A

Project Report

On

ONLINE HOTEL BOOKING SYSTEM

Developed by

GOHIL KAPILBHAI S. (CE015)

CHUDASAMA AKSHAYRAJSINH (CE007)

Department of CE, DD University

Guided By:

Prof. Ankit P. Vaishnav

Department of Computer Engineering

Faculty of Technology, Dharmsinh Desai University

College Road, Nadiad-387001

Department of Computer Engineering
Faculty of Technology, Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical/term work carried out in the subject of **Web Service Development** and recorded in this journal is the bonafide work of **Gohil Kapilbhai, ID No: 22CEUOD002 , Chudasama Akshayrajsinh, ID No: 21CEUOG108** of B.Tech semester **VI** in the branch of Computer Engineering during the academic year **2023-2024.**

Prof. Ankit P. Vaishnav	Dr. C.K. Bhensdadia
Assistant Professor,	Head,
Dept. of Computer Engineering,	Dept. of Computer Engineering,
Faculty of Technology,	Faculty of Technology,
Dharmsinh Desai University,	Dharmsinh Desai University,
Nadiad	Nadiad

List Of Figures

List Of Tables

Table of Contents

1 Introduction

1.1 Purpose

- The software should be useful for the hotel manager to manage the hotel in online mode and also for the growing business.
- The purpose of the online hotel booking system is to allow the customers to self-book the hotel rooms.
- The software should be used for securely storing and managing the data.
- Through the help of the software the customer can do emergency booking.
- This software saves the time from both sides (Hotel and Customer side).

1.2 Intended Audience

- This system is intended for providing room facility in online mode so that the peoples having the online banking or online payment options can do booking in online mode.

1.3 Project Scope

- The online hotel booking system project is intended for the online reservation of the rooms.
- It will be able to do various operation of the hotel booking automatic and in online mode like bill generation, room availability, data management etc.
- There are two user levels in hotel booking system: Customer, and Hotel Manager
- This software is consist of the booking management system and DBMS server.
- Customer will be able to check for room’s availability, select the rooms, and pay for the room.
- Manager will able to update room information such as cost and category.
- Hotel booking system is able to resolve the drawbacks of the reservation of room in offline mode.
- In future updates, software should be able to perform more tasks automatically like financial report generation, growth chart of the system, no of customers visited, shows the regular customer etc.

1.4 SOFTWARE INTERFACES:

- Front end technologies: HTML, CSS, JAVASCRIPT
- Back-end language: c#
- Development Tool: Visual Studio 2022
- Database server: MySQL
- Framework: .Net framework
- Architecture: Service based architecture

2 Software Requirement Specifications

2.1 Product Perspective

- The “Online Hotel Booking System” is a web application used for the book the room using internet via website.
- There manual method for the room booking has so many constrain and issues like more time Consumption, delay in processing, maintaining data (requires all the records of booking, recourses etc. Are maintained in books) etc.
- Hence sometime it is very difficult to book the room in emergency situation
- The purpose of this project is to develop and implement an online hotel booking system for hotels, that will replace the manual method of booking for hotel rooms.
- We are creating application that will used for the processing the online room booking.
- With the help of this web application hotel manager should be able to maintain the hotel with less time and also maintain and store data of the customer and hotel in proper manner without any time-consuming process.

2.2 Product Functions

1) Registration
2) log in
3) display the available rooms
4) Book a room
5) Update Room Details
6) Manage Booking Details
7) Generate bill
8) Store the details of the user in database

2.3 User Classes and Characteristics

There are 2 user classes in our Hotel Booking System:

1.Admin

2.Customer

2.3.1 Admin

- Admin have every access to the hotel system.
- Admin is responsible for managing hotel resources and staffs.
- Admin can view information report such as customer information, booking information, and room information.

2.3.2 Customers

- Customers are vital part of the system.
- Customer have access to view the vacant room information and price range.
- They should be able to confirm the booking and cancel it if necessary.
- Other than this customer can do the following actions:
    - Registration
    - Login
    - Search room availability
    - See the price of rooms and room details
    - Booking
    - Logout

2.4 Operating Environment

- The online hotel booking system is web application so it will operate in all browsers like Google, Chrome, Microsoft Edge, Firefox etc.
- Operating environment for this system will be hosted virtually using any cloud service provider.

2.5 Design And Implementation Constraints

- Internet connection is a constraint for this system because system is available from cloud therefore customer needs to have good network connection to connect to our web interface.

2.6 Assumptions And Dependencies

- It is assumed that system will be developed using .net framework with My SQL database will work perfectly.
- If incase of any difficulties, SRS should be flexible enough to change accordingly.

2.7 External Interface Requirements

2.7.1 User Interfaces

- The user interface for system shall be compatible to any type of web browser such as Mozilla Firefox, Google Chrome, and Internet Explorer.

2.7.2 Hardware Interface

2.7.2.1 Server side

- Monitor: resolution of 1024*768 or above
- Processor: intel or AMD 2GHZ
- Ram: 4 GB or above
- Disk space: 10GB or above

2.7.2.2 Client side

- Monitor: resolution of 1024*768 or above
- Processor: intel or AMD 1GHZ
- Ram: 512 MB or above
- Disk space: 2GB or above

2.7.3 Software Interfaces

- Front end technologies: HTML, CSS, JAVASCRIPT
- Back-end language: c#
- Development Tool: Visual Studio 2022
- Database server: MySQL

2.7.4 Communication Interfaces

- The System shall be using HTTP/HTTPS Or TCP/IP for communication over Internet

2.8 System Features

2.8.1 Functional Requirenments

- Our system has following general requirements:
  1) MANAGING ROOM DETAILS
  2) REGISTRATION
  3) LOGIN
  4) VIEW THE ROOM DETAILS:
  5) ROOM BOOKING

R.1 Managing Room Details

Description: Update the room details.

R.1.1 View available rooms

Input: select view room details option

Output: rooms are displayed

R.1.2 Update room details

Input: New details for room provided

Output: changes made will be affected in room details

R.2 Registration

Description: Users can register themselves

R.2.1 Select register option

Input: Register option selected

Output: User will be asked to enter Full Name, Phone Number, Email, Gender, password and confirm password

R.2.2 Provide details for registration

Input: details provided

Output: User will be registered

Processing: If user has not already registered then registration process will be done.

R.3 Login

Description: User and admin can login to the system

R.3.1 Select login option

Input: Login option selected

Output: User or Admin will be asked to enter username and password

Processing: If user has already registered him/herself then only he/she will be logged in to the system

R.4 View The Room Details

Description: User can check the details about every room before booking

R.4.1 Select book now option

Input: view rooms option selected

Output: shows the details about rooms

R.4.2 Select particular room to view the facilities

Input: room selected

Output: details of the room displayed

R.5 Room Booking

Description: user can book room and make payment

R.5.1 Select the room

Input: room selected

Output: User prompted to provide details

R.5.2 Provide the details

Input: Details provided

Output: User prompted with payment option

R.5.3 Select payment option.

Input: Payment option selected.

Output: requested method will be provided to pay the bill

R.5.4 bill generation

Input: select bill generate option

Output: Bill will be generated

2.9 Other Nonfunctional Requirements

2.9.1 Performance Requirements

NF1. Results for cross checking of availability of rooms in internal database and customer's choice shall be in 5 seconds.

NF2. Web page UI load time should within 3 seconds.

NF3. Redirection page load time should be within 3 seconds.

NF4. Data in database should be updated within 3 seconds.

NF5. Data in database should be updated within 2 seconds

2.9.2 Safety Requirements

NF6. Customer's personal details shall be encrypted.

2.9.3 Software Quality Attributes

- **Correctness:** This system should satisfy the normal regular Hotel Management operations precisely to fulfil the end user objectives
- **Efficiency:** Enough resources to be implemented to achieve the particular task efficiently without any hassle.
- **Flexibility:** System should be flexible enough to provide space to add new features and to handle them conveniently.
- **Integrity:** System should focus on the customer information and avoid data losses as much as possible
- **Portability:** The system should run in any Microsoft windows environment.
- **Maintainability:** The system should be maintainable.
- **Testability**: The system should be able to be tested to confirm the performance and client's specifications.

2.9.4 Goal Of Implementation

In the future we are implementing the following function into our system.

- Alert and notification
- Multiple payment gateway
- Guest history
- Easy retrieved and storage of data and information
- Daily count of guest that arrive and departure from hotel
- Daily weekly and monthly calculation of guest booked, arrived and departed from hotel
- Improved data management efficiency

3 Use-Case Diagram

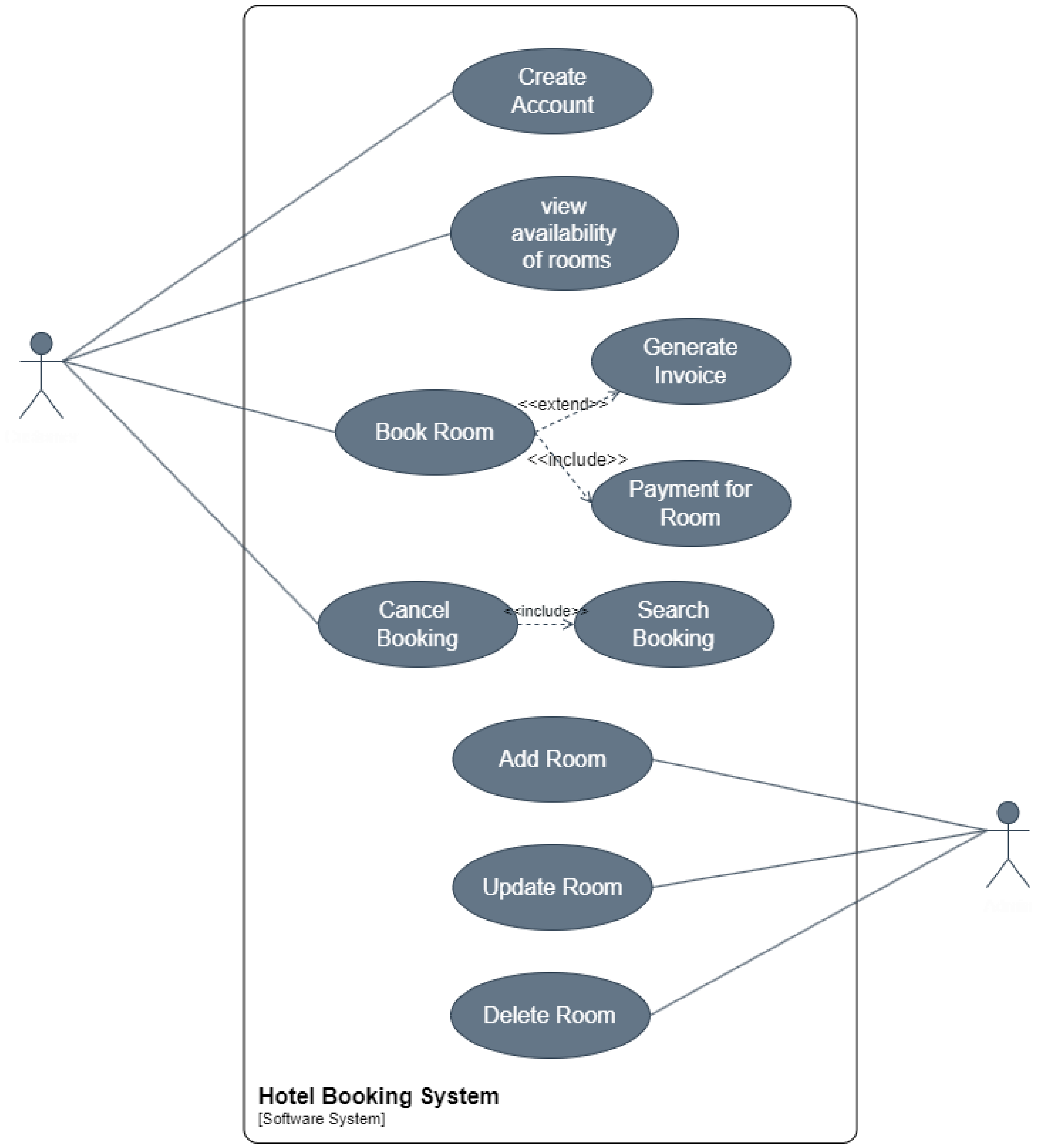


Figure 3.1 Use-Case Diagram

4 Class Diagram

4.1 Class Diagram

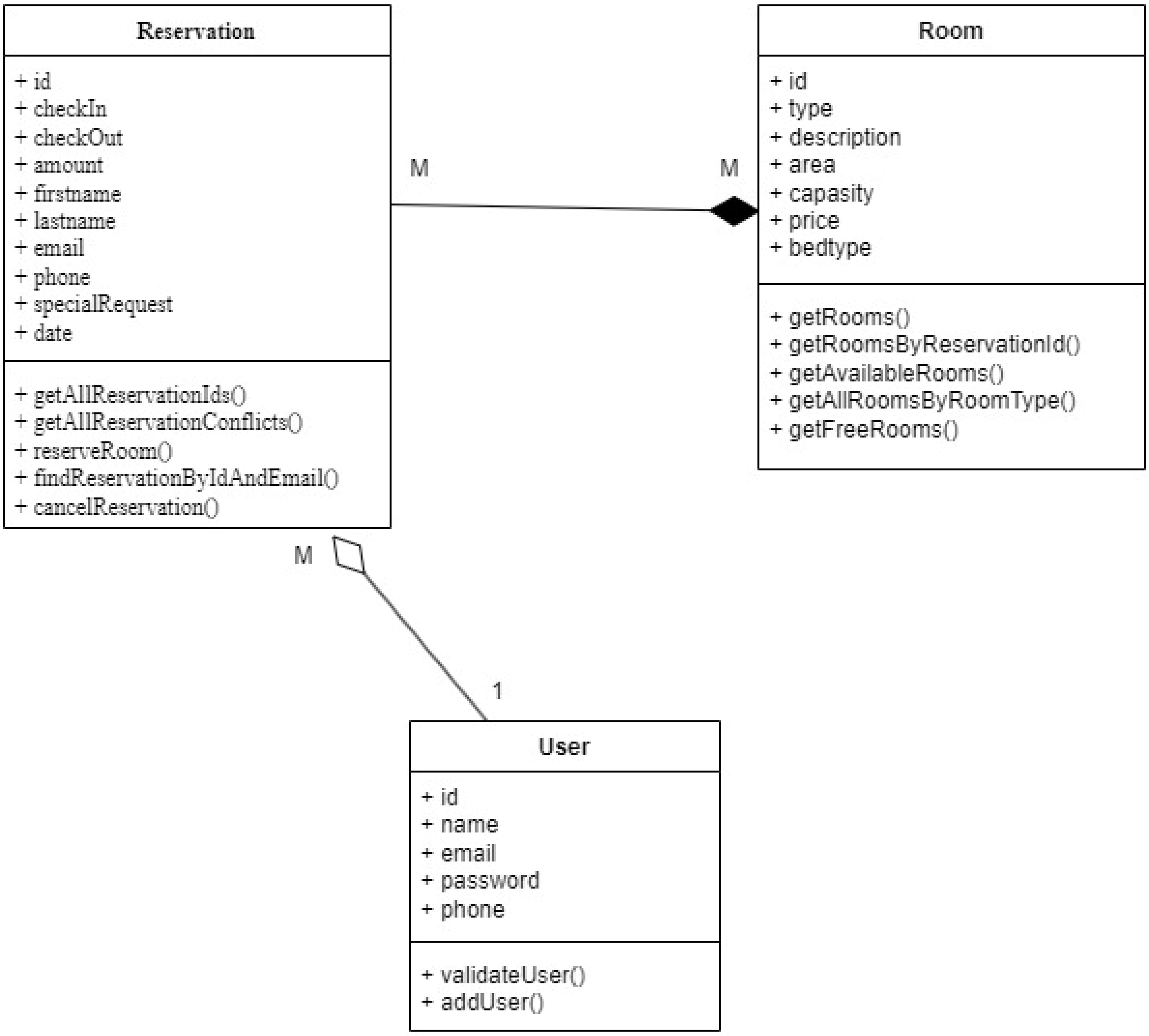


Figure 4.1 Class Diagram

5 Database Schema

5.1 E-R Diagram

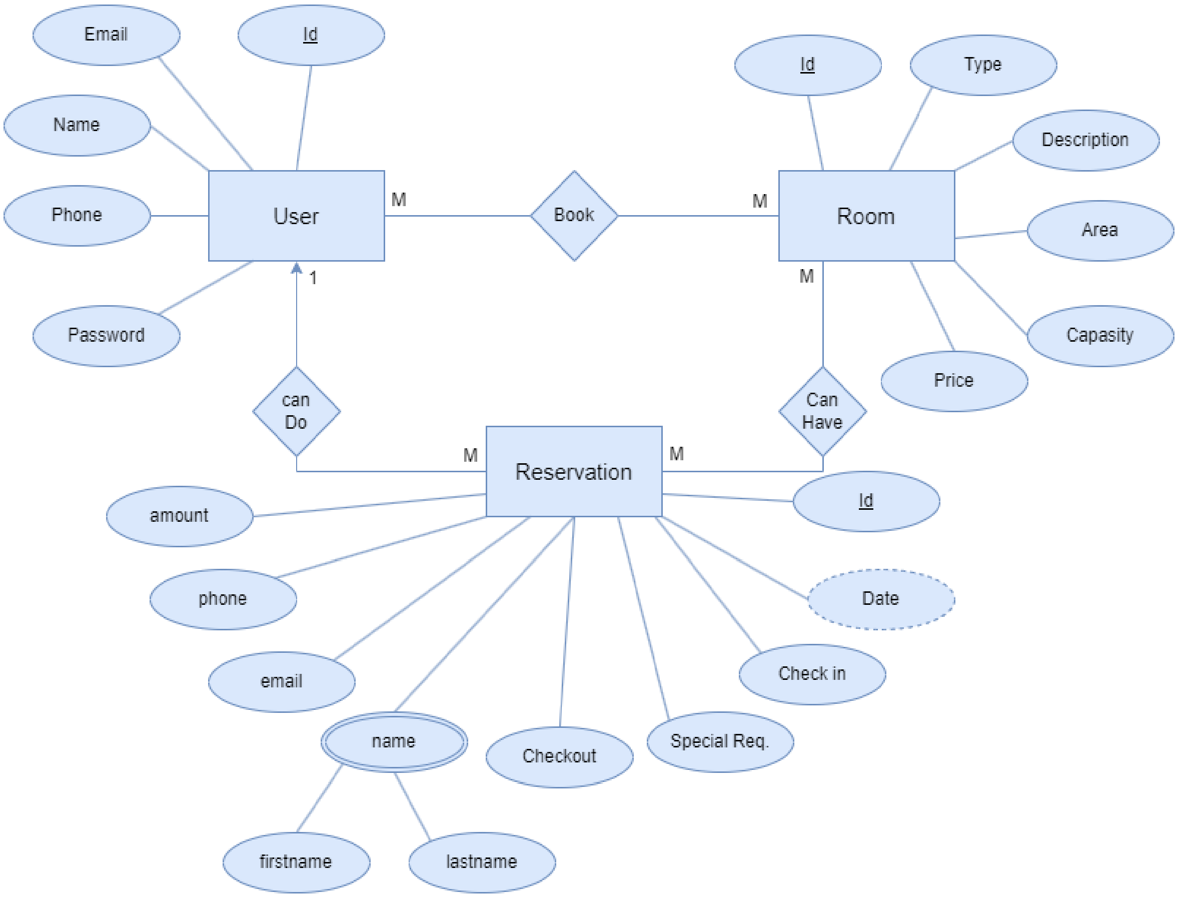


Figure 5.1 E-R Diagram

5.2 Data Dictionary

5.2.1 User

Sr no	Fieldname	Datatype	Field Length	constraints
1.	Id	Int	-	PK
2.	Email	Varchar	50	Not-Null
3.	Name	Varchar	50	Not-Null
4.	Phone	Varchar	50	Not-Null
5.	Password	Varchar	50	Not-Null

Table 5.1 User Data Dictionary

5.2.2 Room

Sr no	Fieldname	Datatype	Field Length	constraints
1.	Id	Int	-	PK
2.	Type	Varchar	50	Not-Null
3.	Desc	Varchar	500	Not-Null
4.	Area	Varchar	50	Not-Null
5.	Capacity	Numeric	(1,0)	Not-Null
6.	Bed Type	Varchar	50	Not-Null
7.	Price	Numeric	(18,0)	Not-Null

Table 5.2 Room Data Dictionary

5.2.3 Reservation

Sr no	Fieldname	Datatype	Field Length	constraints
1.	Id	Int	-	PK
2.	Checkin	Date	-	Not-Null
3.	Checkout	Date	-	Not-Null
4.	Amount	Int	-	Not-Null
5.	Firstname	Varchar	50	Not-Null
6.	Lastname	Varchar	50	Not-Null
7.	Email	Varchar	50	Not-Null
8.	Phone	Varchar	50	Not-Null
9.	SpecialRequest	Varchar	500	-
10.	Date	DateTime	-	Not-Null

Table 5.3 Reservation Data Dictionary

5.2.4 EachRoom:

Sr no	Fieldname	Datatype	Field Length	constraints
1.	Id	Int	-	PK
2.	RoomTypeId	Int	50	FK(Room.Id)

Table 5.4 EachRoom Data Dictionary

5.2.5 ReservationEachRoom:

Sr no	Fieldname	Datatype	Field Length	constraints
1.	Id	Int	-	PK
2.	ReservationId	Int	-	FK(Reservation.Id)
3.	EachRoomId	Int	-	FK(EachRoom.Id)
4.	Adult	Int	-	Null
5.	Children	Int	-	Null
6.	Price	nchar	10	Not-Null

Table 5.5 Reservation-EachRoom Data Dictionary

6 Implementation Details

6.1 Web Services (Interfaces, Contracts)

6.1.1 Iuserservice.Cs

```
namespace HotelService
{
    [ServiceContract]
    public interface IUserService
    {
        [OperationContract]
        User ValidateUser(string email, string pass);
        [OperationContract]
        bool AddUser(User u);
    }
}
```

6.1.2 Iroomservice.Cs

```
namespace HotelService
{
    [ServiceContract]
    public interface IRoomService
    {
        [OperationContract]
        List<Room> GetRooms();
        [OperationContract]
        List<RoomData> getRoomsByReservationId(int resId);
        [OperationContract]
        List<Room> getAvailabeRooms(int noOfRooms, int minCapasity, List<int>
ConflictedResIds);
        [OperationContract]
        List<int> getAllEachRoomsByReservationId(int rid);
        [OperationContract]
        List<int> getAllRoomsByRoomTypeId(int roomTypeId);
        [OperationContract]
        int getFreeRoomIdByRoomHeading(string heading);
    }
}
```

6.1.3 Ireservationservice.Cs

```
namespace HotelService
{
    [ServiceContract]
    public interface IReservationService
    {
        [OperationContract]
        bool cancelReservation(int resId);
        [OperationContract]
        ReservationFull FindReservationByIdAndEmail(int id, string email);
        [OperationContract]
        int BookRoom(FinalStorageData dt);
        [OperationContract]
        List<int> getAllReservationConflicts(DateTime checkin, DateTime checkout);
        [OperationContract]
        List<Reservation> GetALlReservationIds();
    }
}
```

6.2 Host (Form1.Cs)

```
namespace HotelHost
{
    public partial class Form1 : Form
    {
        public Form1()
```

```
{
   InitializeComponent();
}
ServiceHost sh = null;
ServiceHost sh2 = null;
ServiceHost sh3 = null;
private void button1_Click(object sender, EventArgs e)
{
   if (button1.Text=="Stop")
   {
      sh.Close();
      sh2.Close();
      sh3.Close();
      button1.Text = "Start";
      label1.Text = "Services are not Running";
   }
   else
   {
      string baseAddress = "http://localhost/HotelBookingServices/";
      Uri httpUri1 = new Uri(baseAddress+"RoomService");
      Uri httpUri2 = new Uri(baseAddress + "ReservationService");
      Uri httpUri3 = new Uri(baseAddress + "UserService");



      sh = new ServiceHost(typeof(HotelService.RoomService), httpUri1);
      sh2 = new ServiceHost(typeof(HotelService.ReservationService), httpUri2);
      sh3 = new ServiceHost(typeof(HotelService.UserService), httpUri3);

      BasicHttpBinding binding1 = new BasicHttpBinding();
      BasicHttpBinding binding2 = new BasicHttpBinding();
      BasicHttpBinding binding3 = new BasicHttpBinding();

      ServiceMetadataBehavior mBehcave1 = new ServiceMetadataBehavior();
      ServiceMetadataBehavior mBehcave2 = new ServiceMetadataBehavior();
```

```
                ServiceMetadataBehavior mBehcave3 = new ServiceMetadataBehavior();
                mBehcave1.HttpGetEnabled = true;
                mBehcave2.HttpGetEnabled = true;
                mBehcave3.HttpGetEnabled = true;
                sh.Description.Behaviors.Add(mBehcave1);
                sh2.Description.Behaviors.Add(mBehcave2);
                sh3.Description.Behaviors.Add(mBehcave3);

                sh.AddServiceEndpoint(typeof(IMetadataExchange),
MetadataExchangeBindings.CreateMexHttpBinding(), "mex");
                sh2.AddServiceEndpoint(typeof(IMetadataExchange),
MetadataExchangeBindings.CreateMexHttpBinding(), "mex");
                sh3.AddServiceEndpoint(typeof(IMetadataExchange),
MetadataExchangeBindings.CreateMexHttpBinding(), "mex");

                sh.AddServiceEndpoint(typeof(HotelService.IRoomService), binding1, httpUri1);
                sh2.AddServiceEndpoint(typeof(HotelService.IReservationService), binding2,
httpUri2);
                sh3.AddServiceEndpoint(typeof(HotelService.IUserService), binding3, httpUri3);

                sh.Open();
                sh2.Open();
                sh3.Open();

                label1.Text = "Services are Running";
                button1.Text = "Stop";
            }
        }
    }
}
```

6.3 Client (App.Config)

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>

  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008
/define:_MYTYPE=\&quot;Web\&quot; /optionInfer+" />
    </compilers>
  </system.codedom>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="BasicHttpBinding_IRoomService" />
        <binding name="BasicHttpBinding_IUserService" />
        <binding name="BasicHttpBinding_IReservationService" />
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint address="http://localhost/HotelBookingServices/RoomService"
        binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding_IRoomService"
        contract="RoomService.IRoomService" name="BasicHttpBinding_IRoomService" />
      <endpoint address="http://localhost/HotelBookingServices/UserService"
        binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding_IUserService"
        contract="UserService.IUserService" name="BasicHttpBinding_IUserService" />
      <endpoint address="http://localhost/HotelBookingServices/ReservationService"
        binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_IReservationService"
        contract="ReservationService.IReservationService"
name="BasicHttpBinding_IReservationService" />
    </client>
  </system.serviceModel>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="System.Runtime.CompilerServices.Unsafe"
publicKeyToken="b03f5f7f11d50a3a" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-5.0.0.0" newVersion="5.0.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

7 Testing

7.1 Testing Methodology Used

7.1.1 BlackBox Testing

In this section, we'll focus on the testing methodologies utilized in Share-Hub development, with a particular emphasis on black-box testing. Black-box testing is a software testing technique where the internal structure, design, or implementation details of the system are not known to the tester. Instead, the tester interacts with the system's external interfaces and observes its behaviour based on predefined inputs and expected outputs.

7.1.2 Advantages of Black-Box Testing

1. Independence from implementation details: Testers do not need access to the source code or knowledge of the system's internal design, making black-box testing suitable for testing third-party software or components.

2.Encourages thorough testing: Black-box testing encourages testers to explore various input combinations and scenarios, ensuring comprehensive test coverage and uncovering potential defects or inconsistencies in the system.

3.Emphasizes user perspective: By focusing on inputs and outputs from a user's perspective, black-box testing helps ensure that the system meets user requirements and expectations.

7.2 Test Suite

Test Case Id	Test Case Objective	Pre requisite	Steps	Expected Output	Actual Output	Status
TC_1.	Test for the user registration. (For the user not having account.)	User should not have the account.	1.Select the create account button. 2.Enter full name, email, phone and password 3.Submit.	User account should be created.	User account created.	PASS
TC_2.	Test for the user registration.	User should	1.Select the create account button.	The error message should be	The error message shown	PASS

	(For the user having existing account)	have account.	2.Enter full name, email, phone and password 3.Submit.	shown indicating user account already exist.	indicating user account already exist.	
TC_3.	Test for the login with incorrect mail and correct password.	No pre-requisite	1.Go to login page 2.Enter invalid email and valid password.	The error message should be shown regarding invalid credentials.	The error message shown indicating invalid credentials.	PASS
TC_4.	Test for the login with incorrect mail and incorrect password.	No pre-requisite	1.Go to login page 2.Enter invalid email and valid password.	The error message should be shown regarding invalid credentials.	The error message shown indicating invalid credentials.	PASS
TC_5.	Test for the login with correct mail and incorrect password.	No pre-requisite	1.Go to login page 2.Enter invalid email and valid password.	The error message should be shown regarding invalid credentials.	The error message shown indicating invalid credentials.	PASS
TC_6.	Test for the login with correct mail and correct password.	User should already have account.	1.Go to login page 2.Enter valid email and valid password.	The user should be logged in and redirected to the main page.	Successful login and user redirected to the main page.	PASS
TC_7.	Test for the booking room in given dates. (room already booked in given dates)	Room should be booked on give dates.	1.Click book a stay. 2.Enter check-in, no of adults, no of Children, no of rooms, check-out date. 3.Search	The room should be shown as not available.	The room is not available for booking.	PASS
TC_8.	Test for the booking room when no of person is more than	Room should be not booked but the capacity	1.Click book a stay. 2.Enter check-in, no of adults, no of Children, no of rooms, check-out date. 3.Search	The room should be shown as not available as the	The room shown as not available.	PASS

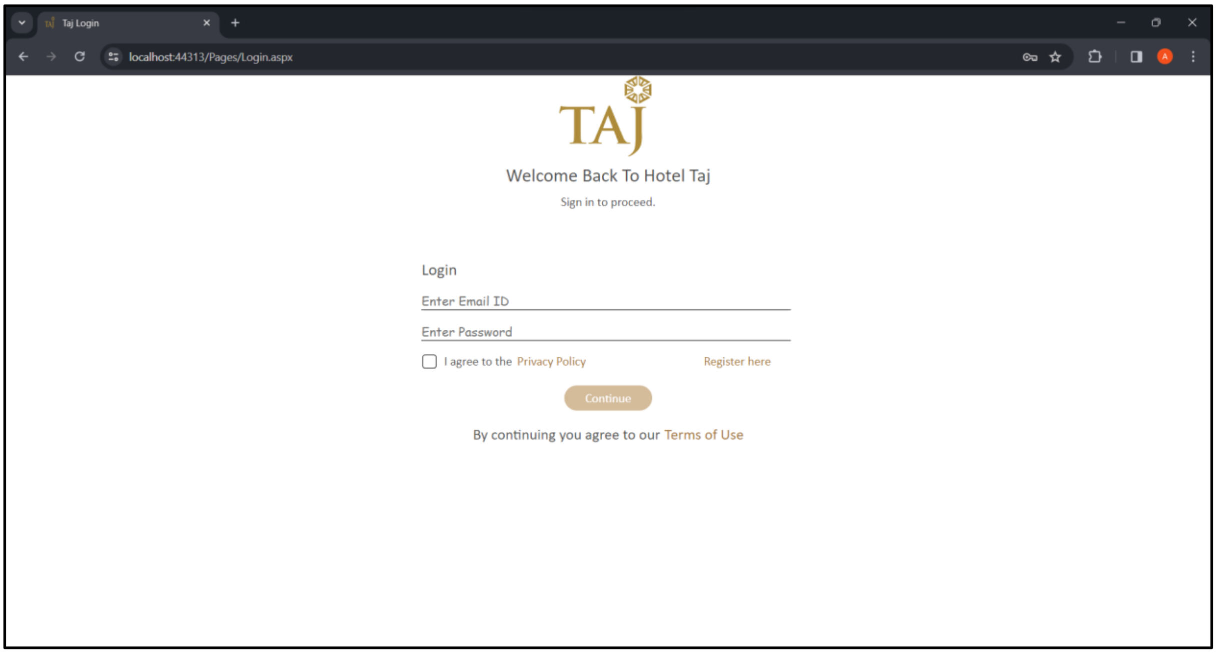
	room capacity.	should be less than the no person entered by user.		room capacity is less than the peoples.		
TC_9.	Test for the booking room. (room criteria matches)	Room should not be booked in given dates.	1.Click book a stay. 2.Enter check-in, no of adults, no of Children, no of rooms, check-out date. 3.Search	The room should be shown for the reservation.	Room shown for the registration.	PASS
TC_10.	Test for booking room with the invalid number entered. (number digit is less than 10)	User should have already selected the room and now filling booking form.	1.Don’ fill all the fields. 2. Submit.	The error message should be shown indicating the field is required.	The error message shown indicating the field is required.	PASS
TC_11.	Test for booking room with the invalid number entered. (number digit is less than 10)	User should have already selected the room and now filling booking form.	1.Enter invalid number. 2. Submit.	The error message should be shown indicating the invalid mobile number.	The error message shown indicating the invalid mobile number.	PASS
TC_12.	Test for the print functionality.	User should have already booked the room.	1.Click on print button.	User should be able to download the pdf of the bill generated.	User is able to view and download the pdf file for the reservation.	PASS
TC_13.	Test for the cancel reservation.	User should have already booked the room.	1.Click on the cancel reservation button.	The booking should be cancelled and room should be available for the booking	The booking cancelled and room is available for the booking in given dates.	PASS

				in given dates.		
TC_14.	Test for the searching room with wrong booking id.	User should have already booked the room.	1.Click on manage room details. 2.Enter the invalid booking id and submit.	The error message should be shown indicating the invalid credentials.	The error message shown indicating the invalid credentials.	PASS
TC_15.	Test for searching room with valid email and booking id.	User should have already booked the room.	1.Click on manage room details. 2.Enter the valid email and booking id and submit.	The booking details should be shown.	The booking details opened.	PASS

Table 7.1 Test Suite

8 User Interface

8.1 Login Page



Taj Login

localhost:44313/Pages/Login.aspx

TAJ

Welcome Back To Hotel Taj

Sign in to proceed.

Login

Enter Email ID

Enter Password

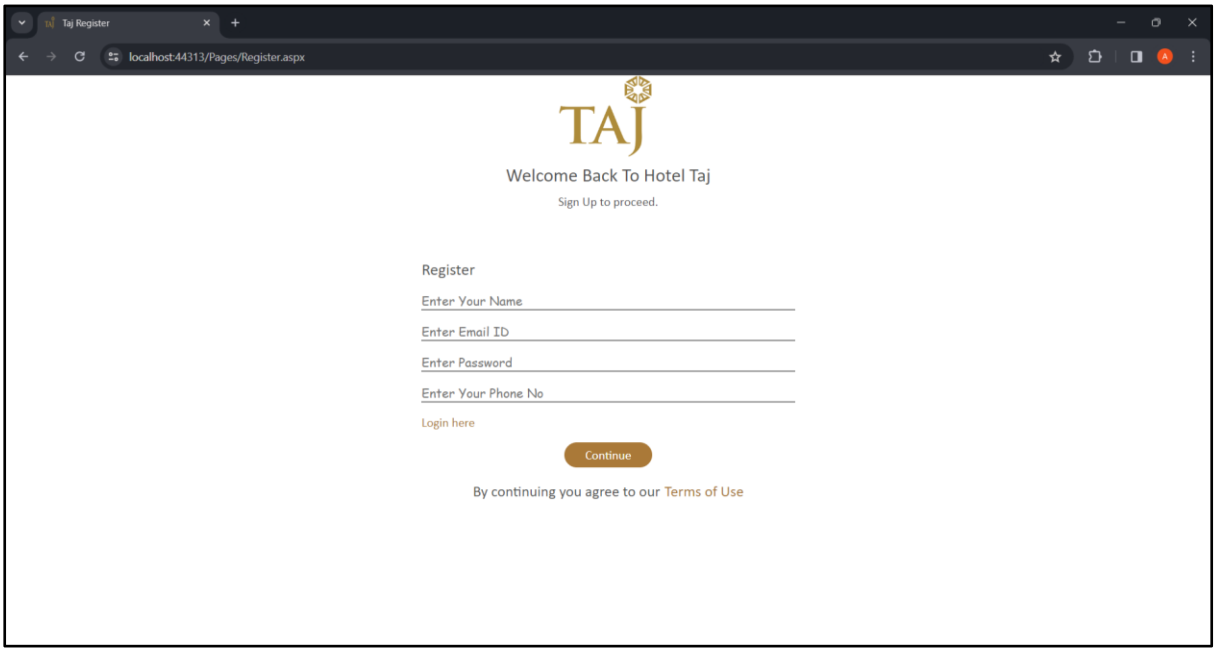
□ I agree to the Privacy Policy Register here

Continue

By continuing you agree to our Terms of Use

Figure 8.1 Login Page UI

8.2 Register Page



Taj Register

localhost:44313/Pages/Register.aspx

TAJ

Welcome Back To Hotel Taj

Sign Up to proceed.

Register

Enter Your Name

Enter Email ID

Enter Password

Enter Your Phone No

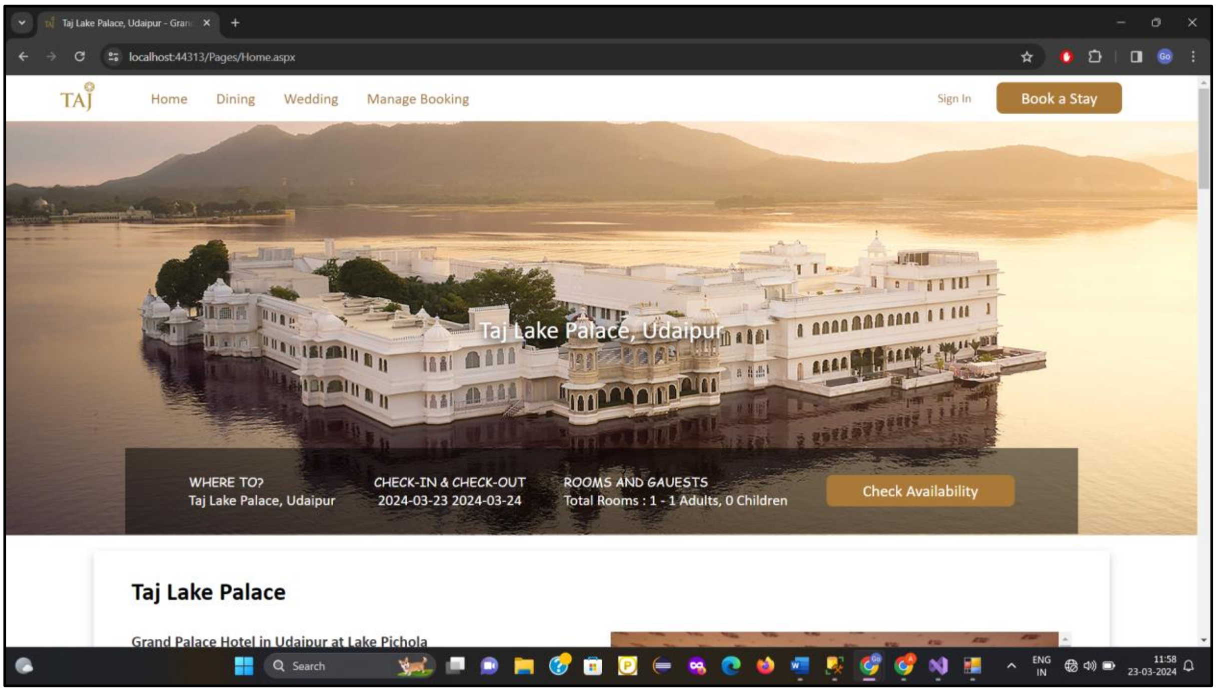
Login here

Continue

By continuing you agree to our Terms of Use

Figure 8.2 Register Page UI

8.3 Home Page



WHERE TO?
Taj Lake Palace, Udaipur

CHECK-IN & CHECK-OUT
2024-03-23 2024-03-24

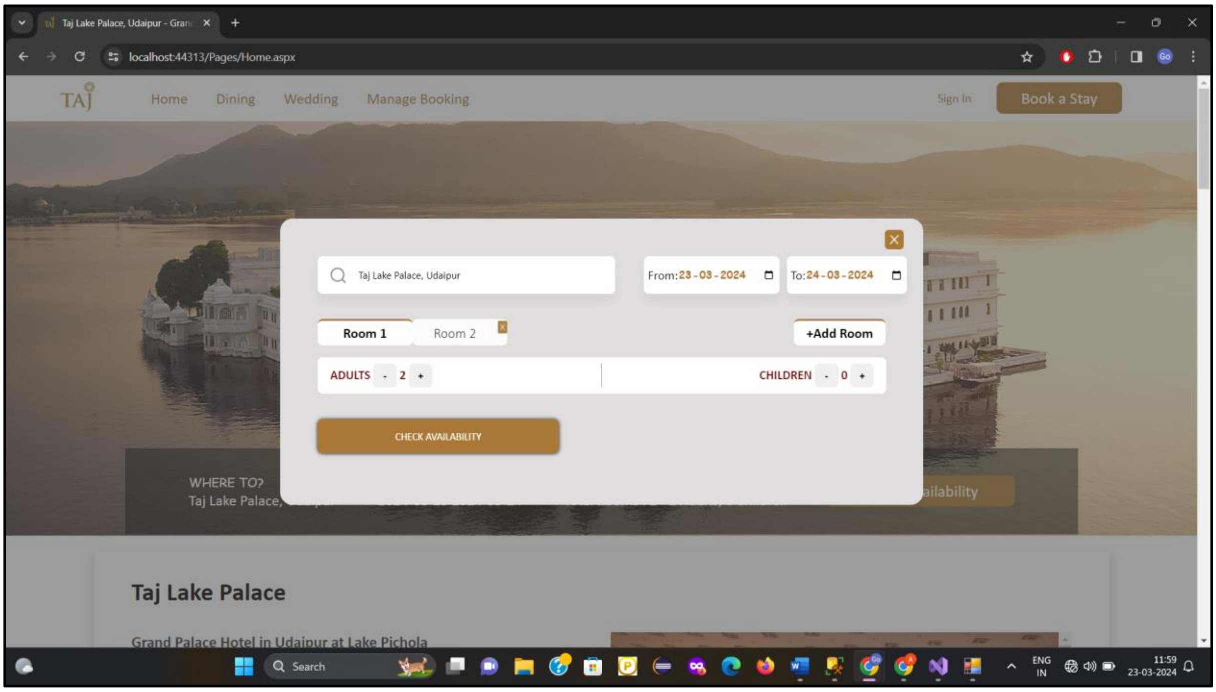
ROOMS AND GAUESTS
Total Rooms : 1 - 1 Adults, 0 Children

Taj Lake Palace

Grand Palace Hotel in Udaipur at Lake Pichola

Figure 8.3 Home Page UI

8.4 Search for Room Availability



Taj Lake Palace, Udaipur

From: 23 - 03 - 2024

To: 24 - 03 - 2024

ADULTS - 2 +

CHILDREN - 0 +

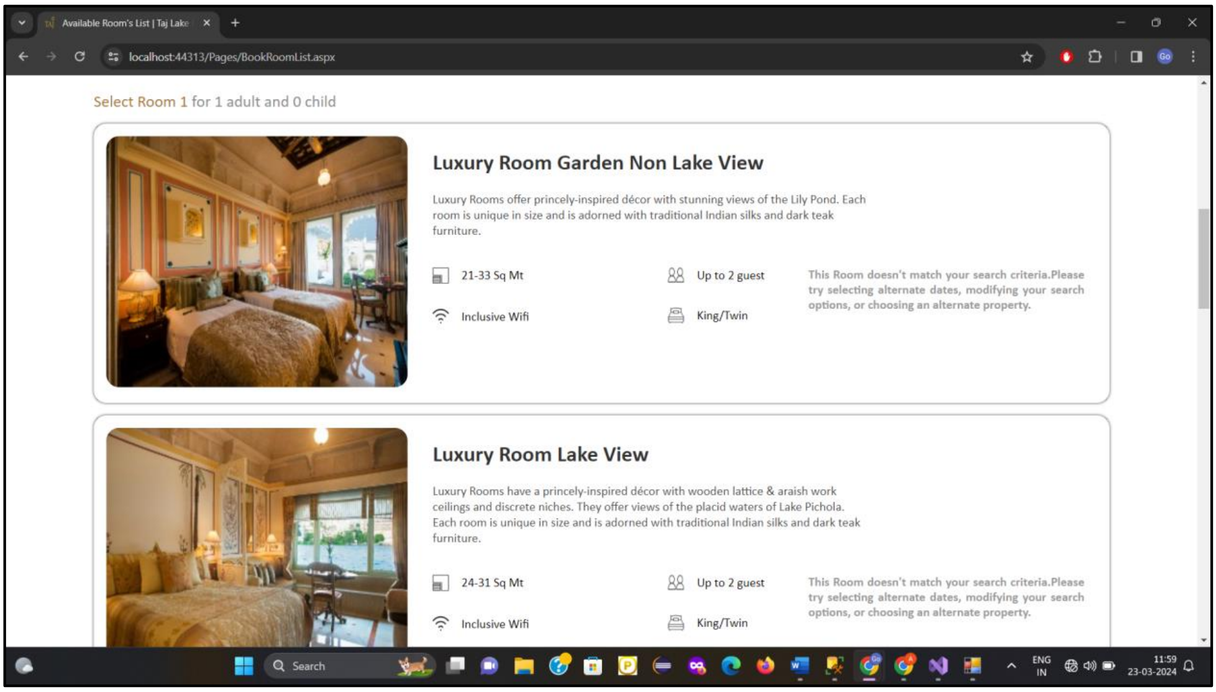
WHERE TO?
Taj Lake Palace,

Taj Lake Palace

Grand Palace Hotel in Udaipur at Lake Pichola

Figure 8.4 Search For Rooms UI

8.5 Available Rooms



Available Room's List | Taj Lake
localhost:44313/Pages/BookRoomList.aspx

Select Room 1 for 1 adult and 0 child

Luxury Room Garden Non Lake View

Luxury Rooms offer princely-inspired décor with stunning views of the Lily Pond. Each room is unique in size and is adorned with traditional Indian silks and dark teak furniture.

21-33 Sq Mt | Up to 2 guest
Inclusive Wifi | King/Twin

This Room doesn't match your search criteria.Please try selecting alternate dates, modifying your search options, or choosing an alternate property.

Luxury Room Lake View

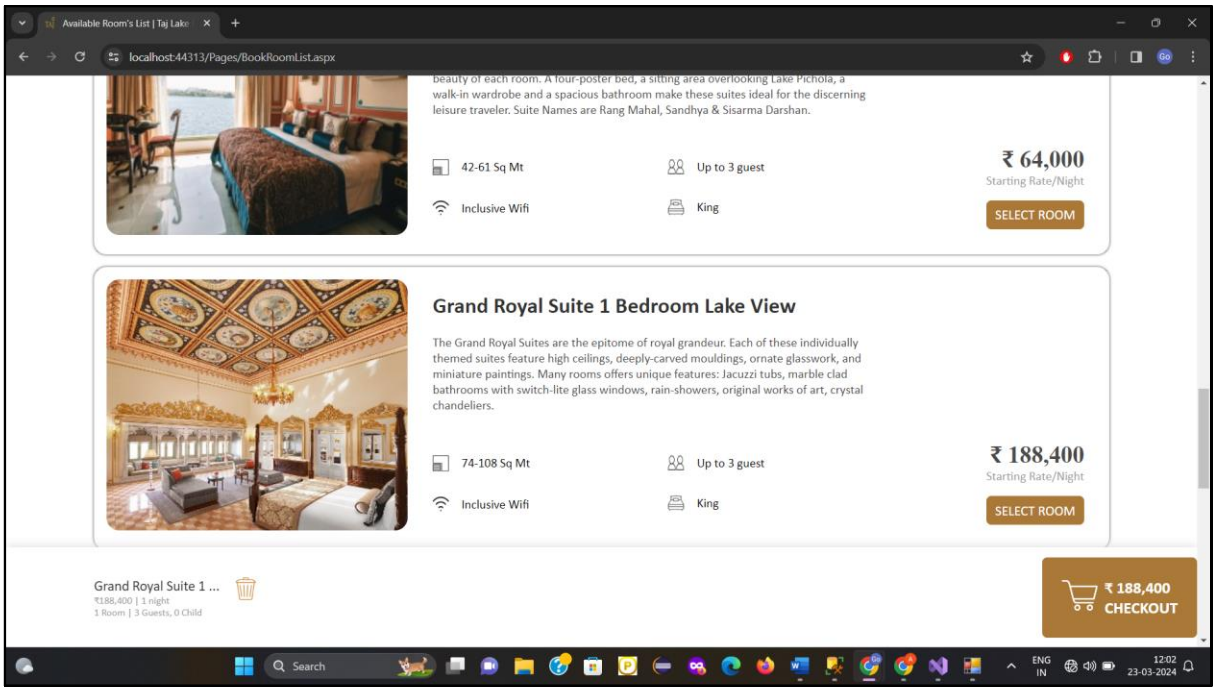
Luxury Rooms have a princely-inspired décor with wooden lattice & araish work ceilings and discrete niches. They offer views of the placid waters of Lake Pichola. Each room is unique in size and is adorned with traditional Indian silks and dark teak furniture.

24-31 Sq Mt | Up to 2 guest
Inclusive Wifi | King/Twin

This Room doesn't match your search criteria.Please try selecting alternate dates, modifying your search options, or choosing an alternate property.

Figure 8.5 Available Rooms UI

8.6 Select Rooms



Available Room's List | Taj Lake
localhost:44313/Pages/BookRoomList.aspx

beauty of each room. A four-poster bed, a sitting area overlooking Lake Pichola, a walk-in wardrobe and a spacious bathroom make these suites ideal for the discerning leisure traveler. Suite Names are Rang Mahal, Sandhya & Sisarma Darshan.

42-61 Sq Mt | Up to 3 guest
Inclusive Wifi | King

₹ 64,000
Starting Rate/Night

SELECT ROOM

Grand Royal Suite 1 Bedroom Lake View

The Grand Royal Suites are the epitome of royal grandeur. Each of these individually themed suites feature high ceilings, deeply-carved mouldings, ornate glasswork, and miniature paintings. Many rooms offers unique features: Jacuzzi tubs, marble clad bathrooms with switch-lite glass windows, rain-showers, original works of art, crystal chandeliers.

74-108 Sq Mt | Up to 3 guest
Inclusive Wifi | King

₹ 188,400
Starting Rate/Night

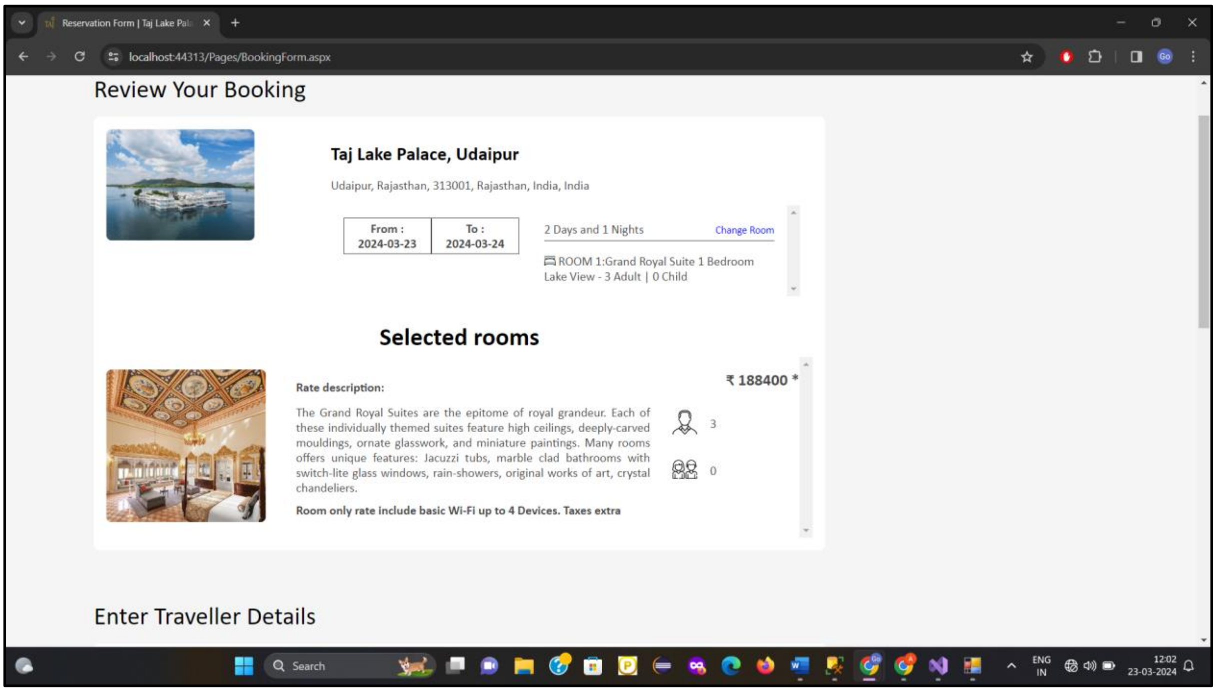
SELECT ROOM

Grand Royal Suite 1 ...
₹188,400 | 1 night
1 Room | 3 Guests, 0 Child

₹ 188,400
CHECKOUT

Figure 8.6 Select Rooms UI

8.7 Booking Details


Reservation Form | Taj Lake Pal
localhost:44313/Pages/BookingForm.aspx
Search
ENG
IN
12:02
23-03-2024

Review Your Booking

Taj Lake Palace, Udaipur

Udaipur, Rajasthan, 313001, Rajasthan, India, India

From :	To :
2024-03-23	2024-03-24

2 Days and 1 Nights Change Room

ROOM 1:Grand Royal Suite 1 Bedroom Lake View - 3 Adult | 0 Child

Selected rooms

₹ 188400 *

Rate description:

The Grand Royal Suites are the epitome of royal grandeur. Each of these individually themed suites feature high ceilings, deeply-carved mouldings, ornate glasswork, and miniature paintings. Many rooms offers unique features: Jacuzzi tubs, marble clad bathrooms with switch-lite glass windows, rain-showers, original works of art, crystal chandeliers.

Room only rate include basic Wi-Fi up to 4 Devices. Taxes extra

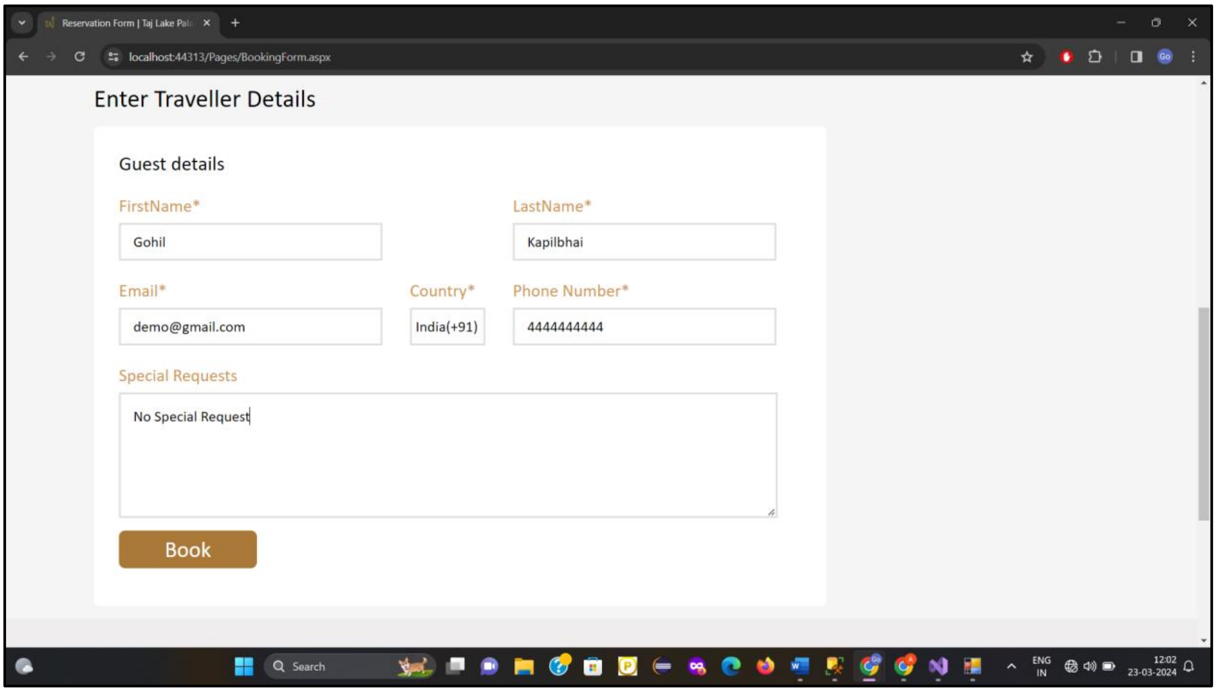
3

0

Enter Traveller Details

Figure 8.7 Booking Details UI

8.8 Booking Form


Reservation Form | Taj Lake Pal
localhost:44313/Pages/BookingForm.aspx
Book
Search
ENG
IN
12:02
23-03-2024

Enter Traveller Details

Guest details

FirstName*: Gohil

LastName*: Kapilbhai

Email*: demo@gmail.com

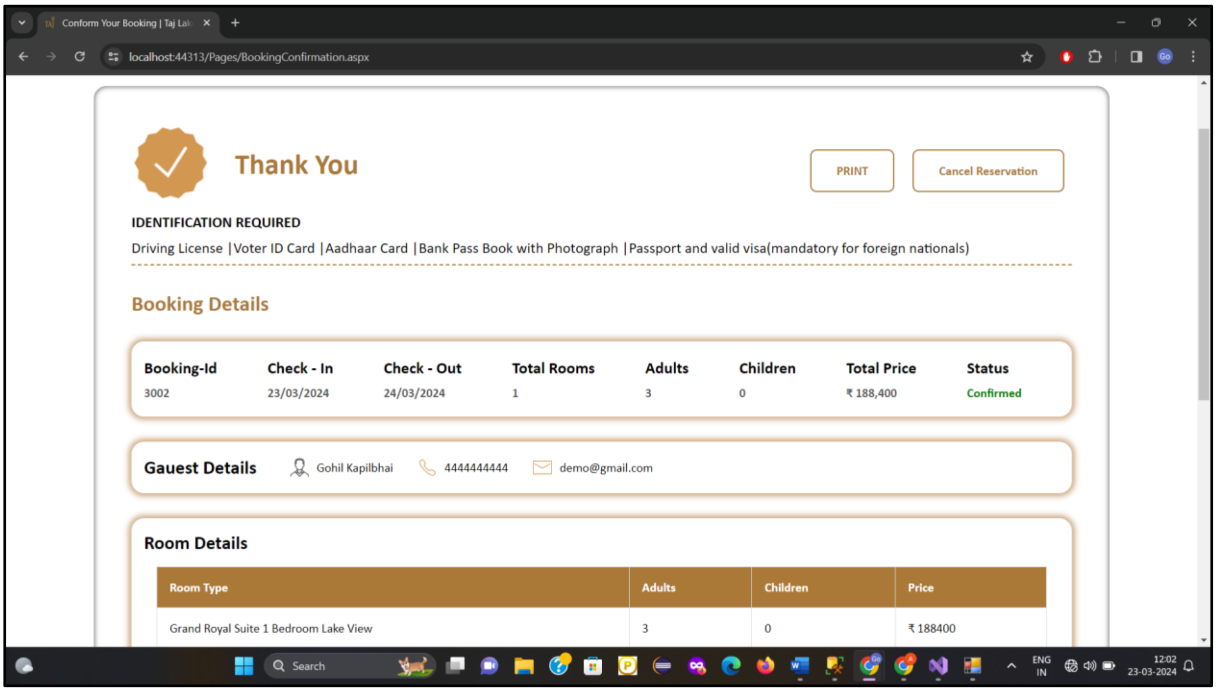
Country*: India(+91)

Phone Number*: 4444444444

Special Requests: No Special Request

Figure 8.8 Booking Form UI

8.9 Reservation


Conform Your Booking | Taj Lak
localhost:44313/Pages/BookingConfirmation.aspx
PRINT
Cancel Reservation
Search
ENG
IN
12:02
23-03-2024

Thank You

IDENTIFICATION REQUIRED

Driving License |Voter ID Card |Aadhaar Card |Bank Pass Book with Photograph |Passport and valid visa(mandatory for foreign nationals)

Booking Details

Booking-Id	Check - In	Check - Out	Total Rooms	Adults	Children	Total Price	Status
3002	23/03/2024	24/03/2024	1	3	0	₹ 188,400	Confirmed

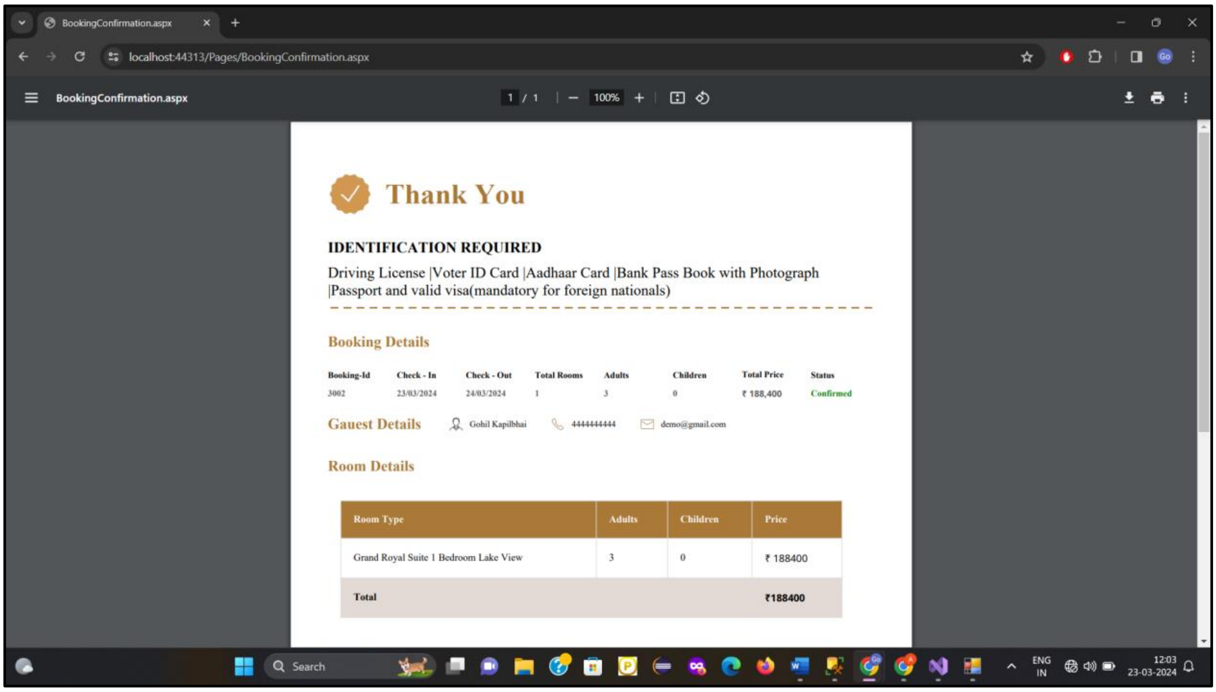
Gauest Details Gohil Kapilbhai 4444444444 demo@gmail.com

Room Details

Room Type	Adults	Children	Price
Grand Royal Suite 1 Bedroom Lake View	3	0	₹ 188400

Figure 8.9 Reservation UI

8.10 Bill Pdf



Thank You

IDENTIFICATION REQUIRED

Driving License |Voter ID Card |Aadhaar Card |Bank Pass Book with Photograph |Passport and valid visa(mandatory for foreign nationals)

Booking Details

Booking-Id	Check - In	Check - Out	Total Rooms	Adults	Children	Total Price	Status
3002	23/03/2024	24/03/2024	1	3	0	₹ 188,400	Confirmed

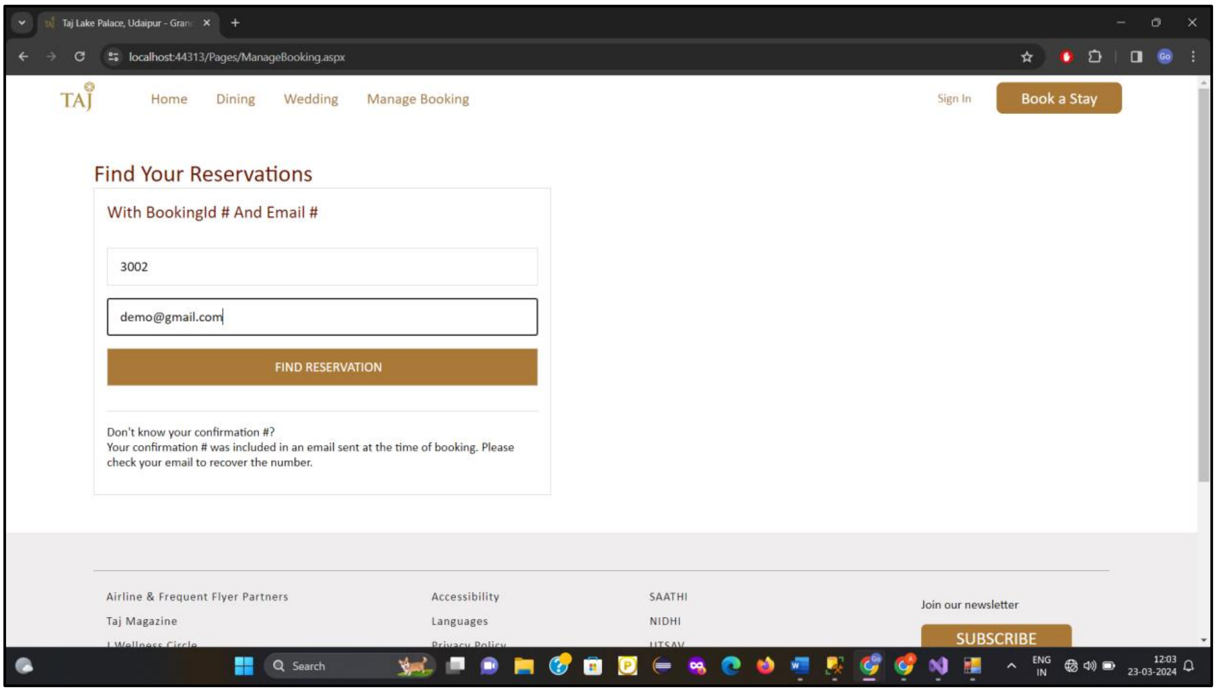
Gauest Details Gohil Kapilbhai 4444444444 demo@gmail.com

Room Details

Room Type	Adults	Children	Price
Grand Royal Suite 1 Bedroom Lake View	3	0	₹ 188400
Total			₹188400

Figure 8.10 Bill PDF

8.11 Find Booking


Taj Lake Palace, Udaipur - Gran
localhost:44313/Pages/ManageBooking.aspx
TAJ
Home
Dining
Wedding
Manage Booking
Sign In
Book a Stay

Find Your Reservations

With BookingId # And Email #

3002

demo@gmail.com

Don't know your confirmation #?
Your confirmation # was included in an email sent at the time of booking. Please check your email to recover the number.

Figure 8.11 Find Booking UI

9 Conclusion

After the end of the project we have successfully implemented the below functionalities

1. **Register and login**
   - User can Register and login in the system
2. **Search room availability**
   - User can search the room based on their criteria no of rooms, adults in one room, children in one room etc.
3. **Select multiple rooms**
   - User can book more than one rooms at a time of their own choice
4. **Book the room**
   - User can book the room by providing the necessary details like name, mobile no, email, any special request etc.
5. **Cancel the booking of the Room**
   - If user wants to cancel the booking then the user can cancel the booking by searching that reservation
6. **Search Booking by booking-id and e-mail**
   - User can search their booking by e-mail and booking id from there they can either print or cancel the reservation
7. **Print the bill for the reservation**
   - User can download the pdf copy of the bill once they have booked room successfully.

10 Limitation and Future Extension

10.1 Limitations

In our project there is some limitations which is listed below

1. We haven't implemented the real payment gateway
2. Once the user have booked the room they can't change the room and switch the booking for another room (they can do by cancelling the previous reservation and book again)
3. There is no charge for cancelling the booking so the ungenuine user can book rooms and cancel reservation any no of times.
4. This project is just for enhancing the skills on the .net webservices so it doesn't put much aspect on the security (data encryption),Real time data ,admin site for the room management ,data processing , various charts for financial and data analysis etc.
5. Lack of authentication while booking room i.e. user can book room just with email and phone no. (not verifying identity like Aadhar no or pan at the time of booking)

At end we made our project for enhancing our skills on the service-based architecture so it may fail in the real time scenarios where authentication, security, performance, scalability would be the most needy things.

10.2 Future Extension

Following extensions possible in our system.

1. Implement the real time payment gateway
2. Make booking functionality more flexible so user can customize or change the selection after booking.
3. Apply the appropriate charge upon cancelling the booking so the ungenuine user can't just book and occupy rooms for times and at end just cancel room. This charge should be based on the no of days remaining for the room check-in date.
4. Add data processing, various financial charts so it would be easy for the admin to analyze the data and make important decision accordingly.

There can be many future extensions to the system according to the need of the Hotel Owner.

11 Bibliography

11.1 Books or Some reading content

Most of the reference taken from the google classroom material provided by the university teachers.

- https://classroom.google.com/u/0/c/NTkwOTgwMjc5MzYx

11.2 Web

We also have followed the other resources from the web which are listed below

- https://learn.microsoft.com/en-us/troubleshoot/developer/visualstudio/csharp/language-compilers/write-web-service
- https://www.c-sharpcorner.com/UploadFile/govind77/introduction-to-service-oriented-architecture/