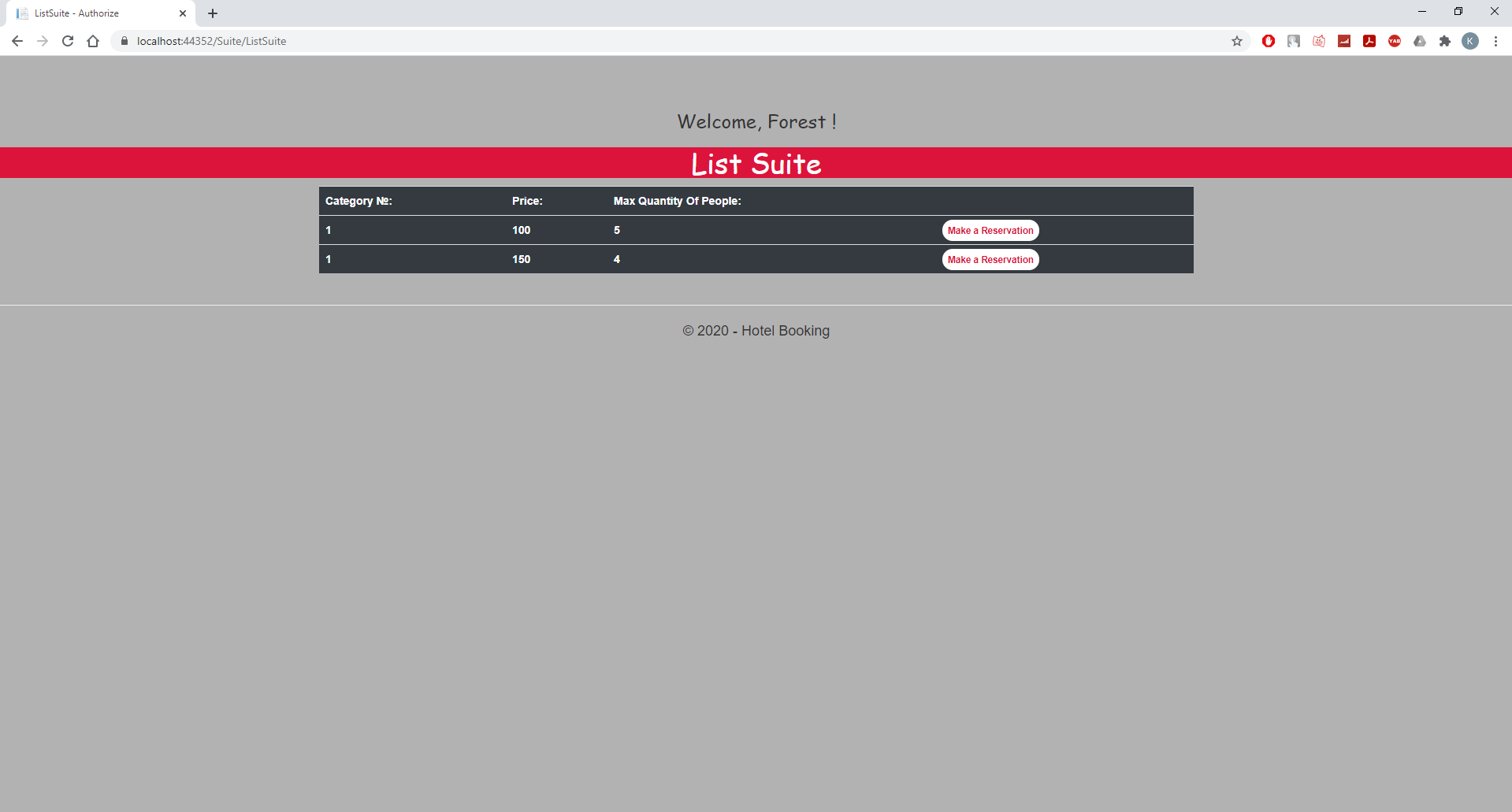


Рисунок 4.3 – Перегляд списку усіх номерів

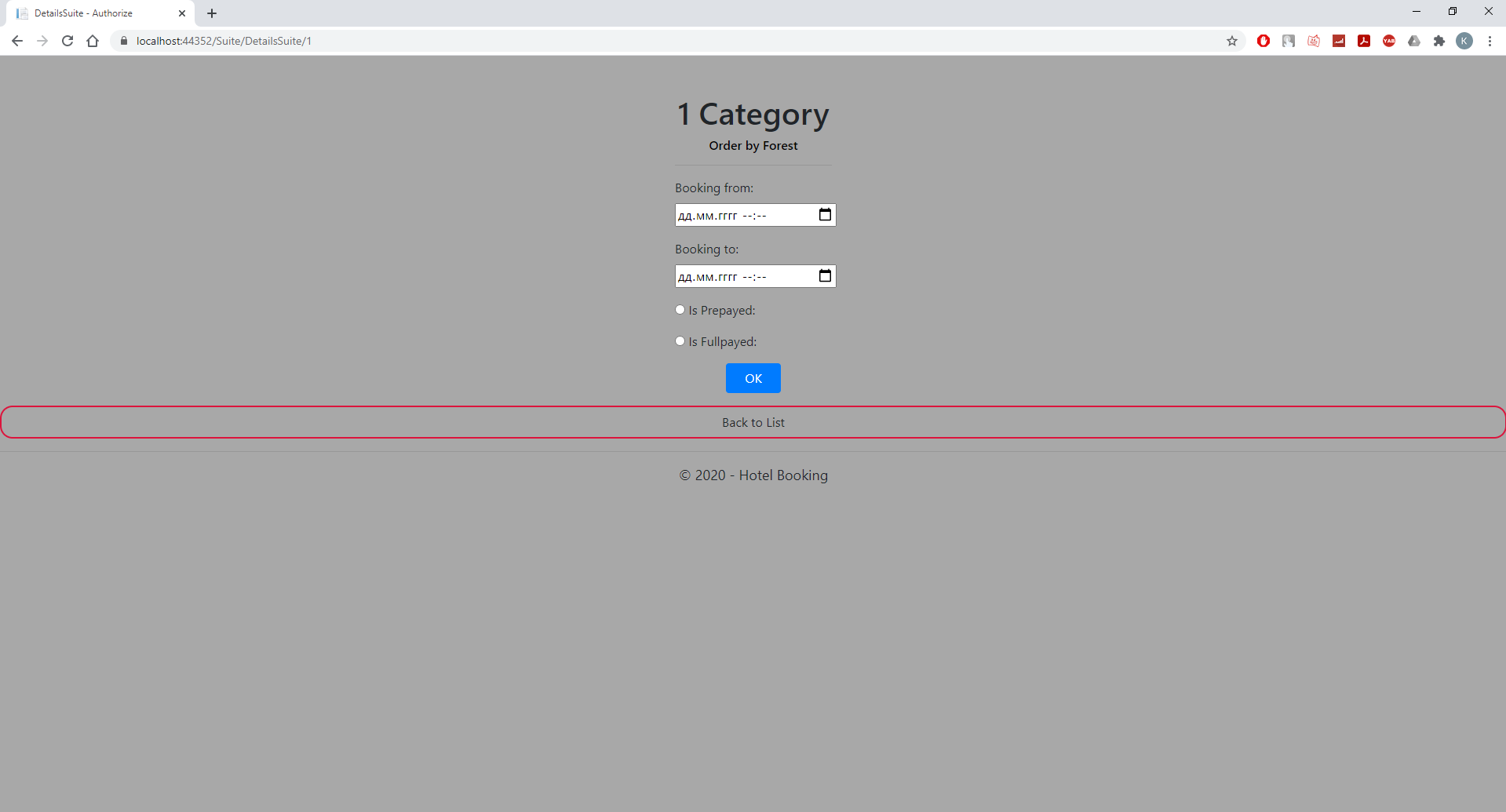


Рисунок 4.4 – Замовлення певного номеру на певний термін

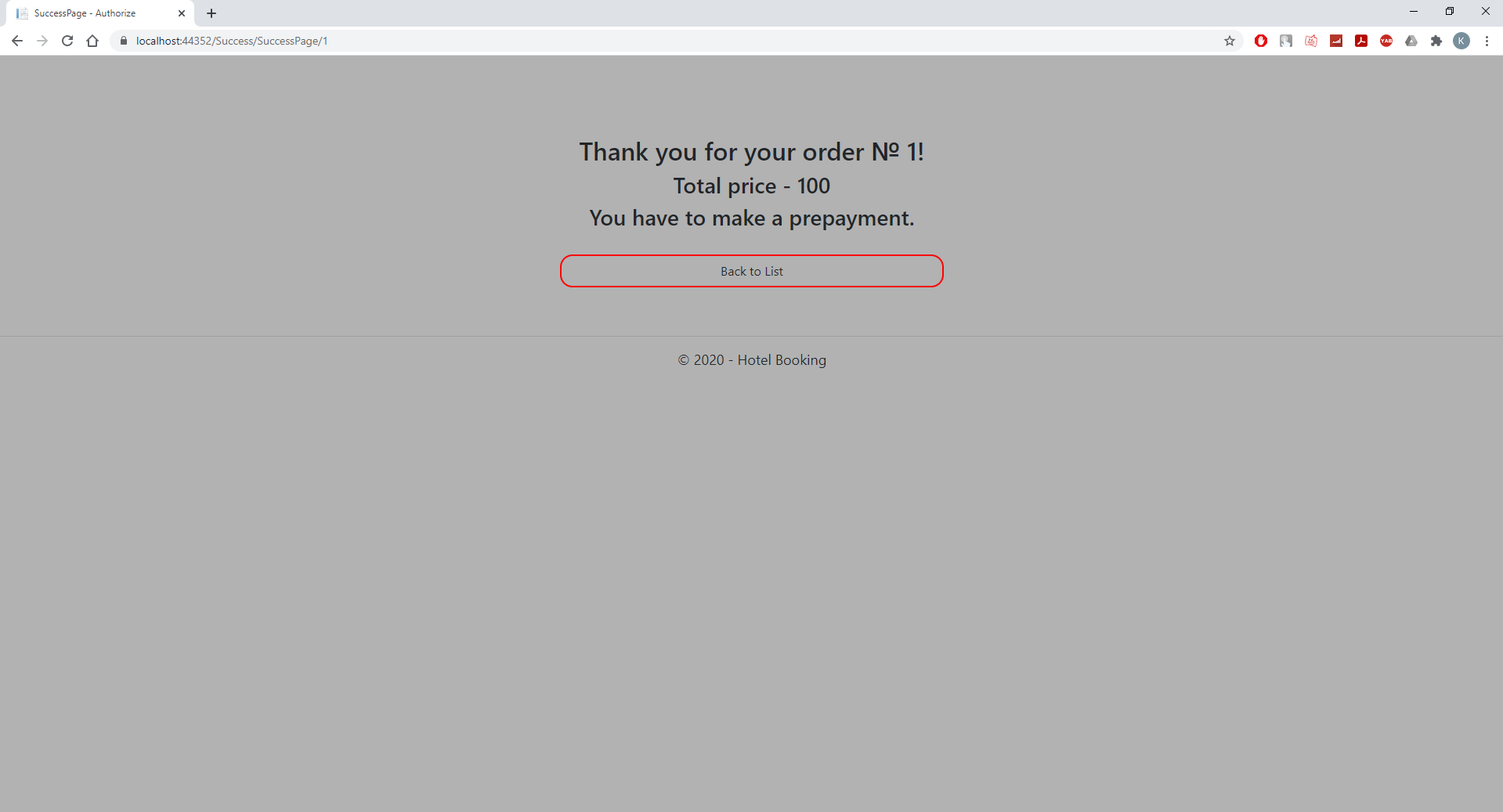


Рисунок 4.5 – Повідомлення про успішне бронювання номеру

Висновки

Під час виконання даної курсової роботи було досліджено принципи роботи з трирівневою n-layer архітектурою та WebAPI.

Окрім того, було розроблено структуру БД, що відповідає вимогам поставленої задачі. Було реалізовано взаємодію веб-застосунку з базою даних засобами Entity Framework.

Під час розробки застосунку було використано трирівневу архітектуру: рівень доступу до даних, що відповідає за взаємодію з БД, рівень бізнес-логіки, що реалізує усі підрахунки та обробку інформації, необхідні для функціонування застосунку, а також рівень представлень, що забезпечує взаємодію з користувачем.

Результатом даного курсового проекту є веб-застосунок для користування електронним гаманцем. Застосунок може бути корисним для використання в реальному житті, допоможе покращити навички планування власних фінансів.

Перелік посилань

1. Повне керівництво ASP.NET MVC 5 [Електронний ресурс]: Metanit – Адреса інтернет-сторінки: <https://metanit.com/sharp/mvc5/>
2. Багаторівнева архітектура [Електронний ресурс]: Metanit – Адреса інтернет-сторінки: <https://uk.wikipedia.org/wiki/Класифікація_комп’ютерних_ігор>.
3. Репозиторій з кодом проекту: https://github.com/Forestieur/CourseWork