



CS 353 - Database Systems

**Hotel Database Management System  
Final Report**

Group 38

Maryam Shahid	21801344
Bora Cün	21802838
Çağrı Eren	21801831
Hammad Khan Musakhel	21801175

# **Contents**

<b>1. Introduction</b>	<b>6</b>
<b>2. Final E/R</b>	<b>7</b>
<b>3. Final List of Tables</b>	<b>8</b>
Users	8
Employee	8
Manager	8
Security-staff	8
Housekeeper	9
Recruiter	9
Receptionist	9
Candidate	9
Guests	9
Room	9
Room-type	9
Reservation	9
Comment	10
QnA	10
Building	10
Security-walk	10
Cleaning-Duty	10
Location	10
Event	10
Guest-Activity	11
Activity	11
Group-Tours	11
Training-Program	11
Ticket	11

Tourist-Attraction	11
Places-to-visit	11
Sec-staff-applies-to	12
Evaluates-sec-staff-application	12
HK-applies-to	12
Evaluates-hk-application	12
Food-Drink	12
Order-contains	12
Food-Order	12
Restaurant	13
Job-Application	13
Approves	13
Security-Report	13
Leave-Request-Form	13
<b>4. Implementation Plan</b>	<b>14</b>
Tools and Technologies	14
Challenges	14
Work Allocation	14
Maryam Shahid	14
Bora Çün	14
Çağrı Eren	14
Hammad Khan Musakhel	14
<b>5. Advanced Database Components</b>	<b>15</b>
5.1. Reports	15
5.2. Views	16
5.3. Triggers	16
5.4. Constraints	16
5.5. Stored Procedures	17
5.6. Secondary Indices	17

<b>6. User's Manual</b>	<b>18</b>
Homepage for All Users	18
Login Page for Guest	18
Incorrect Login	19
Homepage for Guest I	20
Homepage for Guest II (after scrolling)	20
Reservation I	21
Reservation II (after scrolling)	22
Featured Room Type I - Single Deluxe	23
Reservation II - Single Economy	23
Reservation Complete	24
Order Food	25
Add Food to Cart	25
Confirm Food Order	26
Apply for Activities	27
Choose an Activity	27
Comment Card	28
Employee Login	29
Choose from Manager, Housekeeper and Security	30
Manager Login	30
Manager Homepage I	31
Manager Homepage II (after scrolling)	32
Assign Food Order to Housekeeper	32
Assign Order Confirmation	33
Assign Duty - Security Walk	34
Create Event	34
Evaluate Training Application	35
Reject Training Application	35
Evaluate Leave Application	36

Homepage for Housekeeper	37
Assigned Orders	37
Apply Page for Employees	38
Apply Training for Employees	39
Home page for Security	40
Home page for Security	40

## **1. Introduction**

Beach Resort is a hotel database management system which is a web-based project designed to allow user access to a hotel database. The users consist of guests, employees and candidates. Guests are able to book a room by choosing from a variety of room types in the location they prefer. They are also able make comments on their reservations. The system allows them to order food from various restaurants depending on the availability of food and buy tickets for a plethora of events being conducted at the hotel - such as activities, group tours, pool parties etc.

Similarly, employees consist of managers, housekeepers, security staff and recruiters. The system allows them to access their role-specific features. Managers are able to assign tasks to other employees - food orders for delivery, cleaning duties to housekeepers and buildings for security walks to security staff. They can also initiate events - guest activities, group tours, training programs - in various locations of the hotel. The system gives them the authority to accept or reject applications for training programs and leave requests created by housekeepers and security staff. Similarly, housekeepers are assigned cleaning duties to various rooms and are able to deliver food to guests. Housekeepers are also able to create/ view leave and training applications. Security staff have similar functions. Managers are all able to view candidate applications, just as candidates are allowed to apply for available positions in the hotel. All employees are allowed to fill a leave request form by checking their annual leave.

## 2. Final E/R

The ER Diagram from the Design Report , shown in Figure 2.1, has been improved to normalize the tables of the database. The Final ER Diagram is shown in the Figure 2.2 below. The only change is the following:

- Room table has been split into Room table and Room-Type table in order to avoid redundancy in the previous Room table. The old Room table had attributes that were the same for all the room types. This redundancy is avoided in the new version.

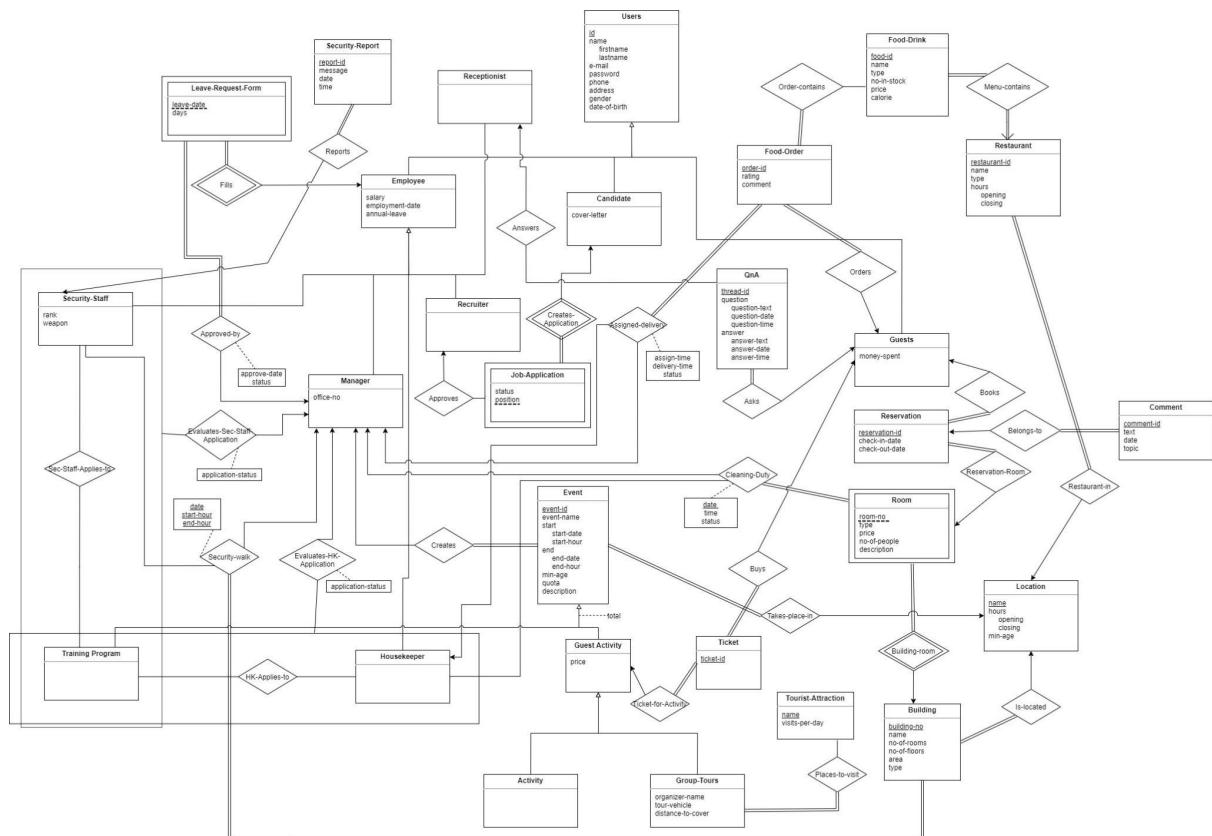


Figure 2.1: ER Diagram from the Design Report

[Link for better image quality](#)

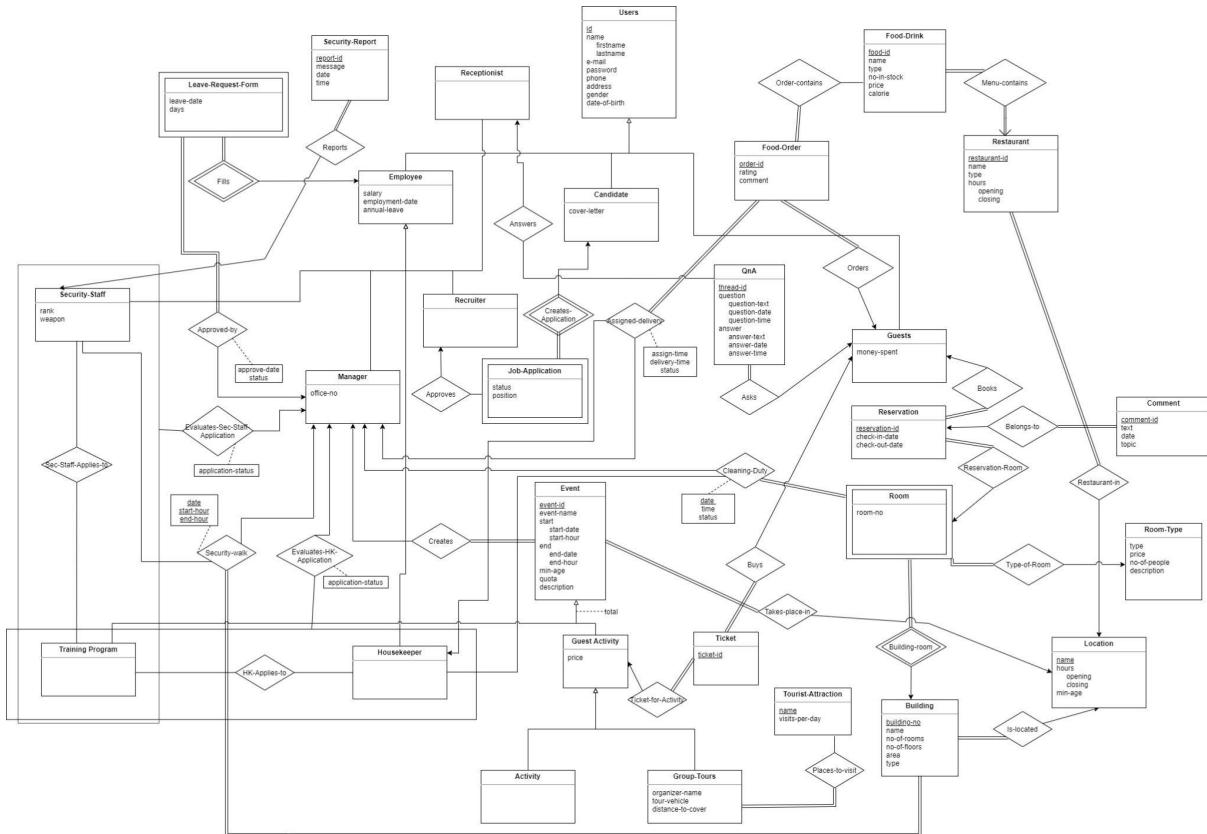


Figure 2.2: Final ER Diagram

[Link for better image quality](#)

### **3. Final List of Tables**

### 3.1. Users

Users(**id**, firstname, lastname, email, password, phone, address, gender, date-of-birth)

\*No foreign keys

### 3.2. Employee

Employee(**id**, salary, employment-date, annual-leave)

\*id is foreign key to Users

### 3.3. Manager

Manager(id, office-no)

\*id is foreign key to Employee

### **3.4. Security-staff**

Security-staff(***id***, rank, weapon)

\*id is foreign key to Employee

### **3.5. Housekeeper**

Housekeeper(id)

\*id is foreign key to Employee

### **3.6. Recruiter**

Recruiter(id)

\*id is foreign key to Employee

### **3.7. Receptionist**

Receptionist(id)

\*id is foreign key to Employee

### **3.8. Candidate**

Candidate( id, cover-letter )

\*id is foreign key to Users

### **3.9. Guests**

Guests(id, money-spent )

\*id is foreign key to Users

### **3.10. Room**

Room(room-no, building-no, type, description)

\*(type) is foreign key to Room-type

\*(building-no) is foreign

### **3.11. Room-type**

Room-type(type, price, no-of-people)

\*No foreign keys

### **3.12. Reservation**

Reservation(reservation-id, guest-id, room-no, building-no, check-in-date, check-out-date )

\*(room-no,building-no) is a foreign key to Room  
\*(guest-id) is a foreign key to Guests

### **3.13. Comment**

Comment(comment-id, reservation-id, text, date, topic )

\*(reservation-id) is a foreign key for Reservation

### **3.14. QnA**

QnA(thread-id, guest-id,question-text, question-date, question-time, answer-text, answer-date, answer-time, receptionist-id)

\*(guest-id) is a foreign key to Guests  
\*(receptionist-id) is a foreign key to QnA

### **3.15. Building**

Building(building-no, location-name, name, no-of-rooms, no-of-floors, area,type)

\*(location-name) is a foreign key to Location

### **3.16. Security-walk**

Security-walk(manager-id,security-staff-id, building-no, date, start-hour, end-hour)

\*(manager-id) is a foreign key to Manager  
\*(security-staff-id) is a foreign key to Security-Staff  
\*(building-no) is a foreign key to Building

### **3.17. Cleaning-Duty**

Cleaning-Duty(housekeeper-id, room-no, building-no, date, time, status, manager-id)

\*(room-no, building-no) is a foreign key to Room  
\*(building-no) is a foreign key to Building  
\*(housekeeper-id) is a foreign key to Housekeeper  
\*(manager-id) is a foreign key Manager

### **3.18. Location**

Location(name, opening, closing, min-age )

### **3.19. Event**

Event(event-id, event-name, location-name, start-date, start-hour, end-date, end-hour, min-age, quota, description, manager-id)

- \*(location-name) is a foreign key to Location
- \*(manager-id) is a foreign key to Manager

### **3.20. Guest-Activity**

Guest-Activity(event-id, price)

- \*(event-id) is a foreign key to Event

### **3.21. Activity**

Activity(event-id)

- \*(event-id) is a foreign key to Guest-Activity

### **3.22. Group-Tours**

Group-Tours(event-id, organizer-name, tour-vehicle, distance-to-cover)

- \*(event-id) is a foreign key to Guest-Activity

### **3.23. Training-Program**

Training-Program(event-id)

- \*(event-id) is a foreign key to Event

### **3.24. Ticket**

Ticket(ticket-id, event-id, guest-id)

- \*(event-id) is a foreign key to Guest-Activity
- \*(guest-id) is a foreign key to Guests

### **3.25. Tourist-Attraction**

Tourist-Attraction(name, visits-per-day)

### **3.26. Places-to-visit**

Places-to-visit(event-id, attraction-name)

- \*(event-id) is a foreign key to Group-Tours
- \*(attraction-name) is a foreign key to Tourist-Attraction

### **3.27. Sec-staff-applies-to**

Sec-staff-applies-to(sec-staff-id, training-program-id)

\*(sec-staff-id) is a foreign key to Security-Staff

\*(training-program-id) is a foreign key to Training-Program

### **3.28. Evaluates-sec-staff-application**

Evaluates-sec-staff-application(sec-staff-id, training-program-id, manager-id, application-status)

\*(sec-staf-id) is a foreign key to Security-Staff

\*(training-program-id) is a foreign key to Training-Program

\*(manager-id) is a foreign key to Manager

### **3.29. HK-applies-to**

HK-applies-to(housekeeper-id, training-program-id)

\*(housekeeper-id) is a foreign key to Housekeeper

\*(training-program-id) is a foreign key to Training-Program

### **3.30. Evaluates-hk-application**

Evaluates-hk-application(housekeeper-id, training-program-id, manager-id, application-status)

\*(housekeeper-id) is a foreign key to Housekeeper

\*(training-program-id) is a foreign key to Training-Program

\*(manager-id) is a foreign key to Manager

### **3.31. Food-Drink**

Food-Drink(food-id, name, type, no-in-stock, price, calorie, restaurant-id)

\*(restaurant-id) is a foreign key to Restaurant

### **3.32. Order-contains**

Order-contains(order-id, food-id)

\*(food-id) is a foreign key to Food-Drink

### **3.33. Food-Order**

Food-Order(order-id, rating, comment, guest-id, manager-id, housekeeper-id, assign-time, delivery-time, status)

\*(guest-id) is a foreign key to Guests

\*(manager-id) is a foreign key to Manager  
\*(housekeeper-id) is a foreign key to Housekeeper

### **3.34. Restaurant**

Restaurant(restaurant-id, location-name, name, type, hours-opening, hours-closing)

\*(location-name) is a foreign key to Location

### **3.35. Job-Application**

Job-Application(id, position, status)

\*(id) is a foreign key to Candidate

### **3.36. Approves**

Approves(candidate-id, position, recruiter-id)

\*(candidate-id) is a foreign key to Candidate  
\*(recruiter-id) is a foreign key to Recruiter

### **3.37. Security-Report**

Places-to-visit(event-id, attraction-name)

\*(event-id) is a foreign key to Security-Staff

### **3.38. Leave-Request-Form**

Leave-Request-Form(id, leave-date, days, manager-id, approve-date, status)

\*(id) is a foreign key to Employee  
\*(manager-id) is a foreign key to Manager

## **4. Implementation Plan**

### **4.1. Tools and Technologies**

We used React.js to build the interface and the front-end of the application. For backend Spring Boot is used and for database Bilkent University's MariaDB is used. For the Spring Boot Java 11 is used and SQL is used to write queries. For database operations JDBC and Java's SQL library are used.

### **4.2. Challenges**

Since we divided the workload into two teams - frontend and backend, the most difficult challenge of all was to connect the two via requests. The frontend team found it difficult to learn the syntax used by the backend team and vice versa. This created a lot of conflicts and errors while fetching data, and sending/receiving requests. To overcome this, the backend team created an API document consisting of URL links, type, description, request and response. This allowed for a smoother process of integration for the members working on frontend and helped in fetching data, and handling requests for the frontend.

### **4.3. Work Allocation**

#### **4.3.1. Maryam Shahid**

Maryam worked on the frontend of the project. She designed and created the UI for guests and managers. UI for guests consisted of making a reservation, making food orders, applying for activities and giving comments; UI for managers consisted of assigning food orders and duties, and evaluating training and leave applications.

#### **4.3.2. Bora Cün**

Bora was in the backend team and was responsible for creating the tables and triggers in the database. Implemented methods and queries for inserting user types to their respective tables, database reports, all job applications, security staff, guest activity and training program features, and part of the initial database connection.

#### **4.3.3. Çağrı Eren**

Çağrı worked on backend, implemented java code and SQL queries for food related functionalities, reservations, comments, login, register and leave request related functionalities.

#### **4.3.4. Hammad Khan Musakhel**

Hammad was in the team to develop the user interface for the application and also coordinate the connection between the front and backend. He designed the UI for Housekeeper and Security. He also designed the login page for all users to allow registered people to log in.

# 5. Advanced Database Components

## 5.1. Reports

- The first report is a query to find out the average, minimum, and maximum age of each user type. SQL query:

```
SELECT CASE
```

```
    WHEN (Users.id IN (SELECT id FROM Guests)) THEN 'Guest'  
  
    WHEN (Users.id IN (SELECT id FROM Manager)) THEN 'Manager'  
  
    WHEN (Users.id IN (SELECT id FROM Housekeeper)) THEN 'Housekeeper'  
  
    WHEN (Users.id IN (SELECT id FROM Security_Staff)) THEN 'Security Staff'  
  
    WHEN (Users.id IN (SELECT id FROM Recruiter)) THEN 'Recruiter'  
  
    WHEN (Users.id IN (SELECT id FROM Receptionist)) THEN 'Receptionist'  
  
    WHEN (Users.id IN (SELECT id FROM Candidate)) THEN 'Candidate'  
  
    END AS user_type, AVG((UNIX_TIMESTAMP() * 1000 - date_of_birth) / 31556952000) AS avg_age, MIN((UNIX_TIMESTAMP() * 1000 - date_of_birth) / 31556952000) AS min_age, MAX((UNIX_TIMESTAMP() * 1000 - date_of_birth) / 31556952000) max_age  
  
FROM Users  
  
GROUP BY user_type;
```

Example Output:

	user_type	avg_age	min_age	max_age
1	Candidate	51.340597434410014	51.340597434410014	51.340597434410014
2	Guest	51.351683734289885	51.340597434410014	51.37580714068963
3	Housekeeper	50.56731608574237	47.47419069107181	51.340597434410014
4	Manager	49.59771681653538	47.854836198660756	51.340597434410014
5	Receptionist	20.47875472891045	20.47875472891045	20.47875472891045
6	Recruiter	51.340597434410014	51.340597434410014	51.340597434410014
7	Security Staff	49.59771681653538	47.854836198660756	51.340597434410014

Figure 5.1.1: Output of the first report.

- The second report is a query for finding out which room type gets how many reservations and how much money the guests who book these rooms spend on average. SQL query:

```
SELECT type, reservation_count, avg_money_spent  
  
FROM (SELECT type, COUNT(DISTINCT reservation_id) AS reservation_count, AVG(money_spent) AS avg_money_spent  
  
FROM Room_Type NATURAL JOIN Room NATURAL JOIN Reservation, Guests  
  
WHERE guest_id = Guests.id  
  
GROUP BY type) AS Room_Type_Stats
```

```
WHERE reservation_count > 0;
```

Example Output:

	type	reservation_count	avg_money_spent
1	good	3	45.300000
2	Single Economy	1	28.800000
3	Single Standard	2	28.800000

Figure 5.1.2: Output of the second report

## 5.2. Views

Our system currently does have a view for guests getting employee data which hides private employee data like password, salary etc. since those data should not be known by guests. SQL Query:

```
CREATE VIEW Employee_Info AS SELECT id, firstname, lastname, email, phone, address, gender, date_of_birth, employment_date
FROM Users NATURAL JOIN Employee;
```

	id	firstname	lastname	email	phone	address	gender	date_of_birth	employment_date
1	2	Michael	Scott	ms@gmail.com	123	Scranton	Male	111111111111	2222222222
2	3	Hank	SecurityMan	hs@gmail.com	312	Scranton	Male	111111111111	2222222222
3	24	Peter	Rasmussen	dev@ast2	9063	Copenhagen	Male	111111111111	1620770066
4	7353	HK2	HK2	hk2	9063	Paris	Male	111111111111	1620901179
5	8004	HK4	HK4	hk4	9063	Paris	Male	111111111111	1620901192
6	10564	HK1	HK1	hk1	9063	Paris	Male	111111111111	1620901204
7	15331	Nicolai	Reedtz	dev@ast	9063	Copenhagen	Male	111111111111	1111111111
8	25405	bora	cun	@.com	123123	address	female	123123123123	1620581947
9	53680	HK3	HK3	hk3	9063	Paris	Male	111111111111	1620901218
10	55178	Kenny	Schrub	kennys	9063	Paris	Male	111111111111	1620836042
11	67505	bora	cun	bora@bilkent.edu.tr	532	Izmir	male	975016800000	10151612

## 5.3. Triggers

Currently, the system has only one trigger implemented. When a guest orders food from one of the restaurants, this trigger activates and decrements the stock number of that food and charges the guest with money. There are other triggers we designed, however, could not implement because of the time limit. Some of these would have been used for promoting candidates to their respective jobs when their job applications are approved. One of them would have been used for adjusting guest activity properties similar to ordering food trigger described above.

## 5.4. Constraints

Although, system currently does not use any constraints, there are already designed constraints for age checking and time(deadline) checking functionalities. For example, age checking for an event that has an age limit.

## **5.5. Stored Procedures**

Although, system currently does not use any stored procedures, we have designed procedures for frequently executed queries, like upgrading the salary of a housekeeper according to their feats.

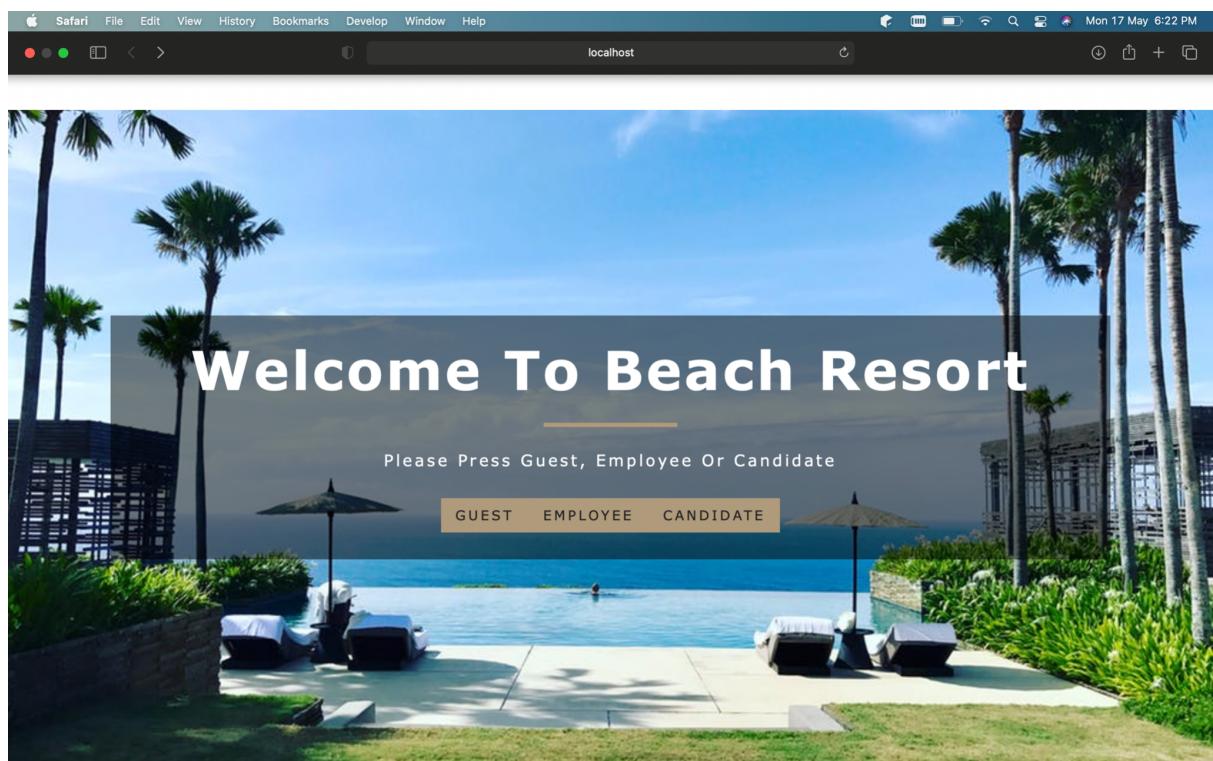
## **5.6. Secondary Indices**

We have set secondary indices for the Users table on the email and password attributes to reduce access time to the table.

## 6. User's Manual

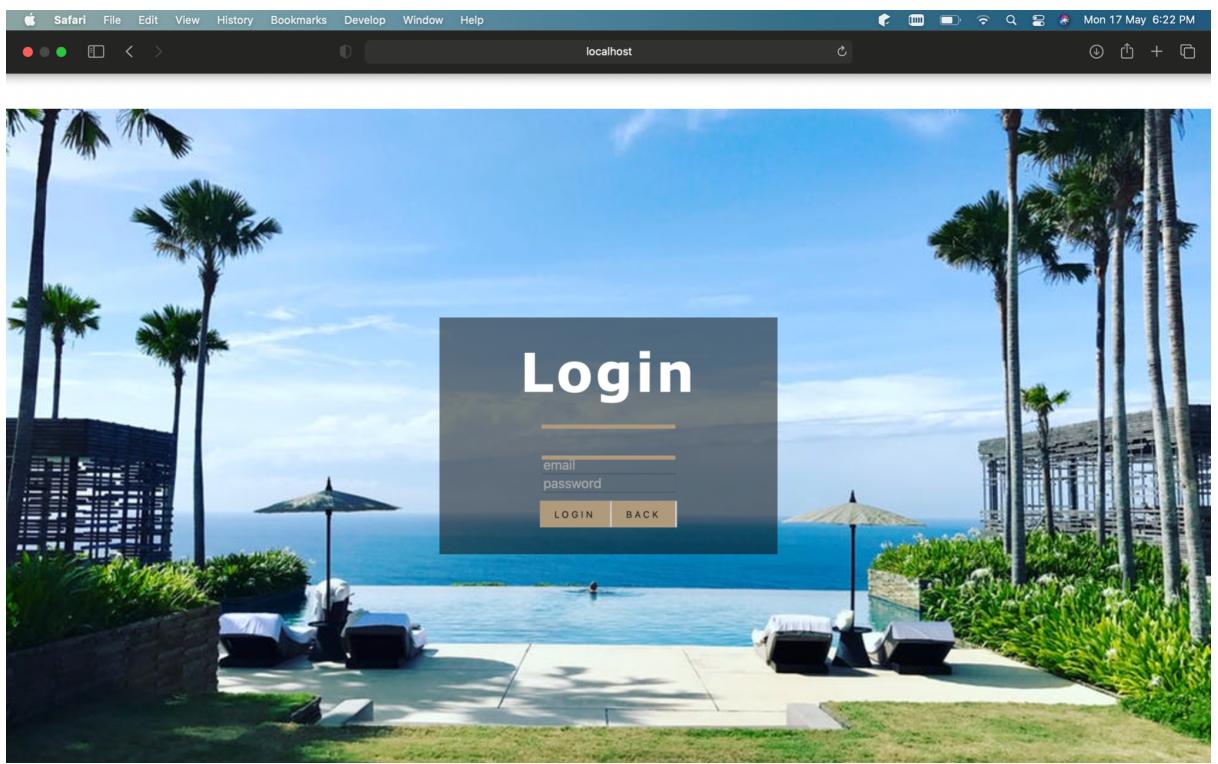
### 6.1. Homepage for All Users

This page will be displayed when the React application is launched on a local host. The page will have three choices for a user to choose from: guest, employee, and candidate. The button for guest and candidate will directly take the user to the login page whereas, the employee will introduce the user to choose from the three categories of employees. This page serves as a homepage for all, as the heading 6.1. indicates.



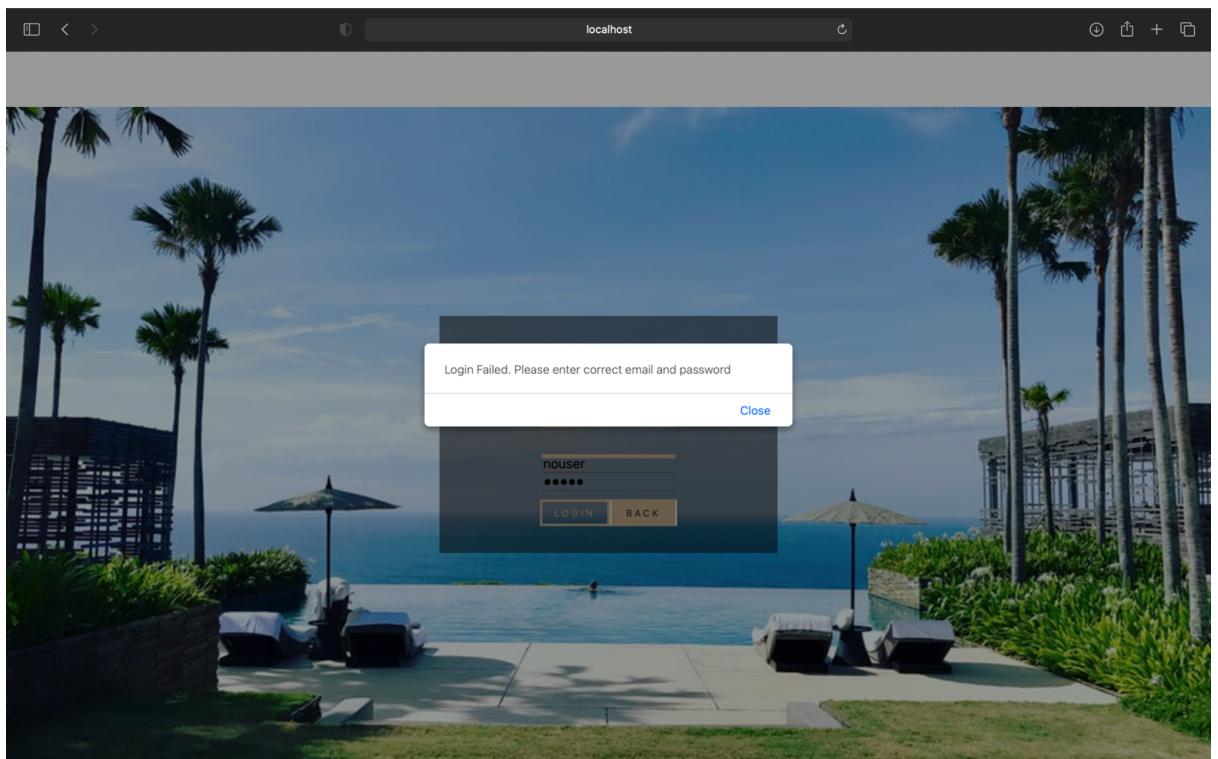
### 6.2. Login Page for Guest

The login page is similar for all users. The same template is used to maintain consistency in the UI design of the application. The login page is connected to the backend and thus, if an invalid combination of email and password is inserted, a window alert appears giving an error message. The 'Back' button will redirect you to the homepage for all if one is directed to the login page by pressing the 'guest' or 'candidate' button. For the employee, which contains the manager, housekeeper and security as the option, pressing the 'Back' button will return the user back to the employee homepage.



### 6.3. Incorrect Login

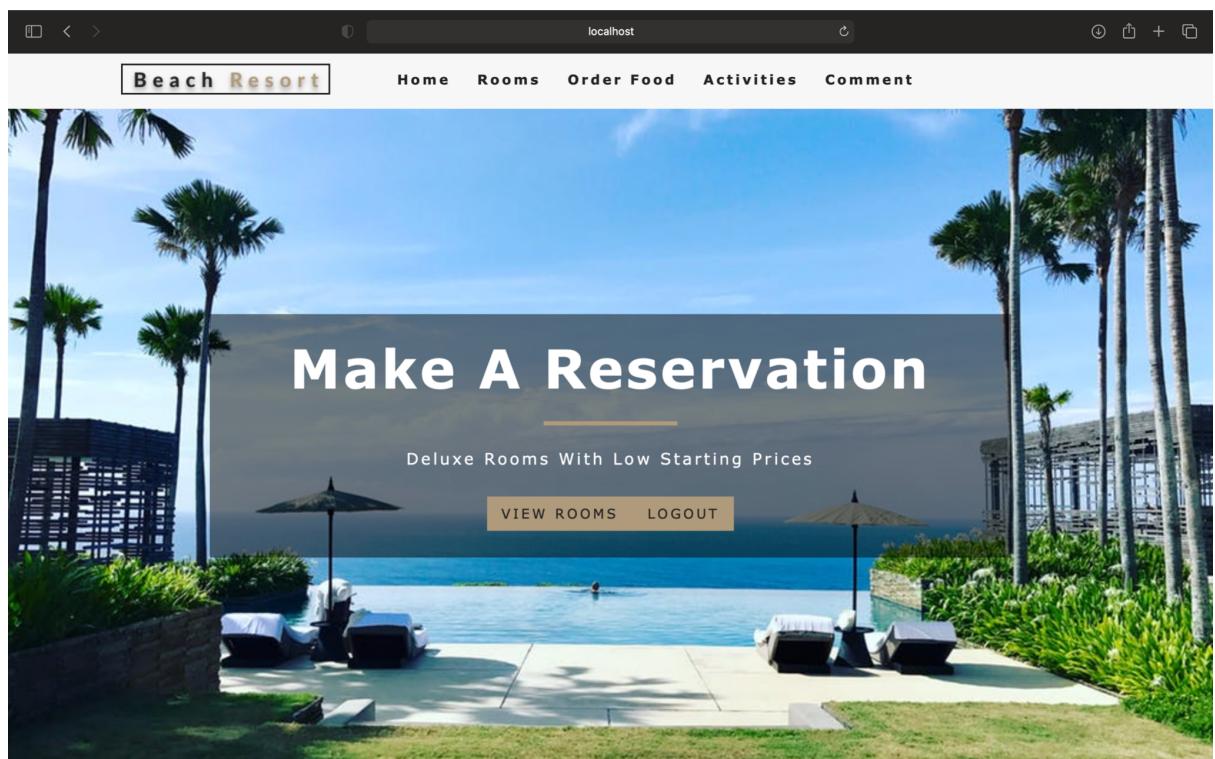
The picture below demonstrates the result after inserting an invalid combination of inputs in the fields of email and password.



## 6.4. Homepage for Guest I

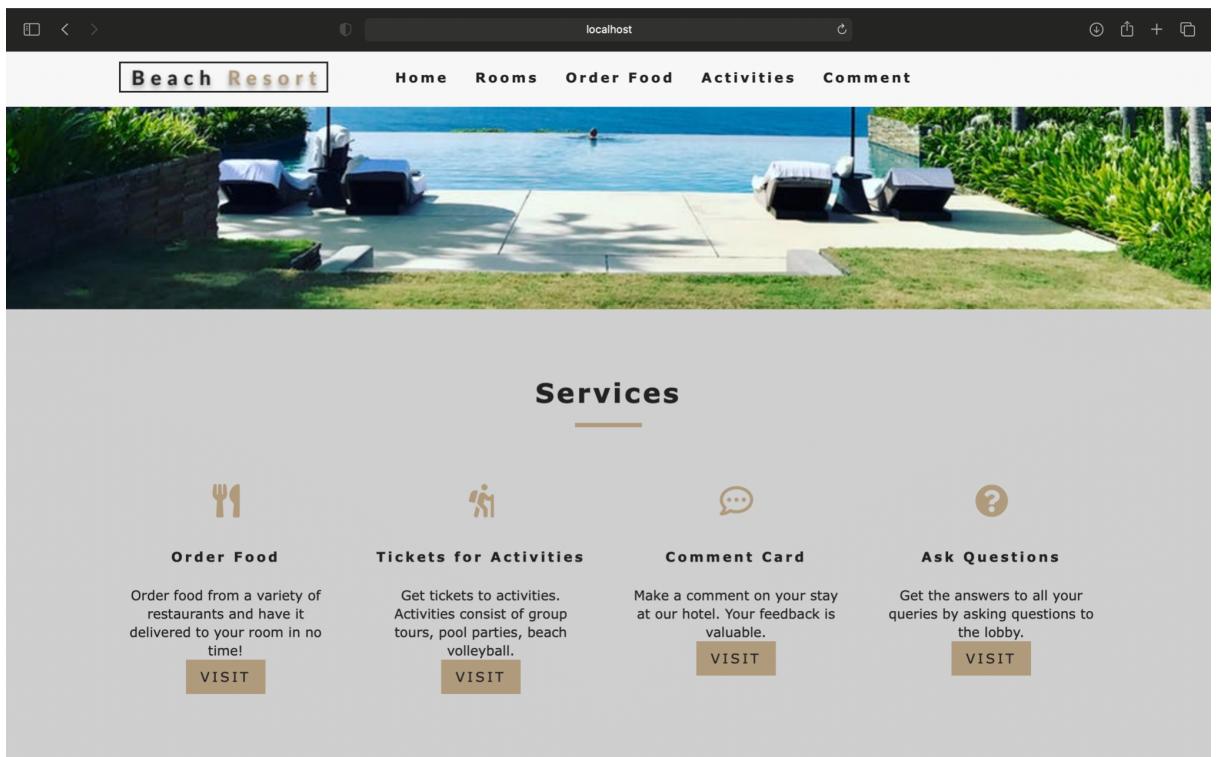
After logging in as a guest, the user has various links to other pages for numerous functionalities. The navigation bar on the top of the page acts as a quick guide for the user to access different options. The ‘Home’ link on the navigation bar is the quick way to return to the homepage for guests, the very page shown in the picture below. The button ‘view rooms’ and link ‘Rooms’ directs the guest to a page where the user can then choose a room of their own choice for reservation. The ‘Order food’ link on the navigation bar directs the guest to a page where the guest can order from database listed restaurants. The guest can also pass comments regarding the stay by clicking on the ‘Comment’ link on the navigation bar.

The badge ‘Beach Resort’ will direct the user back to the homepage for all. Lastly, the button ‘logout’ directs the guest back to homepage for all



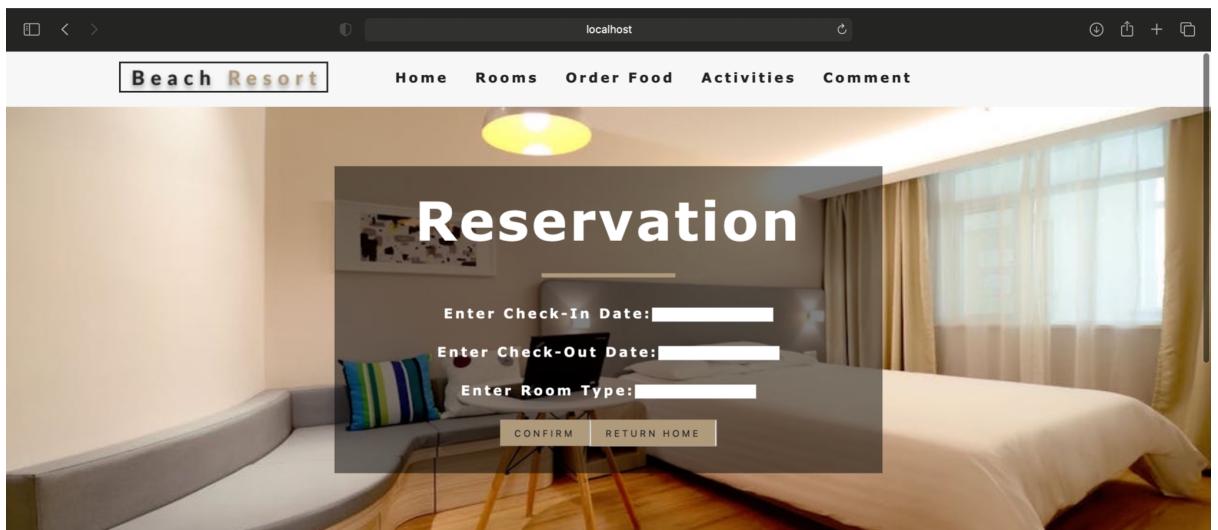
## 6.5. Homepage for Guest II (after scrolling)

After scrolling down on the guest’s homepage, the quest can view services offered. The services provided by the application are to book tickets to different events created by the manager, pass comments, ask questions, and to order food.



## 6.6. Reservation I

After clicking on the ‘view rooms’ button, the guest is directed to the reservation page. The guest has to be specific with date and room type. The date form accepted by the application is 21022021 rather than 21/02/2021. The room types need to be specific as Single Economy, Deluxe, or Double Economy. The string entered is case-sensitive.

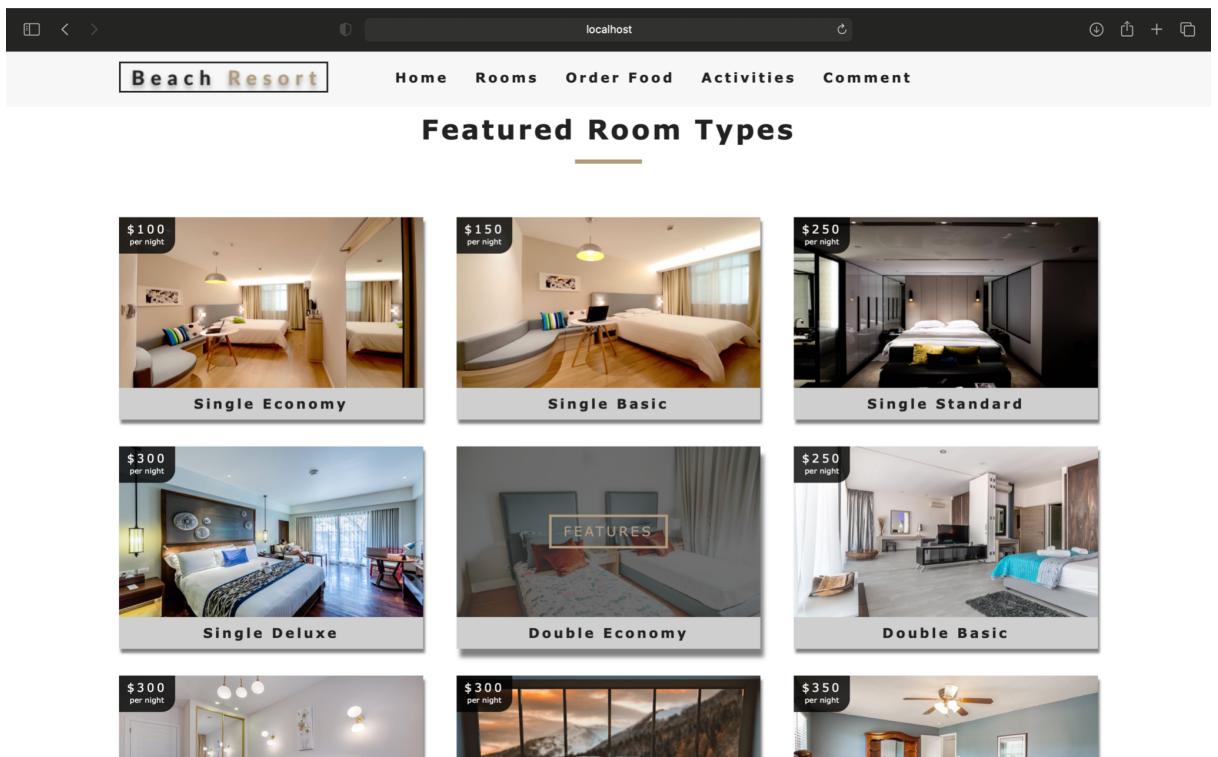


## Featured Room Types



### 6.7. Reservation II (after scrolling)

After scrolling down the reservation page, featured rooms appear. It gives a visual of the specific room type with also relevant details important to the guest, such as price, capacity, etc. As one can see, the navigation bar remains there even after scrolling.



## 6.8. Featured Room Type I - Single Deluxe

In the picture below, details for a Single Deluxe are shown.

The screenshot shows a web browser window with a dark header bar containing icons and the text "localhost". Below the header are three small images: a kitchen area with white cabinets and stainless steel appliances, a bathroom with a white vanity and a walk-in shower, and a living room with blue walls, a wooden desk, and bookshelves. The main content area has a light gray background. It features a large block of placeholder text under the heading "Details". To the right are two columns: "Info" and "Extras". The "Info" column lists "Price : \$300", "Size : 400 SQFT", "Max Capacity : 1 Person", "Pets Allowed", and "Free Breakfast Included". The "Extras" column lists several amenities: "Plush pillows and breathable bed linens", "Complimentary refreshments", "Comfortable beds", "Soft, oversized bath towels", "Adequate safety/security", "Full-sized, pH-balanced toiletries", and "Internet".

Details		
Street art edison bulb gluten-free, tofu try-hard lumbersexual brooklyn tattooed pickled chambray. Actually humblebrag next level, deep v art party wolf tofu direct trade ready-made sustainable hell of banjo. Organic authentic subway tile cliche palo santo, street art XOXO dreamcatcher retro sriracha portland air plant kitsch stumptown. Austin small batch squid gastropub. Pabst pug tumblr gochujang offal retro cloud bread bushwick semiotics before they sold out sartorial literally mlkshk. Vaporware hashtag vice, sartorial before they sold out pok pok health goth trust fund cray.		

Info		
Price : \$300	Size : 400 SQFT	Max Capacity : 1 Person
Pets Allowed		Free Breakfast Included

Extras		
- Plush pillows and breathable bed linens	- Soft, oversized bath towels	- Full-sized, pH-balanced toiletries
- Complimentary refreshments	- Adequate safety/security	- Internet
- Comfortable beds		

## 6.9. Reservation II - Single Economy

In the picture below, details for a Single Deluxe are shown.

**Details**

Street art edison bulb gluten-free, tofu try-hard lumbersexual brooklyn tattooed pickled chambray. Actually humblebrag next level, deep v art party wolf tofu direct trade readymade sustainable hell of banjo. Organic authentic subway tile cliché palo santo, street art XOXO dreamcatcher retro sriracha portland air plant kitsch stumptown. Austin small batch squid gastropub. Pabst pug tumblr gochujang offal retro cloud bread bushwick semiotics before they sold out sartorial literally milkshk. Vaporware hashtag vice, sartorial before they sold out pok pok health goth trust fund cray.

**Info**

Price : \$150  
Size : 250 SQFT  
Max Capacity : 1 Person  
No Pets Allowed

## 6.10. Reservation Complete

When the guest clicks on the ‘reserve room’ button, a window alert appears with a message of successful reservation. An error message will appear if the room type tried to reserve is not available.

**Beach Resort**

Home   Rooms   Order Food   Activities   Comment

**Reservation**

Enter Check-In Date: 16/01/2020

Reservation Complete!

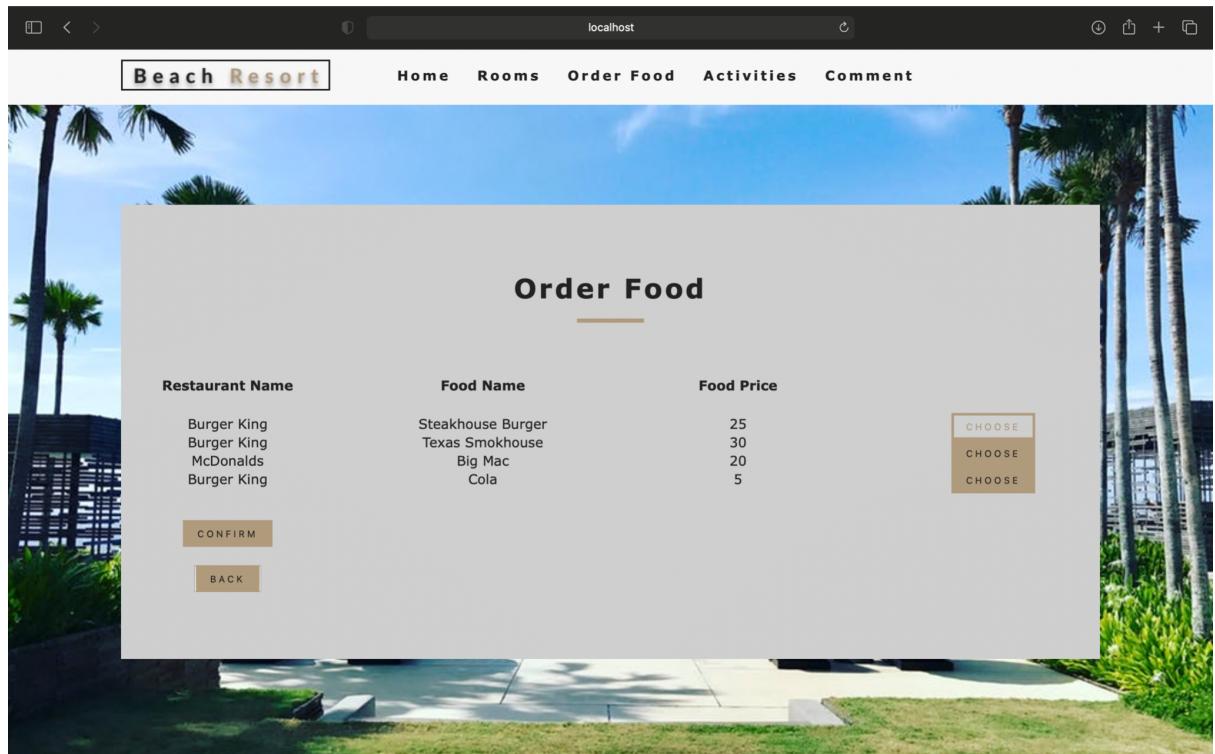
CONFIRM   RETURN HOME   Close

**Featured Room Types**

\$100 per night   \$150 per night   \$250 per night

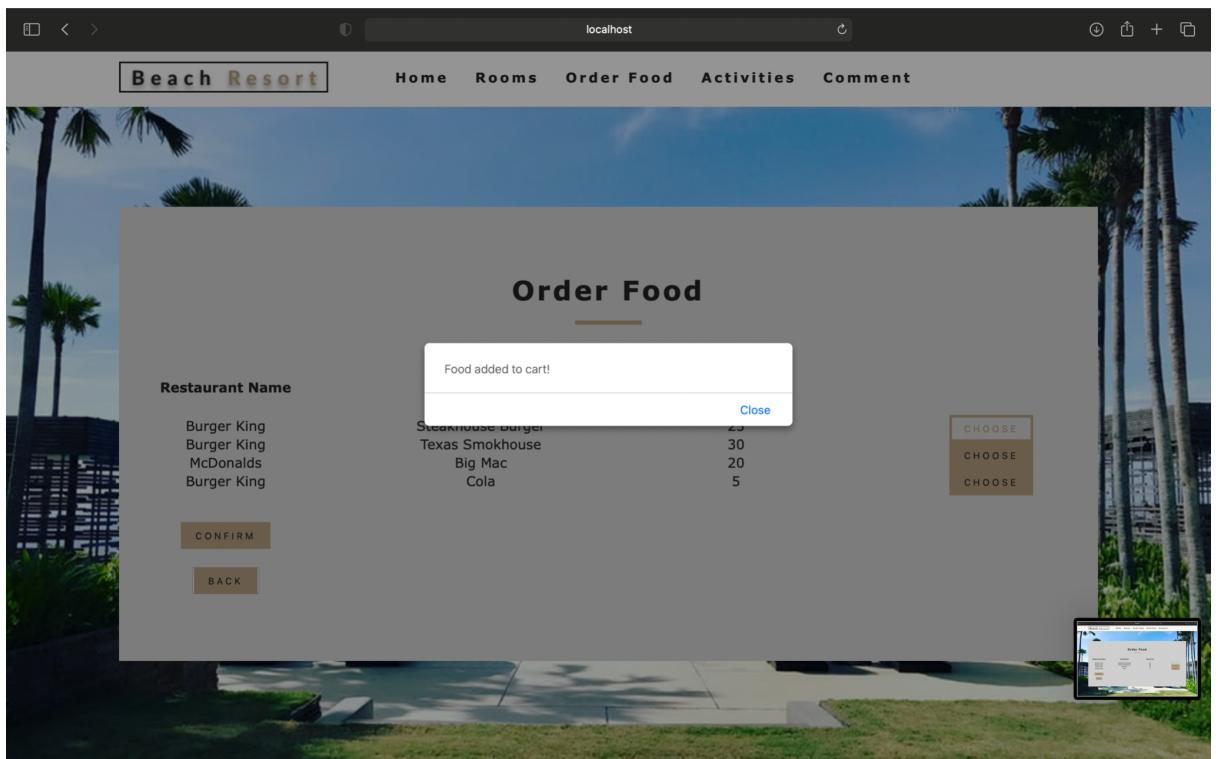
## 6.11. Order Food

When the guest is directed to the order food page, the guest can view the restaurants with the food name available there along with the price of each food item.



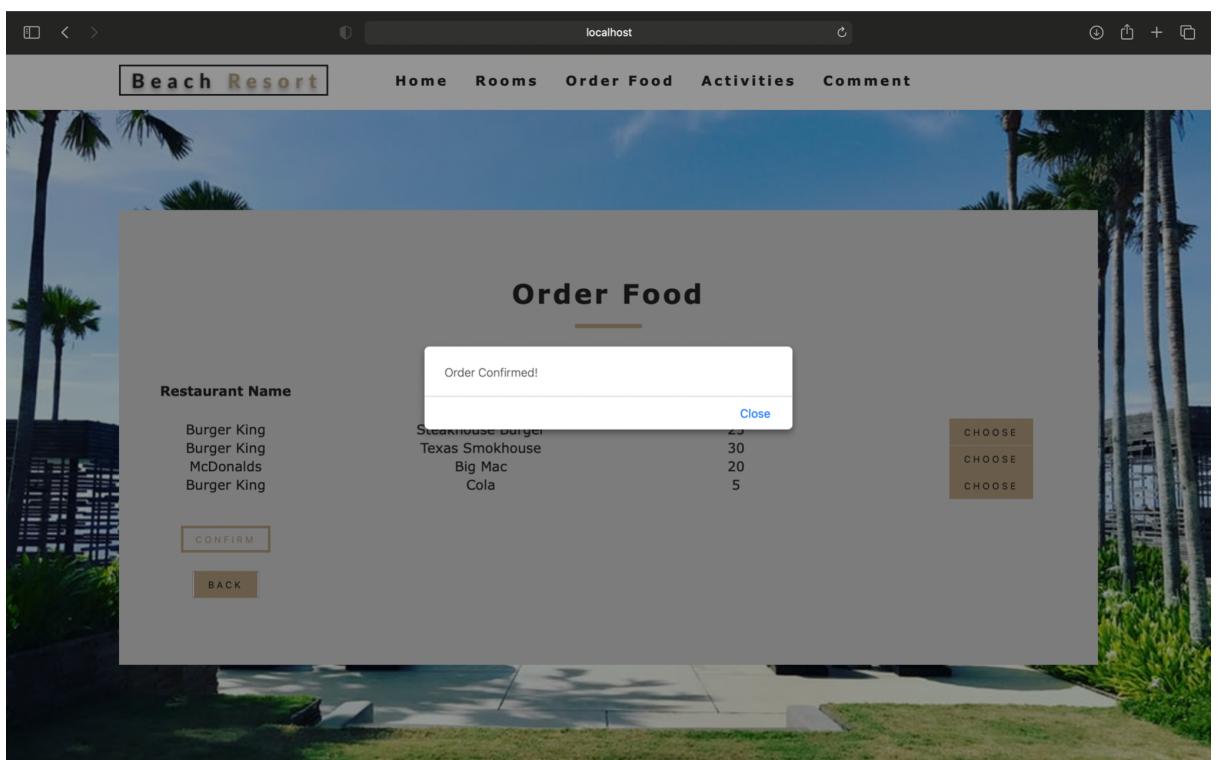
## 6.12. Add Food to Cart

By clicking on the choose 'button', the food order is added to the food cart.



### 6.13. Confirm Food Order

After food is added to the cart, the confirm button confirms the food order. This way, more than one food order can be added to the cart.



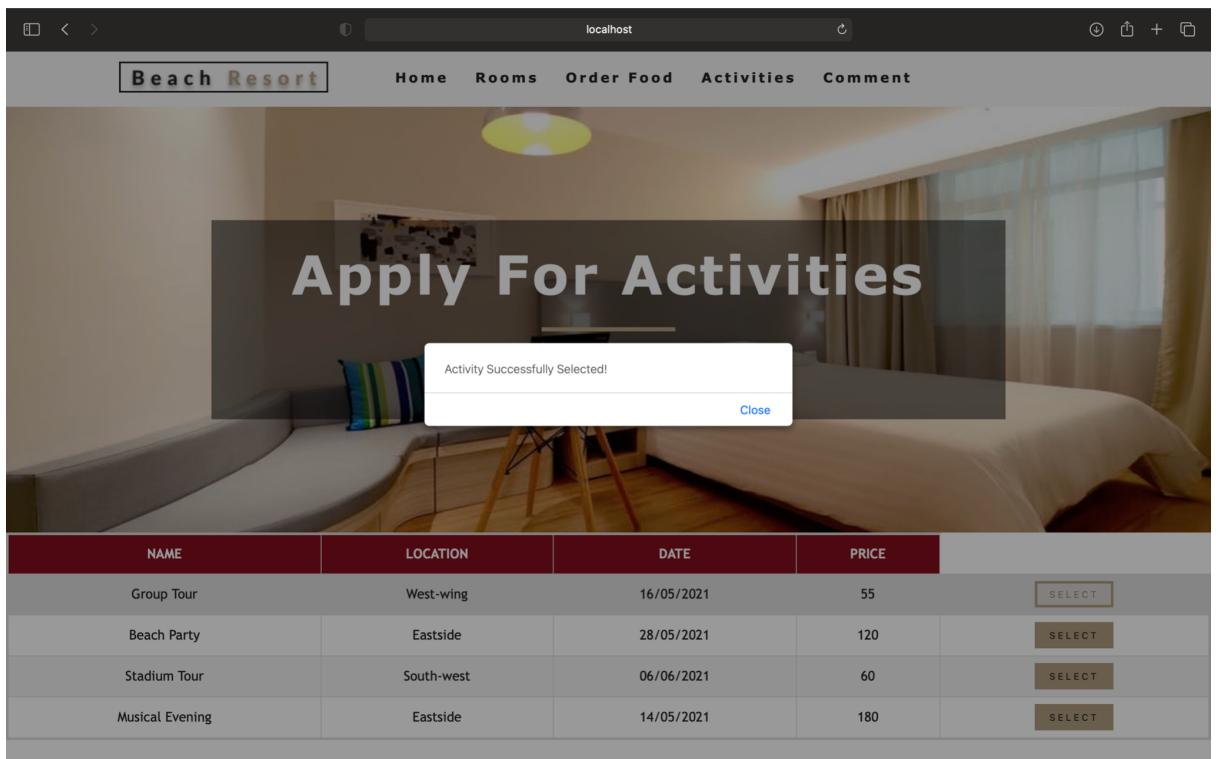
## 6.14. Apply for Activities

The apply for activities page has the list of activities created by the hotel management. It is updated with rendering to enlist all the activities created by the management to date. By selecting an activity, the guest registers himself to the activity.

NAME	LOCATION	DATE	PRICE	
Group Tour	West-wing	16/05/2021	55	<button>SELECT</button>
Beach Party	Eastside	28/05/2021	120	<button>SELECT</button>
Stadium Tour	South-west	06/06/2021	60	<button>SELECT</button>
Musical Evening	Eastside	14/05/2021	180	<button>SELECT</button>

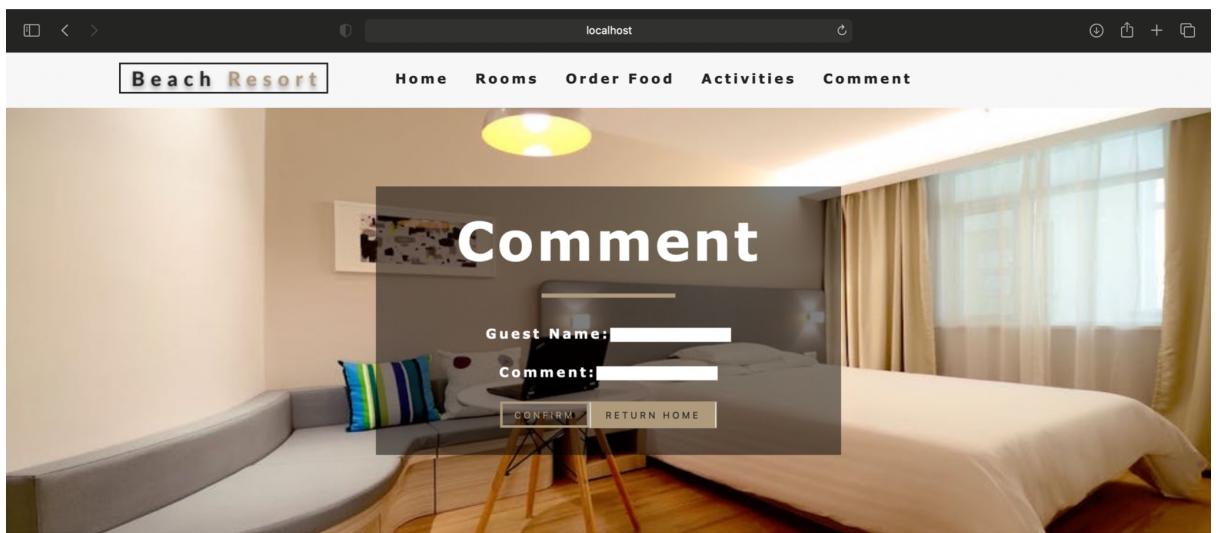
## 6.15. Choose an Activity

The window alert message after registering to an activity.



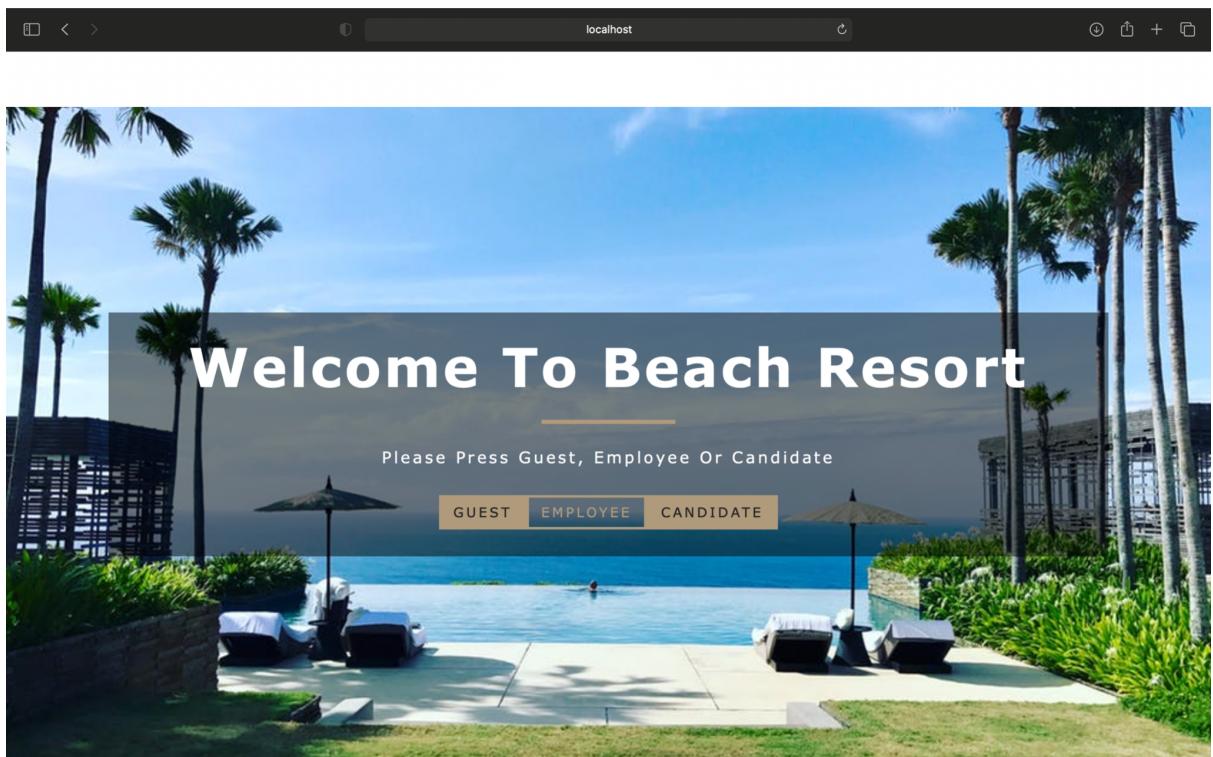
## 6.16. Comment Card

The comment card allows the guest to make comments about their stay at the hotel. The guest has to enter their full name and their comment in the text fields. By clicking the ‘confirm’ button, the comment is sent to the management as feedback.

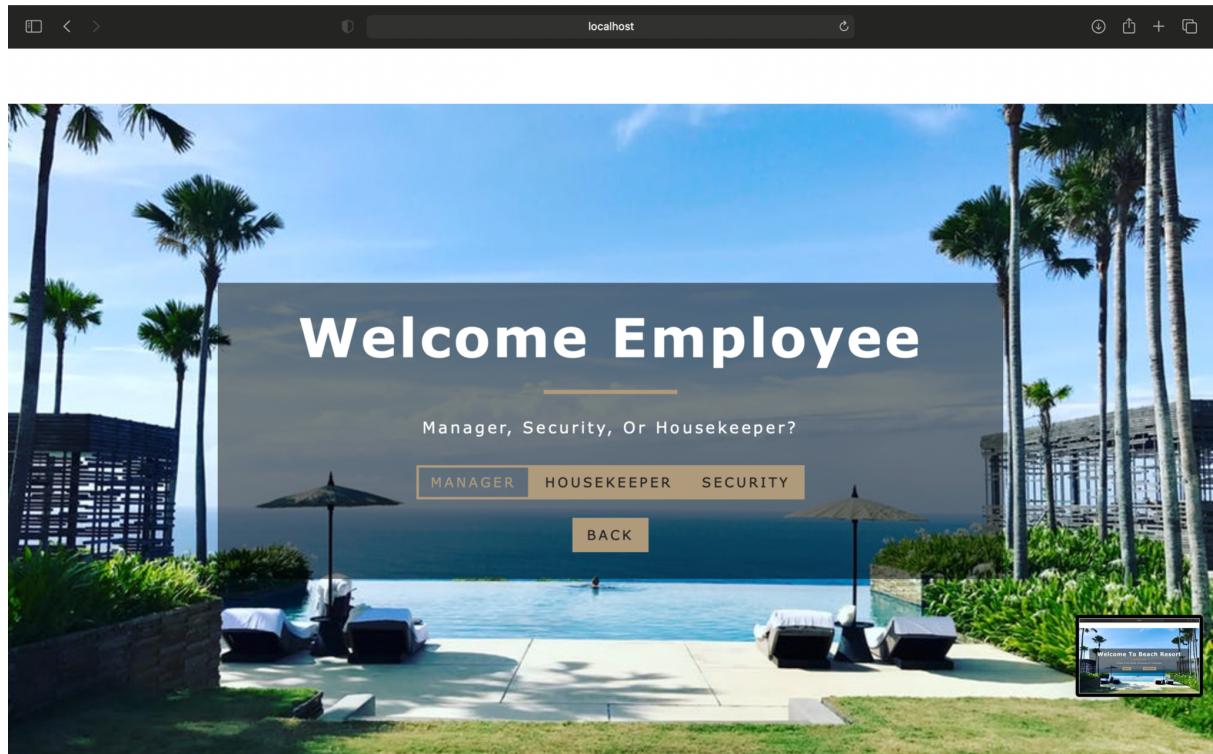


## 6.17. Employee Login

As mentioned above, the employee page will have three buttons that will link each type of employee to their respective login pages, and from there they can administer the management system.

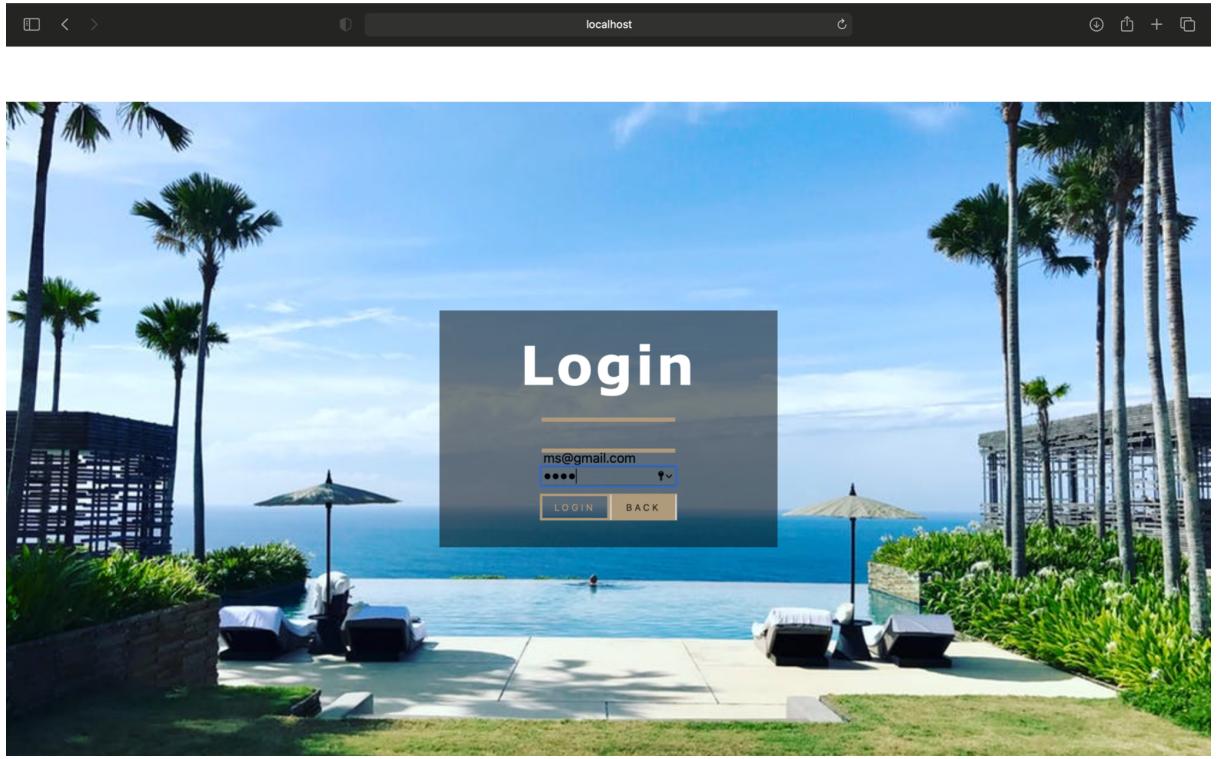


## 6.18. Choose from Manager, Housekeeper and Security



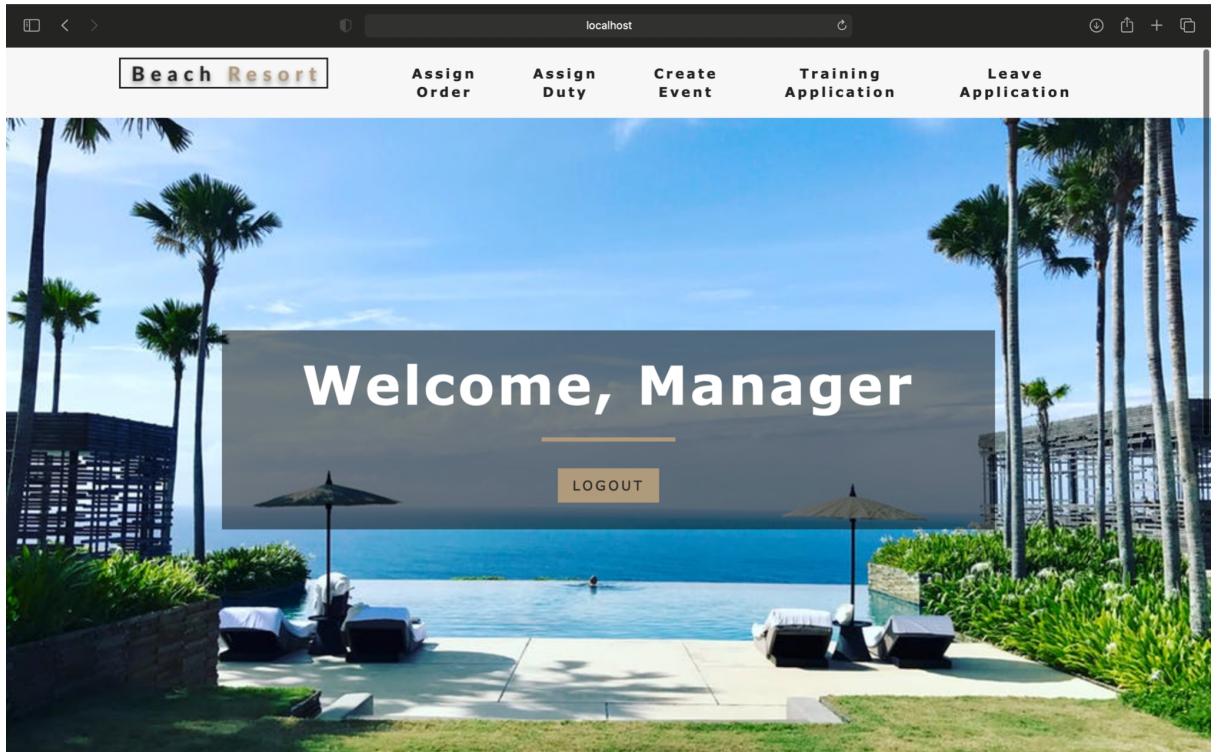
## 6.19. Manager Login

Login page for the manager. As one can observe, the template for the login is the same for all login pages across the application.



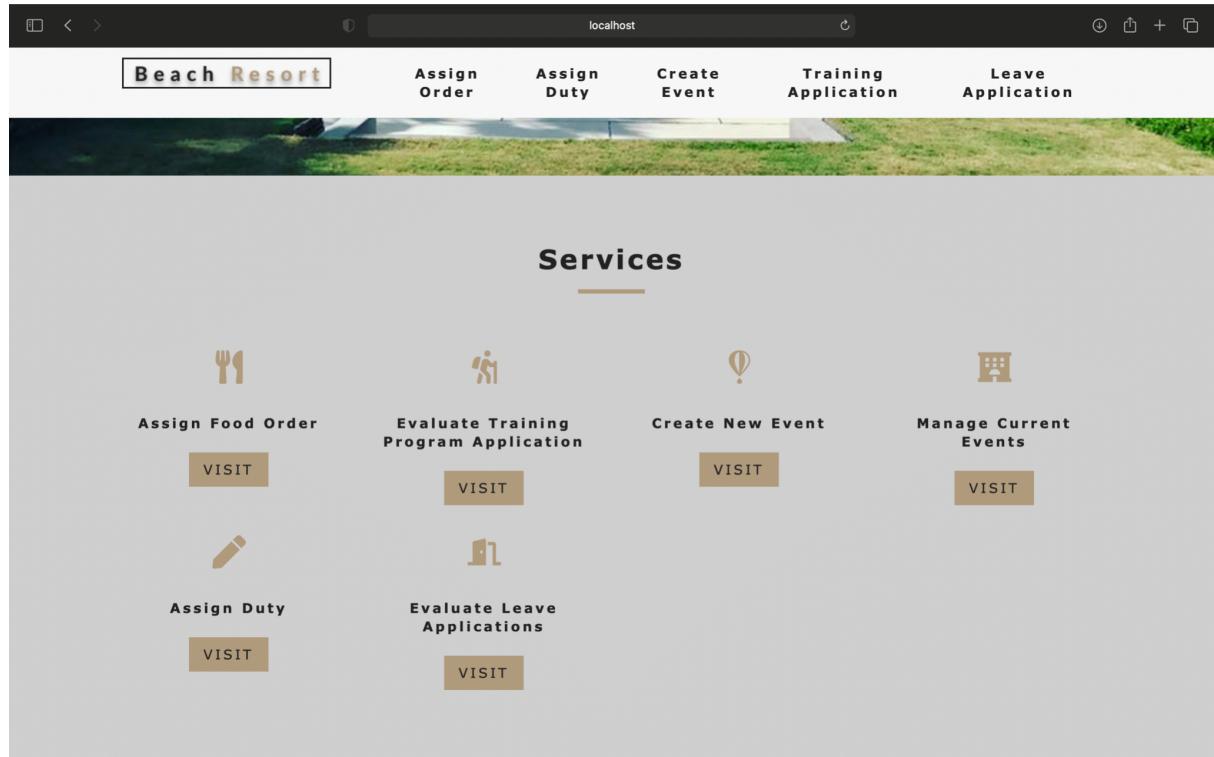
## 6.20. Manager Homepage I

When logged in as manager, the user has numerous options to choose from. The manager is the most defined user in the application. The navigation bar provides an excellent introduction to the general options available to the manager.



## 6.21. Manager Homepage II (after scrolling)

After scrolling down, the manager can also see the services with guiding badges to assist in understanding of the service.



## 6.22. Assign Food Order to Housekeeper

After clicking on ‘assign order’ from the navigation bar or the services section, the manager is directed to the assign order page. The ‘Assign Order’ page shows the available housekeepers to assign the task of delivering the food from the restaurant to the guest’s room. Only one order can be assigned to a housekeeper until the housekeeper has not delivered it from his side. The ‘assign’ button assigns the food order to the housekeeper.

The screenshot shows a web browser window for 'localhost' with the title 'Beach Resort'. The main content is titled 'Assign Order' with a 'RETURN HOME' button. Below this is a table with four columns: HOUSEKEEPER, ROOM, ORDER, and DATE. The table contains the following data:

HOUSEKEEPER	ROOM	ORDER	DATE
HK3	W104	10	16/05/2021
Micheal Scott	E702	12	28/05/2021
HK3	E416	16	06/06/2021
Micheal Scott	W112	21	14/05/2021

Each row has an 'ASSIGN' button on the right. A small inset image in the bottom right corner shows a close-up of the 'ASSIGN' button.

## 6.23. Assign Order Confirmation

Window alert after food is assigned.

The screenshot shows the same 'Assign Order' page as above, but with a modal dialog box in the center. The dialog box contains the text 'OrderAssigned!' and a 'Close' button. The rest of the page, including the table and buttons, remains visible.

## 6.24. Assign Duty - Security Walk

After clicking on ‘assign duty’ from the navigation bar or the services section, the manager is directed to the assign duty page. The ‘Assign Duty’ page shows the available security staff to assign the task of walks around certain buildings. Only one walk can be assigned to a security staff. The ‘assign’ button assigns the food order to the housekeeper.

The screenshot shows a web browser window titled 'localhost'. The top navigation bar includes a logo for 'Beach Resort' and links for 'Assign Order', 'Assign Duty', 'Create Event', 'Training Application', and 'Leave Application'. The main content area features a photograph of a modern hotel room with a bed, a sofa, and a desk. Overlaid on the image is a dark rectangular box containing the text 'Assign Duty' in large white letters and a smaller 'RETURN HOME' button below it. Below the image, the text 'Assign Security Walk' is displayed. A table below the text lists three security staff members with their details: SS3 (Building B12, Location West-wing, Date 22/10/2021), SS2 (Building E2, Location West-wing, Date 08/01/2021), and ms (Building M16, Location South, Date 31/01/2021). Each row in the table has an 'ASSIGN' button in the last column.

SECURITYSTAFF	BUILDING	LOCATION	DATE	
SS3	B12	West-wing	22/10/2021	<button>ASSIGN</button>
SS2	E2	West-wing	08/01/2021	<button>ASSIGN</button>
ms	M16	South	31/01/2021	<button>ASSIGN</button>

## 6.25. Create Event

The create event page allows the manager to create events for the guests. It takes the following data and sends it to the page for guests - where they can apply for the created event.

**Create Event**

Enter Event Name

event name

Enter Event Location

event location

Enter Event Start Date

event start date

Enter Event End Date

event end date

Enter Event Quota

event quota

Enter Event Price

event price

**CONFIRM** **BACK**

## 6.26. Evaluate Training Application

This page allows the manager to accept or reject applications sent by the employees for training. It depicts the employees who had applied for a training program.

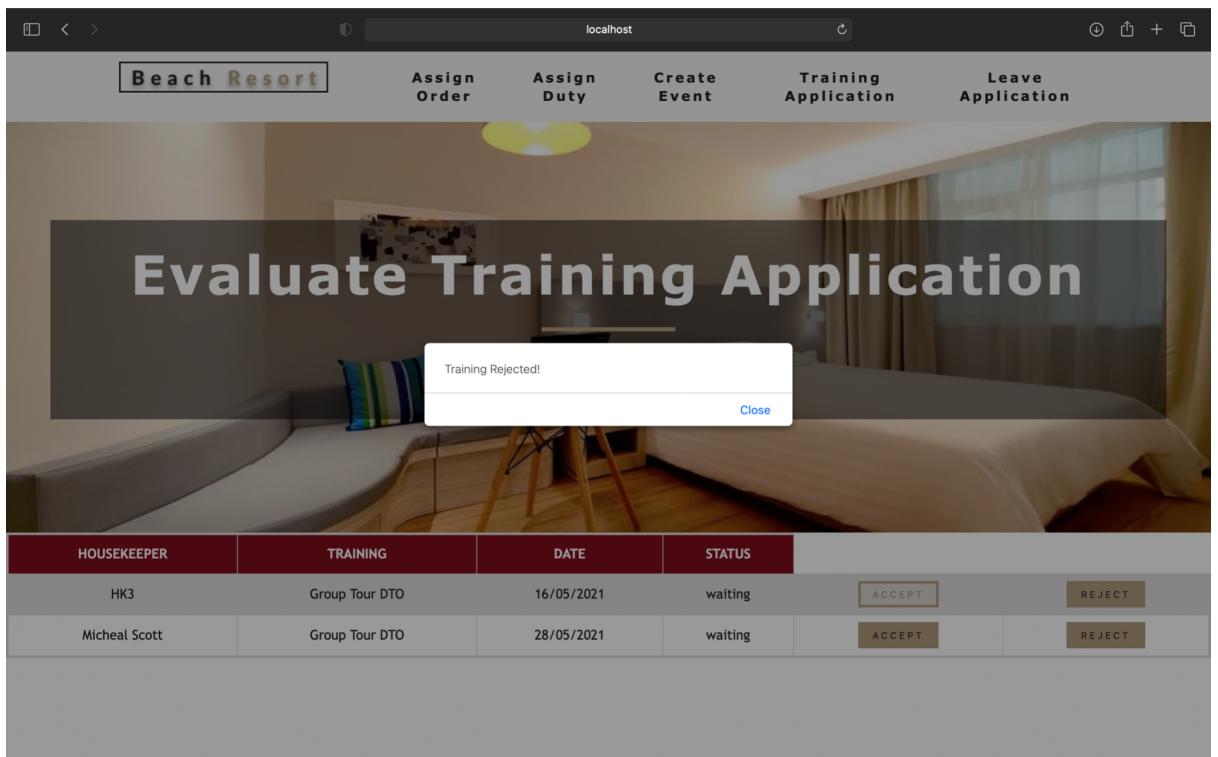
**Evaluate Training Application**

**RETURN HOME**

HOUSEKEEPER	TRAINING	DATE	STATUS	ACCEPT	REJECT
HK3	Group Tour DTO	16/05/2021	waiting	<b>ACCEPT</b>	<b>REJECT</b>
Micheal Scott	Group Tour DTO	28/05/2021	waiting	<b>ACCEPT</b>	<b>REJECT</b>

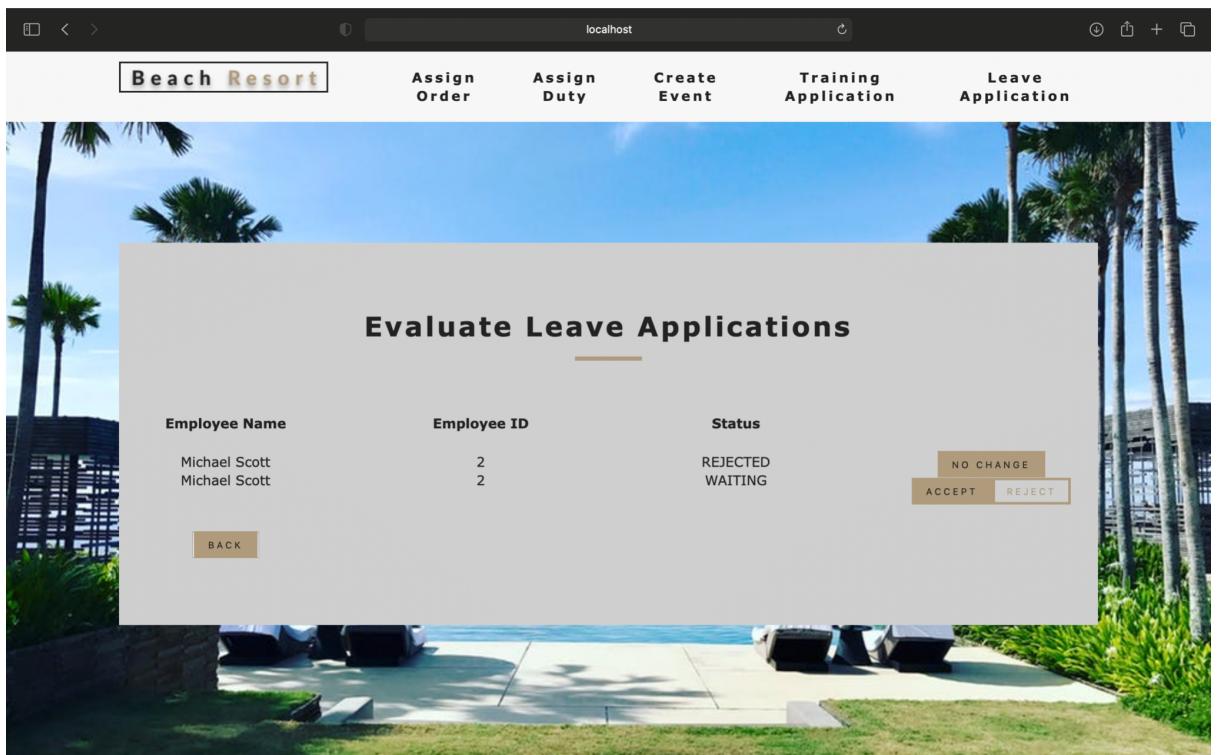
## 6.27. Reject Training Application

Window alert for rejecting an application.



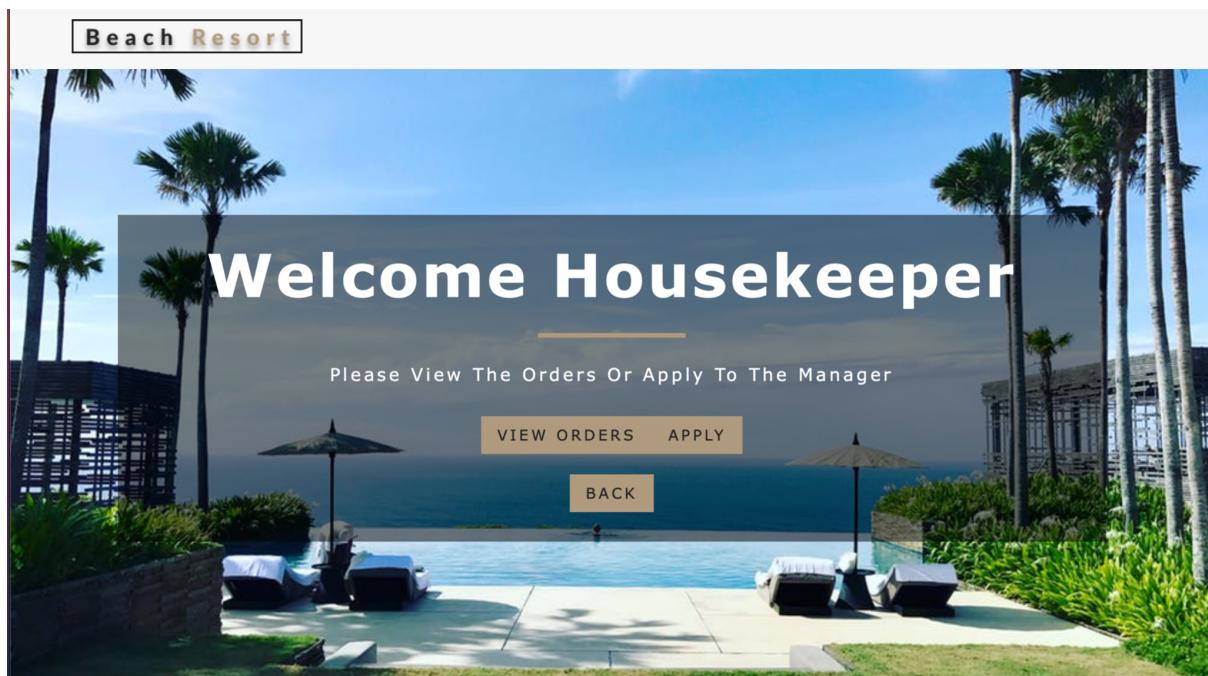
## 6.28. Evaluate Leave Application

This page allows the manager to accept or reject applications sent by the employees for leave. It depicts the employees who had applied for a leave. The accept and reject button determine the status of the applicant. If the application has already been accepted or rejected, no further change can be made



## 6.29. Homepage for Housekeeper

This page appears when the housekeeper logs in through the login page after clicking on the ‘Housekeeper’ button. This page will have three buttons: Orders, Apply, and Back. The ‘orders’ button will lead the housekeeper to view the food order assigned to him/her. Secondly, the ‘apply’ button will direct the housekeeper to the applications’ page where employees can apply for a leave or a training program. The ‘Back’ button will lead the user back to the employee homepage.



## 6.30. Assigned Orders

The assigned order page will depict the order assigned to the housekeeper. The details of the order such as guest name, order number, building number, room number will be displayed. When the housekeeper presses the ‘delivered’ button, the order is removed and the housekeeper is available to be assigned again. The picture below shows the assign order page.

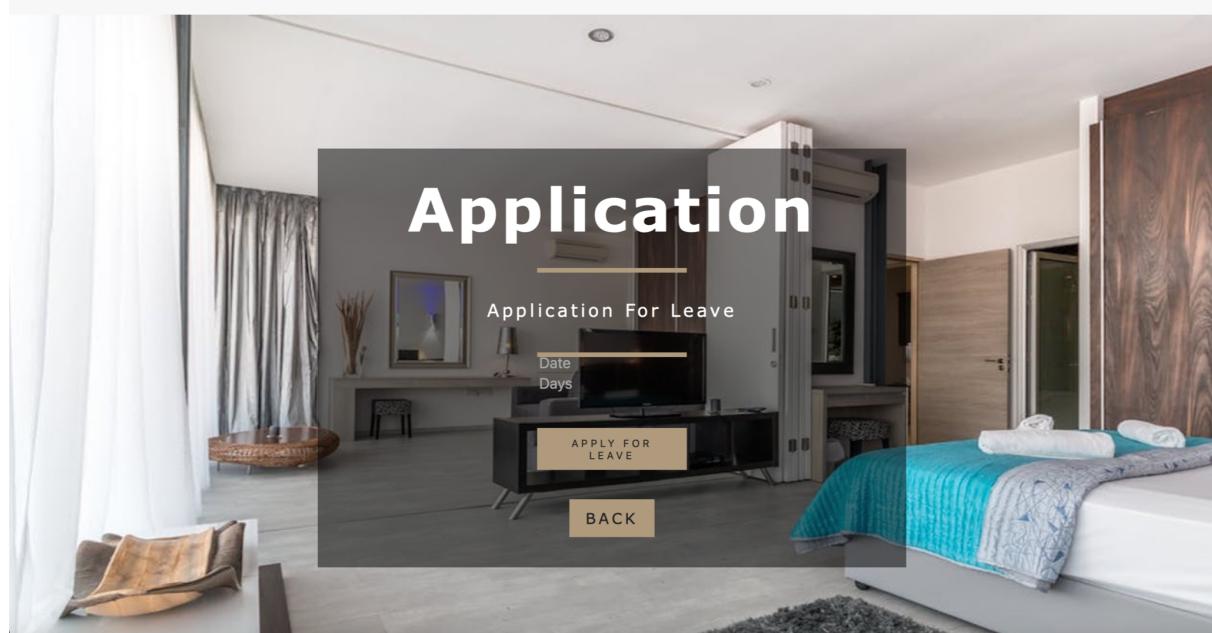
### Beach Resort



### 6.31. Apply Page for Employees

This page will be the same for housekeeper and security. The two buttons, leave and training, will redirect the employees to the training application and leave application pages. The leave application page will require the employee to enter the date from when the leave is required and the duration of the leave. Lastly, the application for training will have relevant training programs for which employees can apply to. The images below hold reference to the leave application page and training program page.

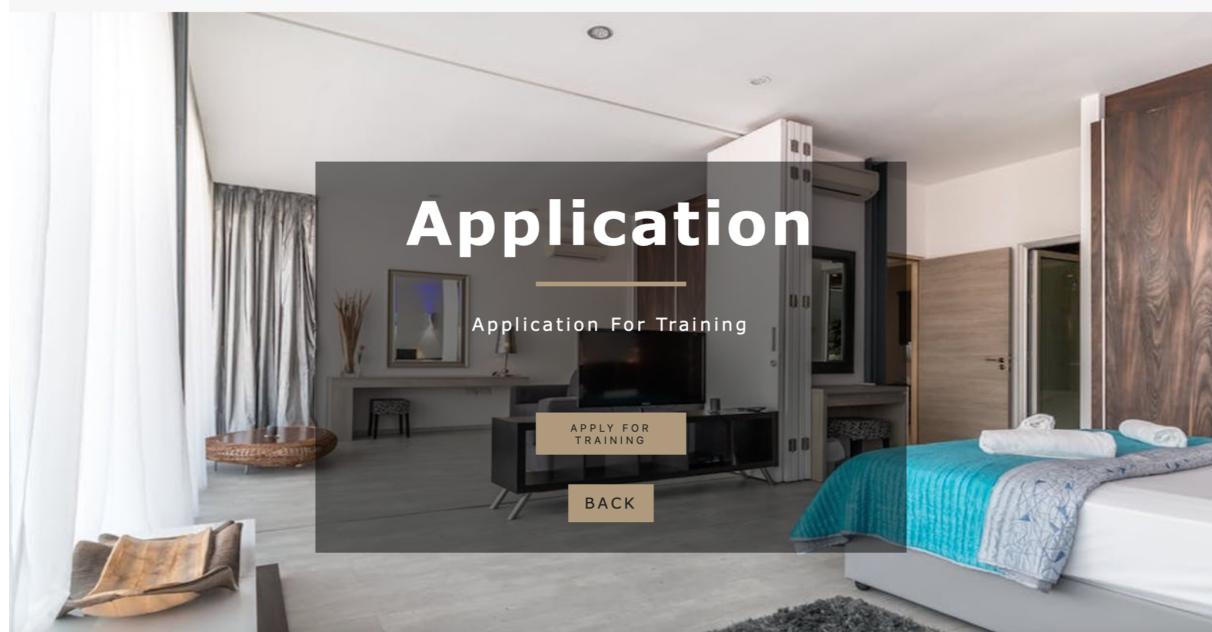
**Beach Resort**



### 6.32. Apply Training for Employees

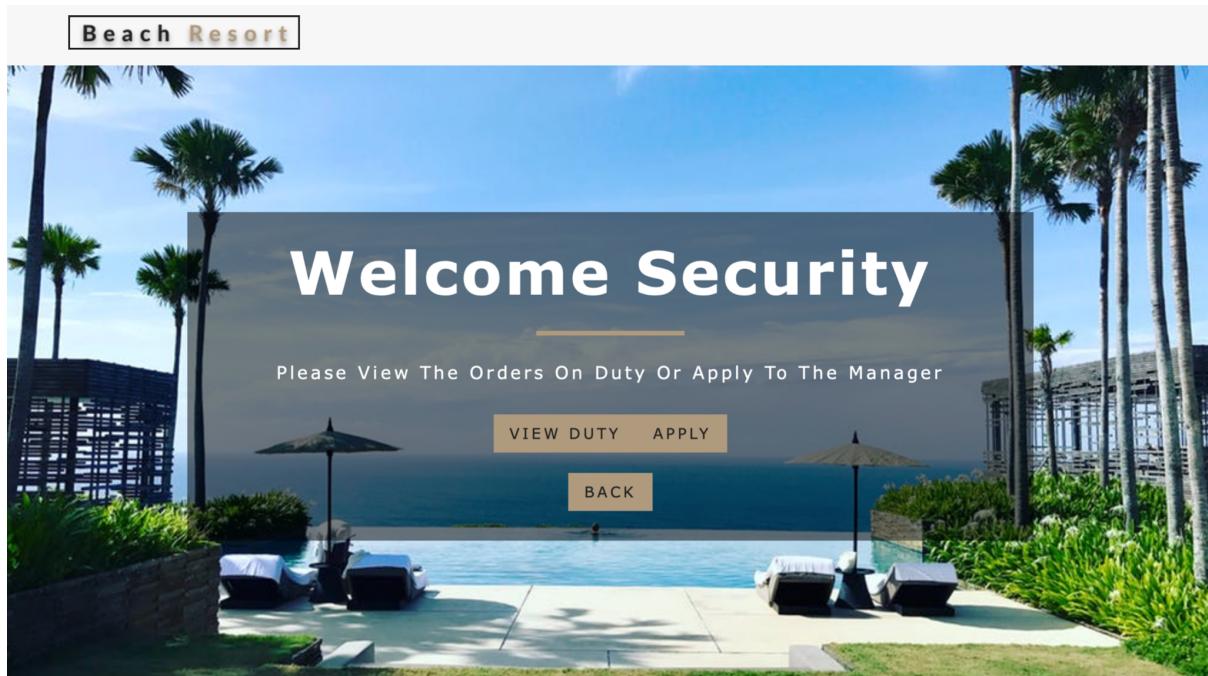
The list of training programs will appear when they are listed. The employee will select one and then press the 'apply for training' button.

**Beach Resort**



### **6.33. Home page for Security**

This page will have three buttons: view duty, apply, and back. Apply and Back are the same as Housekeeper. However, the ‘view duty’ button directs the security staff to the page where the security staff member can see the duties assigned to him/her.



### **6.34. Home page for Security**

The view duty page allows the security staff member to view the duties assigned to him/her with relevant details. The image below is an example of duties assigned to a security staff member.

## **Beach Resort**

<b>Building</b>	<b>Start Date</b>	<b>End Date</b>	<b>Manager assigned by</b>
West-1	1231231231231	123123	Michael
West-1	1231231231231	1231231231232	Michael

**BACK**