

Project Title : Hotel Booking System

Table of Contents

Sectio n No	Content	Page No
1	Introduction	3
2	Project Features	4
3	ER/EER Diagram	5
4	Schema Diagram	6
5	Frontend Development	7-15
6	Backend Development	15-31
7	Source Code Repository	32
8	Conclusion	32
9	References	32

Introduction

In today's world, when people travel, they need an easy way to book hotels. That's where a hotel booking system comes in handy. This system helps hotels manage their bookings efficiently, making it simple for guests to reserve rooms. Whether you're a hotel owner or someone planning a trip, having a reliable and user-friendly hotel booking system makes the whole process smoother. This project focuses on creating a straightforward and effective system to make booking hotels a breeze for everyone involved.

HTML, PHP, BootStrap and MYSQL is the language that the project is based on. The project uses PHP to connect to the database and write sql queries and retrieve, then uses HTML and Bootstrap to show those retrieve data more elegantly. The development of Hotel Booking System guided by keeping following features in mind

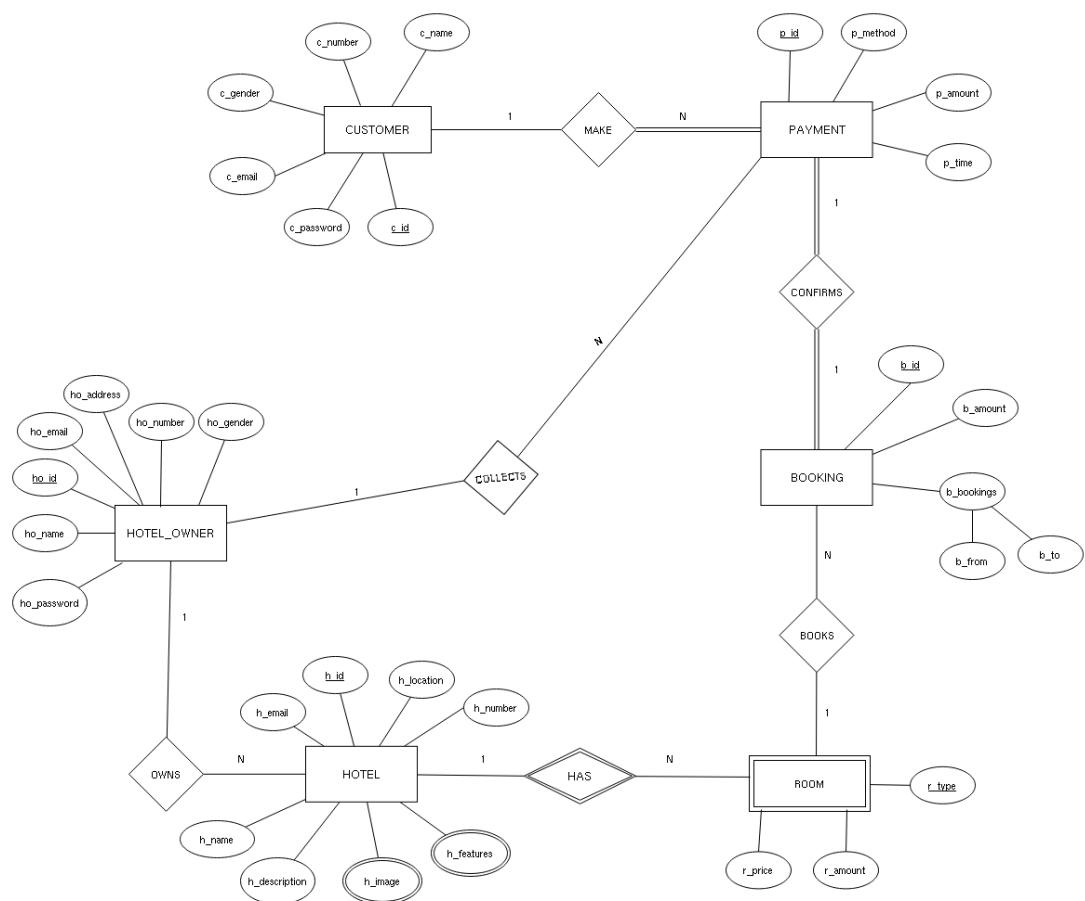
Enhancing User Experience : This project uses HTML and PHP to minimize the delay to load pages. Also this website has a search bar, using the search bar users can find the hotel based on their necessity. Not to mention, for users convenience there is an All Hotel button, which shows all the Hotels that are registered on this website, so that a user can see our vast collection of hotels easily. This website also has a navigation bar on the top of the page, which makes signup-login and overall navigation much easier.

Data Management and Security : To keep this website easier to navigate, the login and signup panel for both User and Hotel Owner are separated. Both user and owner login's are protected by password. If email and password don't match with the databases, it will show 'Invalid Username or Password'. Not to mention, no one can skip the mandatory fields of the signup process, if anyone misses any of the fields or puts wrong data in it, it will show an error. Using PHP and sql the website gets real time data from the database, which makes the hotel room booking more convenient as the user can see which rooms are booked on that particular day and based on that he can place an order. This website also provides options to update password, hotel info and more.

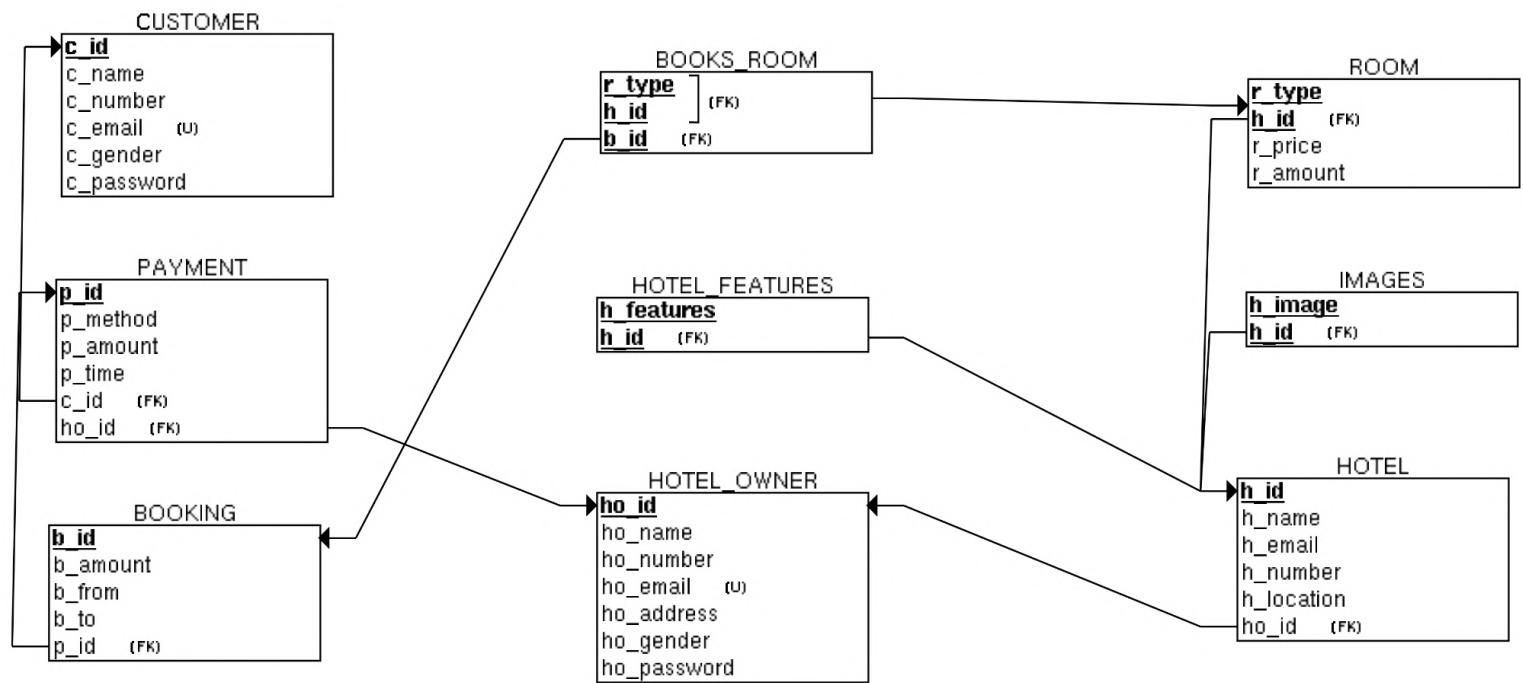
Project Features

1. User friendly interface to see all hotels and their info
2. Login and signup using own mail and password
3. User dashboard to show the reservation
4. Owner dashboard to maintain his hotel
5. Updating password after login
6. Easy to use interface to book hotel room
7. Session handling for keeping track of order and hotel
8. Hotel Registration with multiple room
9. User and Owner Profile
10. Hotel search based on Name, Location and Features

ER/EER Diagram



Schema Diagram

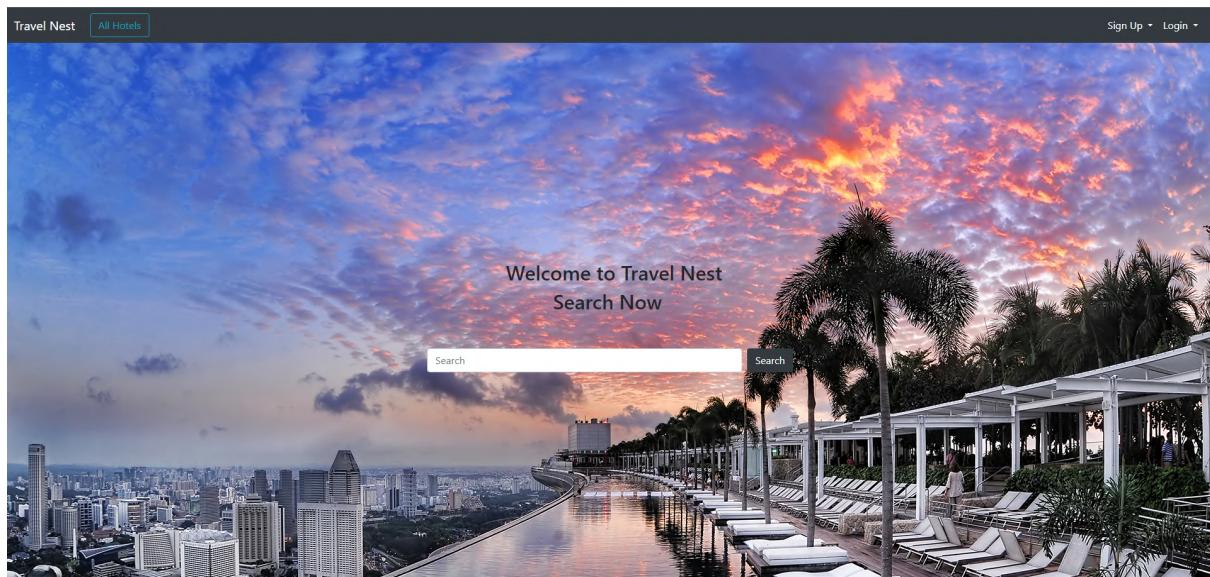


Frontend Development

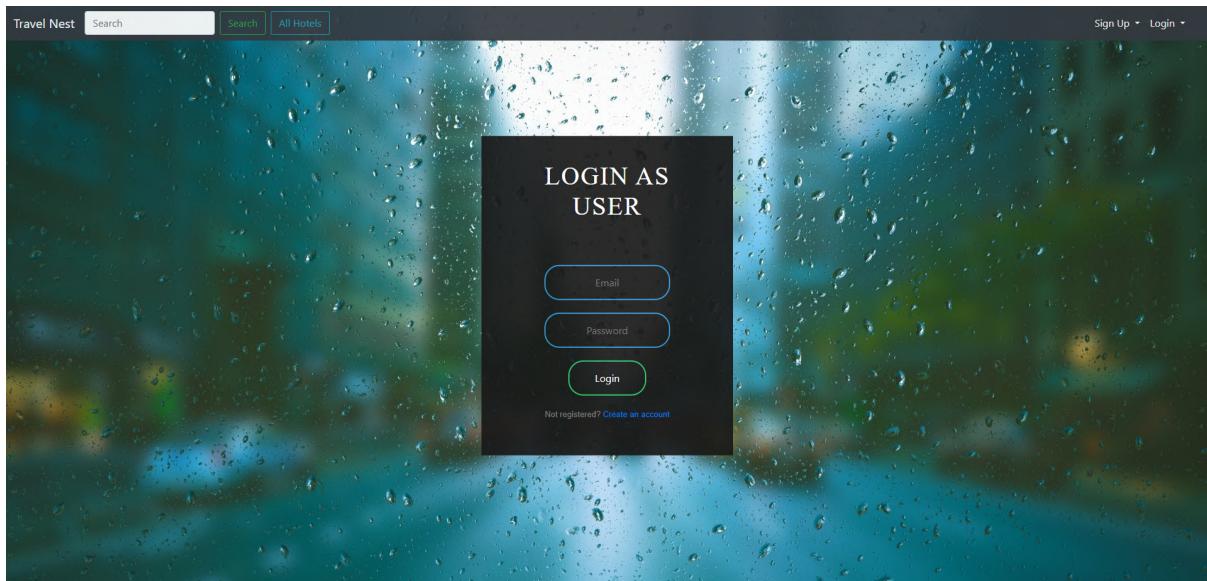
Global

This is what **everyone can see** without signing into the website

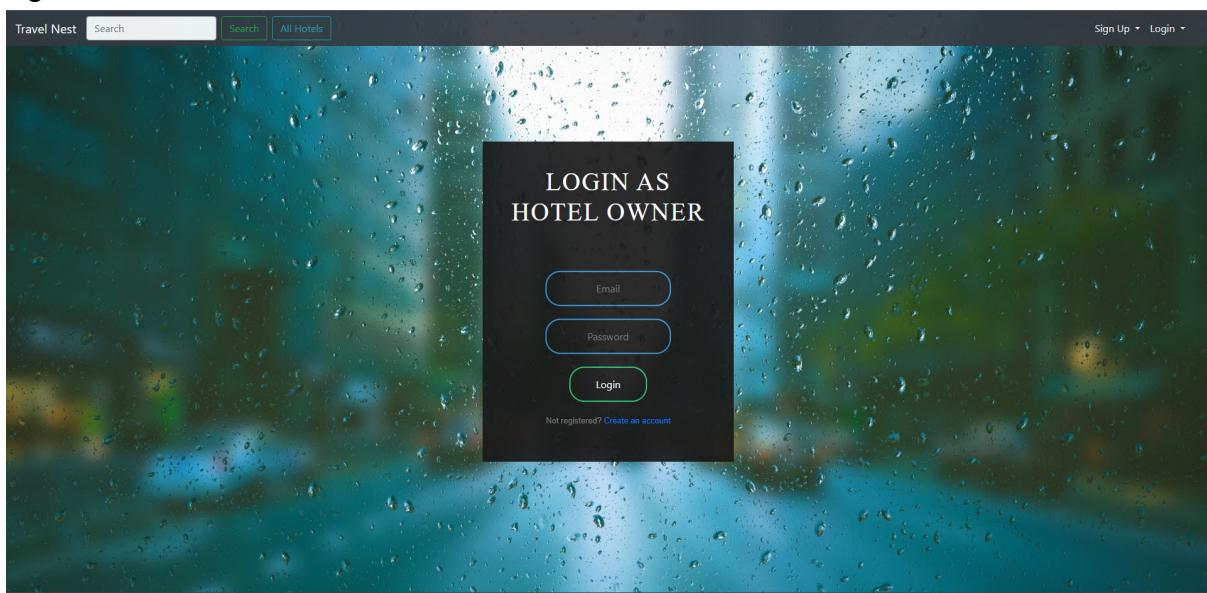
This is the **Home Page** when a viewer visits our website. From this page the user can search Hotels or view a list of All Hotels available on our website. Also they can login, Sign up as User or as Hotel Owner. This is implemented using HTML, CSS, Bootstrap and can be found in the Global folder of the project source code.



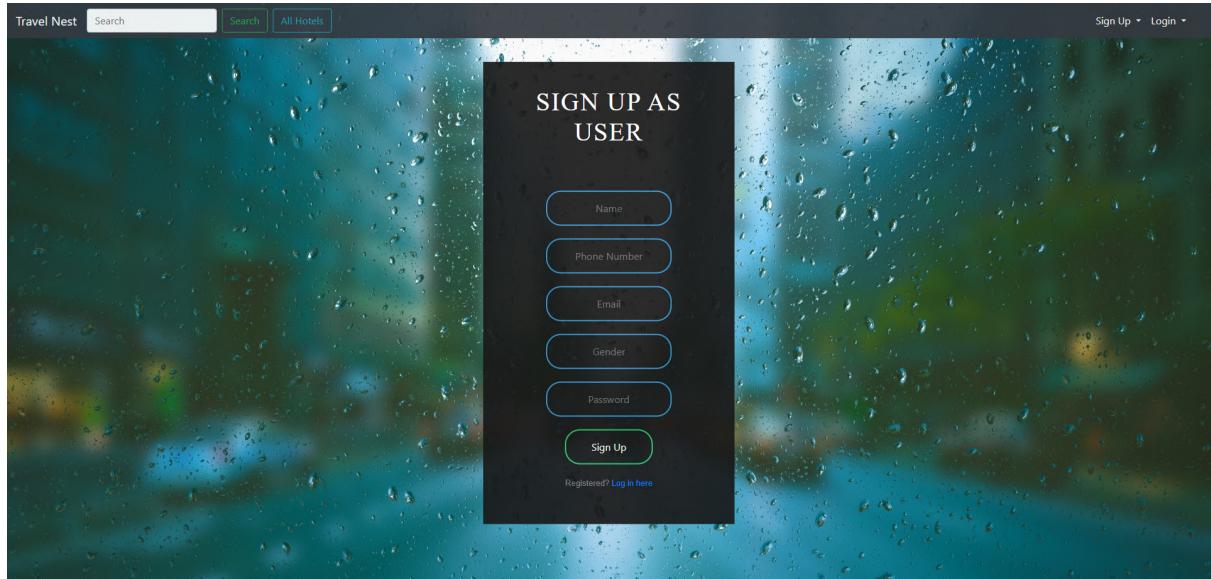
Login page of a user. Here also, the Navigation bar is noticeable and using the ‘Create an Account’ button users can redirect to the signup page if they want to.



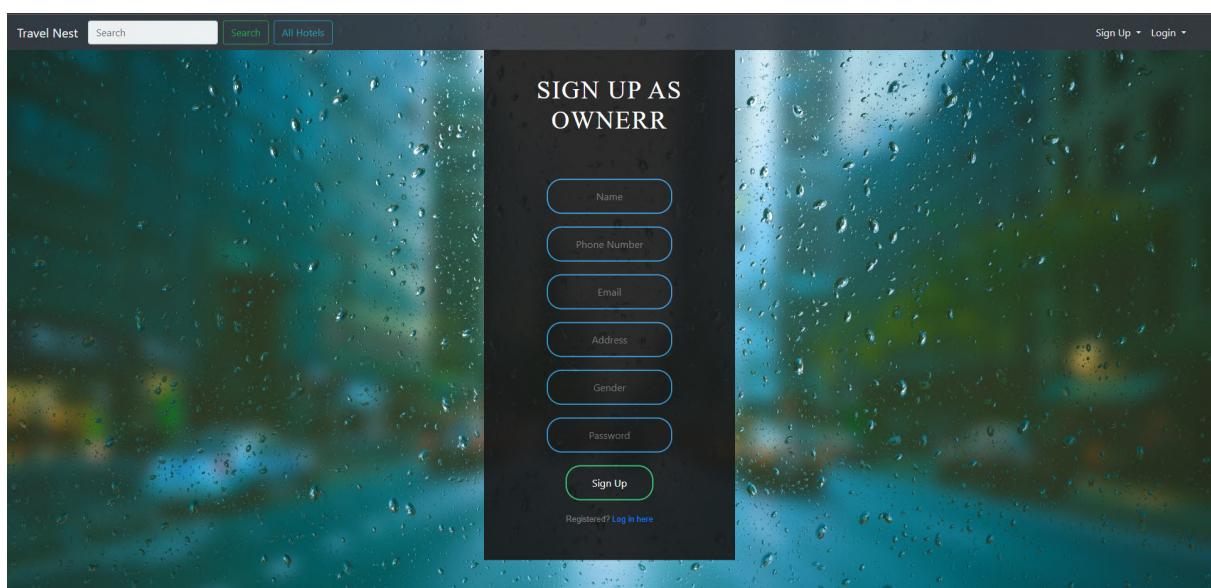
Login Page of a Hotel Owner. It also provides the same feature set as an User login.



Sign Up as a User. Here a user can sign up to the website filling up this form with a unique name, email, phone number, gender and password. If everything is filled up perfectly, it will return a ‘sign up successful’ message. Otherwise it will show, ‘All fields are required’.



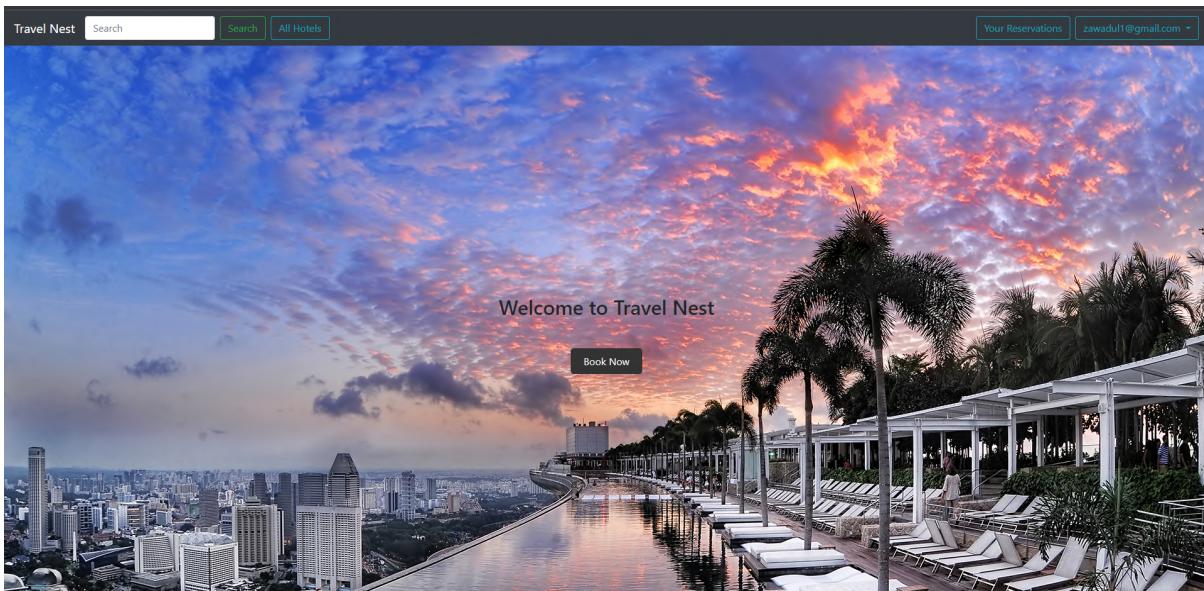
Sign Up as Owner. Same as the User. Using this a hotel owner can sign up to the website.



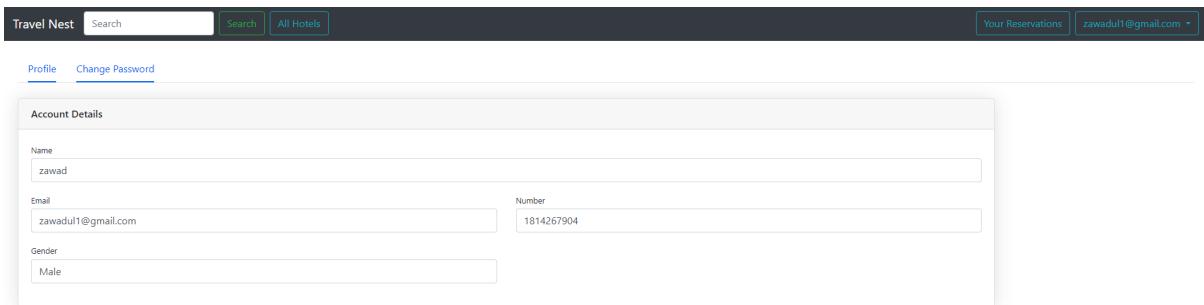
User

This is what a **user will see** after login

Logged in As User. Here it is logged in with the mail 'zawadul1@gmail.com' which can be noticed in the top right dropdown box of the navbar. Using this dropdown box users also can log out or go to the user profile. There is also a 'Your Reservation' button which provides previous booking information of the user.



User Profile Information. After clicking the profile information from the dropdown box of the mail. It will show the account details of the data. Using change password, the user will be able to change password.



All Hotel. After clicking the ‘All Hotel’ button, this page will show up. As the registered hotel is not registered with a picture so it is showing the default ‘No Image Available’ photo.

Hotel Details. After clicking into a hotel, a user will be able to see the hotel's name, features, phone number, mail and description. Most importantly they will be able to see the room type and how many rooms are available on a particular day. Before placing an order an user must have to select a from and to date. After selecting the date and pressing check availability, users will be able to see From and To on the page like the screenshot. If there are any rooms booked on the particular day, the user will notice a reduction in the quantity.

Successful Booking. After booking successfully, the user will redirect to this success page, which will show relevant details based on the type of room the user booked.

Travel Nest	<input type="text" value="Search"/>	<input type="button" value="Search"/>	<input type="button" value="All Hotels"/>	Your Reservations	zawadut@gmail.com
Room ID: 29 Room Type: Family Quantity: 2 Total Price: 10000 From: 2023-12-14 To: 2023-12-18 Hotel ID: 29 Room Type: Sea View Quantity: 1 Total Price: 5000 From: 2023-12-14 To: 2023-12-18 Booking confirmed!					

User Reservation. It is placed as 'your reservation' on the website. Here users can see their previous booking details. Here, different types of rooms

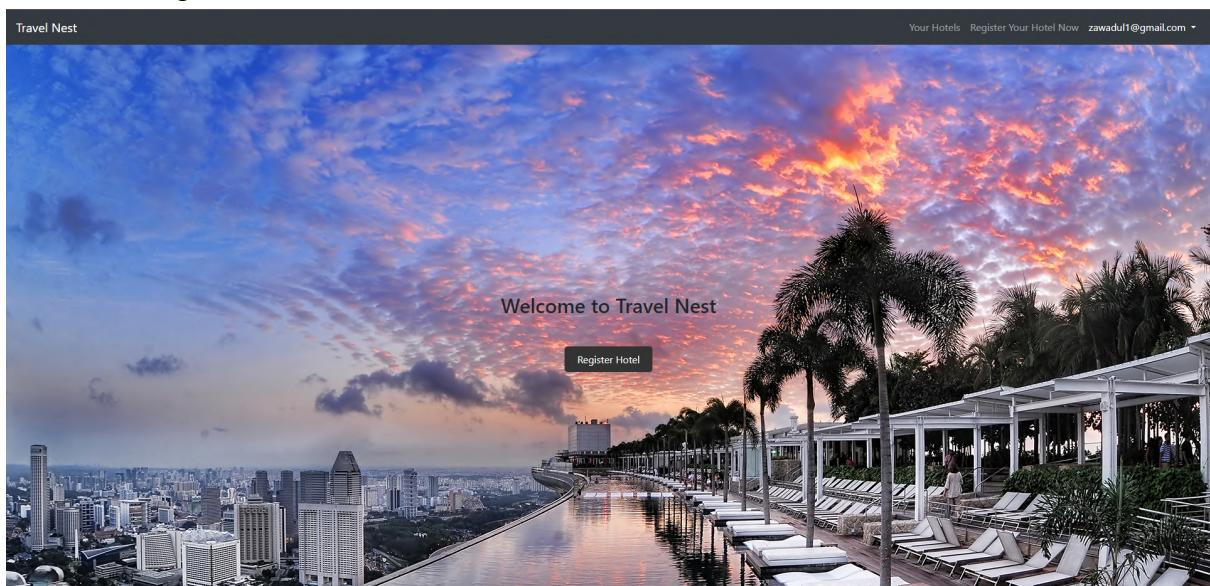
will create different types of transaction id.

Travel Nest	<input type="text" value="Search"/>	<input type="button" value="Search"/>	<input type="button" value="All Hotels"/>	Your Reservations	zawadut@gmail.com
Billing History					
Transaction ID Date Amount Method Status					
47	2023-12-10 21:07:46	5000	CASH	Paid	
50	2023-12-10 21:08:15	5000	CASH	Paid	
53	2023-12-10 21:09:21	10000	CASH	Paid	
56	2023-12-10 21:15:16	10000	CASH	Paid	
59	2023-12-10 21:17:16	10000	CASH	Paid	
60	2023-12-10 21:18:06	35000	CASH	Paid	
61	2023-12-10 21:18:06	20000	CASH	Paid	
62	2023-12-10 21:18:06	14000	CASH	Paid	

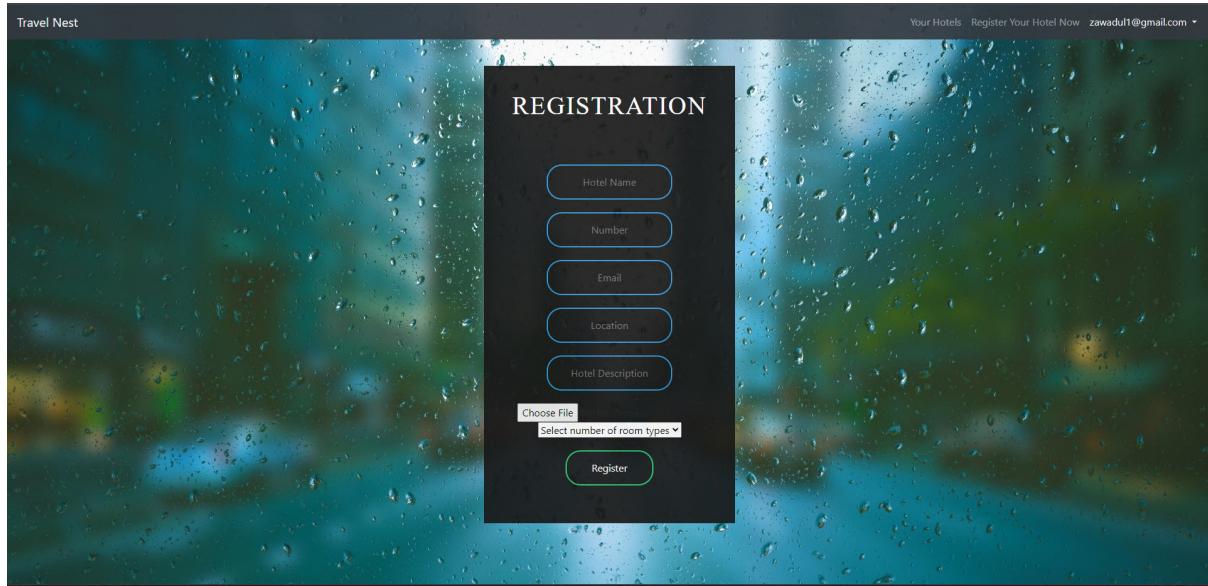
Owner

This is what a **hotel owner will see** after login

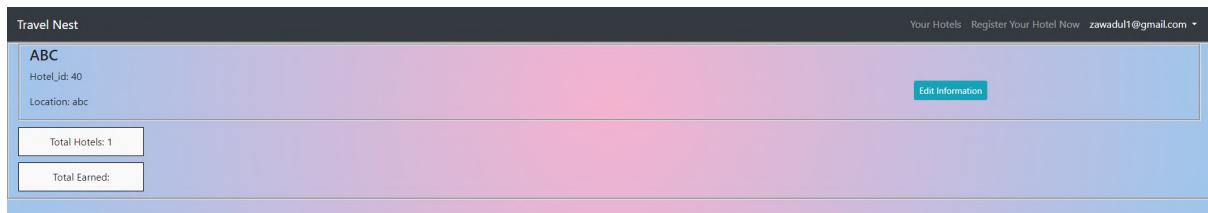
Logged in as Owner. This is the homepage an hotel owner will see after login. It is the same as User with slight modification in the navbar. In the navbar, the search bar and ‘your reservation’ is removed for the owner and added two buttons which are ‘Your Hotel’ and ‘Register Your Hotel’. Also instead of a search bar in the homepage, it shows ‘Register Now’.



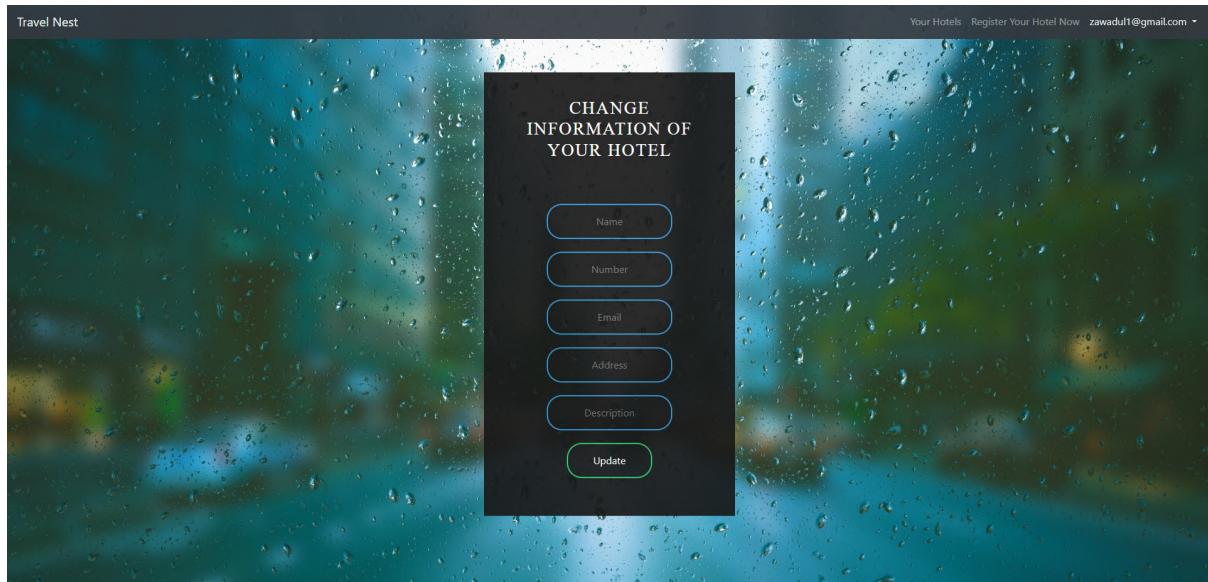
Register Your Hotel. Here the hotel owner can register his/her hotel. This is a form like signup or login page with some added features like choosing image and room types. Here room types are written in JavaScript, so that a hotel owner can choose to add up to 4 different types of room with their price, name and quantity dynamically.



Your Hotel. Here an user will be able to see all of his hotel listings, with total earnings from all the hotels combined. Also there is an 'edit information' button, which will show a form to update the hotel information.



Edit Information: The Owner of the hotel can change the necessary information of his hotel if needed



Change Password. Using this, the user will be able to update his/her password. This is the same as a user. Owner also has a user profile dashboard.

The screenshot shows a user profile page titled "Travel Nest". At the top right, there are links for "Your Hotels", "Register Your Hotel Now", and an email address "zawadul1@gmail.com". Below the header, there are two tabs: "Profile" (which is selected) and "Settings". The main content area is titled "Account Details" and contains a form for changing a password. The form includes fields for "New Password" and "Confirm Password", both with placeholder text "Enter New Password" and "Confirm New Password" respectively. A blue "Save changes" button is positioned to the right of the "Confirm Password" field. The entire form is enclosed in a light gray box.

Backend Development

Sign Up as Owner Front. Because of this code the form is created. To make it attractive we also added some css from the 'owner_signup_design.css' file. It is storing data against the 'name' which can be accessed from backend.

```
owner > 🗂 owner_signup_front.php > ...
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <meta charset="utf-8">
5       <title>Sign Up as Owner</title>
6       <link rel="stylesheet" href="css/owner_signup_design.css">
7   </head>
8   <body>
9       <?php include "...\\global_navbar.php" ?>
10      <form action="owner_signup_back.php" class="box" method="post">
11          <h1>Sign Up as Owner</h1>
12          <br>
13          <hr>
14          <input type="text" name="ho_name" placeholder="Name">
15          <input type="int" name="ho_number" placeholder="Phone Number">
16          <input type="text" name="ho_email" placeholder="Email">
17          <input type="text" name="ho_address" placeholder="Address">
18          <input type="text" name="ho_gender" placeholder="Gender">
19          <input type="password" name="ho_password" placeholder="Password">
20          <input type="submit" name="" value="Sign Up">
21          <p class="message">Registered? <a href="owner_login_front.php">Log in here</a></p>
22      </form>
23
24  </body>
25  </html>
26
27
```

Sign Up as Owner Back. Accessing the data which is passed from front and assigning relevant variables against it. For example it will store the owner's name to \$name variable and so on. Using if condition checking if they fulfilled all necessary columns or not. Here \$stmt contains the sql query. There can be many ways a sql can be written, we choose this way because it prevents sql injection attacks.

```
<?php
require_once('dbconnect.php');
$name = $_POST['ho_name'];
$phone = $_POST['ho_number'];
$email = $_POST['ho_email'];
$password = $_POST['ho_password'];
$gender = $_POST['ho_gender'];
$address = $_POST['ho_address'];

if(!empty($name) && !empty($phone) && !empty($email) && !empty($password) && !empty($gender) && !empty($address)){
    $stmt = $conn->prepare("INSERT INTO Hotel_Owner VALUES (NULL, ?, ?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("ssssss", $name, $phone, $email, $gender, $address, $password);
    $stmt->execute();
    header("Location: owner_homepage.php");
    $stmt->close();
    $conn->close();
} else {
    echo "All fields are required";
    die();
}

?> You, 2 weeks ago • v0.1
```

Login as Owner. Login as owner also has the same type of frontend because of that we are only adding backend code, here as every other code, we used ‘dbconnect.php’ to connect our code with the database. Here we are also assigning a variable against the value. And checking if the user fulfilled all the necessary information or not. Then we are running a sql query. If the value matches with the database then it will redirect the owner to the owner homepage. Otherwise it will throw an error.

```
<!-- login related backend work and sql query -->

<?php

require_once('dbconnect.php');
$email = $_POST['ho_email'];
$password = $_POST['ho_password'];
$login = False;

if (!empty($email) && !empty($password)) {

    $stmt = $conn->prepare("SELECT * FROM Hotel_Owner WHERE ho_email = ? AND ho_password = ?");
    $stmt->bind_param("ss", $email, $password);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $login = true;
        session_start();
        $_SESSION['loggedin'] = true;
        $_SESSION['email'] = $email;
        header("Location: owner_homepage.php");
    } else {
        echo 'Invalid email or password';
    }

    $stmt->close();
    $conn->close();
} else {
    echo "All fields are required";
    die();
}
```

User signup

```
100, 1 second ago | + author (100)
<?php

require_once('dbconnect.php');
$name = $_POST['c_name'];
$phone = $_POST['c_number'];
$email = $_POST['c_email'];
$password = $_POST['c_password'];
$gender = $_POST['c_gender'];

if (!empty($name) && !empty($phone) && !empty($email) && !empty($password) && !empty($gender)) {

    $stmt = $conn->prepare("INSERT INTO Customer VALUES (NULL, ?, ?, ?, ?, ?)");
    $stmt->bind_param("sssss", $name, $phone, $email, $gender, $password);
    $stmt->execute();
    echo 'signup successful';
    $stmt->close();
    $conn->close();
} else {
    echo "All fields are required";
    die();
}
|
```

User Login. Same as the owner. Just some values changed based on the scenario. Here after a successful login it will redirect to 'home.php'.

```
<?php

require_once('dbconnect.php');
$email = $_POST['c_email'];
$password = $_POST['c_password'];
$login = False;

if(!empty($email) && !empty($password)){
    $stmt = $conn->prepare("SELECT * FROM Customer WHERE c_email = ? AND c_password = ?");
    $stmt->bind_param("ss", $email, $password);
    $stmt->execute();
    $result = $stmt->get_result();

    if($result->num_rows > 0){
        $login = true;
        session_start();
        $_SESSION['loggedin'] = true;
        $_SESSION['email'] = $email;
        header("Location: home.php");
    } else {
        echo 'Invalid email or password';
    }

    $stmt->close();
    $conn->close();
} else {
    echo "All fields are required";
    die();
}
?>      You, 2 weeks ago • v0.1
```

Session. Session handling is necessary to track things like who is registering a hotel or who is booking a hotel. Here the particular screenshot shows the session handling of the user. Hotel owner session handling will be the same as this. First screenshot shows the code of session_start and the second screenshot shows the code of session destruction. The session will be destroyed whenever the user presses the logout button.

```
<?php
//session handling
session_start();

if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] != true) {
    header("location: user_login_front.php");
    exit;
}

?>
```

```
<?php
session_start();
session_unset();
session_destroy();
header("Location: user_login_front.php");

?> You, 4 days ago • v0.7
```

Search: Using Search, the user can see his desired hotel based on name/ location and feature. Here, if no results are found, then proper error handling is used. We used a join query as the hotel_features is a different table and the hotel is another table. \$query used for search term that user provided. Using the first sql query we are finding hotel id. Then based on hotel id we are running multiple queries to find the relevant details. Also we are using loop, as there can be multiple hotel id.

```

<?php
require_once('dbconnect.php');
$query = $_POST['query'];
$noresults = true;

if (!empty($query)) {
    // Query to find the hotel id
    $stmt = "SELECT h.h_id FROM Hotel h
    WHERE h_name LIKE '%$query%' OR h_location LIKE '%$query%'
    UNION
    SELECT h.h_id AS 'Hotel Name' FROM Hotel h
    JOIN Hotel_features hf ON h.h_id = hf.h_id
    WHERE hf.h_features LIKE '%$query%';

    $result = mysqli_query($conn, $stmt);
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {

            // Finding the hotel name
            $stmt2 = "SELECT h_name, h_location, h_email FROM Hotel WHERE h_id = " . $row["h_id"];
            $result2 = mysqli_query($conn, $stmt2);
            $row2 = mysqli_fetch_assoc($result2);

            $Hotel_name = $row2["h_name"];
            $Hotel_location = $row2["h_location"];
            $Hotel_email = $row2["h_email"];

            // Finding the hotel features
            $stmt3 = "SELECT h_features FROM Hotel_features WHERE h_id = " . $row["h_id"];
            $result3 = mysqli_query($conn, $stmt3);
            $Hotel_features = array(); // initialize as an empty array

            while ($row3 = mysqli_fetch_assoc($result3)) {
                array_push($Hotel_features, $row3["h_features"]);
            }

        }
    }
}

// Finding the hotel images
$stmt4 = "SELECT h_image FROM Hotel_image_archive WHERE h_id = " . $row["h_id"];
$result4 = mysqli_query($conn, $stmt4);
$Hotel_images = array(); // initialize as an empty array

while ($row4 = mysqli_fetch_assoc($result4)) {
    array_push($Hotel_images, $row4["h_image"]);
}

$stmt5 = "SELECT MAX(r_price) as max_price, MIN(r_price) as min_price FROM Room WHERE h_id = " . $row["h_id"];
$result5 = mysqli_query($conn, $stmt5);

if ($row5 = mysqli_fetch_assoc($result5)) {
    $max_price = $row5["max_price"];
    $min_price = $row5["min_price"];
}

$noresults = false; // to make sure that the no results found , return what went wrong

//choosing the first image from the array of images
$imageSrc = !empty($Hotel_images) ? './owner/hotel_pic/' . $Hotel_images[0] : './owner/hotel_pic/default_hotel.jpg';

echo
//Printing Hotel Details
<div class="col-md-12 col-md-pull-3" style="border-style:groove; width:180%">

    <div class="search-result-item-body">
        <div class="row">
            <div class="col-sm-9">
                <a class="image-link" href="hotel_details.php?h_id=' . $row["h_id"] . '"> </a>
                <h4 class="search-result-item-heading"><a href="hotel_details.php?h_id=' . $row["h_id"] . '">' . $row["h_id"] . '> ' . $Hotel_name . '</a></h4>
                <p class="info">location: ' . $Hotel_location . '</p>
                <p class="description">Features: ' . implode(", ", $Hotel_features) . '</p>
                <p class="description">Email: ' . $Hotel_email . '</p>
            </div>
            <div class="col-sm-3 text-align-center" style="margin-top:9%">

```

If search returns nothing then this condition will work.

```

        <p class="fs-mini text-muted" >Per Night</p>
        <a class="btn btn-primary btn-info btn-sm" href="hotel_details.php?h_id=' . $row["h_id"] . '">View Rooms</a>
    </div>
</div> ';
}

if ($noresults) {
    echo '<div class="jumbotron jumbotron-fluid">
        <div class="container">
            <p class="display-4">No Results Found</p>
            <p class="lead"> Suggestions: <ul>
                <li>Make sure that all words are spelled correctly.</li>
                <li>Try different keywords.</li>
                <li>Try more general keywords. </li></ul>
        </div>
    </div>';
}

mysqli_close($conn);

```

Owner Dashboard: Owner Dashboard represents the number of hotels he owns, proper amount calculation is done for the room booking he received from the customer. We are gathering owner emails from the session. Using the email we are finding the id. And using the id we are running multiple loop to find the hotel names and other relevant info

```

<!-- here he will see all the hotel info under him-->
<?php

// include 'authentication.php';
require_once('dbconnect.php');

// Get the user ID from the session
$owner_email = $_SESSION['email'];
$stmt = $conn->prepare("SELECT ho_id FROM Hotel_Owner WHERE ho_email = ?");
$stmt->bind_param("s", $owner_email);
$stmt->execute();
$result = $stmt->get_result();
$row = $result->fetch_assoc();
$owner_id = $row['ho_id'];

$stmt = $conn->prepare("SELECT SUM(p_amount) FROM Payment WHERE ho_id = ?");
$stmt->bind_param("i", $owner_id);
$stmt->execute();
$result = $stmt->get_result();
$row = $result->fetch_assoc();
$total_amount = $row['SUM(p_amount)'];

// Get the hotel info from the database
$stmt = $conn->prepare("SELECT * FROM Hotel WHERE ho_id = ?");
$stmt->bind_param("s", $owner_id);
$stmt->execute();
$result = $stmt->get_result();

?>

```

Hotel Confirmation: Again we got id from session handling and using the id. Extracted from the multiple tables to show the details of the booking.

```
<?php include 'authentication.php' ?>

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

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="css/user_profile.css">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aokXAx+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAw/dAiS6JXm" crossorigin="anonymous">

    <title>User Profile</title>
</head>

<body>
    <?php include 'user_navbar.php' ?>

    <?php
        require_once('dbconnect.php');

        $current_user = $_SESSION['email'];
        $sql = "SELECT c_id FROM Customer WHERE c_email='$current_user'";
        $result = mysqli_query($conn, $sql);
        $row = mysqli_fetch_assoc($result);
        $c_id = $row['c_id'];

        // Check if form is submitted
        if ($_SERVER['REQUEST_METHOD'] == "POST") {
            // Get the from and to dates
            $from = $_POST['from'];
            $to = $_POST['to'];
            $total_price = 0;
            // Loop through each room type
            foreach ($_POST['quantity'] as $room_type => $quantity) {
                if ($quantity!=0){

                    foreach ($_POST['quantity'] as $room_type => $quantity) {
                        if ($quantity!=0){
                            // Get the hotel id, room type, and room price for this room type
                            $hotel_id = $_POST['hotel_id'][$room_type];
                            $room_price = $_POST['room_price'][$room_type];
                            $total_price = $quantity * $room_price;

                            echo "Hotel ID: " . $hotel_id . "<br>";
                            echo "Room Type: " . $room_type . "<br>";
                            echo "Quantity: " . $quantity . "<br>";
                            echo "Total Price: " . $total_price . "<br>";
                            echo "From: " . $from . "<br>";
                            echo "To: " . $to . "<br>";

                            // Get the hotel owner id
                            $stmt = $conn->prepare("SELECT ho_id FROM Hotel WHERE h_id = ?");
                            $stmt->bind_param("i", $hotel_id);
                            $stmt->execute();
                            $result = $stmt->get_result();
                            $row = $result->fetch_assoc();
                            if ($row) {
                                $ho_id = $row['ho_id'];
                            }

                            $current_time = date('Y-m-d H:i:s', time());
                            $default_payment = 'CASH';
                            $stmt = $conn->prepare(
                                "INSERT INTO Payment (p_id, p_amount, p_method, p_time, c_id, ho_id)
                                VALUES (NULL, ?, ?, ?, ?, ?)
                            ");
                            $stmt->bind_param("ddiiii", $total_price, $default_payment, $current_time, $c_id, $ho_id);
                            $stmt->execute();
                            $stmt->close();
                        }
                    }
                }
            }
        }
    }
    </?php>
```

```

$stmt->bind_param("issii", $total_price, $default_payment, $current_time, $c_id, $ho_id);
$stmt->execute();

$p_id = $conn->insert_id;

$stmt2 = $conn->prepare(
    INSERT INTO Booking (b_id, b_amount, b_from, b_to, p_id)
    VALUES (NULL, ?, ?, ?, ?)
);

$stmt2->bind_param("issi", $quantity, $from, $to, $p_id);
$stmt2->execute(); // Corrected here
$b_id = $conn->insert_id;

$stmt3 = $conn->prepare(
    INSERT INTO Books_room (b_id, r_type, h_id)
    VALUES (?, ?, ?)
);
$stmt3->bind_param("isi", $b_id, $room_type, $hotel_id);
$stmt3->execute();
}

// Redirect to confirmation page
echo "Booking confirmed!";
else {
    echo "No booking information received.";
}

```

Hotel Information Update: Using <form></form>, redirecting values to backend, we used update query to update hotel information an redirecting to owner dashboard to see the new updated information

```

<?php

require_once('dbconnect.php');
include "authentication.php";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $h_id = $_POST['h_id'];
    $h_name = $_POST['h_name'];
    $h_number = $_POST['h_number'];
    $h_email = $_POST['h_email'];
    $h_location = $_POST['h_location'];
    $h_description = $_POST['h_description'];

    // Update the hotel data in the database

    $stmt = $conn->prepare("UPDATE Hotel SET h_name = ?, h_number = ?, h_email = ?, h_location = ?, h_description = ? WHERE h_id = ?");
    $stmt->bind_param("sssssi", $h_name, $h_number, $h_email, $h_location, $h_description, $h_id);
    $stmt->execute();
    header('Location: owner_dashboard_front.php');
    exit;
}

?>

```

Hotel Registration - Here we are also extracting id from the session. Based on the selected owner id of the, if an owner fills up this form with relevant details, it will register an hotel against his id. With this the if condition we are checking if the owner given all the necessary info or not.

```
<!-- Hotel Registration Query Handeling -->

<?php
//session handling
session_start();

if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] != true) {
    header("location: user_login_front.php");
    exit;
}

require_once('dbconnect.php');

// Get the user ID from the session
$owner_email = $_SESSION['email'];
$stmt = $conn->prepare("SELECT ho_id FROM Hotel_Owner WHERE ho_email = ?");
$stmt->bind_param("s", $owner_email);
$stmt->execute();
$result = $stmt->get_result();
$row = $result->fetch_assoc();
$owner_id = $row['ho_id'];

//Actual Login Code
$name = $_POST['h_name'];
$phone = $_POST['h_number'];
$email = $_POST['h_email'];
$location = $_POST['h_location'];
$description = $_POST['h_description'];
$roomCount = $_POST['roomCount'];
```

```

// Handle the image upload
$image = $_FILES['h_image']['name'];
$target_dir = "hotel_pic/";
$target_file = $target_dir . basename($image);

// Checking img and moving it to uploads folder
if (!empty($image) && move_uploaded_file($_FILES['h_image']['tmp_name'], $target_file)) {
    if (!empty($name) && !empty($phone) && !empty($email) && !empty($location) && !empty($description)) {

        // Insert the hotel into the database

        $stmt = $conn->prepare("INSERT INTO Hotel VALUES (NULL, ?, ?, ?, ?, ?, ?)");
        $stmt->bind_param("sssssi", $name, $phone, $email, $location, $description, $owner_id);

        if ($stmt->execute()) {
            // Get the hotel ID from the last insert AND insert the image
            $last_id = $conn->insert_id;
            $stmt = $conn->prepare("INSERT INTO Hotel_image_archive VALUES (?, ?)");
            $stmt->bind_param("ss", $last_id, $image);
            $stmt->execute();

            // Get the number of room types

            // Prepare the statement to insert the room data
            $stmt = $conn->prepare("INSERT INTO Room VALUES (?, ?, ?, ?, ?)");

            // Insert each room type into the database
            for ($i = 0; $i < $roomCount; $i++) {
                $r_type = $_POST['r_type' . $i];
                $r_price = $_POST['r_price' . $i];
                $r_quantity = $_POST['r_quantity' . $i];
                $stmt->bind_param("isii", $last_id, $r_type, $r_price, $r_quantity);
                $stmt->execute();
            }
            header('Location: owner_dashboard_front.php');
            exit;
        }
    }
}
?>

```

Hotel Booking. Based on the hotel selected by the customer, we executed many sql commands to gather info. For handling how many rooms are vacant we used a condition, if the selected to and from is less or more than the all the booking date of the hotel then it will show full amount otherwise, it will count all the number of booked rooms on a variable then subtract it from total number of rooms.

```
<?php
require_once('dbconnect.php');

if (isset($_GET['h_id'])) {
    $h_id = $_GET['h_id'];

    // sql query to find information based on the hotel id
    $stmt1 = $conn->prepare("SELECT * FROM Hotel WHERE h_id = ?");
    $stmt1->bind_param("i", $h_id);
    $stmt1->execute();
    $hotel = $stmt1->get_result()->fetch_assoc();

    // sql query to find features based on the hotel id
    $stmt2 = $conn->prepare("SELECT h_features FROM Hotel_features WHERE h_id = ?");
    $stmt2->bind_param("i", $h_id);
    $stmt2->execute();
    $features = $stmt2->get_result()->fetch_all(MYSQLI_ASSOC);

    // sql query to find rooms based on the hotel id
    $stmt3 = $conn->prepare("SELECT * FROM Room WHERE h_id = ?");
    $stmt3->bind_param("i", $h_id);
    $stmt3->execute();
    $rooms = $stmt3->get_result()->fetch_all(MYSQLI_ASSOC);

    // sql query to find images based on the hotel id
    $stmt4 = $conn->prepare("SELECT h_image FROM Hotel_image_archive WHERE h_id = ?");
    $stmt4->bind_param("i", $h_id);
    $stmt4->execute();
    $result4 = $stmt4->get_result();
    $Hotel_images = array(); // initialize as an empty array

    while ($row = $result4->fetch_assoc()) {
        array_push($Hotel_images, $row["h_image"]);
    }
}

if (isset($_POST['submit'])) {
    $from = $_POST['from'];
    $to = $_POST['to'];
```

```

// sql query to find b_from, b_to from booking table and used join query to get the h_id from booking table with the help of b_id
$stmt = $conn->prepare("
SELECT Booking.b_from, Booking.b_to, Books_room.h_id
FROM Booking
JOIN Books_room ON Booking.b_id = Books_room.b_id");
$stmt->execute();
$result = $stmt->get_result();
$bookings = $result->fetch_all(MYSQLI_ASSOC);

// array to store the count of how many rooms are booked based on the room type
$room_bookings = array();
$quantity = array();
$visited_b_ids = array();

// loop to check the amount of availability of the rooms
foreach ($bookings as $booking) {
    // if the hotel id is same and b_from and b_to is in between the selected dates then it will execute the query
    if ($booking['h_id'] == $h_id && !($booking['b_from'] < $from && $booking['b_to'] < $from) || ($booking['b_from'] > $to && $booking['b_to'] > $to)) {

        // sql query to find the room type and b_amount from room table and used first join query to get the r_type and h_id from room table with the help of h_id and second join query to get
        $stmt5 = $conn->prepare("
SELECT Books_room.r_type, Booking.b_amount, Booking.b_id
FROM Books_room
JOIN Booking ON Books_room.b_id = Booking.b_id
WHERE Books_room.h_id = ? AND Booking.b_from = ? AND Booking.b_to = ?
");
        $stmt5->bind_param("iss", $h_id, $booking['b_from'], $booking['b_to']);
        $stmt5->execute();
        $result = $stmt5->get_result();
        while ($row = $result->fetch_assoc()) {
            $quantity[] = $row;
        }
    }
}

// checking of how many rooms are available ended

```

```

foreach ($quantity as $booking) {
    //if the room type is already in the array then it will add the b_amount to the previous b_amount else it will add the room type and b_amount to the array.
    if (!in_array($booking['b_id'], $visited_b_ids)) {
        // Add the b_id to the visited array
        $visited_b_ids[] = $booking['b_id'];

        if (array_key_exists($booking['r_type'], $room_bookings)) {
            $room_bookings[$booking['r_type']] += $booking['b_amount'];
        } else {
            $room_bookings[$booking['r_type']] = $booking['b_amount'];
        }
    }
}

```

All Hotel list: Run a sql query on hotel, features and image table and extracted all the necessary information. Handled default image, if no image found.

```
<?php
require_once('dbconnect.php');

// query to find all hotels
$stmt = "SELECT h.h_id, h.h_name, h.h_location, h.h_email FROM Hotel h";
$result = mysqli_query($conn, $stmt);

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $Hotel_name = $row["h_name"];
        $Hotel_location = $row["h_location"];
        $Hotel_email = $row["h_email"];

        // finding the hotel features
        $stmt3 = "SELECT h_features FROM Hotel_features WHERE h_id = " . $row["h_id"];
        $result3 = mysqli_query($conn, $stmt3);
        $Hotel_features = array(); // initialize as an empty array

        while ($row3 = mysqli_fetch_assoc($result3)) {
            array_push($Hotel_features, $row3["h_features"]);
        }

        // finding the hotel images
        $stmt4 = "SELECT h_image FROM Hotel_image_archive WHERE h_id = " . $row["h_id"];
        $result4 = mysqli_query($conn, $stmt4);
        $Hotel_images = array(); // initialize as an empty array

        while ($row4 = mysqli_fetch_assoc($result4)) {
            array_push($Hotel_images, $row4["h_image"]);
        }

        $stmt5 = "SELECT MAX(r_price) as max_price, MIN(r_price) as min_price FROM Room WHERE h_id = " . $row["h_id"];
        $result5 = mysqli_query($conn, $stmt5);

        if ($row5 = mysqli_fetch_assoc($result5)) {
            $max_price = $row5["max_price"];
            $min_price = $row5["min_price"];
        }

        $noresults = false; // to make sure that the no results found jumbotron is not displayed

        //choosing the first image from the array of images
        $imageSrc = !empty($Hotel_images) ? '..\owner\hotel_pic\' . $Hotel_images[0] : '..\owner\hotel_pic\default_hotel.jpg';

        echo

```

```
echo
//Printing Hotel Details
'<div class="col-md-12 col-md-pull-3" style="border-style:groove; width:180%">

    <div class="search-result-item-body">
        <div class="row">
            <div class="col-sm-9">
                <a class="image-link" href="hotel_details.php?h_id=' . $row["h_id"] . '"><img class="image" src="" . $imageSrc . '" style="height:200px; width:200px;" /> </a>
                <h4 class="search-result-item-heading"><a href="hotel_details.php?h_id=' . $row["h_id"] . '">' . $row["h_name"] . '</a></h4>
                <p class="info">Location: ' . $Hotel_location . '</p>
                <p class="description">Features: ' . implode(", ", $Hotel_features) . '</p>
                <p class="description">Email: ' . $Hotel_email . '</p>
            </div>
            <div class="col-sm-3 text-align-center" style="margin-top:9%">
                <p class="value3 mt-sm">' . $min_price . 'TK - ' . $max_price . 'TK</p>
                <p class="fs-mini text-muted">Per Night</p>
                <a class="btn btn-primary btn-info btn-sm" href="hotel_details.php?h_id=' . $row["h_id"] . '">View Rooms</a>
            </div>
        </div>
    </div> ';

}

else { //it will show error when no hotel is in db
    echo '<div class="jumbotron jumbotron-fluid">
        <div class="container">
            <p class="display-4">No Results Found</p>
        </div>
    </div>';
}

mysqli_close($conn);
?>
```

Previous Order list: Retrieved from the payment table where users can see all their reservations up to date. First we retrieved customer id and based on that we executed a SQL query and retrieved all the payment info of that customer.

```
<?php
require_once('dbconnect.php');

$user_email=$_SESSION['email'];

$sql= ("SELECT c_id FROM Customer WHERE c_email = '$user_email'");
$result= mysqli_query($conn,$sql);
if ($result) {

    $row = mysqli_fetch_assoc($result);
    $user_id=$row['c_id'];

}

// sql query to find paid information based on the customer_id
$sql2= ("SELECT * FROM Payment WHERE c_id = $user_id");
$result= mysqli_query($conn,$sql2);
```

```
<?php
if ($result) {

    while ($row = mysqli_fetch_assoc($result)) {
        if ($row['p_amount']!=0){
            echo
                '<tbody>
                    <tr>
                        <td>'.$row["p_id"].'</td>
                        <td>'.$row["p_time"].'</td>
                        <td>'.$row["p_amount"].'</td>
                        <td>'.$row["p_method"].'</td>
                        <td><span class="badge bg-success">Paid</span></td>
                    </tr></tbody>'
    ;}}}?>
```

Password Change

Owner: Updated the password using UPDATE MySQL query in hotel_owner table

```
<?php      You, 4 days ago • v0.7
session_start();

if(!isset($_SESSION['loggedin']) || $_SESSION['loggedin']!=true){
    header("location: login.php");
    exit;
}

include 'dbconnect.php';
$current_user=$_SESSION['email'];

$pass1=$_POST['newpass'];
$pass2=$_POST['newpass2'];

if($pass1==$pass2){
    $sql="UPDATE hotel_owner SET ho_password = '$pass1' WHERE ho_email='$current_user'";
    $result=mysqli_query($conn,$sql);
    header("location: owner_profile.php");
}
else if ($pass1!=$pass2) {
    echo 'invalid';
}
?>
```

User password change: similar process like owner, but table and attribute changed in accordance with 'customer' table

```
<?php
|      You, 2 weeks ago • v0.1
require_once('dbconnect.php');
$name = $_POST['c_name'];
$phone = $_POST['c_number'];
$email = $_POST['c_email'];
$password = $_POST['c_password'];
$gender = $_POST['c_gender'];

if (!empty($name) && !empty($phone) && !empty($email) && !empty($password) && !empty($gender)) {

    $stmt = $conn->prepare("INSERT INTO Customer VALUES (NULL, ?, ?, ?, ?, ?)");
    $stmt->bind_param("sssss", $name, $phone, $email, $gender, $password);
    $stmt->execute();
    echo 'signup successful';
    $stmt->close();
    $conn->close();
} else {
    echo "All fields are required";
    die();
}
```

Profile

Owner: Retrieved information of owner from hotel_owner

```
<?php
include 'dbconnect.php';
$currentUserId = $_SESSION['email'];

$sql = "SELECT * FROM Hotel_Owner WHERE ho_email='$currentUserId'";
$result = mysqli_query($conn, $sql);
if ($result) {
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_array($result)) {
            $name = $row['ho_name'];
            $email = $row['ho_email'];
            $number = $row['ho_number'];
            $gender = $row['ho_gender'];
            $address = $row['ho_address'];
        }
    }
}
?>
```

User: Retrieved information of owner from 'customer' table

```
<?php
include 'dbconnect.php';
$currentUserId = $_SESSION['email'];

$sql = "SELECT * FROM Customer WHERE c_email='$currentUserId'";
$result = mysqli_query($conn, $sql);
if ($result) {
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_array($result)) {
            $name = $row['c_name'];
            $email = $row['c_email'];
            $number = $row['c_number'];
            $gender = $row['c_gender'];
        }
    }
}
?>
```

Conclusion

In summary, our project's hotel booking system database effectively tackles the complexities inherent in the modern hotel booking system. By offering an intuitive user interface, optimizing resource allocation, delivering real-time updates, enforcing data security measures, and preparing for scalability, we've established a comprehensive solution that benefits both travelers and hotel owners.³⁰ While this report captures the project's scope and accomplishments, the path toward a fully functional system continues. Our initiative paves the way for future enhancements, geared toward delivering seamless travel experiences and operational excellence for airlines. Moving ahead, collaboration and adaptability will remain pivotal. The insights gleaned from this project will prove invaluable for upcoming ventures in database management and aviation technology. Essentially, our undertaking showcases how technology has the potential to heighten efficiency and user satisfaction in the aviation realm, contributing to the ever-evolving global air travel landscape

References

[PHP Tutorials in Hindi - YouTube](#)

[Bootstrap html snippet. bs5 edit profile account details](#)

[Bootstrap snippet and html example. ecommerce products with image description colors and add button](#)

[Introduction · Bootstrap v5.0](#)