

CAN THO UNIVERSITY
COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY



**PROJECT – FUNDAMENTAL TOPICS
INFORMATION TECHNOLOGY
(HIGH-QUALITY PROGRAM)**

**BUILDING A WEBSITE FOR SEARCH AND HOTEL
BOOKING USING SPRINGBOOT & REACTJS**



Advisor:
TS. Lâm Nhựt Khang

Student:
Nguyễn Gia Linh
Student ID: B2111934
Class: 2021-2025 (Cohort K47)

Cần Thơ 11/2024

INSTRUCTOR'S COMMENTS



Cần Thơ, ngày tháng ... năm 2024
Advisor

Lâm Nhựt Khang

THANK YOU!

I would like to extend my heartfelt gratitude to Ms. Lam Nhut Khang, whose guidance and support greatly influenced my implementation process. Her valuable advice and dedication provided me with the encouragement I needed to overcome many challenges along the way.

Additionally, I would like to express my sincere thanks to the faculty of the School of Information and Communications Technology for imparting foundational knowledge that enabled me to complete my thesis. Their unwavering support, guidance, and commitment have been instrumental throughout my academic journey.

I am also deeply grateful to Can Tho University, which has been not only a place of learning but also a community for my personal and professional growth. I am proud to be a student of this institution and will carry the values and skills I have acquired here throughout my life and career.

I wish Ms. Khang continued health, happiness, and success in her teaching career.

Thank you sincerely,

Nguyễn Gia Linh

TABLE OF CONTENT

A. INTRODUCTION	11
1. Problem statement:	11
2. Research objectives:.....	11
3. Subjects and scope of research:.....	12
3.1. Research subjects	12
3.2. Research scope.....	12
4. Research Methodology:.....	13
5. Research content:.....	13
6. Layout:.....	13
B. CONTENT.....	15
CHAPTER 1. REQUIREMENT DESCRIPTION	15
1.1. Related theoretical basis:.....	15
1.1.1. ReactJS.....	15
1.1.2. Redux	15
1.1.3. Tailwind CSS	16
1.1.4. Database MySQL	16
1.1.5. Client – Server	16
1.1.6. SpringBoot framework.....	17
1.2. Description system:.....	17
1.3. Require function:	18
1.3.1. User management.....	18
1.3.2. Administrator Functions	18
1.3.3. Hotel Owner Functions.....	18
1.3.4. Customer Functions.....	18
CHAPTER 2. SOLUTION DESIGN	19
2.1. Overall architecture:.....	19

2.2. Conceptual data model (CDM).....	20
2.3. Flowchart diagram of the system:	21
2.4. System functions:	22
2.4.1. Functions for customers	22
2.4.2. Functions for hotel owners	23
2.4.3. Functions for administrators	24
2.5. Database.....	25
2.5.1. Hotels.....	25
2.5.2. Customers	25
2.5.3. Rooms.....	26
2.5.4. Reviews.....	26
2.5.5. Bookings.....	26
2.5.6. Accounts.....	27
CHAPTER 3. SOLUTION IMPLEMENTATION	28
3.1. Interface.....	28
3.1.1. Home	28
3.1.2. Hotel	30
3.1.3. Service	31
3.1.4. Review	32
3.1.5. Contact.....	33
3.1.6. Booking	34
3.1.7. Favorite	35
3.1.8. Profile	36
3.1.9. Log-in	37
3.1.10. Register	38
3.1.11. Admin Hotel	38
3.1.12. Admin Room.....	40
3.1.13. Admin Booking.....	42
3.1.14. Admin User.....	44
3.2. Design Description using operating system and hardware	46
3.2.1. Operating environment	46
3.2.2. Requires external communication	46
CHAPTER 4. TESTING AND EVALUATION	48

4.1. Testing objectives:	48
4.2. Testing scenarios:	48
4.2.1. Register function	48
4.2.2. Login function	49
4.2.3. Update profile function	49
4.2.4. Logout function	49
4.2.5. Create Admin User function	49
4.2.6. Create Admin Booking function	50
4.2.7. Create Admin Hotel & Room function	50
4.2.8. Create Add to Favorite function	50
4.2.9. Create Booking function	50
4.2.10. Create Payment function	51
4.2.11. Create Filter Hotel function	51
4.3. Testing results:	51
4.3.1. Register function	51
4.3.2. Login function	51
4.3.3. Update profile function	52
4.3.4. Logout function	52
4.3.5. Admin User function	52
4.3.6. Admin Booking function	52
4.3.7. Admin Hotel & Room function	52
4.3.8. Add to Favorite function	53
4.3.9. Booking function	53
4.3.10. Payment function	53
4.3.11. Filter Hotel function	53
C. CONCLUSION	54
1. Achieved results :	54
2. Future directions:	54
TÀI LIỆU THAM KHẢO	55
PHỤ LỤC	56

Image category

Figure 1: Over Architecture.....	19
Figure 2 CDM.....	20
Figure 3 Flowchart	21
Figure 4: Customer function	22
Figure 5: Hotel Owner function	23
Figure 6: Admin funciton	24
Figure 7: view Home / header	28
Figure 8: view Home / Welcome.....	28
Figure 9: view Home / Discount.....	29
Figure 10: view Home / Popular Hotel.....	29
Figure 11: view Home / Footer	29
Figure 12: view Hotel	30
Figure 13: view Hotel / Discount.....	30
Figure 14: view Hotel / Filter & List Hotel.....	30
Figure 15: view Hotel / Footer	31
Figure 16: view Service / Header	31
Figure 17: view Service	31
Figure 18: view Service / Footer	32
Figure 19: view Review / Header	32
Figure 20: view Review	32
Figure 21: view Review / Footer	33
Figure 22: view Contact / Header	33
Figure 23: view Contact	33
Figure 24: view Contact / Footer	34
Figure 25: view Booking / Header.....	34
Figure 26: view Booking	34
Figure 27: view Booking	35
Figure 28: view Booking / Footer.....	35
Figure 29: view Favorite / Header.....	35
Figure 30: view Favorite.....	36
Figure 31: view Favorite / Footer.....	36
Figure 32: view Profile / Header	36
Figure 33: view Profile	37
Figure 34: view Profile / edit Profile	37
Figure 35: view Profile / Footer	37
Figure 36: view LogIn	38
Figure 37: view Register.....	38
Figure 38: view Admin DashBoard.....	39
Figure 39: view Admin Hotel.....	39
Figure 40: view Admin Hotel / add Hotel.....	40
Figure 41: view Admin Hotel / edit Hotel.....	40

Figure 42: view Admin Hotel / delete Hotel	40
Figure 43: view Admin Room	41
Figure 44: view Admin Room / add Room	41
Figure 45: view Admin Room / edit Room.....	42
Figure 46: view Admin Room / delete Room	42
Figure 47: view Admin Booking	43
Figure 48: view Admin Booking / add Booking	43
Figure 49: view Admin Booking / edit Booking.....	44
Figure 50: view Admin Booking / delete Booking.....	44
Figure 51: view Admin User	44
Figure 52: view Admin User / add User	45
Figure 53: view Admin User / edit User	45
Figure 54: view Admin User / delete User	46

Table category

Table 1 Database hotels.....	25
Table 2 Database customers.....	25
Table 3 Database rooms	26
Table 4 Database reviews	26
Table 5 Database bookings	26
Table 6 Database accounts	27
Table 7 Register funcion	48
Table 8: Login function.....	49
Table 9: Update profile function.....	49
Table 10: Logout function	49
Table 11 Create hotel function	49
Table 12: Edit hotel function.....	50
Table 13 : Delete hotel function	50
Table 14 : Create Add to Favorite	50
Table 15: Create Booking	50
Table 16: Payment	51
Table 17: Filter	51
Table 18: Testing register function	51
Table 19: Testing login function.....	51
Table 20: Testing update profile function.....	52
Table 21: Testing logout function	52
Table 22: Testing create hotel function	52
Table 23: Testing edit hotel function	52
Table 24: Testing delete hotel function	52
Table 25: Add to Favorite.....	53
Table 26: Booking.....	53
Table 27: Payment	53
Table 28: Filter Hotel.....	53

ABSTRACT

In the era of Technology 4.0, the Internet has become an essential tool for managing and sharing information, whether on a national level or within local communities. As Internet usage expands, tourists' expectations continue to rise. Many travelers face challenges in finding suitable hotels before they arrive or experience dissatisfaction when accommodations fail to meet expectations, leading to frustration and wasted time.

To address these needs, our project, titled "**Building a Hotel Search and Booking Website Using Spring Boot & ReactJS**" aims to provide a convenient solution. This platform enables tourists to find and reserve accommodations in advance, offering a seamless experience. With just a few steps on their device, users can easily locate a comfortable place to stay that perfectly matches their preferences, ensuring a more satisfying travel experience.

A. INTRODUCTION

1. Problem statement:

- Vietnam boasts a thriving tourism industry, with rising numbers of both domestic and international tourists. This growth drives an increasing demand for efficient hotel search and booking solutions.
- While numerous hotel search and booking websites exist, many fall short of fully meeting user needs. Some platforms lack user-friendly interfaces, are challenging to navigate, or provide insufficient hotel information.
- In response to these issues, the "Hotel Search and Booking Website" project was developed to offer users a fresh perspective on local accommodations. This platform not only aims to make finding and booking hotels easier but also brings a unique, vibrant experience to users' lives. Additionally, the website serves as a tool to promote local culture and attractions to a broader audience.

2. Research objectives:

- The Hotel Search and Booking Website using Spring Boot is designed to comprehensively meet the needs of both users and hotel providers. Key features include:
 - User-Friendly Interface: The website will offer an attractive, intuitive interface that is easy to navigate, enabling users to effortlessly search for and book hotel rooms.
 - Comprehensive Hotel Information: Detailed information on each hotel, including addresses, photos, prices, and available amenities, will be provided to ensure users can make informed choices.
 - Simple Booking Process: A streamlined, straightforward booking process will allow users to complete reservations easily and efficiently.
- This website aims to enhance the user experience while addressing the needs of the hospitality industry.

3. Subjects and scope of research:

To address the issues identified, this project will focus on researching the following topics:

- **Theoretical Research:**

- ReactJS: Understand how to use ReactJS for building the system's user interface
- Supporting Libraries and APIs for ReactJS: Explore libraries and APIs that integrate with ReactJS to enhance functionality.
- Spring Boot Framework: Study the Spring Boot framework for back-end development and integration.

- **Technical Research:**

- Database Design: Develop a database schema for effective data storage and retrieval.
- Framework and Library Utilization: Utilize frameworks and libraries to streamline development.

3.1. Research subjects

- **Users:** Focus on website users who search for and book hotels, including groups such as tourists and business travelers,...
- **Hotel:** Investigate hotels that register on the website to enable room searching and booking.

3.2. Research scope

- **User Needs Analysis:** Investigate user requirements for a hotel search and booking website, considering interface design, features, pricing, and more
- **Competitive Analysis:** Study existing hotel search and booking websites to identify their strengths and weaknesses and apply best practices.

4. Research Methodology:

- **Literature Review:** Conduct research through existing documents and literature.
- **Field Survey:** Perform surveys with real users and stakeholders.
- **Data Analysis:** Analyze collected data for insights and trends.

5. Research content:

- **User Needs Analysis:** Analyze user requirements by examining existing websites. Follow a structured process of design, development, testing, deployment, and maintenance for the website.

- **Technology Stack:**

- Back-end: Spring Boot Framework
- Front-end: ReactJS
- Database: MySQL

- **Development Tools:**

- Tool design data: MySQL Workbench 8.0 CE
- Tool installer: VSCode
- Implementation support tools: Web Browser

6. Layout:

The contents of the thesis include:

- ❖ A. Introduction:
 - Problem statement
 - Research objectives
 - Object and scope of the study

- Research Methods
- Research content
- Layout

❖ B. Content:

- Chapter 1: REQUIREMENTS DESCRIPTION
- Chapter 2: SOLUTION DESIGN
- Chapter 3: SOLUTION IMPLEMENTATION
- Chapter 4: TESTING AND EVALUATION

❖ C. Conclusion:

- Result
- Develop

B. CONTENT

CHAPTER 1. REQUIREMENT DESCRIPTION

1.1. Related theoretical basis:

1.1.1. ReactJS

- ReactJS is an open source JavaScript library, developed by Facebook, used to build user interfaces (UI) for web applications. React's main goal is to help develop fast, efficient, and scalable web applications, with a special focus on efficiently re-rendering UI elements when data changes.
- React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.
- Declarative views make your code more predictable and easier to debug.
- Build encapsulated components that manage their own state, then compose them to make complex UIs.
- React has a rich ecosystem with many supporting libraries and tools, such as React Router (for handling navigation), Redux (managing application state), and many others for integrating with the backend.

1.1.2. Redux

- React Redux is maintained by the Redux team, and kept up-to-date with the latest APIs from Redux and React
- Designed to work with React's component model. You define how to extract the values your component needs from Redux, and your component updates automatically as needed.
- Provides APIs that enable your components to interact with the Redux store, so you don't have to write that logic yourself.
- Automatically implements complex performance optimizations, so that your own component only re-renders when the data it needs has actually changed.

1.1.3.

Tailwind CSS

- Tailwind CSS is a utility-first CSS framework, designed to help developers build web interfaces quickly and easily. Instead of providing ready-made interface component classes like many other CSS frameworks (Bootstrap, Foundation), Tailwind focuses on small utility classes that can be flexibly combined to create any design.
- Tailwind CSS works by scanning all of your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file. It's fast, flexible, and reliable
- With its widget-based approach, it reduces the need to write manual CSS and allows creating complex interfaces quickly and efficiently

1.1.4. Database MySQL

- MySQL Workbench is an integrated graphical tool (GUI - Graphical User Interface) developed by Oracle, providing MySQL database development and management features. It is mainly used to design, develop, manage and operate MySQL databases more easily, without having to work through the command line. MySQL Workbench is one of the most popular tools for developers and database administrators.
- It is suitable for storing and managing structured data, such as customer information, product data, and business-related data

1.1.5.

Client – Server

- The Client-Server model is a computer network architecture in which computers (called clients) send requests to a central computer (called server - server) to receive a specific response or service. This model is popular in network systems, web applications, database services, and many other applications.
- The client sends a request to the server via the network. The server receives the request, processes it (performs necessary actions such as searching for

data, processing the request or performing a service) then the server sends a response back to the client with the results. handle.

1.1.6. **SpringBoot framework**

- Spring Boot is a Java-based framework, built on the Spring Framework foundation. It makes building web applications and microservices faster and easier.
- Spring Boot provides a number of features such as automatic configuration, dependency management, and integrated development tools, which help reduce the amount of configuration required when building applications.
- It is commonly used for the backend of an application, which handles business logic, database management, and API services.

1.2. Description system:

- The "Hotel Search and Booking Website using Spring Boot" is developed using the MVC (Model-View-Controller) architecture
- **Front-end:** This component, built with ReactJS, is responsible for displaying the user interface and managing user interactions.
- **Back-end:** This component, built with Java Spring Boot, handles business logic and database access.
- System Workflow:
 - Users visit the website to search and book hotel rooms, entering search details such as location, check-in and check-out dates, number of guests, etc.
 - The back-end receives the search criteria, queries the database, and retrieves suitable accommodation options.
 - The front-end displays the relevant search results to the user.
 - The user selects their preferred hotel and proceeds to make a reservation

1.3. Require function:

1.3.1. User management

- The system categorizes users into three main groups:
 - Customers
 - Hotel Owners
 - Administrators

1.3.2. Administrator Functions

- Dedicated Account: Administrators have a unique account for system management
- User Account Management: Administrators can view, edit, and delete user accounts

1.3.3. Hotel Owner Functions

- Log in, register an account
- Update hotel information: check-in and check-out time, address, updated room availability

1.3.4. Customer Functions

- Account Access: Customers can register and log in to their accounts.
- Personal Information Management: Manage personal information and upload profile pictures.
- Hotel Browsing: View detailed information on hotels.
- Get Directions: Find directions to the selected hotel.
- Hotel Reviews: Submit reviews and ratings after their stay experience.

CHAPTER 2. SOLUTION DESIGN

2.1. Overall architecture:

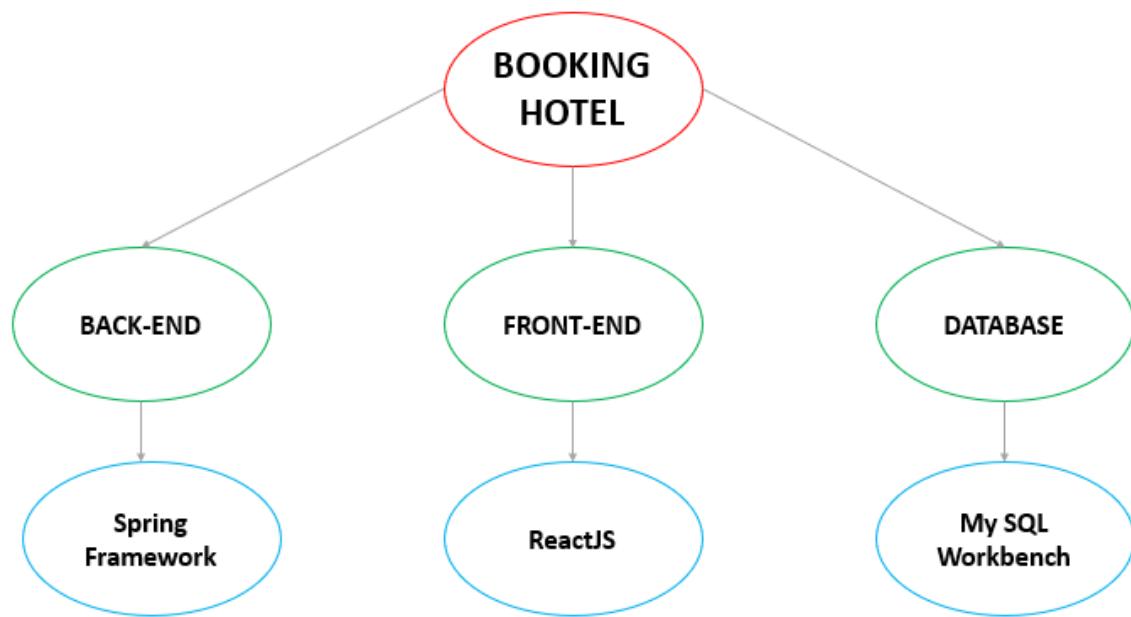


Figure 1: Over Architecture

2.2. Conceptual data model (CDM)

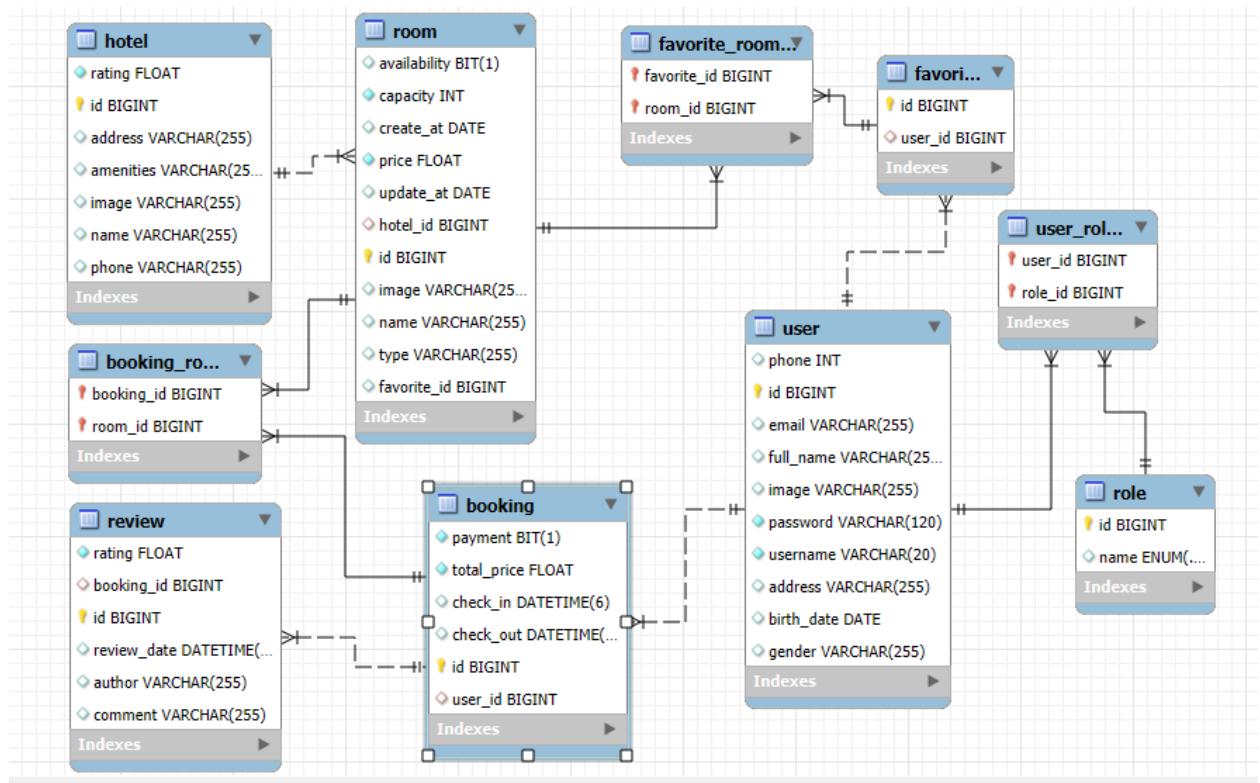


Figure 2 CDM

2.3. Flowchart diagram of the system:

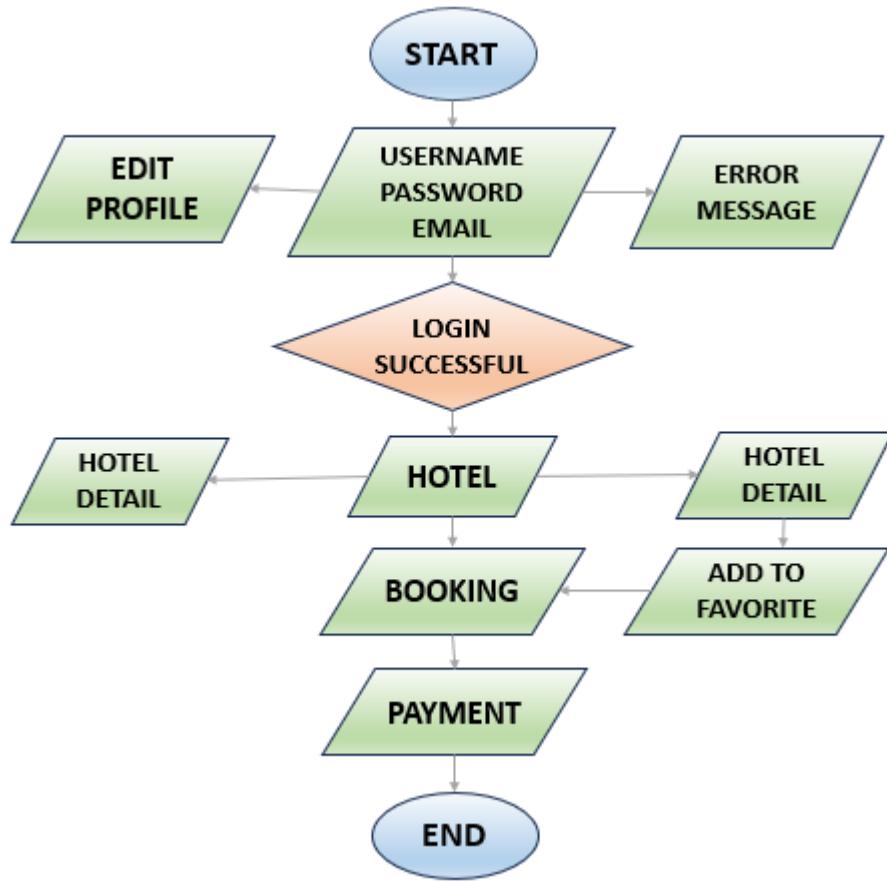


Figure 3 Flowchart

2.4. System functions:

2.4.1.

Functions for customers

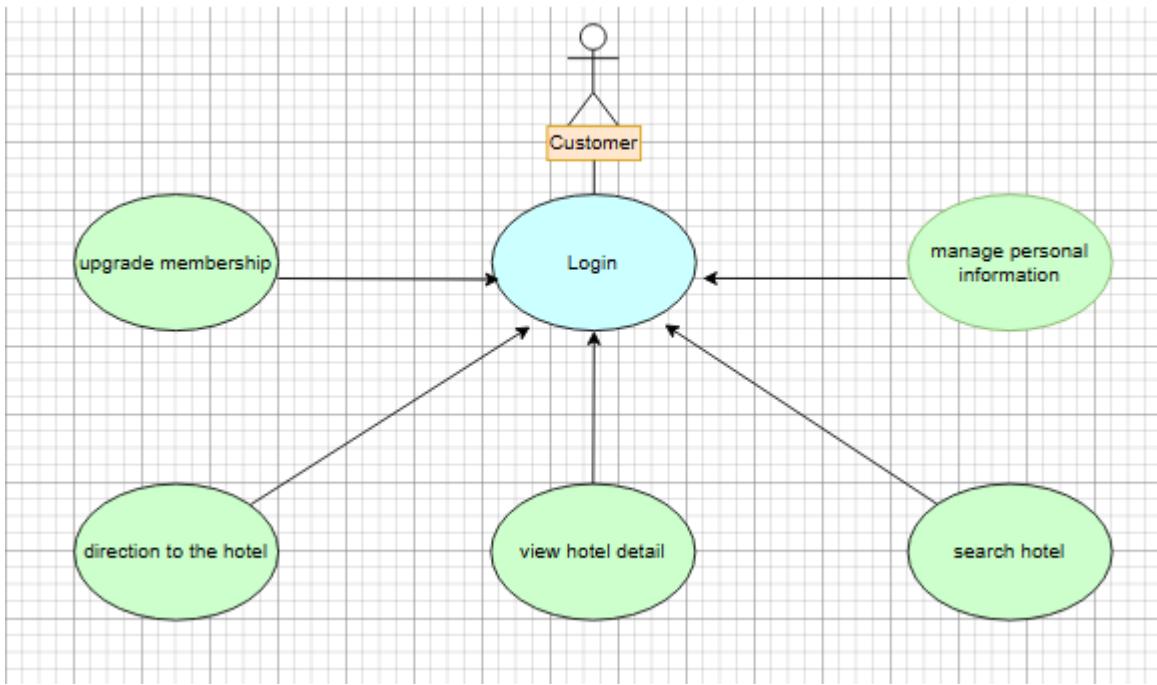


Figure 4: Customer function

- Register and Login an account.
- Manage personal information, upload avatar.
- View details about the hotel
- Search for hotels
- Directions to the hotel
- Rate the hotel after experiencing the stay

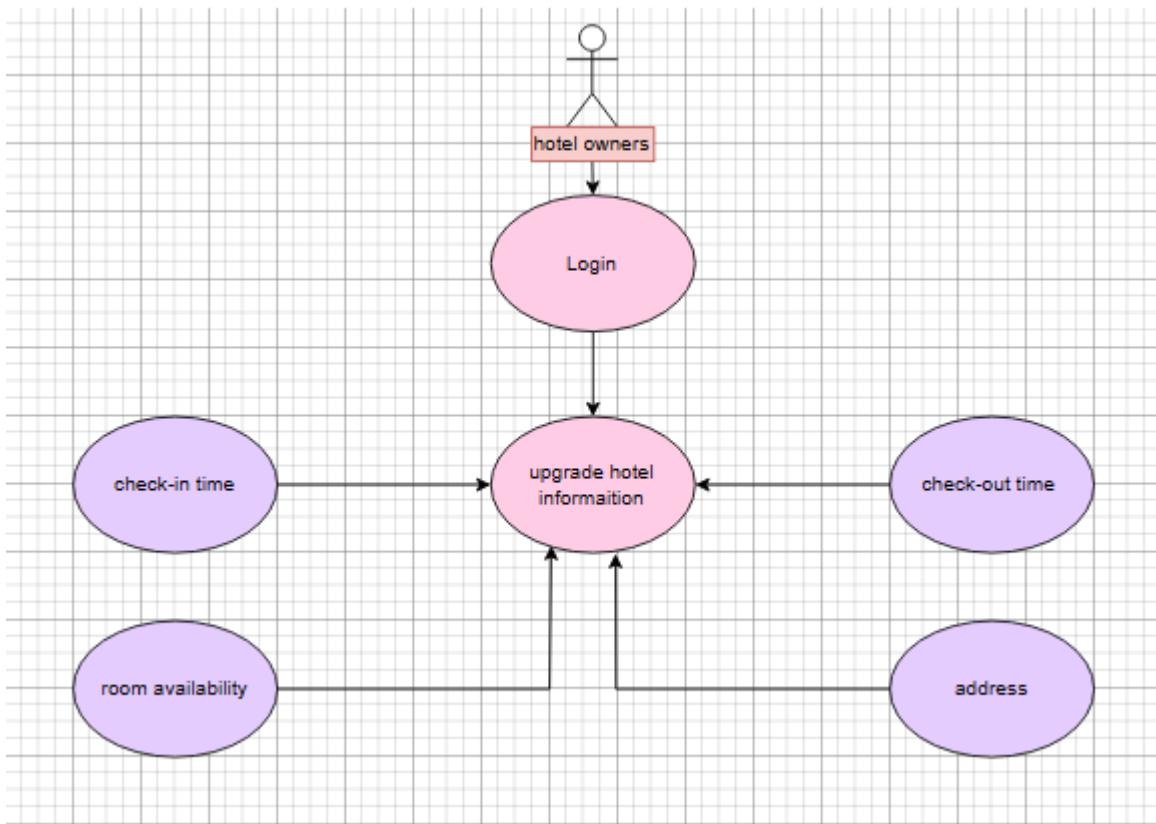
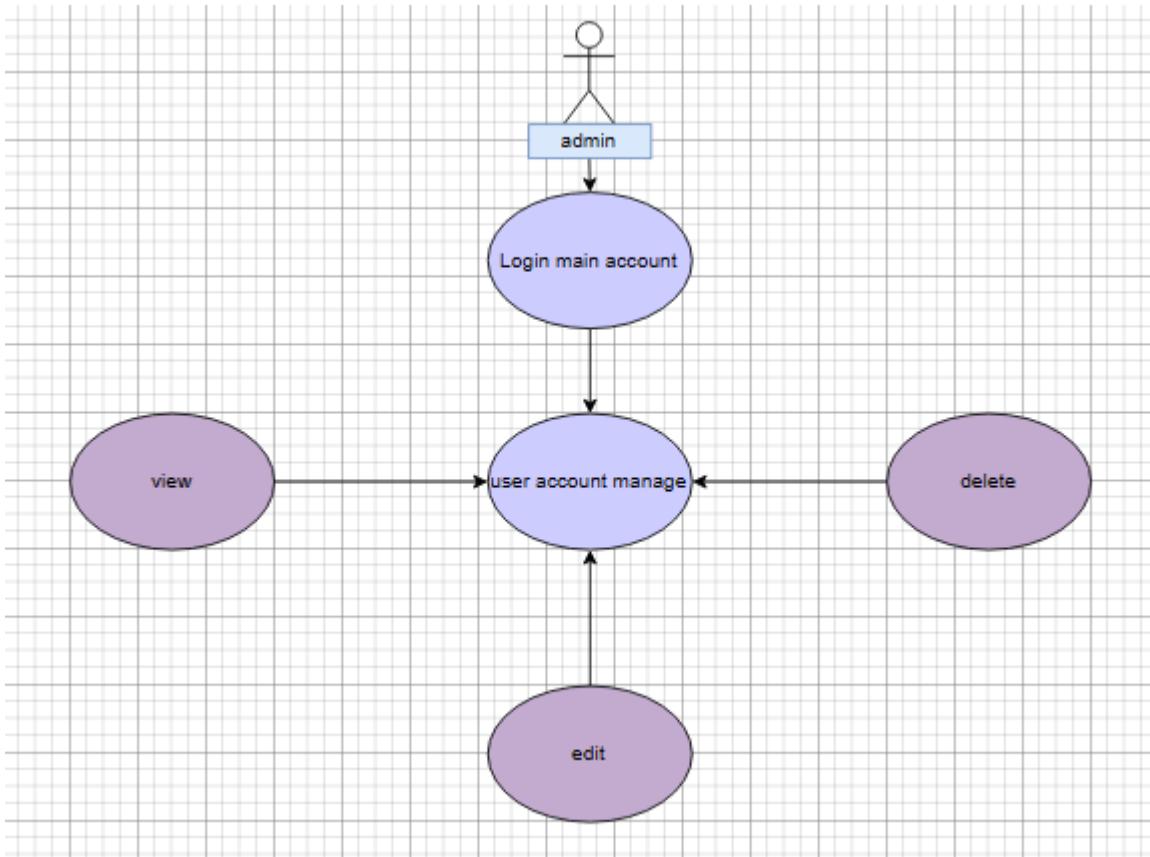
2.4.2.**Functions for hotel owners**

Figure 5: Hotel Owner function

Register and login an account

Update hotel information: check-in and check-out times, address, update room availability

2.4.3.**Functions for administrators***Figure 6: Admin funciton*

Be given a separate account

Manage user accounts: view, edit, delete

2.5. Database

2.5.1.

Hotels

Properties	Datatypes	Primary Key	Not null
Hotel_id	INT	X	X
Hotel_name	VARCHAR(255)		X
Address	VARCHAR(255)		X
Rating	FLOAT		X
Number_rooms	INT		X
Amenities	VARCHAR(255)		X
Hotel_image	VARCHAR(255)		X

Table 1 Database hotels

2.5.2.

Customers

Properties	Datatypes	Primary Key	Not null
Customer_id	INT	X	X
First_name	VARCHAR(255)		X
Last_name	VARCHAR(255)		X
Email	VARCHAR(255)		X
Phone	INT		X
Address	VARCHAR(255)		X
Customer_image	VARCHAR(255)		X
Account_id	INT		X

Table 2 Database customers

2.5.3.**Rooms**

Properties	Datatypes	Primary Key	Not null
Room_id	INT	X	X
Hotel_id	INT		X
Room_type	VARCHAR(255)		X
Price_per_night	INT		X
Availability	BOOLEAN		X
Room_image	VARCHAR(255)		X

*Table 3 Database rooms***2.5.4.****Reviews**

Properties	Datatypes	Primary Key	Not null
Review_id	INT	X	X
Customer_id	INT		X
Rating	FLOAT		X
Comment	VARCHAR(255)		X
Date	DATE		X

*Table 4 Database reviews***2.5.5.****Bookings**

Properties	Datatypes	Primary Key	Not null
Booking_id	INT	X	X
Customer_id	INT		X
Room_id	INT		X
Check-in-date	DATE		X
Check-out-date	DATE		X
Total_price	INT		X
Pay	BOOLEAN		X

Table 5 Database bookings

2.5.6.**Accounts**

Properties	Datatypes	Primary Key	Not null
Account_id	INT	X	X
Role	ENUM		X
Username	VARCHAR(255)		X
Password	VARCHAR(255)		X
Rank_account	ENUM		X

Table 6 Database accounts

CHAPTER 3. SOLUTION IMPLEMENTATION

3.1. Interface

3.1.1.

Home



Figure 7: view Home / header

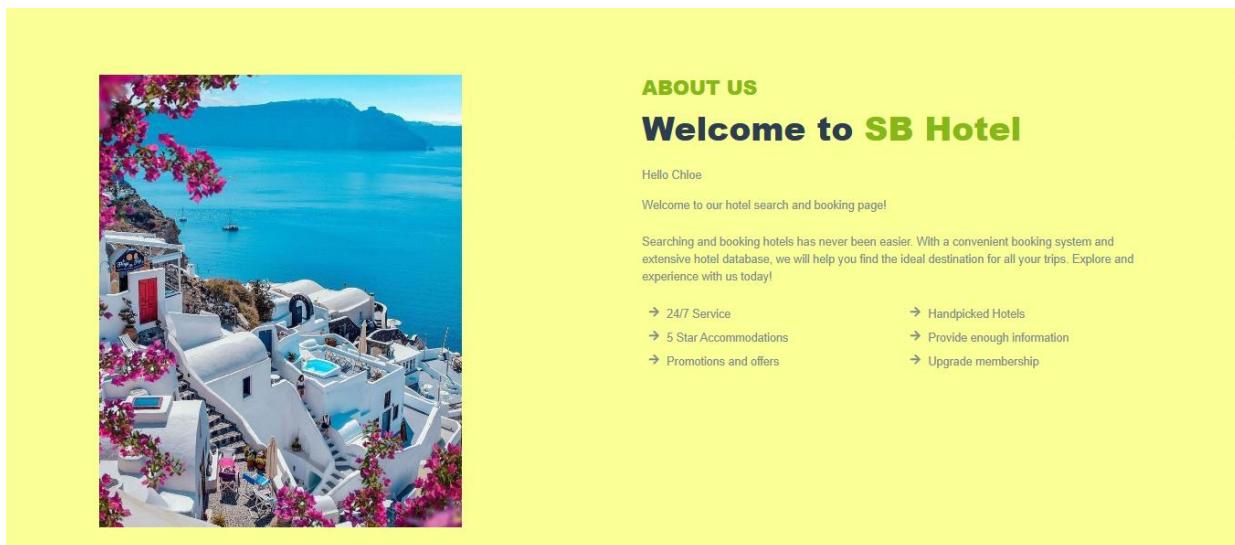


Figure 8: view Home / Welcome

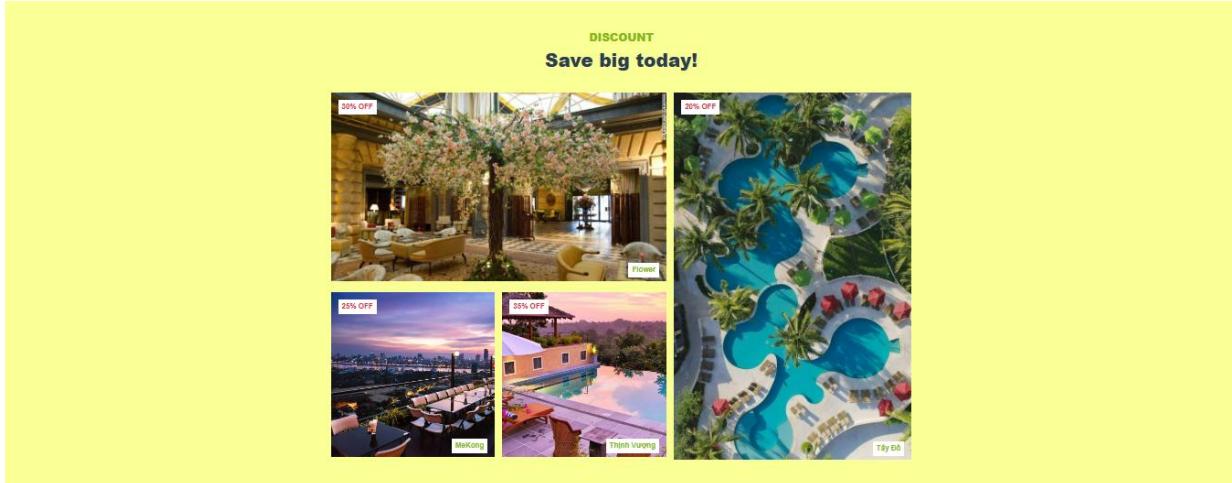


Figure 9: view Home / Discount

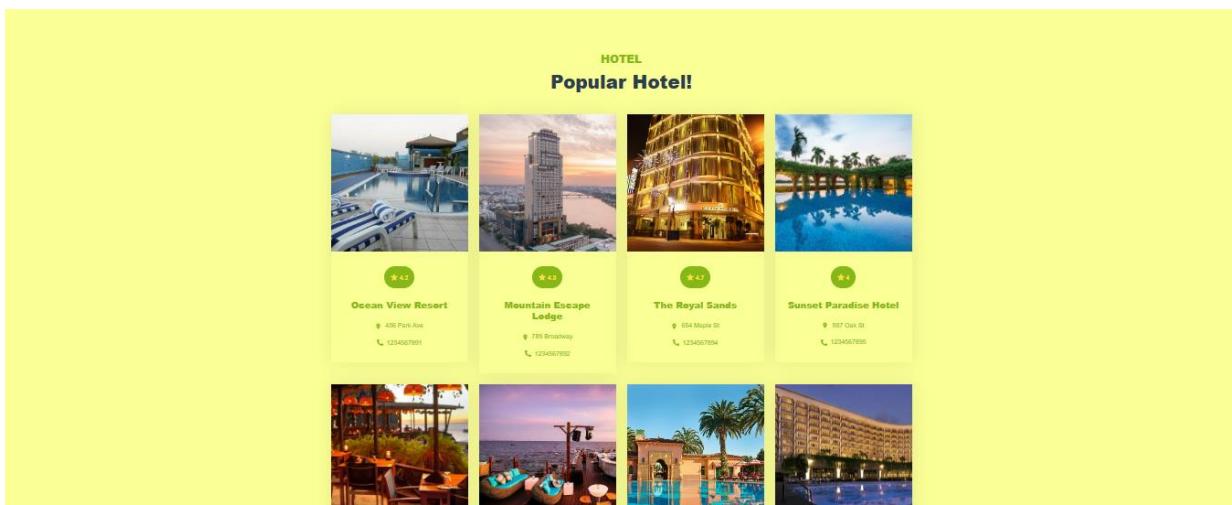


Figure 10: view Home / Popular Hotel

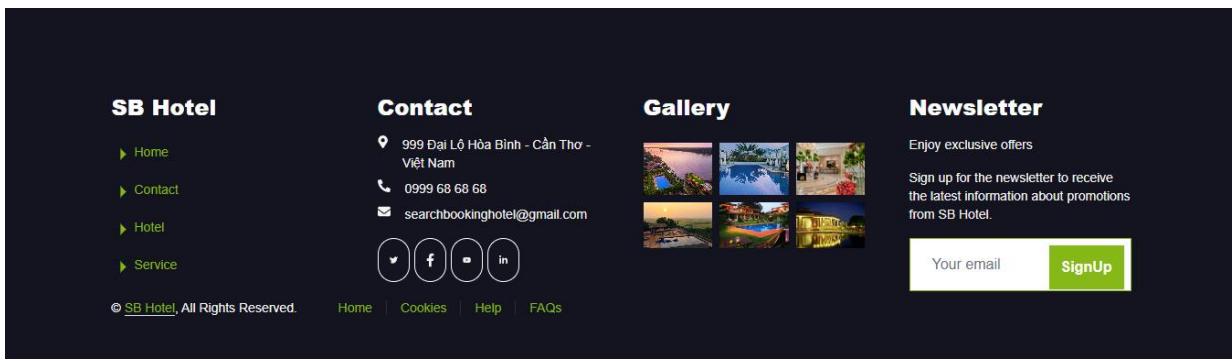


Figure 11: view Home / Footer

3.1.2.

Hotel



Figure 12: view Hotel

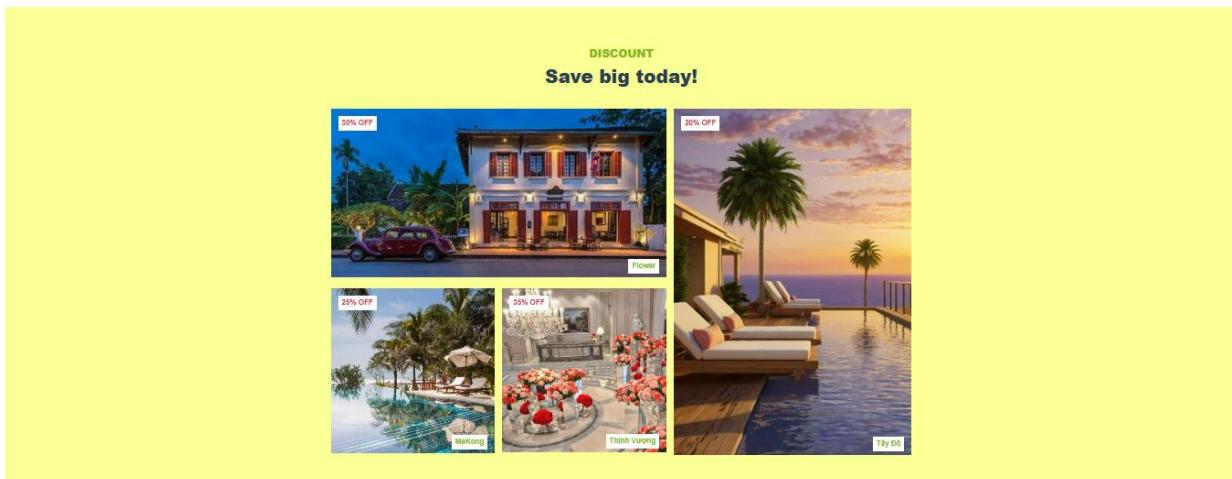


Figure 13: view Hotel / Discount

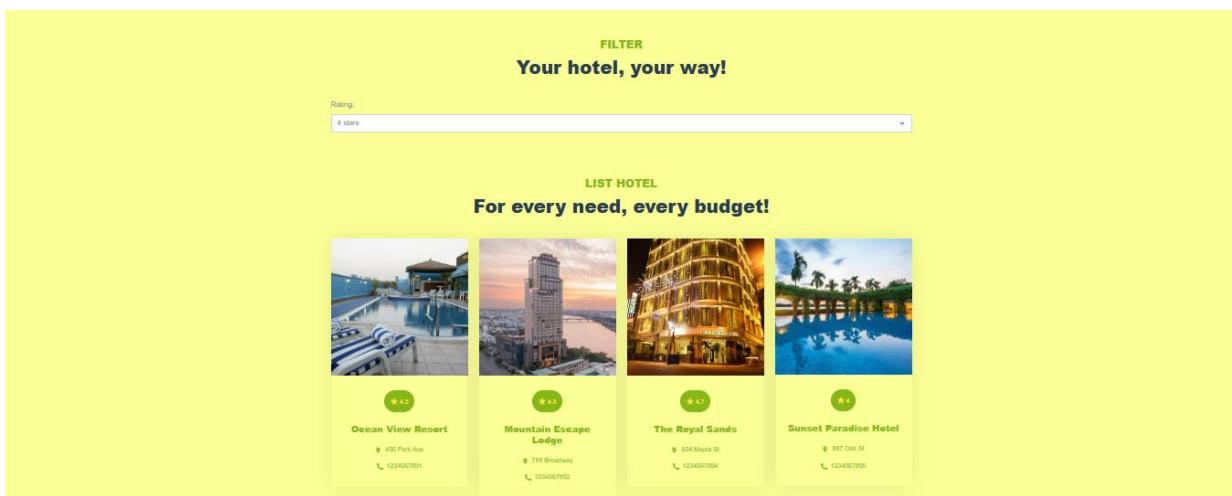


Figure 14: view Hotel / Filter & List Hotel

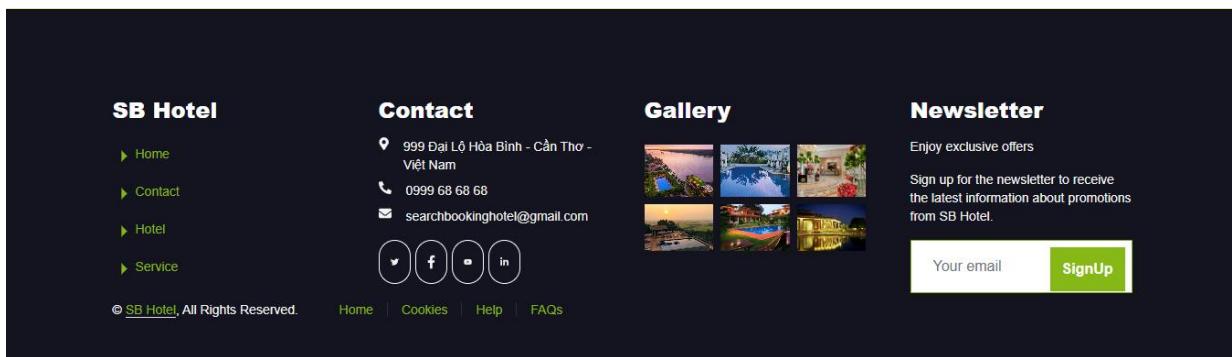


Figure 15: view Hotel / Footer

3.1.3.

Service

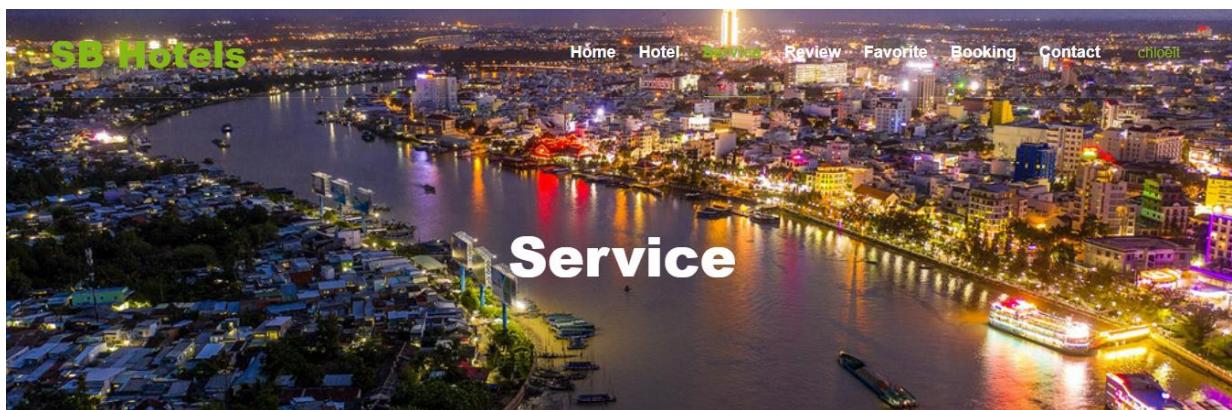


Figure 16: view Service / Header

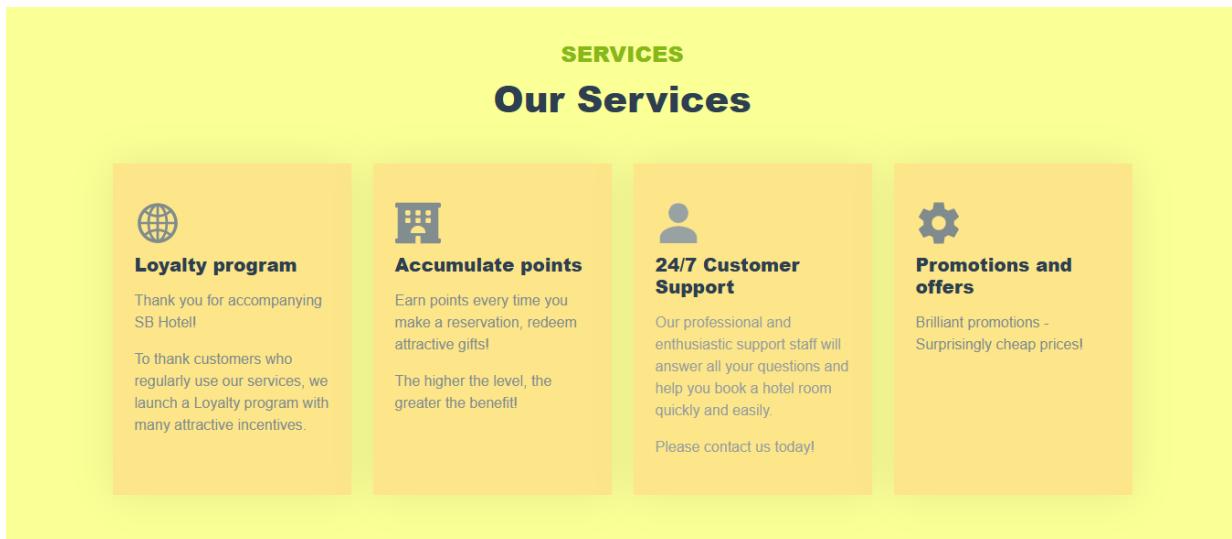


Figure 17: view Service

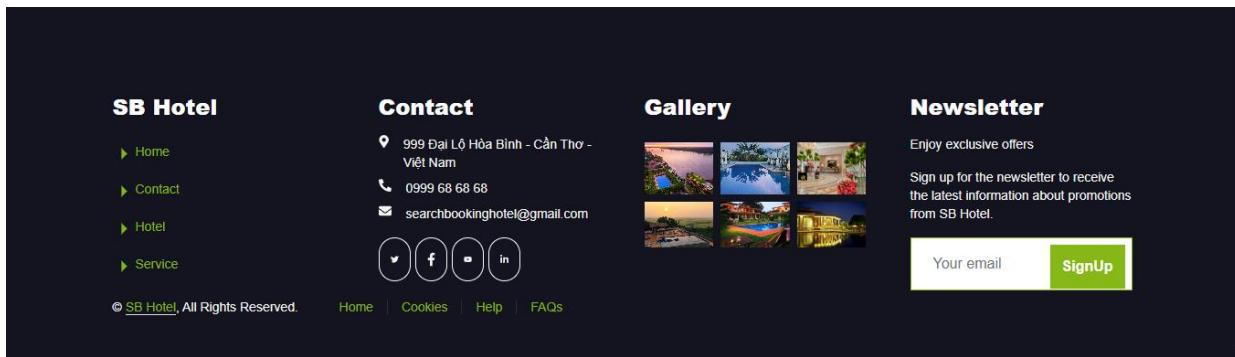


Figure 18: view Service / Footer

3.1.4. Review

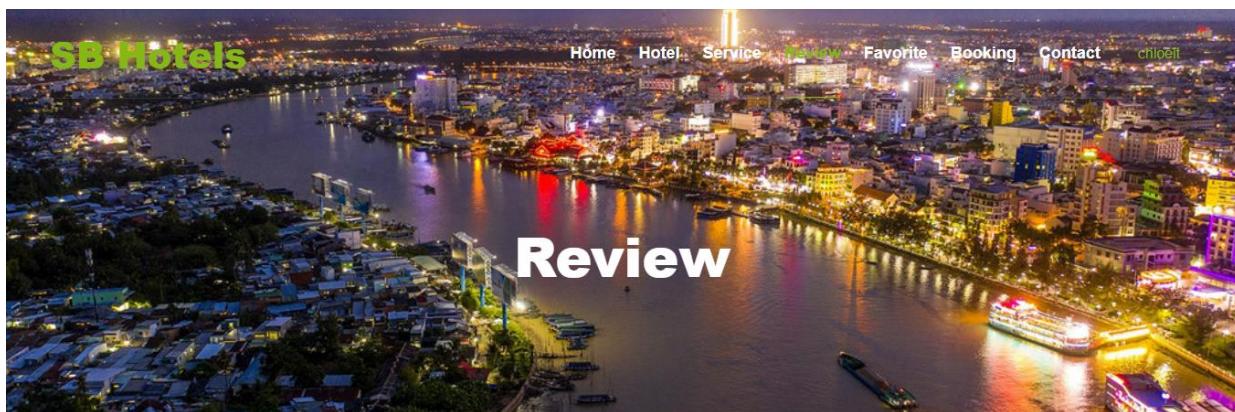


Figure 19: view Review / Header



Figure 20: view Review

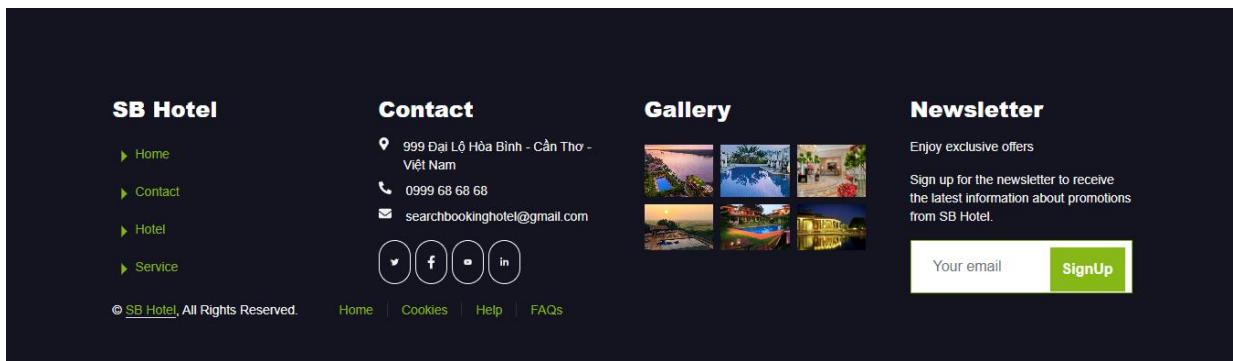


Figure 21: view Review / Footer

3.1.5.

Contact



Figure 22: view Contact / Header

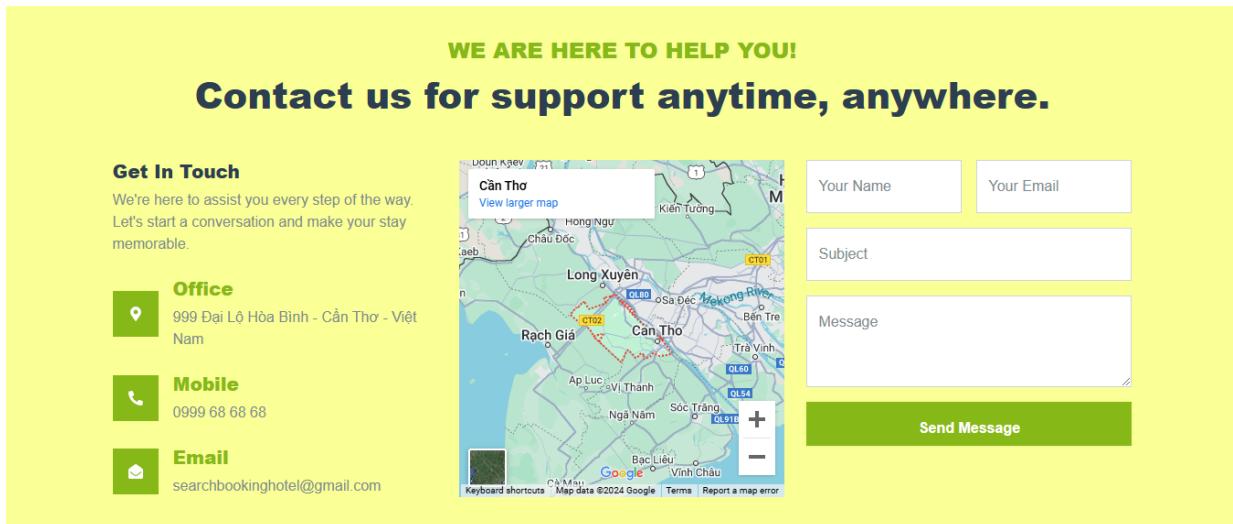


Figure 23: view Contact

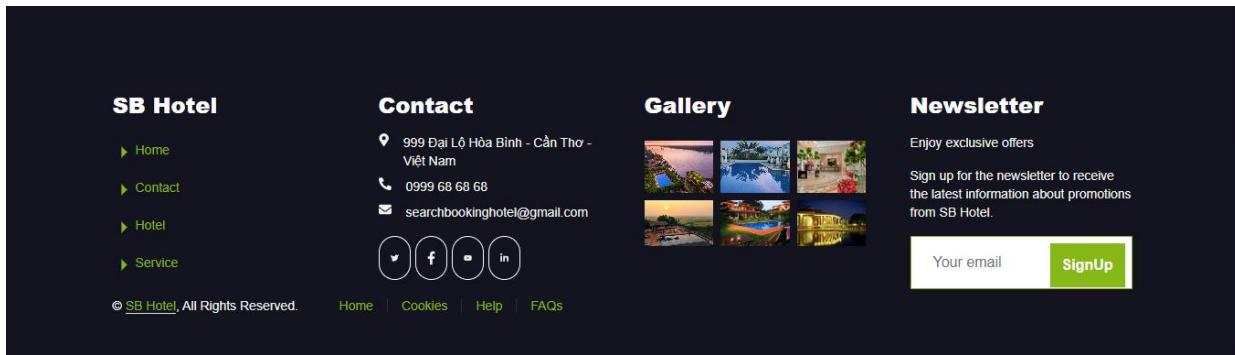


Figure 24: view Contact / Footer

3.1.6.

Booking

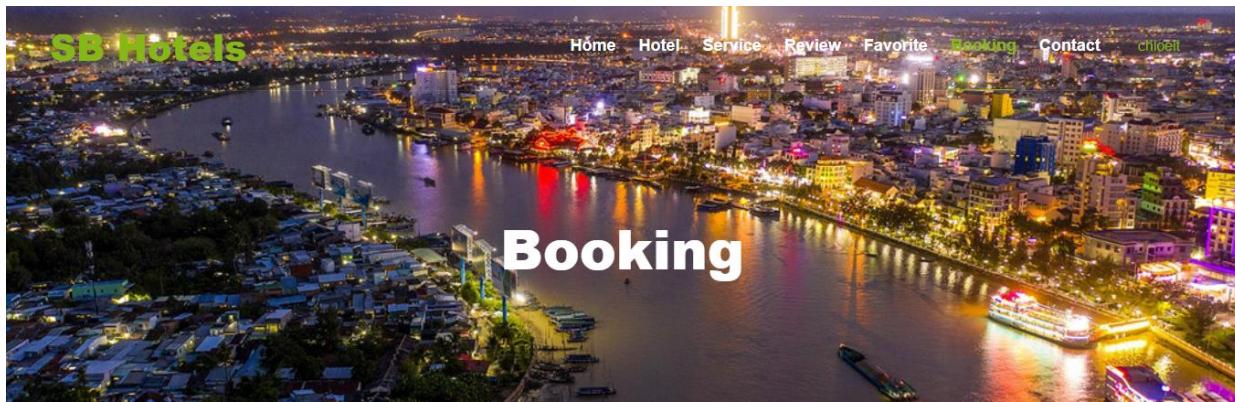


Figure 25: view Booking / Header

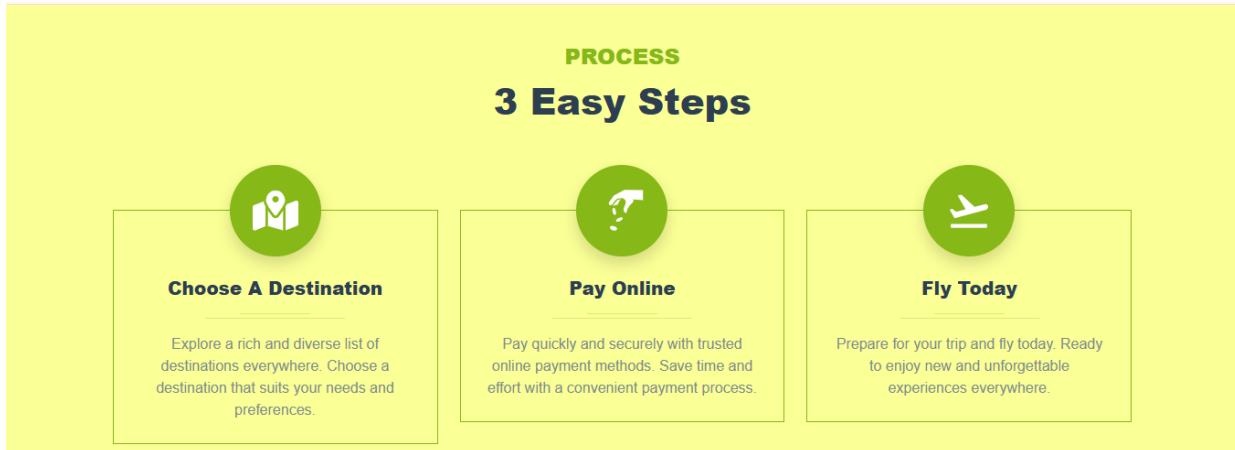


Figure 26: view Booking

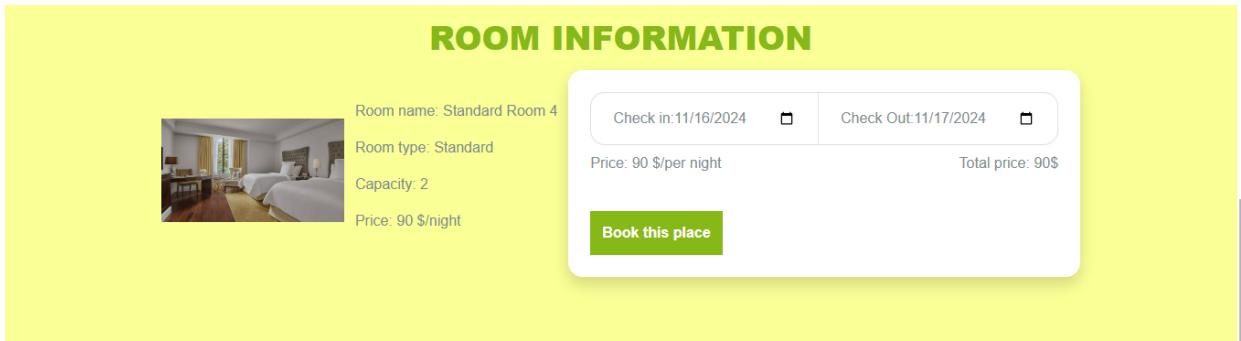


Figure 27: view Booking

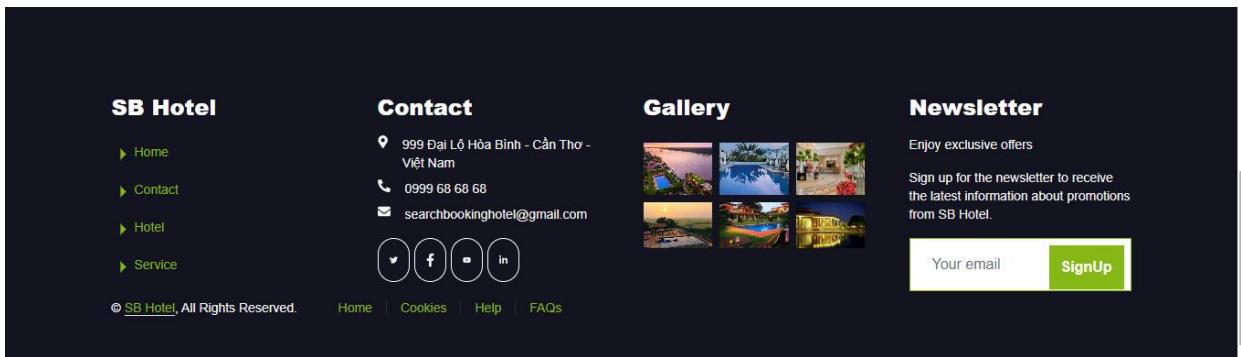


Figure 28: view Booking / Footer

3.1.7. Favorite



Figure 29: view Favorite / Header

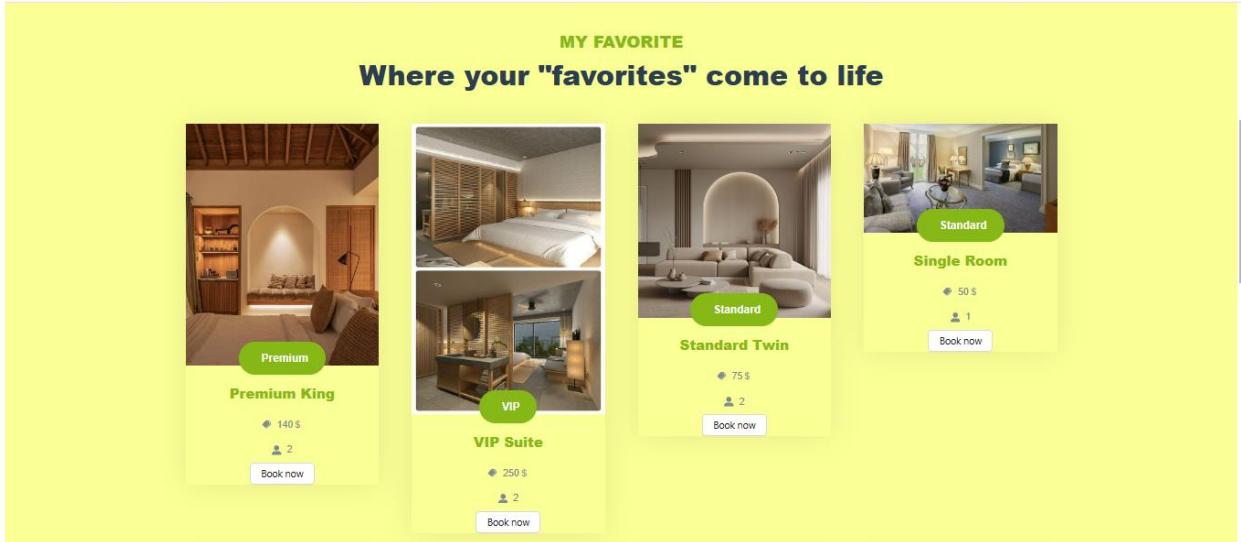


Figure 30: view Favorite

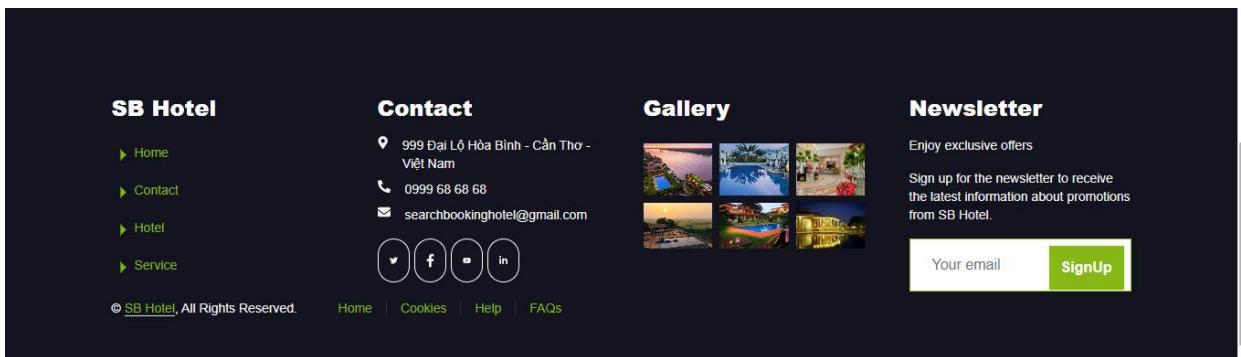


Figure 31: view Favorite / Footer

3.1.8.

Profile



Figure 32: view Profile / Header

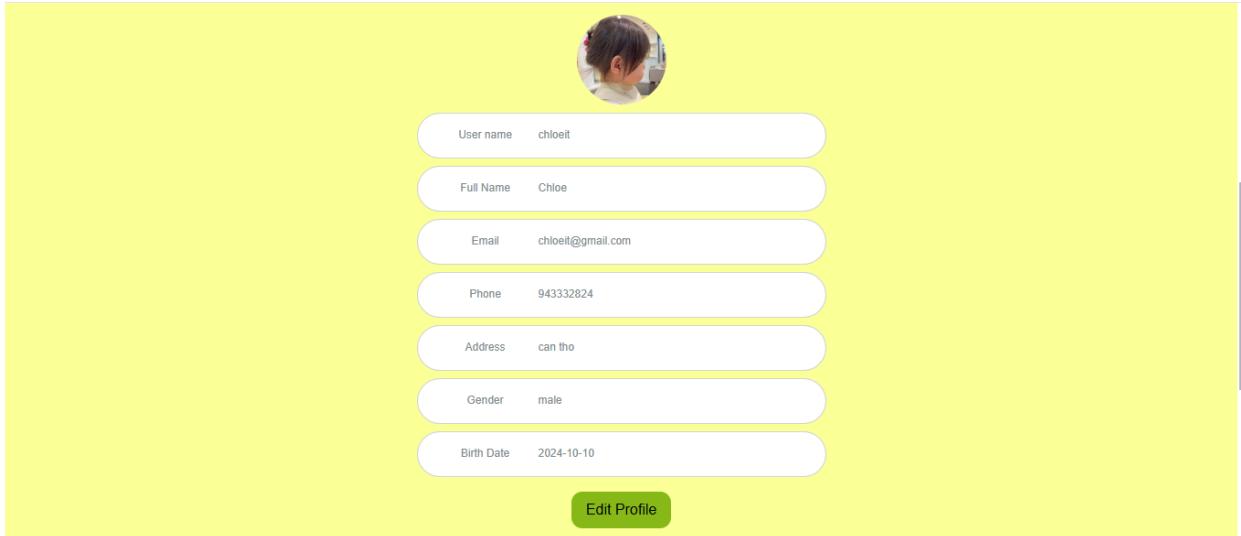


Figure 33: view Profile

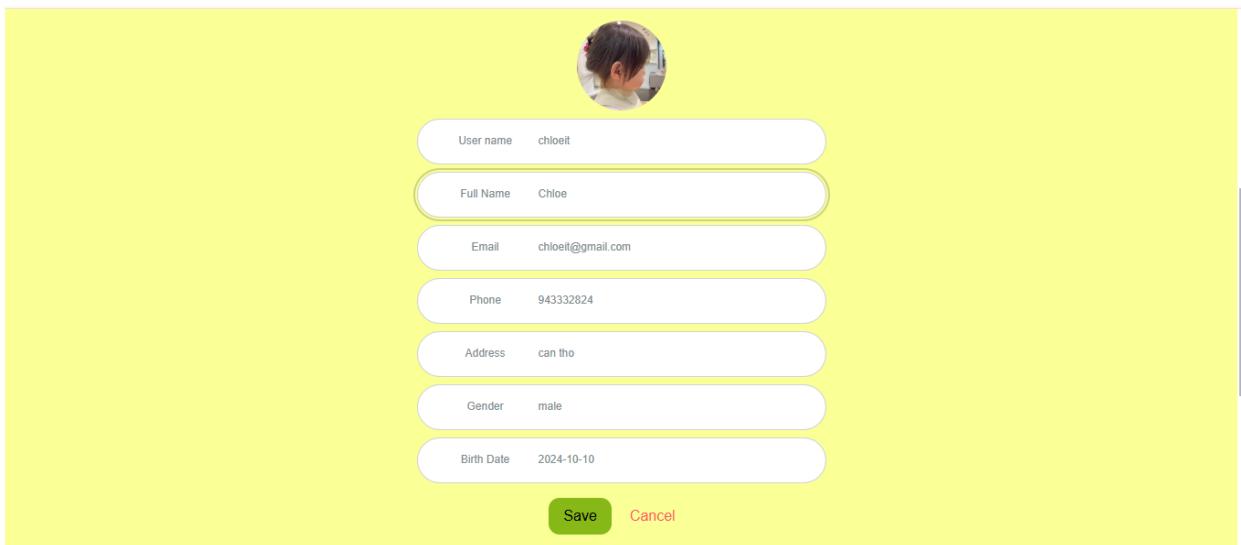


Figure 34:view Profile / edit Profile

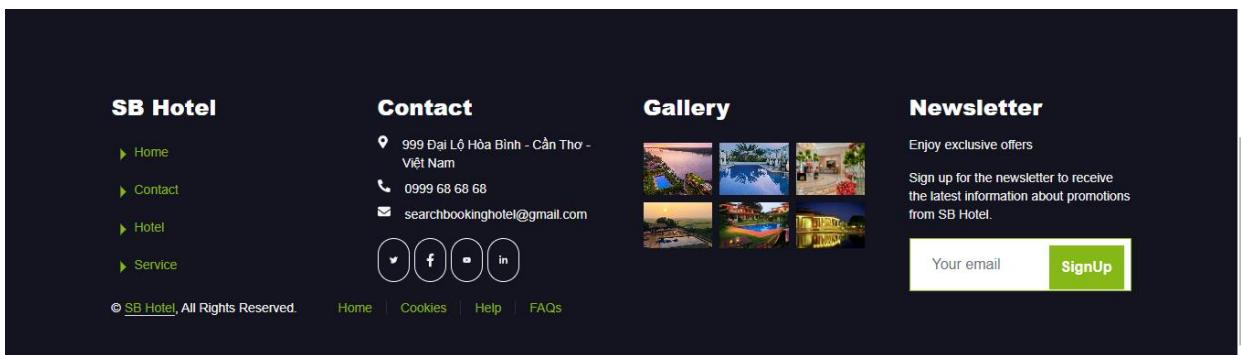


Figure 35: view Profile / Footer

3.1.9.

Log-in

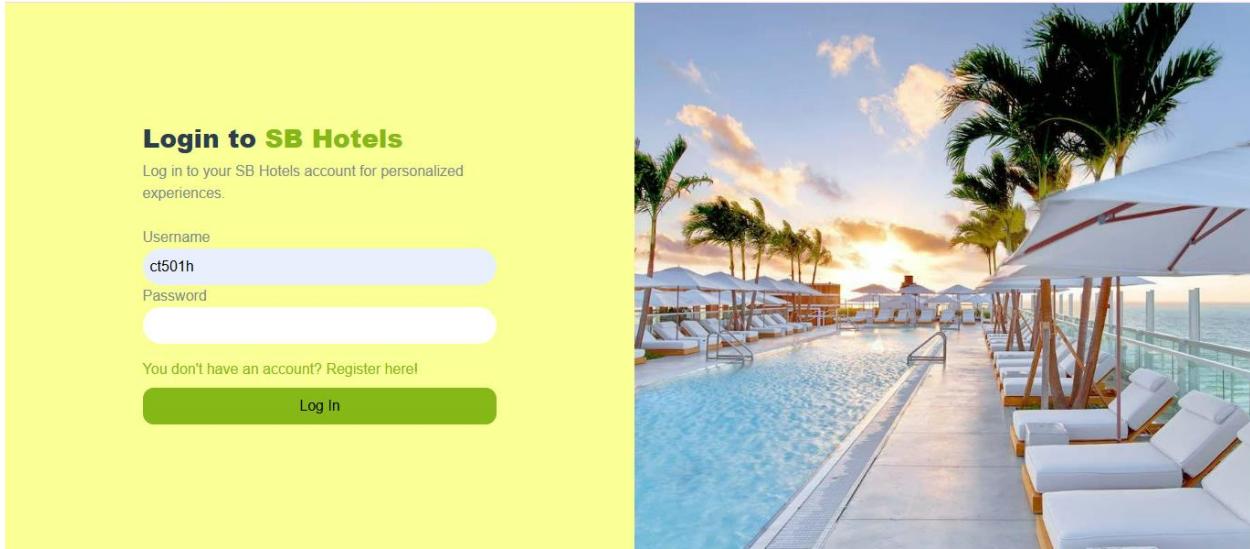


Figure 36: view Login

3.1.10.

Register

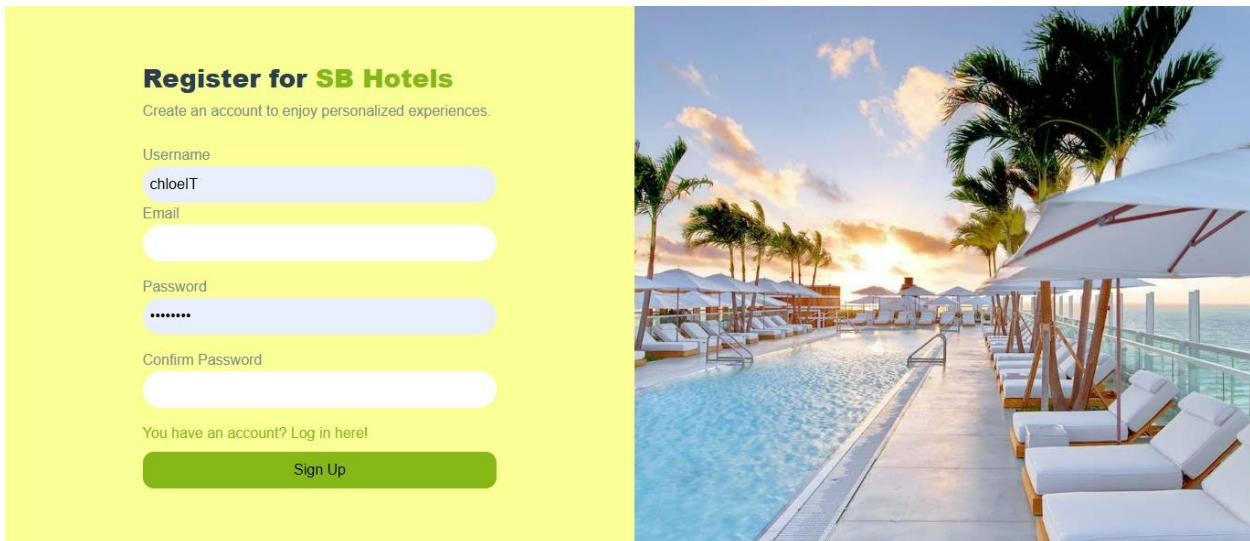


Figure 37: view Register

3.1.11.

Admin Hotel

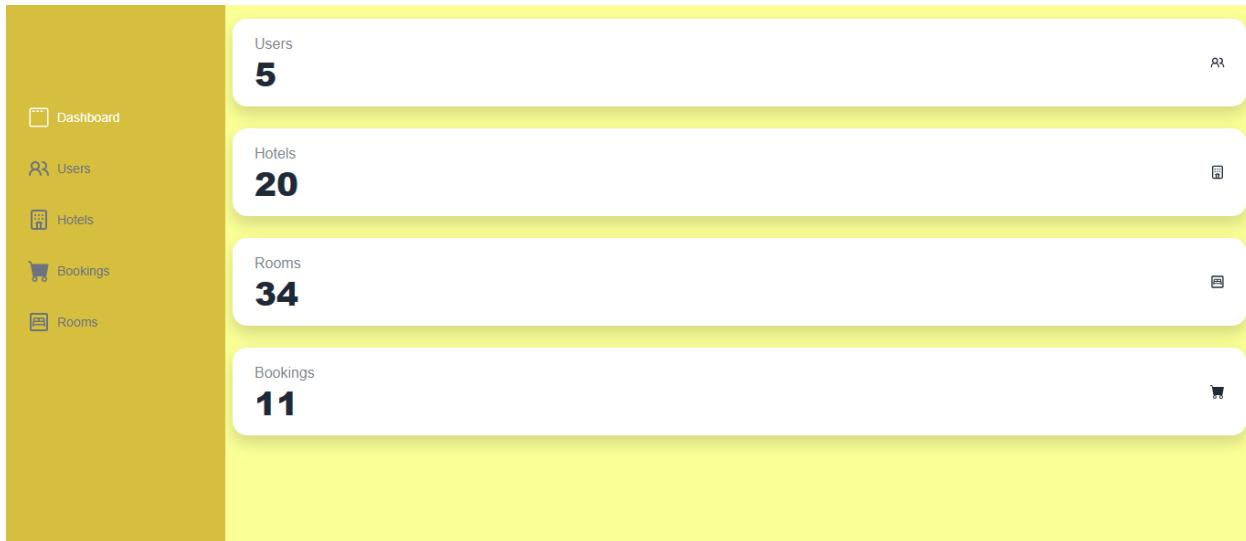


Figure 38: view Admin Dashboard

Hotels							
Image	Name	Address	Amenities	Rating	Phone	Rooms	Action
	Ocean View Resort	456 Park Ave	Free WiFi Gym Spa	4.2	1234567891	2	Edit Delete
	Mountain Escape Lodge	789 Broadway	Free WiFi Parking Restaurant	4.8	1234567892	2	Edit Delete
	City Center Inn	321 Elm St	Gym Spa Pool	3.9	1234567893	4	Edit Delete
	The Royal Sands	654 Maple St	Free WiFi Parking Gym	4.7	1234567894	4	Edit Delete
	Sunset Paradise Hotel	987 Oak St	Free WiFi Gym Pool	4	1234567895	2	Edit Delete
	Green Valley Retreat	555 Pine St	Free WiFi Parking	3.8	1234567896	2	Edit Delete
	Golden Gate Suites	222 Cedar St	Free WiFi Spa Gym	4.6	1234567897	2	Edit Delete
	Blue Lagoon Resort	111 Birch St	Free WiFi Pool Gym	4.3	1234567898	2	Edit Delete
	Riverside Inn	888 Spruce St	Free WiFi Parking Spa	4.1	1234567899	1	Edit Delete
	Emerald Bay Hotel	777 Redwood St	Free WiFi Gym Pool	4.5	1234567800	1	Edit Delete

Figure 39: view Admin Hotel

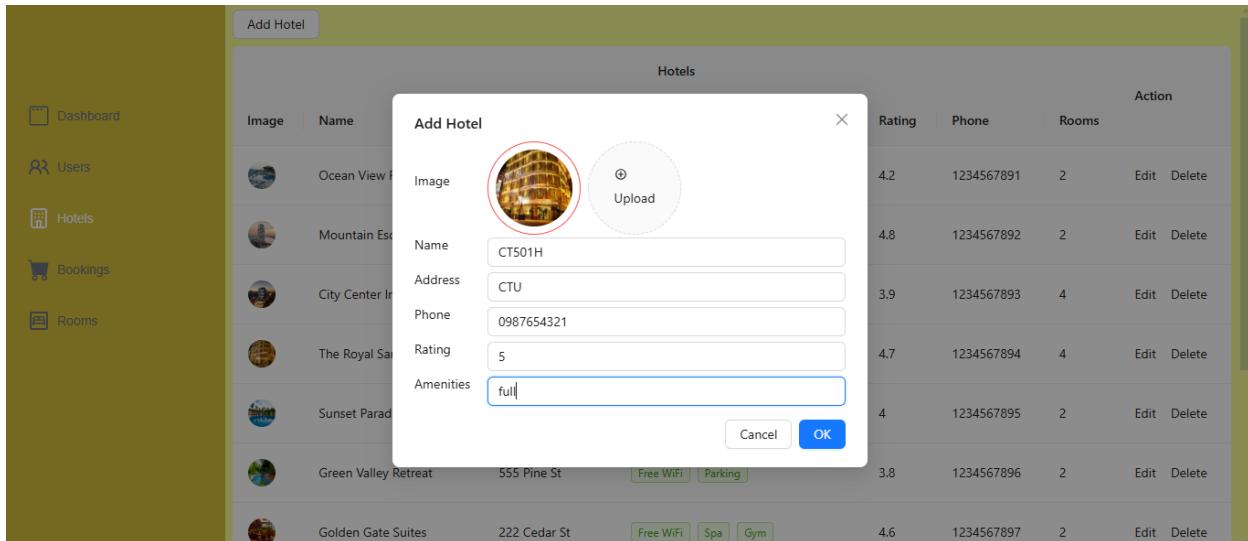


Figure 40: view Admin Hotel / add Hotel

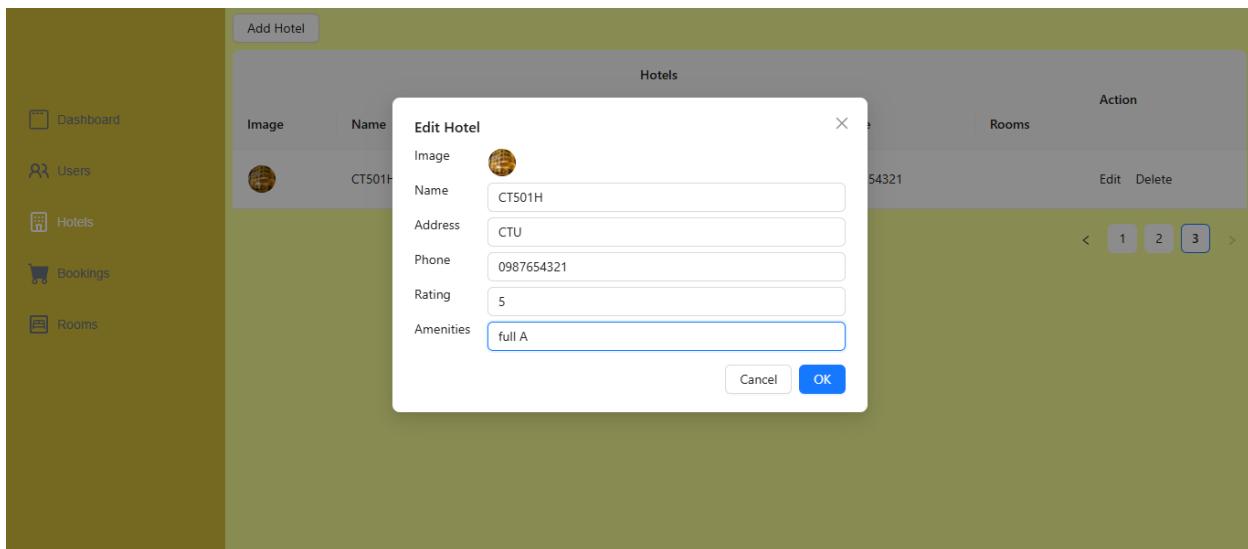


Figure 41: view Admin Hotel / edit Hotel

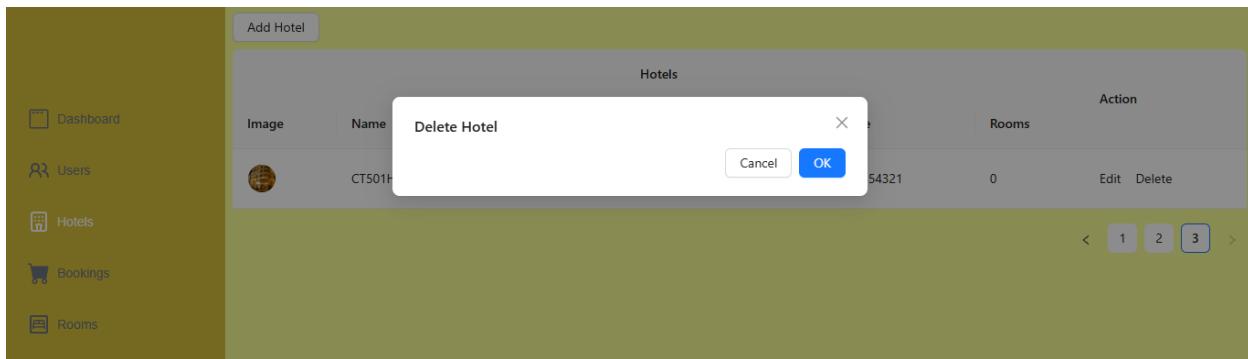
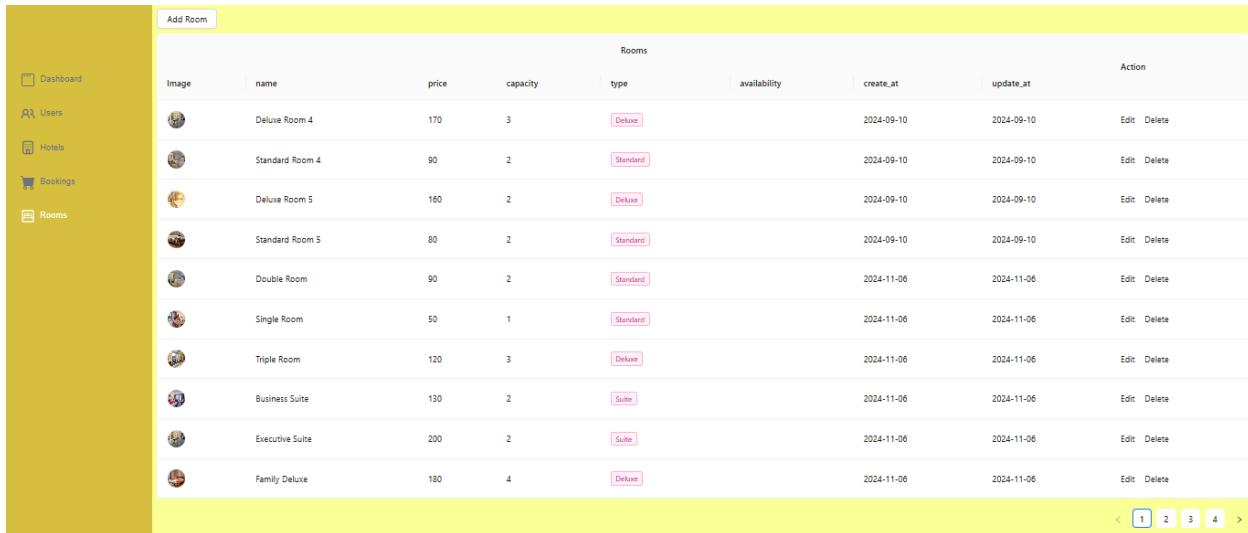


Figure 42: view Admin Hotel / delete Hotel

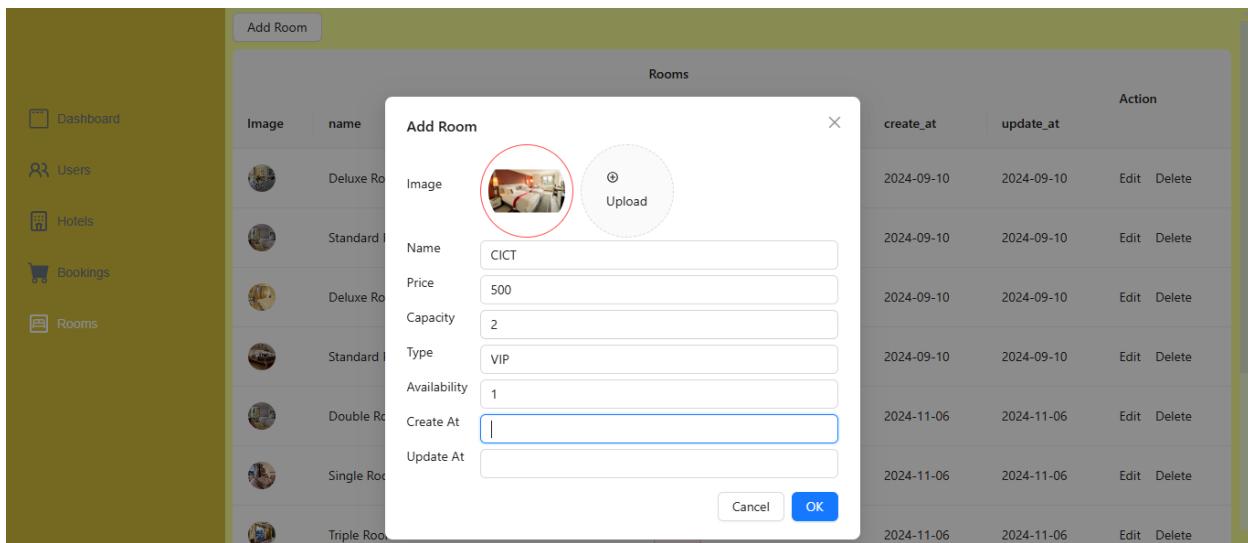
3.1.12.

Admin Room



Rooms								Action
Image	name	price	capacity	type	availability	create_at	update_at	
	Deluxe Room 4	170	3	Deluxe		2024-09-10	2024-09-10	Edit Delete
	Standard Room 4	90	2	Standard		2024-09-10	2024-09-10	Edit Delete
	Deluxe Room 5	160	2	Deluxe		2024-09-10	2024-09-10	Edit Delete
	Standard Room 5	80	2	Standard		2024-09-10	2024-09-10	Edit Delete
	Double Room	90	2	Standard		2024-11-06	2024-11-06	Edit Delete
	Single Room	50	1	Standard		2024-11-06	2024-11-06	Edit Delete
	Triple Room	120	3	Deluxe		2024-11-06	2024-11-06	Edit Delete
	Business Suite	130	2	Suite		2024-11-06	2024-11-06	Edit Delete
	Executive Suite	200	2	Suite		2024-11-06	2024-11-06	Edit Delete
	Family Deluxe	180	4	Deluxe		2024-11-06	2024-11-06	Edit Delete

Figure 43: view Admin Room



Add Room

Image	name	Add Room	Action
	Deluxe Room	<input type="text" value="CICT"/> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid red; border-radius: 50%; width: 40px; height: 40px; margin-right: 10px;"></div> <div style="border: 1px dashed #ccc; border-radius: 50%; width: 40px; height: 40px; margin-left: 10px;"></div> </div>	<input type="button" value="Create_at"/> <input type="button" value="Update_at"/>
	Standard Room	<input type="text" value="500"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Deluxe Room	<input type="text" value="2"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Standard Room	<input type="text" value="VIP"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Double Room	<input type="text" value="1"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Single Room	<input type="text" value=""/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Triple Room	<input type="text" value=""/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 44: view Admin Room / add Room

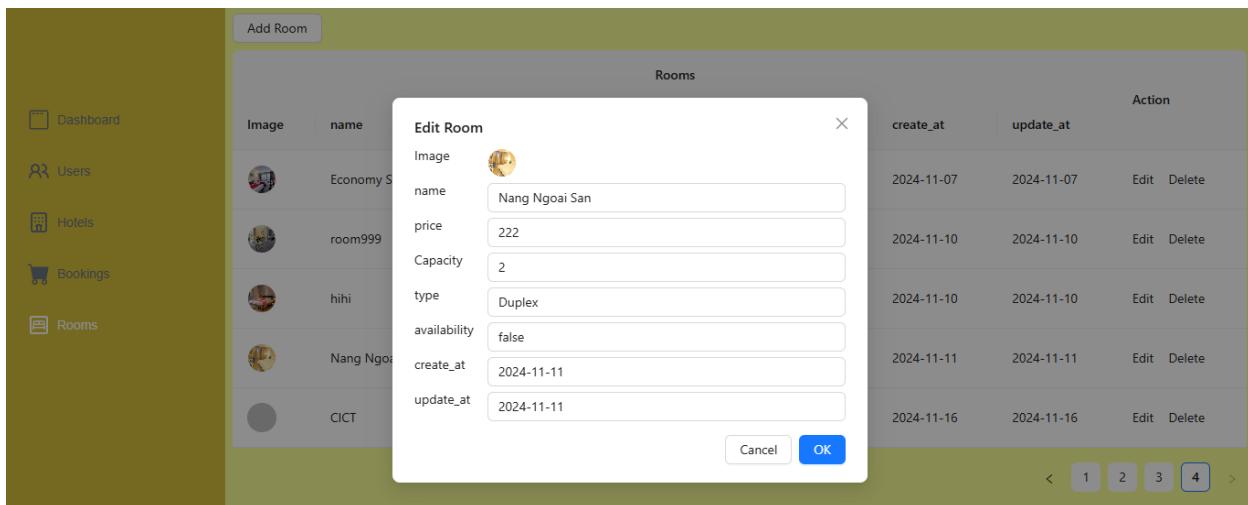


Figure 45: view Admin Room / edit Room

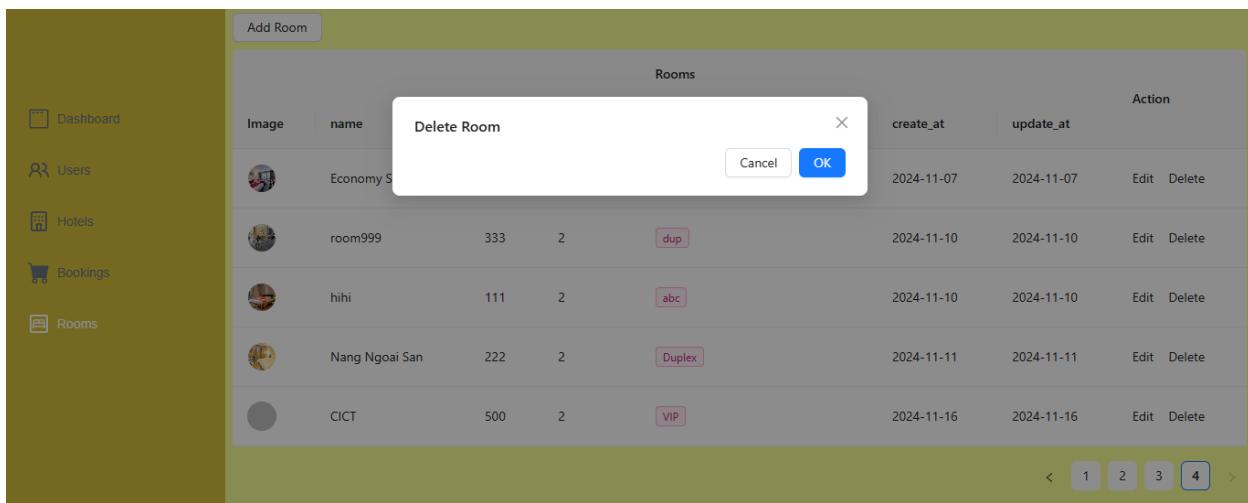


Figure 46: view Admin Room / delete Room

3.1.13.**Admin Booking**

Bookings					Action
Check In	Check Out	Total Price	Payment		
2024-10-10T15:00:00	2024-10-12T13:00:00	610	paid	Edit	Delete
2024-09-30T15:30:00	2024-10-01T10:00:00	520	not yet paid	Edit	Delete
2024-10-30T07:00:00	2024-10-31T10:00:00	490	paid	Edit	Delete
2024-10-30T07:00:00	2024-10-30T08:00:00	4449	paid	Edit	Delete
2024-09-30T15:30:00	2024-09-30T17:30:00	22	not yet paid	Edit	Delete
2024-11-11T07:00:00	2024-11-12T07:00:00	90	paid	Edit	Delete
2024-11-11T07:00:00	2024-11-12T07:00:00	999	paid	Edit	Delete
2024-11-15T07:00:00	2024-11-27T07:00:00	1680	paid	Edit	Delete
2024-11-15T07:00:00	2024-11-16T07:00:00	180	paid	Edit	Delete
2024-11-15T07:00:00	2024-11-16T07:00:00	130	paid	Edit	Delete

1 2 >

Figure 47: view Admin Booking

Add Booking

Check In	2024-10-10T15:00:00
Check Out	2024-11-10T15:00:00
Total Price	555
Payment	true

Figure 48: view Admin Booking / add Booking

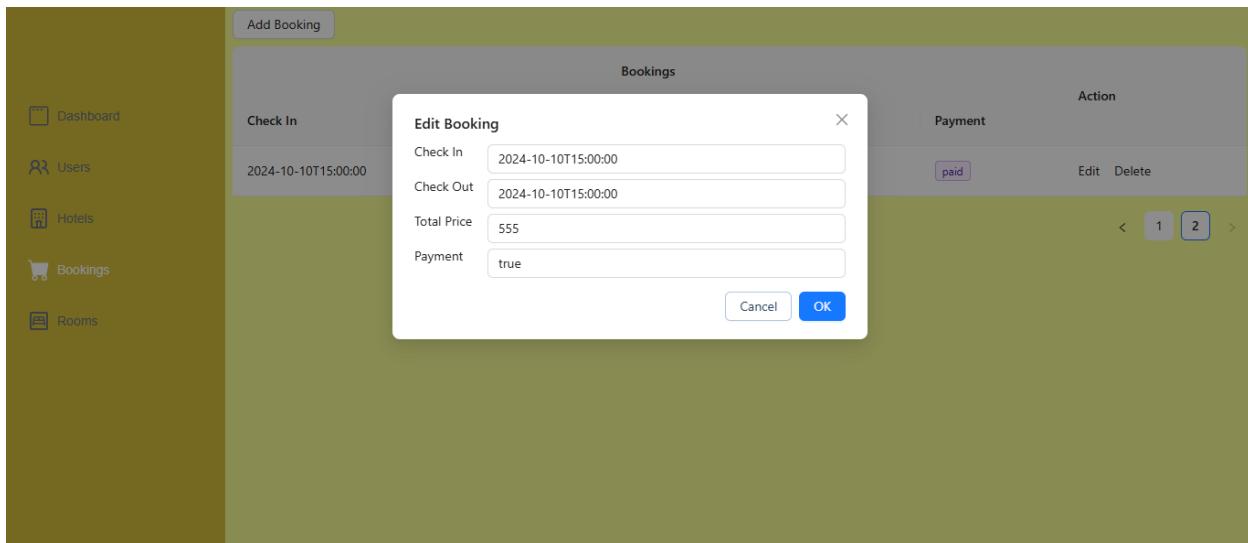


Figure 49: view Admin Booking / edit Booking

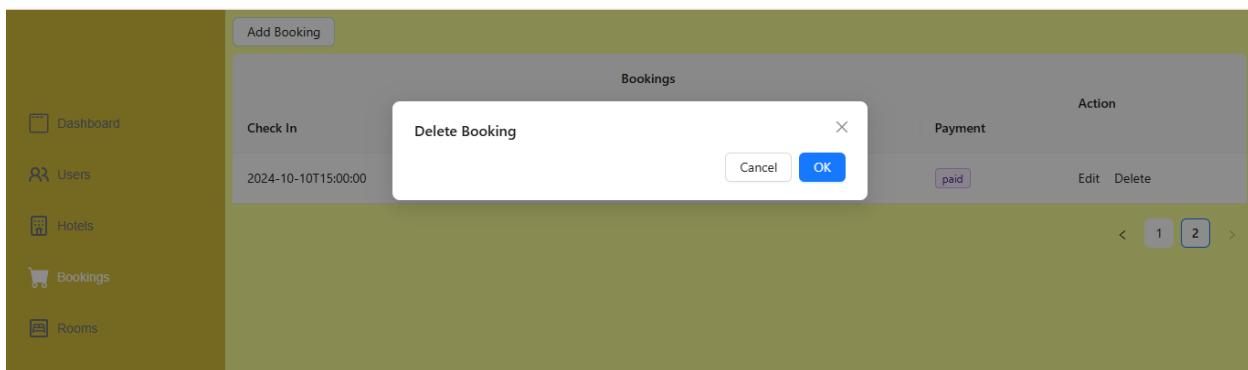


Figure 50: view Admin Booking / delete Booking

3.1.14.

Admin User

Users								Action	
Image	Username	Full name	Gender	Phone	Birth date	Roles	Address	Edit	Delete
	chloet	Chloe	male	943332824	2024-10-10	<button>ROLE_ADMIN</button> <button>ROLE_MODERATOR</button>	can tho	<button>Edit</button>	<button>Delete</button>
	admin1	dasda	male	922445654		<button>ROLE_ADMIN</button>	can tho	<button>Edit</button>	<button>Delete</button>
	pamela	HD pamela	male	987654321		<button>ROLE_USER</button>	pam	<button>Edit</button>	<button>Delete</button>
	linhlinh			828239292		<button>ROLE_ADMIN</button>		<button>Edit</button>	<button>Delete</button>

Figure 51: view Admin User

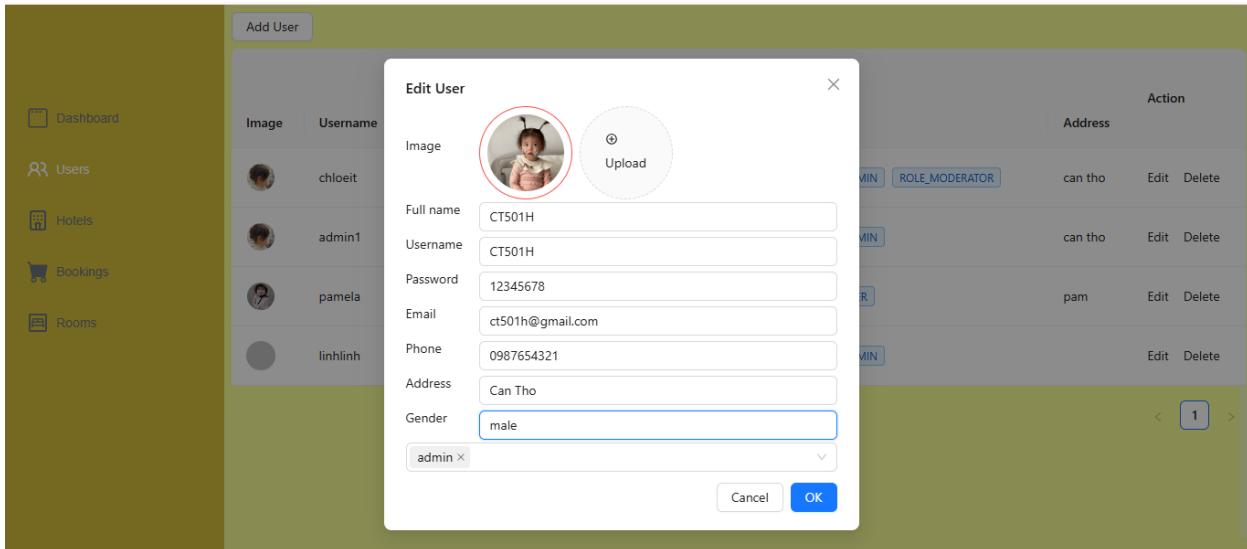


Figure 52: view Admin User / add User

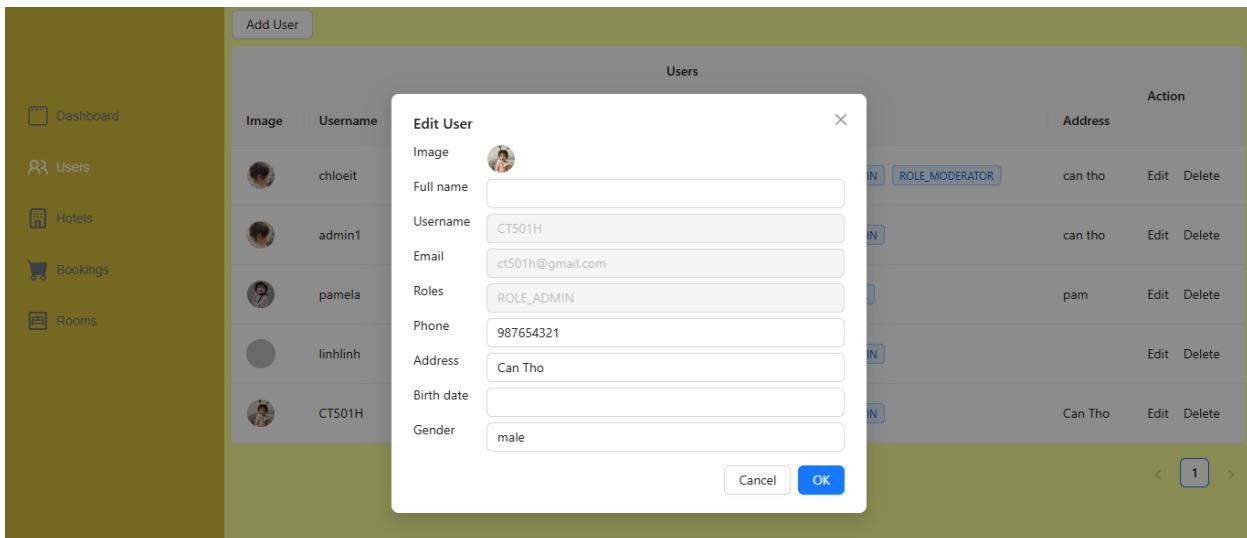


Figure 53: view Admin User / edit User

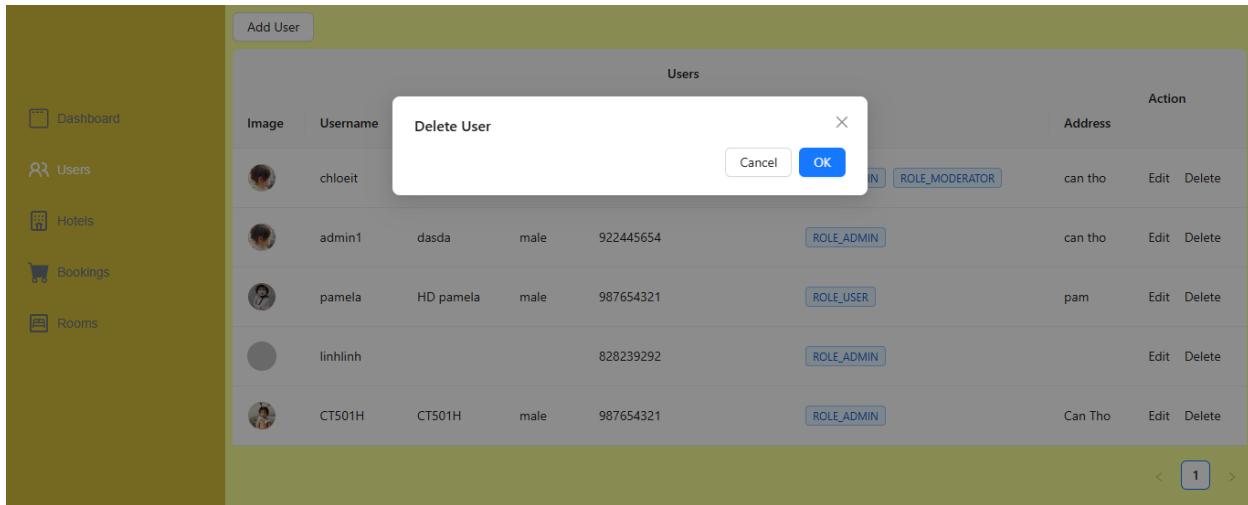


Figure 54: view Admin User / delete User

3.2. Design Description using operating system and hardware

3.2.1. Operating environment

- Database system: MySQL
- Window 10, 11 operating system
- RAM: 8GB
- SSD: 512GB
- CPU: Intel Core i7
- Screen 14 inches or more
- Supports Internet connection

3.2.2. Requires external communication

- Hardware communication:
 - Interacts via mouse and keyboard
 - Other supported devices: Wifi Modern
- Software communication:

- Use MySQL database management system
- Windows operating systems

CHAPTER 4. TESTING AND EVALUATION

4.1. Testing objectives:

- The main goal of testing is to detect errors and check whether the program meets the established requirements or not.
- The testing process includes usability testing, functional testing, compatibility testing, database testing, and security testing.
- Usability testing is checking whether the website and application are user friendly or not? Can new users understand the application easily or not.
 - Links to the home page are available on each page.
 - A confirmation message is displayed for any update operation.
 - Check if the static path displayed on the browser is correct or not.
 - Test all links to see if they work and if they go to the expected screen.
 - Check the ability to force data entry, empty data, bind data, display a message when entering incorrect data.
 - Check content: images, text on other devices.
 - The content is accurate, without any spelling or grammatical errors

4.2. Testing scenarios:

4.2.1. Register function

Order	Test case description	Implementation steps	Implementation date
1	Customer register	1. Open web 2. Register 3. Enter username & password 4. Click register	17/09/2024

Table 7 Register function

4.2.2.**Login function**

Order	Test case description	Implementation steps	Implementation date
1	Customer login	<ol style="list-style-type: none"> 1. Open web 2. Login 3. Enter username & password 4. Click login 	17/09/2024

Table 8: Login function

4.2.3.**Update profile function**

Order	Test case description	Implementation steps	Implementation date
1	Customer update profile	<ol style="list-style-type: none"> 1. Open web 2. Login 3. Update profile 4. Click save changes 	17/10/2024

Table 9: Update profile function

4.2.4.**Logout function**

Order	Test case description	Implementation steps	Implementation date
1	Customer logout	<ol style="list-style-type: none"> 1. Access website 2. Logout 	17/10/2024

Table 10: Logout function

4.2.5.**Create Admin User function**

Order	Test case description	Implementation steps	Implementation date
1	Admin User	<ol style="list-style-type: none"> 1. Access admin rights 2. Start add, edit, delete user 	02/10/2024

Table 11 Create hotel function

4.2.6.**Create Admin Booking function**

Order	Test case description	Implementation steps	Implementation date
1	Admin Booking	<ol style="list-style-type: none"> 1. Access admin rights 2. Start add, edit, delete booking 	02/10/2024

*Table 12: Edit hotel function***4.2.7.****Create Admin Hotel & Room function**

Order	Test case description	Implementation steps	Implementation date
1	Admin Hotel & Room	<ol style="list-style-type: none"> 1. Access admin rights 2. Start access Admin Hotel & Room function 	30/10/2024

*Table 13 : Delete hotel function***4.2.8.****Create Add to Favorite function**

Order	Test case description	Implementation steps	Implementation date
1	Add to Favorite	<ol style="list-style-type: none"> 1. Access admin rights 2. Start access Admin Hotel & Room function 	30/10/2024

*Table 14 : Create Add to Favorite***4.2.9.****Create Booking function**

Order	Test case description	Implementation steps	Implementation date
1	Booking	<ol style="list-style-type: none"> 1. Access admin rights 2. Start access Admin Hotel & Room function 	30/10/2024

Table 15: Create Booking

4.2.10.**Create Payment function**

Order	Test case description	Implementation steps	Implementation date
1	Payment	<ol style="list-style-type: none"> 1. Access admin rights 2. Start access Admin Hotel & Room function 	30/10/2024

*Table 16: Payment***4.2.11.****Create Filter Hotel function**

Order	Test case description	Implementation steps	Implementation date
1	Payment	<ol style="list-style-type: none"> 1. Access admin rights 2. Start access Admin Hotel & Room function 	30/10/2024

*Table 17: Filter***4.3. Testing results:****4.3.1.****Register function**

order	Test case description	Implementation steps	Implementation date	Result
1	Customer register	<ol style="list-style-type: none"> 1. Open web 2. Register 3. Enter username & password 4. Click register 	17/03/2024	success

*Table 18: Testing register function***4.3.2.****Login function**

order	Test case description	Implementation steps	Implementation date	Result
1	Customer login	<ol style="list-style-type: none"> 1. Open web 2. Login 3. Enter username & password 4. Click login 	30/03/2024	success

Table 19: Testing login function

4.3.3.**Update profile function**

order	Test case description	Implementation steps	Implementation date	Result
1	Customer update profile	1. Open web 2. Login 3. Update profile 4. Click save changes	10/04/2024	success

Table 20: Testing update profile function

4.3.4.**Logout function**

order	Test case description	Implementation steps	Implementation date	Result
1	Customer logout	1. Access web 2. Logout	30/04/2024	success

Table 21: Testing logout function

4.3.5.**Admin User function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start creating a new hotel	02/05/2024	success

Table 22: Testing create hotel function

4.3.6.**Admin Booking function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start edit hotel	02/05/2024	success

Table 23: Testing edit hotel function

4.3.7.**Admin Hotel & Room function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start delete hotel	02/05/2024	success

Table 24: Testing delete hotel function

4.3.8.**Add to Favorite function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start delete hotel	02/05/2024	success

*Table 25: Add to Favorite***4.3.9.****Booking function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start delete hotel	02/05/2024	success

*Table 26: Booking***4.3.10.****Payment function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start delete hotel	02/05/2024	success

*Table 27: Payment***4.3.11.****Filter Hotel function**

order	Test case description	Implementation steps	Implementation date	Result
1	Create hotel	1. Access admin rights 2. Start delete hotel	02/05/2024	success

Table 28: Filter Hotel

C. CONCLUSION

1. Achieved results :

- Basically, "Hotel search and booking website" has achieved its set goals.
- Build a customer-friendly interface with features: allowing customers to register, log in to their account, and view hotel information.

2. Future directions:

- For the system to work better, there needs to be a future orientation develop. The goal is set to improve and develop the system.
 - Build more diverse data sources, closer to reality
 - Build and develop more functions for administrators, hotels and customers
 - Add advanced functions such as: Chatbox, reviews...
 - Improved interface, increased user experience

TÀI LIỆU THAM KHẢO
abc

PHỤ LỤC

abc