StayDine

WHERE COMFORT MEETS CULINARY DELIGHT!

Integrated Hotel and Restaurant Reservation Platform

PROJECT BY —
SHAIK MASTHAN JILANI, 23075074
SOHEL, 23075081

Objective

- This project helped us gain experience creating a website and developing an integrated platform for managing hotel room bookings, restaurant reservations, and organizing events. This
- application ‘StayDine’ provides a host of features and the goal is to create an intuitive and user-friendly interface so special, covering both hotel and restaurant needs. The main motive for this idea, was a
- dream project for both of us, a business idea that sparked and has ignited the demand for online hotel and restaurant booking, enhancing operational efficiency for hotel and restaurant management. ‘StayDine’ is made to provide both features at one
- place.

Motivation for StayDine

Demand

We often find that, many make project on Ecommerce or Library Management sites, when it comes to DBMS Development Projects. For a change, giving our fullest to a successful site is our motive.

Accessibility

Though the project may seem simple, the complexity of all the tables and how they're accessed is quite amusing and has given us the excitement to give our fullest in making this.

Problems

To contribute to the growth famous hotels and restaurants by providing them a digital platform to connect with customers and to keep track of things easily by the staff.

Database Design (Tables)

A total of 12 Tables, 7 Triggers and 8 views are created for the complete usage of DBMS, however, more might be added further.

- ***User's Table***: Stores user information, including unique identifiers, contact details, and account creation timestamp.
- ***Role's Table***: Contains different staff roles with unique identifiers and role names for categorization.
- ***Staff's Table***: Manages staff details, including personal information, role assignments, and manager relationships.
- ***Suite's Table***: Represents hotel room information, including type, pricing, capacity, availability status, and staff assignments.
- ***Diner's Table***: Stores details about restaurant tables, including capacity, availability, and staff assignments.
- ***Staff's Assignment Table***: Tracks assignments of the staff to specific rooms and tables, including the date of assignment.

Database Design (Tables)

- ***Bookings Table***: Records reservations made by users for rooms or tables, including booking status and associated event type.
- ***Menu Item's Table***: Contains details about menu items available in the restaurant, including their names, prices, and categories.
- ***Orders' Table***: Manages customer orders linked to bookings, tracking total prices and order status.
- ***Ordered Items' Table***: Details individual items within an order, including quantity and price for each menu item.
- ***Event Types Table***: Stores information about events available for booking, including names, pricing, guest capacity, and creation timestamp.
- ***Event Bookings' Table***: Records reservations for events made by users, including event details, total price, and booking status.

USER'S TABLE

```sql
CREATE DATABASE IF NOT EXISTS StayDine;
USE StayDine;

CREATE TABLE Users (
user_id INT PRIMARY KEY AUTO_INCREMENT,
username CHAR(75) UNIQUE NOT NULL,
email CHAR(150) UNIQUE NOT NULL,
password CHAR(100) NOT NULL,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50),
mobile_no CHAR(15) NOT NULL,
aadhaar_no VARCHAR(16) UNIQUE NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

STAFF'S TABLE

```sql
CREATE TABLE Staff_roles (
role_id INT PRIMARY KEY,
role_name VARCHAR(30) NOT NULL UNIQUE
);

CREATE TABLE Staffs (
staff_id INT PRIMARY KEY AUTO_INCREMENT,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50),
email CHAR(100) UNIQUE NOT NULL,
mobile_no DECIMAL(10,0) NOT NULL,
manager_id INT,
role_id INT NOT NULL,
hire_date DATE NOT NULL,

CONSTRAINT fk_manager FOREIGN KEY (manager_id) REFERENCES Staffs(staff_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_role FOREIGN KEY (role_id) REFERENCES Staff_roles(role_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

SUITE'S TABLE

```sql
CREATE TABLE Suites (
room_number INT PRIMARY KEY,
room_type VARCHAR(50) NOT NULL,
bed_type VARCHAR(20) DEFAULT 'Double',
price_per_night DECIMAL(15,2),
max_occupants TINYINT DEFAULT 2,
is_available BOOLEAN DEFAULT TRUE,
staff_id INT,

CONSTRAINT fk_staff_hotel FOREIGN KEY (staff_id)
    REFERENCES Staffs(staff_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

RESTAURANT DINER'S

```sql
CREATE TABLE Restaurant_diner (
table_number INT PRIMARY KEY,
capacity INT NOT NULL DEFAULT 2,
is_available BOOLEAN DEFAULT TRUE,
staff_id INT,

CONSTRAINT fk_staff_diner FOREIGN KEY (staff_id)
    REFERENCES Staffs(staff_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

STAFF'S ASSIGNMENT'S

```sql
CREATE TABLE StaffAssignments (
assignment_id INT PRIMARY KEY AUTO_INCREMENT,
staff_id INT NOT NULL,
room_number INT,
table_number INT,
assigned_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

CONSTRAINT fk_staff FOREIGN KEY (staff_id)
    REFERENCES Staffs(staff_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_room_st FOREIGN KEY (room_number)
    REFERENCES Suites(room_number)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_table_st FOREIGN KEY (table_number)
    REFERENCES Restaurant_diner(table_number)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

BOOKINGS' TABLE

```sql
CREATE TABLE Bookings (
booking_id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
room_id INT,
table_id INT,
booking_date DATE NOT NULL,
event_type VARCHAR(50),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
stat ENUM('Pending', 'Completed', 'Cancelled') DEFAULT 'Pending',

CONSTRAINT fk_cust_id FOREIGN KEY (user_id)
    REFERENCES Users(user_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_room FOREIGN KEY (room_id)
    REFERENCES Suites(room_number)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_table FOREIGN KEY (table_id)
    REFERENCES Restaurant_diner(table_number)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

MENU ITEMS' TABLE

```sql
CREATE TABLE MenuItems (
menu_item_id INT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
price DECIMAL(15, 2) NOT NULL,
category VARCHAR(50) NOT NULL
);
```

ORDERS' TABLE

```sql
CREATE TABLE Orders (
order_id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
booking_id INT,
total_price DECIMAL(15, 2) DEFAULT 0,
order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
stat ENUM('Pending', 'Completed', 'Cancelled') DEFAULT 'Pending',

CONSTRAINT fk_cust_ord_id FOREIGN KEY (user_id)
    REFERENCES Users(user_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_booking_id FOREIGN KEY (booking_id)
    REFERENCES Bookings(booking_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

ORDERED ITEM'S

```sql
CREATE TABLE OrderItems (
order_item_id INT PRIMARY KEY AUTO_INCREMENT,
order_id INT NOT NULL,
menu_item_id INT NOT NULL,
quantity INT NOT NULL DEFAULT 1,
price DECIMAL(15, 2) NOT NULL,

CONSTRAINT fk_ord_id FOREIGN KEY (order_id)
    REFERENCES Orders(order_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_menu_id FOREIGN KEY (menu_item_id)
    REFERENCES MenuItems(menu_item_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

EVENTS' TABLE

```sql
CREATE TABLE Eventz (
event_id INT PRIMARY KEY,
event_name VARCHAR(100) NOT NULL,
price DECIMAL(15, 2) NOT NULL,
max_guests INT NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

EVENT BOOKING'S

```sql
CREATE TABLE EventBookings (
booking_id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
event_id INT,
event_date DATE NOT NULL,
total_price DECIMAL(10, 2) NOT NULL,
stat ENUM('Pending', 'Completed', 'Cancelled') DEFAULT 'Pending',
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

CONSTRAINT fk_cust_ev_id FOREIGN KEY (user_id)
  REFERENCES Users(user_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_ev_id FOREIGN KEY (event_id)
  REFERENCES Eventz(event_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

Database Design (Triggers)

- ***To validate email format before inserting or updating in users***: Ensuring valid format of email is used when a new user tries to enter or if an existing user is trying to update.
- ***To validate password strength before inserting or updating in users***: To validate if the password is at least 8 characters long, contains a combination of numbers and alphabets (at least one upper case) and also special characters.
- ***Assigning a Waiter to a Table/Diner***: To automate assignment of a waiter to a table, when a customer has occupied that particular dining table.
- ***Assigning Housekeeper to a Occupied Room***: To automate assigning of a housekeeper to a room, when a customer has booked/is assigned a particular room.
- ***Freeing the staff (Housekeeper or Waiter) After the Customer's Payment***: Automating the task to free staff after customer leaves.

Email constraints, before insert and update in USERS Table

```
DELIMITER //
CREATE TRIGGER ValidateEmailFormatBeforeInsertUsers
BEFORE INSERT ON Users
FOR EACH ROW
BEGIN
    IF NOT (NEW.email LIKE '%_@__%.com') THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid email format for Users.';
    END IF;
END //

CREATE TRIGGER ValidateEmailFormatBeforeUpdateUsers
BEFORE UPDATE ON Users
FOR EACH ROW
BEGIN
    IF NOT (NEW.email LIKE '%_@__%.com') THEN
        SIGNAL SQLSTATE '45001' SET MESSAGE_TEXT = 'Invalid email format for Users.';
    END IF;
END //
```

Password Strength, before inserting in USERS Table

```
CREATE TRIGGER ValidatePasswordStrengthBeforeInsertUsers
BEFORE INSERT ON Users
FOR EACH ROW
BEGIN
    IF LENGTH(NEW.password) < 8 OR
       NOT (NEW.password REGEXP '[0-9]') OR
       NOT (NEW.password REGEXP '[A-Z]') OR
       NOT (NEW.password REGEXP '[!@#$%^&*()]') THEN
        SIGNAL SQLSTATE '45002'
        SET MESSAGE_TEXT = 'Password must be at least 8 characters long,
        contain at least one number, one uppercase letter,
        and one special character.';
    END IF;
END //
```

Password Strength, before updating in USERS Table

```
CREATE TRIGGER ValidatePasswordStrengthBeforeUpdateUsers
BEFORE UPDATE ON Users
FOR EACH ROW
BEGIN
    IF LENGTH(NEW.password) < 8 OR
       NOT (NEW.password REGEXP '[0-9]') OR
       NOT (NEW.password REGEXP '[A-Z]') OR
       NOT (NEW.password REGEXP '[!@#$%^&*()]') THEN
        SIGNAL SQLSTATE '45003'
        SET MESSAGE_TEXT = 'Password must be at least 8 characters long,
        contain at least one number, one uppercase letter,
        and one special character.';    END IF;
END //
```

Assigning Housekeeper to a room, after updation in SUITES Table

```
CREATE TRIGGER Assign_Housekeeper_To_Room
AFTER UPDATE ON Suites
FOR EACH ROW
BEGIN
  DECLARE housekeeper_id INT;
  IF NEW.is_available = TRUE AND NEW.staff_id IS NULL THEN
    SELECT staff_id INTO housekeeper_id
    FROM Staffs
    WHERE role_id = (SELECT role_id FROM Staff_roles WHERE role_name = 'Housekeeping')
    AND staff_id NOT IN (SELECT staff_id FROM StaffAssignments WHERE room_number = NEW.room_number)
    ORDER BY (SELECT COUNT(*) FROM StaffAssignments WHERE staff_id = Staffs.staff_id) ASC
    LIMIT 1;

    IF housekeeper_id IS NOT NULL THEN
      INSERT INTO StaffAssignments (staff_id, room_number)
      VALUES (housekeeper_id, NEW.room_number);
    END IF;
  END IF;
END //
```

Assigning Waiter to a diner(table), after updation in RESTAURANT_DINER Table

```
CREATE TRIGGER Assign_Waiter_To_Table
AFTER UPDATE ON Restaurant_diner
FOR EACH ROW
BEGIN
  DECLARE waiter_id INT;
  IF NEW.is_available = TRUE AND NEW.staff_id IS NULL THEN
    SELECT staff_id INTO waiter_id
    FROM Staffs
    WHERE role_id = (SELECT role_id FROM Staff_roles WHERE role_name = 'Waiter')
    AND staff_id NOT IN (SELECT staff_id FROM StaffAssignments WHERE table_number = NEW.table_number)
    ORDER BY (SELECT COUNT(*) FROM StaffAssignments WHERE staff_id = Staffs.staff_id) ASC
    LIMIT 1;

    IF waiter_id IS NOT NULL THEN
      INSERT INTO StaffAssignments (staff_id, table_number)
      VALUES (waiter_id, NEW.table_number);
    END IF;
  END IF;
END //
```

Freeing Staff after the Customer pays the bill. After Updation in BOOKINGS Table

```
CREATE TRIGGER Free_Staff_After_Payment
AFTER UPDATE ON Bookings
FOR EACH ROW
BEGIN
    IF NEW.stat = 'Completed' THEN
        IF NEW.room_id IS NOT NULL THEN
            DELETE FROM StaffAssignments WHERE room_number = NEW.room_id;
        END IF;
        IF NEW.table_id IS NOT NULL THEN
            DELETE FROM StaffAssignments WHERE table_number = NEW.table_id;
        END IF;
    END IF;
END //
DELIMITER ;
```

Database Design (Views)

Frequent Customers: Finding customers, with more than 3 bookings, and providing them with offers and discounts!

Active Bookings: Shows all active bookings, including pending and confirmed ones. Useful to keep track of active bookings, for the admin panel.

Pending Orders And Bookings: Combines pending bookings and orders in one view, displaying bookings and any associated orders that are still in a pending status

Staff Current Assignments: To keep track of current room or table assignments for all staff members.

***Staff Workload*:** Provides a summary of how many rooms and tables are assigned to each staff member.

***Menu Overview*:** Displays an organized list of all menu items along with their category, price, and ID, sorted by category and price to give a clear menu structure.

***Event Bookings Overview*:** Shows a summary of all event bookings that are either pending or confirmed.

***Event Participation*:** To keep track of all the guests arrived/participated in an event.

***Revenue Overview*:** To analyse revenue from bookings, orders and events for business insights

FREQUENT CUSTOMERS VIEW

```sql
CREATE VIEW FrequentCustomers AS
SELECT
    u.user_id,
    u.username,
    u.email,
    COUNT(b.booking_id) AS total_bookings
FROM
    Users u
JOIN
    Bookings b ON u.user_id = b.user_id
GROUP BY
    u.user_id
HAVING
    COUNT(b.booking_id) > 3;
```

ACTIVE BOOKINGS VIEW

```sql
CREATE VIEW ActiveBookings AS
SELECT
    b.booking_id,
    u.username,
    u.email,
    b.room_id,
    b.table_id,
    b.booking_date,
    b.event_type,
    b.stat AS booking_status
FROM
    Bookings AS b
JOIN
    Users AS u ON b.user_id = u.user_id
WHERE
    b.stat IN ('Pending', 'Confirmed');
```

PENDING ORDERS AND BOOKINGS VIEW

```sql
CREATE VIEW PendingOrdersAndBookings AS
SELECT
    b.booking_id,
    u.username,
    b.room_id,
    b.table_id,
    b.booking_date,
    b.stat AS booking_status,
    o.order_id,
    o.total_price AS order_total,
    o.stat AS order_status
FROM
    Bookings AS b
LEFT JOIN
    Orders AS o ON b.booking_id = o.booking_id
JOIN
    Users AS u ON b.user_id = u.user_id
WHERE
    b.stat = 'Pending' OR o.stat = 'Pending';
```

CURRENT STAFF ASSIGNMENTS VIEW

```sql
CREATE VIEW StaffCurrentAssignments AS
SELECT
    s.staff_id,
    CONCAT(s.first_name, ' ', s.last_name) AS staff_name,
    sr.role_name,
    sa.room_number,
    sa.table_number,
    sa.assigned_date
FROM
    StaffAssignments AS sa
JOIN
    Staffs AS s ON sa.staff_id = s.staff_id
JOIN
    Staff_roles sr ON s.role_id = sr.role_id;
```

WORKLOAD OF A STAFF'S VIEW

```sql
CREATE VIEW StaffWorkload AS
SELECT
    s.staff_id,
    CONCAT(s.first_name, ' ', s.last_name) AS staff_name,
    sr.role_name,
    COUNT(sa.room_number) AS total_rooms_assigned,
    COUNT(sa.table_number) AS total_tables_assigned
FROM
    Staffs AS s
LEFT JOIN
    Staff_roles AS sr ON s.role_id = sr.role_id
LEFT JOIN
    StaffAssignments AS sa ON s.staff_id = sa.staff_id
GROUP BY
    s.staff_id;
```

CLEAR MENU VIEW

```sql
CREATE VIEW MenuOverview AS
SELECT
    category,
    menu_item_id AS ID,
    name,
    price
FROM
    MenuItems
ORDER BY
    category,
    price;
```

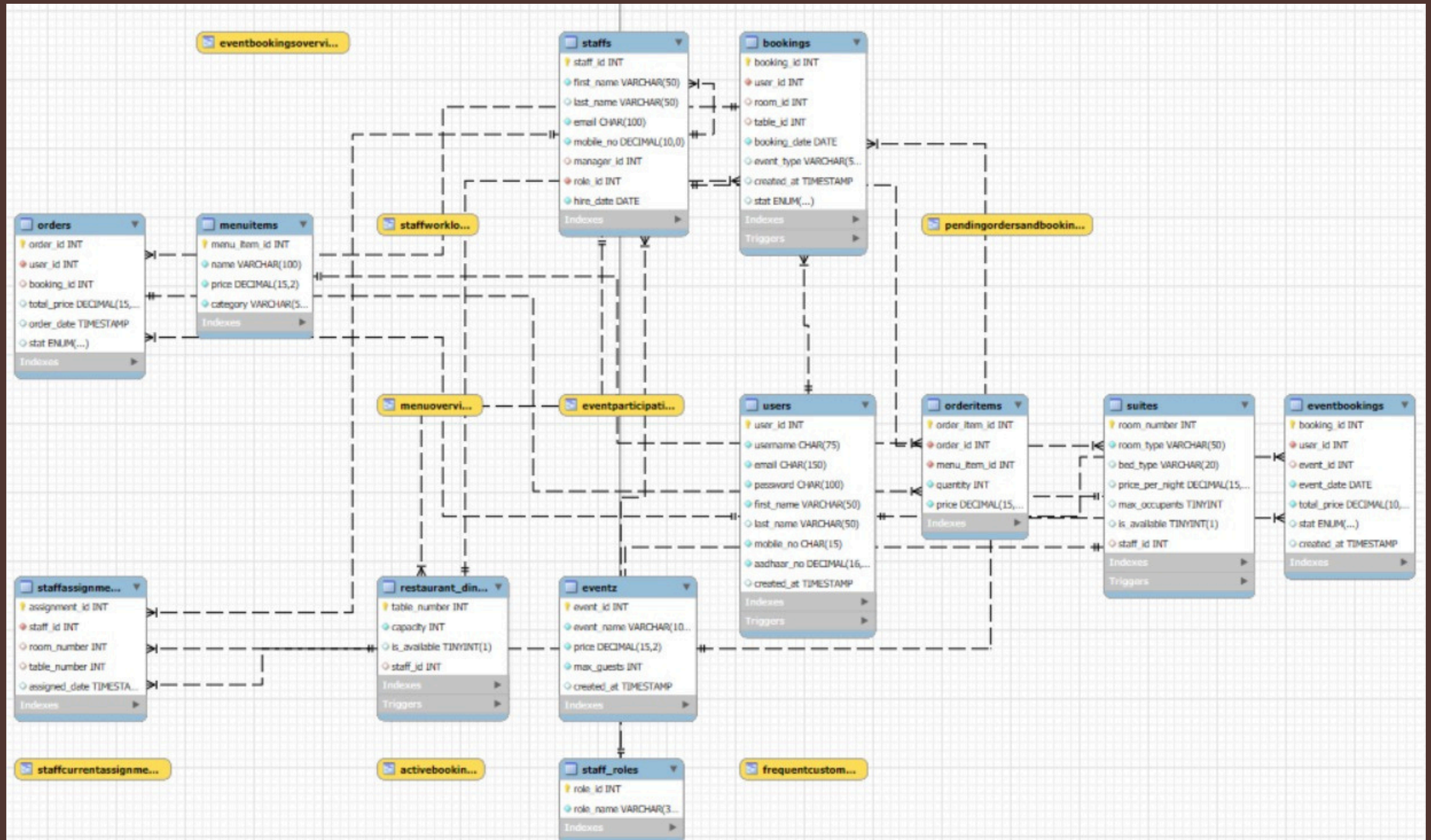
EVENT BOOKINGS' VIEW

```sql
CREATE VIEW EventBookingsOverview AS
SELECT
    eb.booking_id,
    u.username,
    e.event_name,
    eb.event_date,
    eb.total_price,
    eb.stat
FROM
    EventBookings AS eb
JOIN
    Users AS u ON eb.user_id = u.user_id
JOIN
    Eventz AS e ON eb.event_id = e.event_id
WHERE
    eb.stat IN ('Pending', 'Confirmed');
```

GUESTS IN EVENT VIEW

```sql
CREATE VIEW EventParticipation AS
SELECT
    u.user_id,
    u.username,
    e.event_name,
    eb.event_date,
    eb.total_price
FROM
    EventBookings AS eb
JOIN
    Users AS u ON eb.user_id = u.user_id
JOIN
    Eventz e ON eb.event_id = e.event_id;
```

Entity Relationship Diagram

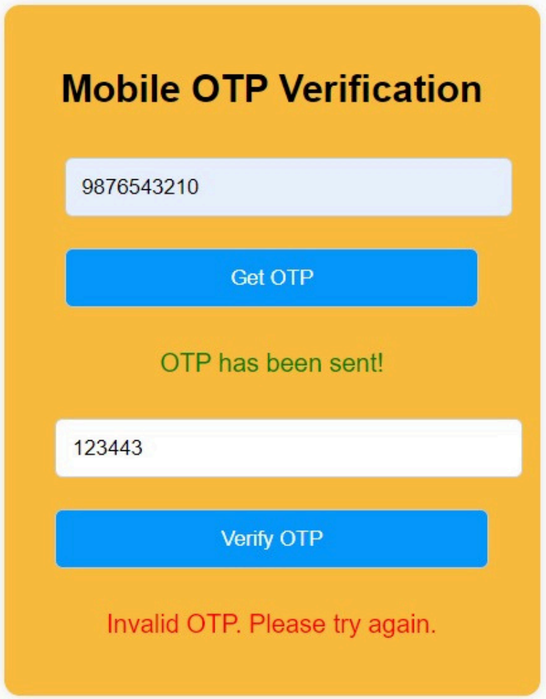


Frontend Design(Preliminary)

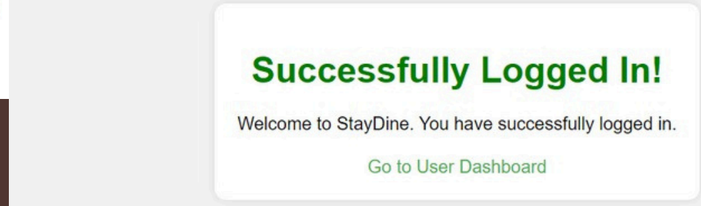
Current Features:

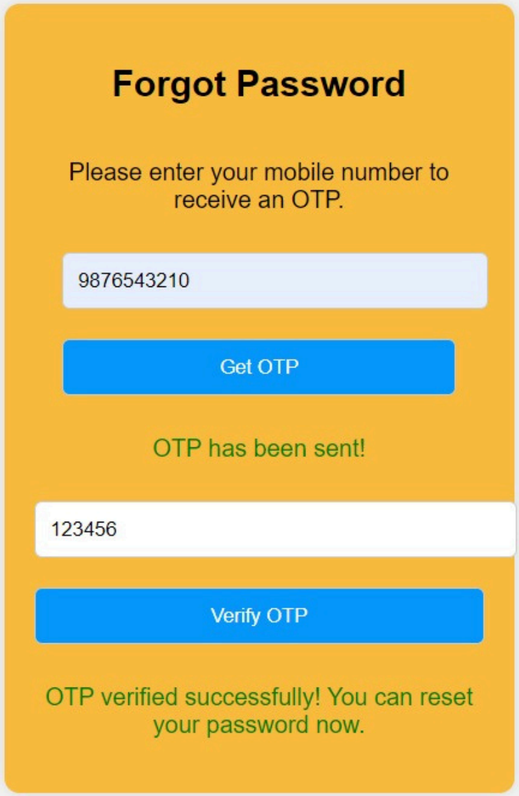
- ***Login/Sign Up Page***: Easy to understand login page, with social media redirections.
- ***Home Page***: A clean and intuitive layout showcasing hotel and restaurant services.
- ***Booking Page***: Easy-to-use reservation system for suites and restaurant tables.
- ***Menu Interface***: 5-star menu with Indian Rupees (INR) pricing.

For Sign-up



After Inserting login credentials





StayDine Table Reservations

Reserve a Table

Select Date:

dd-mm-yyyy

Select Time:

--:--

Table Size:

Table for 2

Your Name:

Enter your name

Email Address:

Enter your email

Reserve Now

Welcome to StayDine,
Where Comfort meets Culinary Delight!

Experience the Best of StayDine ;)

Great Cuisine!

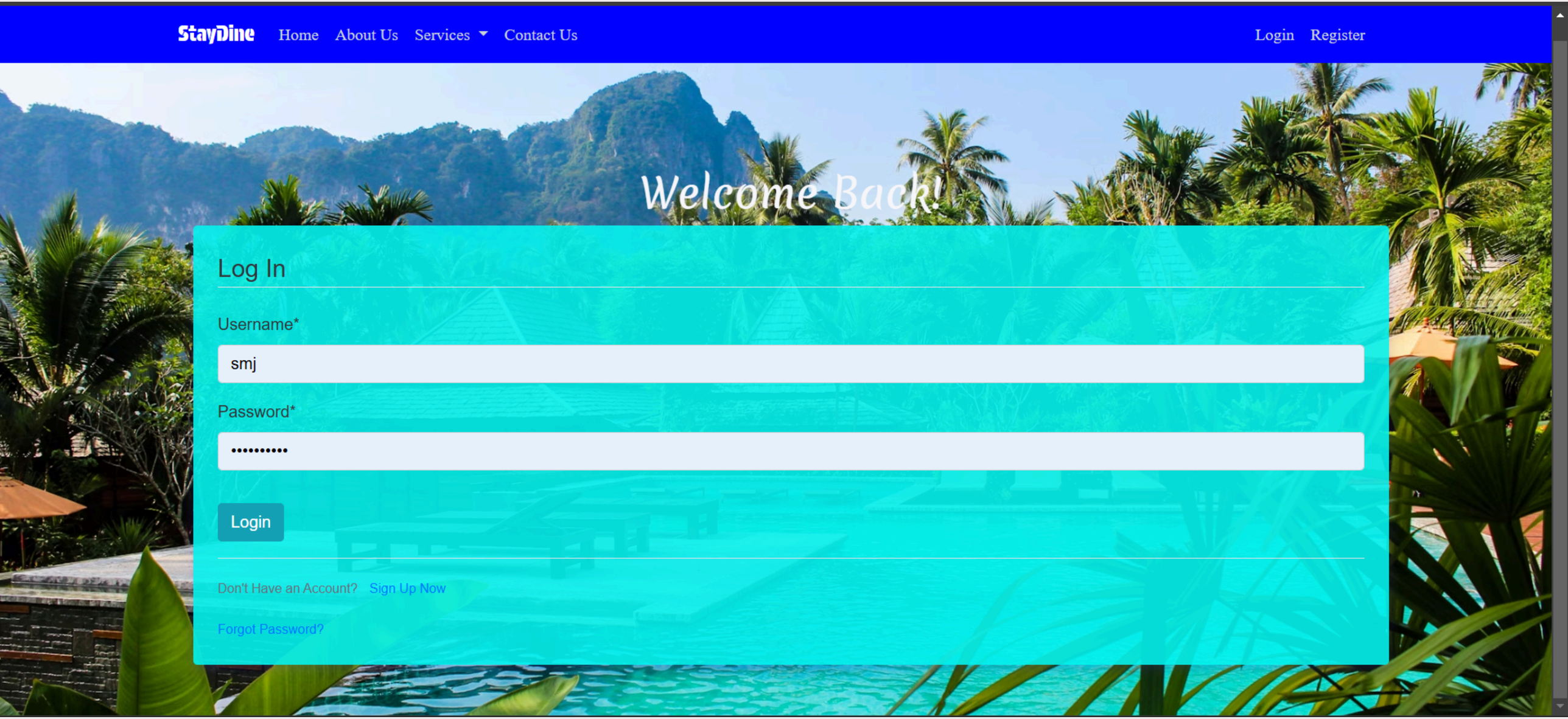
"Indulge in a culinary journey with dishes crafted to perfection. Every meal is an experience, blending flavors and freshness."

Exceptional Hospitality!

"Feel at home with our dedicated staff, ready to make your stay comfortable and memorable with thoughtful service."

Unmatched Comfort!

"Relax in our elegantly designed spaces, offering you the utmost in comfort, luxury, and peace for a restful retreat."


StayDine
Home
About Us
Services
Contact Us
Login
Register
Welcome Back!
Log In
Username*
smj
Password*
••••••••••
Login
Don't Have an Account?
Sign Up Now
Forgot Password?

Join Us!

Join Today

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Please fill out this field.

Email*

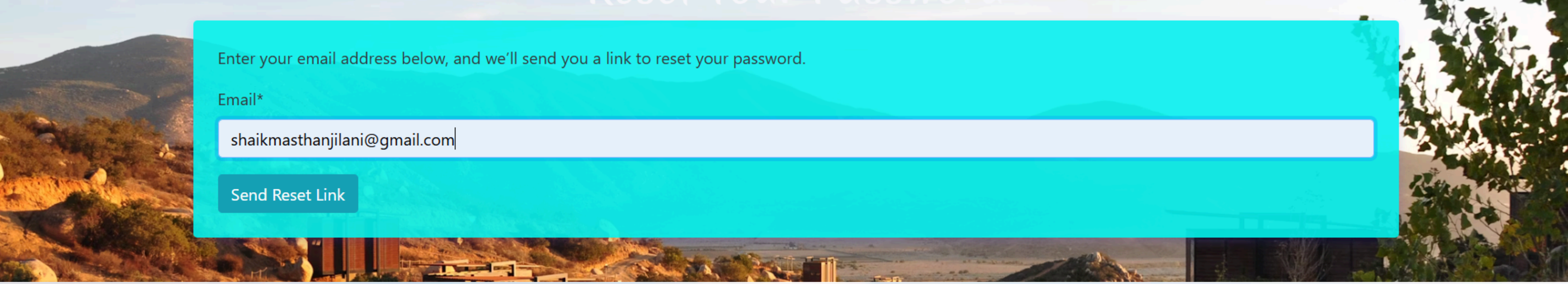
Password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password Updated

Your password has been updated successfully.

Back to Profile



Check Your Email

If an account exists with the email provided, a password reset link has been sent.

Return to Login

Password Updated

Your password has been updated successfully.

Back to Profile

About Us

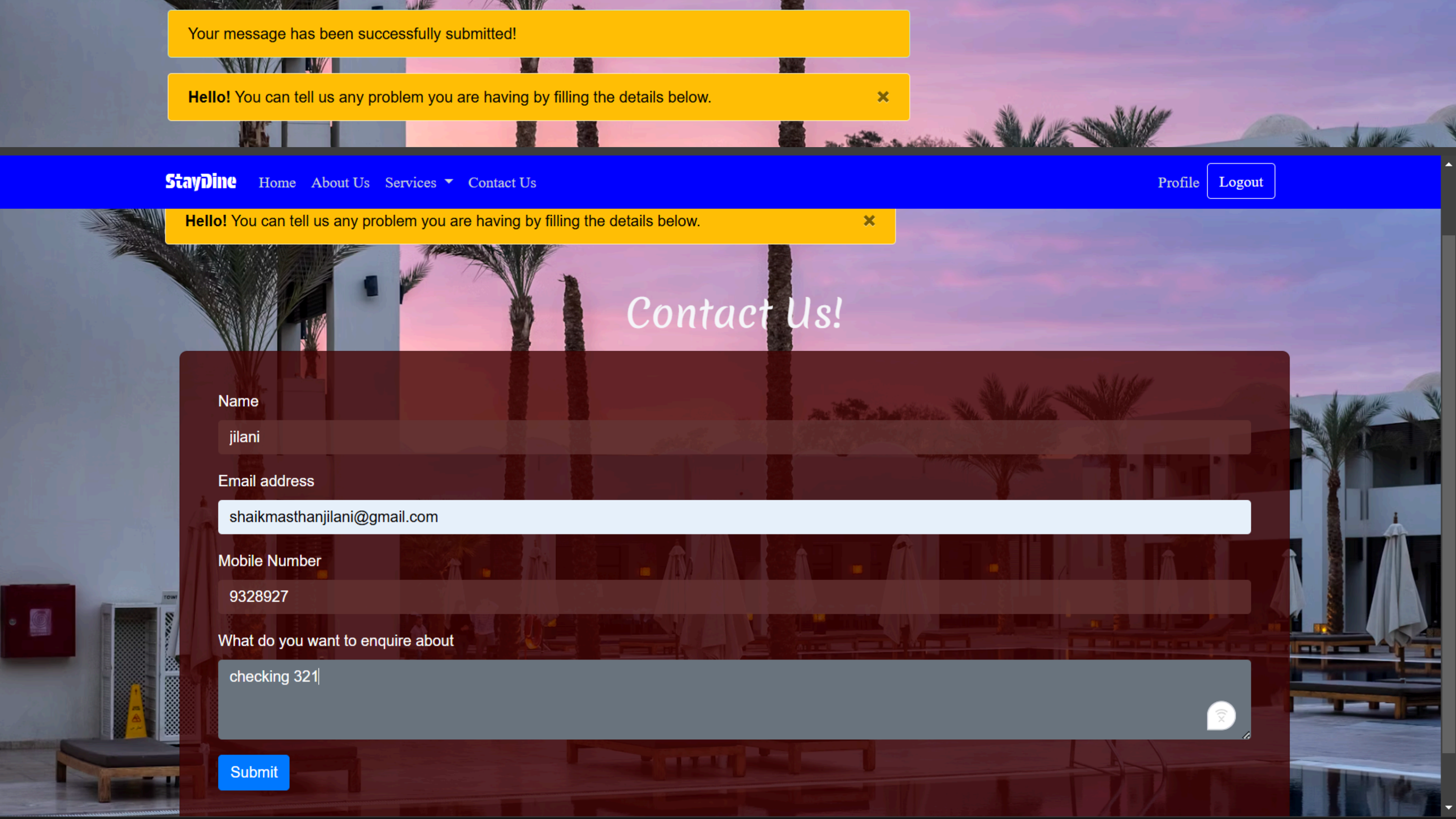
Welcome to our website "StayDine" Where Comfort Meets Culinary Delights!
We are dedicated to providing you with the best service possible, ensuring that every visit is memorable and enjoyable. Our commitment to quality extends beyond just our food and hospitality; we aim to create an environment where comfort meets exceptional culinary experiences.

At StayDine, we are passionate about what we do, and we strive for excellence in every project. Our team works tirelessly to ensure that our offerings are not only delicious but also innovative. We believe that food is not just sustenance; it is an experience that brings people together.

The Founders

Founder 1: Shaik Masthan Jilani
This is my first project on making a website, and I am very excited about it. StayDine is our dream project, and while there may not be much to show in this initial frontend portion, I am committed to working hard to make this website attractive and to utilize all the features necessary for a smooth user experience. Thank you for joining us on this journey!

Founder 2: Sohel
Hi, I am a student from IIT(BHU) Varanasi, pursuing BTech in CSE. I am thrilled to have such a wonderful opportunity to contribute to this project! With my technical background, I aim to integrate cutting-edge technology into our platform, making it more accessible and efficient for our users.


Your message has been successfully submitted!
Hello! You can tell us any problem you are having by filling the details below.
StayDine
Home
About Us
Services
Contact Us
Profile
Logout
Hello! You can tell us any problem you are having by filling the details below.
Contact Us!
Name
jilani
Email address
shaikmasthanjilani@gmail.com
Mobile Number
9328927
What do you want to enquire about
checking 321
Submit

Available Room Types

Presidential Suite

Price Per Night: ₹ 5000.00

The Presidential Suite is the epitome of luxury, designed for guests seeking an unforgettable experience. This expansive suite combines sophistication and comfort, with panoramic views, a spacious living area, and high-end furnishings. Ideal for dignitaries and VIPs, it offers exclusive amenities, including a private office, dining area, and concierge services to attend to every detail.

Superior Suite

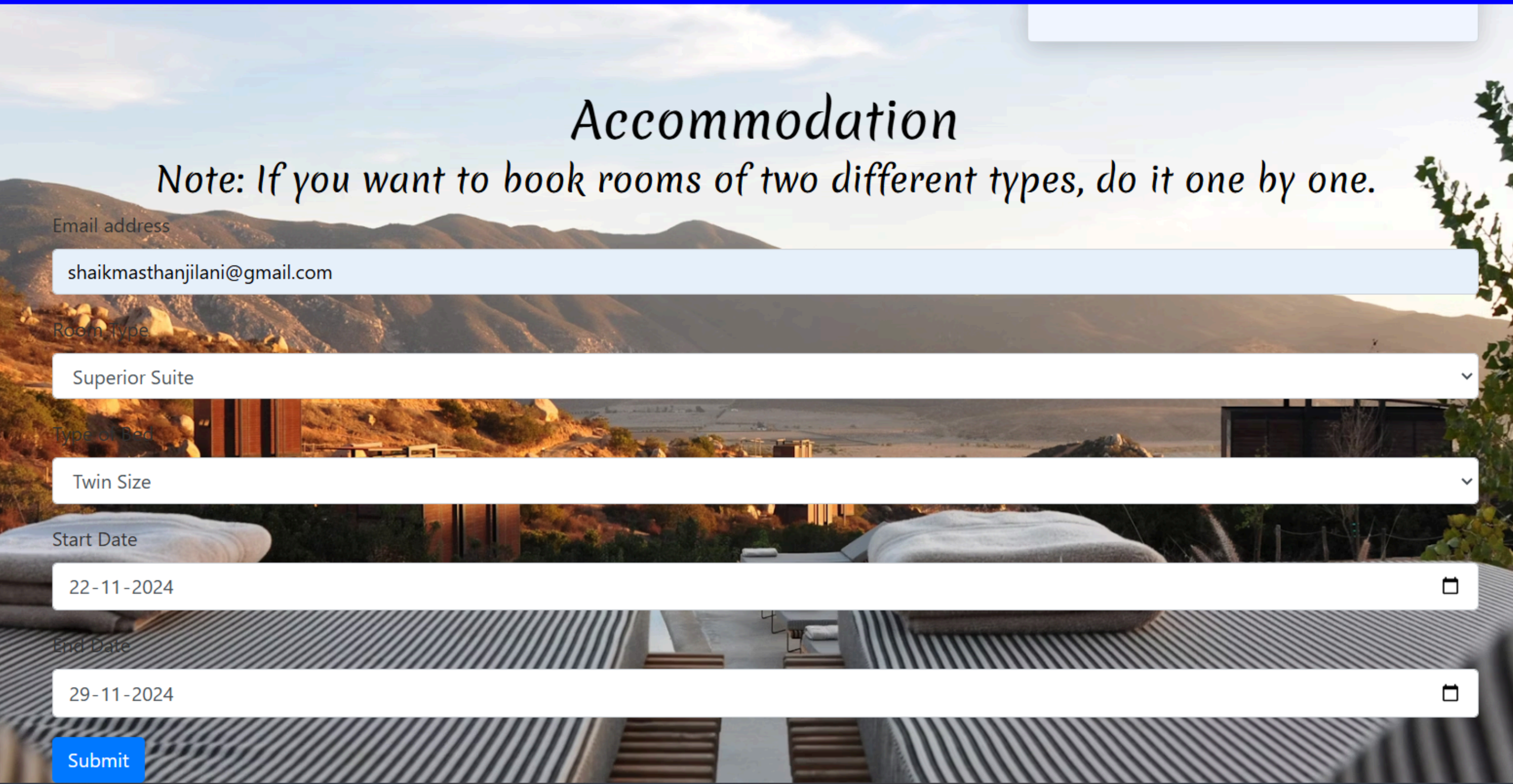
Price Per Night: ₹ 3000.00

The Superior Suite offers a cozy and practical space, perfect for solo travelers or couples looking for comfort without excess. Thoughtfully designed, it includes a queen-sized bed, a modern bathroom, and a dedicated workspace, providing everything needed for a restful stay. With warm decor and essential amenities, it's the ideal choice

Executive Suite

Price Per Night: ₹ 3999.99

Experience the pinnacle of sophistication in our Executive Suite. This elegant space is perfect for business travelers and offers a seamless blend of comfort and functionality. Enjoy modern amenities, a spacious work area, and stunning city views, ensuring every stay is productive and relaxing.



Accommodation

Note: If you want to book rooms of two different types, do it one by one.

Email address

shaikmasthanjilani@gmail.com

Room Type

Superior Suite

Type of Bed

Twin Size

Start Date

22-11-2024

End Date

29-11-2024

Submit

Menu & Order

Select items and specify quantities for your order.

Available Menu Items

Margherita Pizza

Category: Pizza

Price: ₹299.00

A classic favorite that never goes out of style! Topped with fresh mozzarella, fragrant basil, and a drizzle of extra virgin olive oil, our Margherita Pizza delivers a taste of Italy in

BBQ Chicken Delight

Category: Pizza

Price: ₹399.00

Get ready to savor the smoky goodness! This mouthwatering pizza features tender grilled chicken smothered in zesty barbecue sauce, complemented by red onions and cilantro

Spicy Pepperoni Extravaganza

Category: Pizza

Price: ₹349.00

Bring on the heat with our Spicy Pepperoni Extravaganza! Loaded with crispy, spicy

spices and yogurt, then simmered in a luscious tomato-butter gravy. Best enjoyed with naan or steamed rice, it's comfort food at its finest!

smokiness, these skewers are served with mint chutney and onion rings, making for a perfect appetizer or side dish!

a blend of spices. Garnished with fried onions and served with raita, this hearty meal is a feast for the senses!

Email address

Enter your email

Select Items to Order

Margherita Pizza - ₹299.00

Quantity

Enter quantity for Margherita Pizza

BBQ Chicken Delight - ₹399.00

Quantity

Enter quantity for BBQ Chicken Delight

Enter quantity for Mango Mousse

Butter Chicken - ₹350.00

Quantity

Enter quantity for Butter Chicken

Paneer Tikka - ₹250.00

Quantity

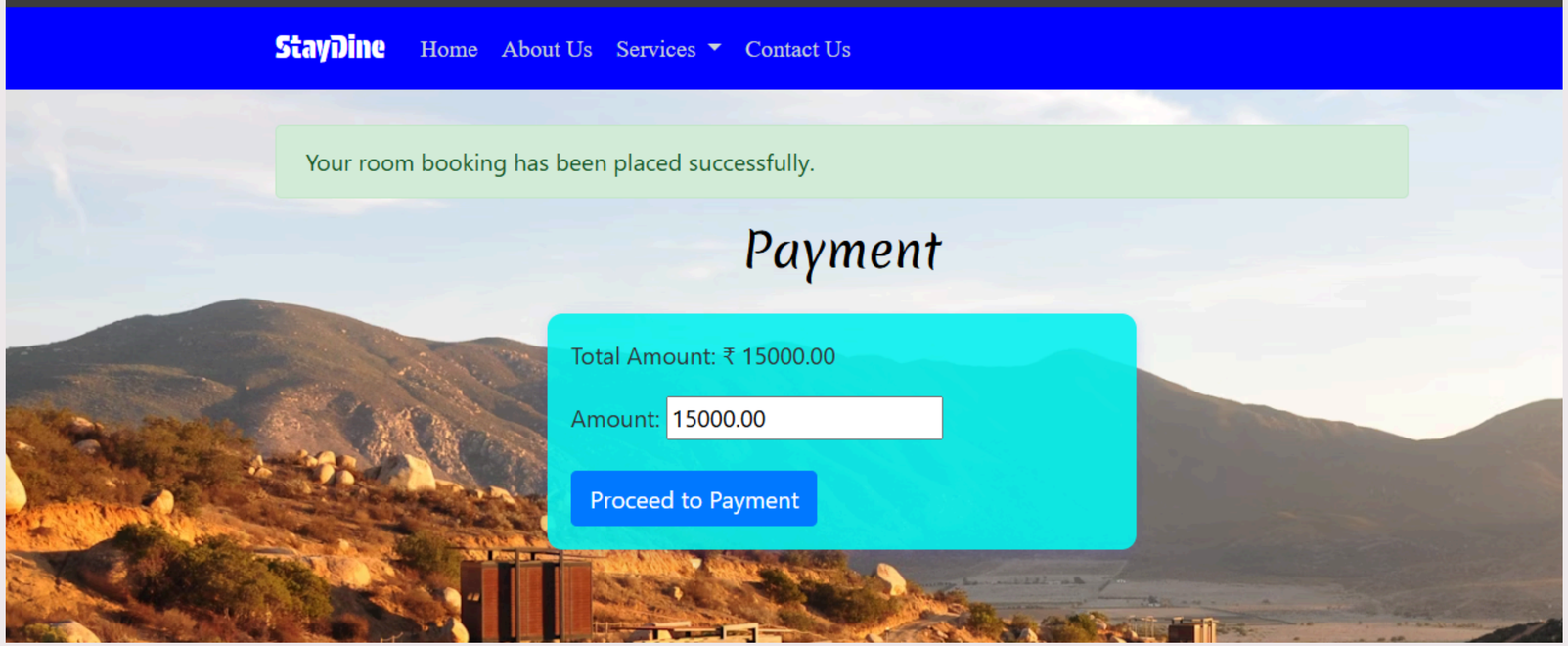
Enter quantity for Paneer Tikka

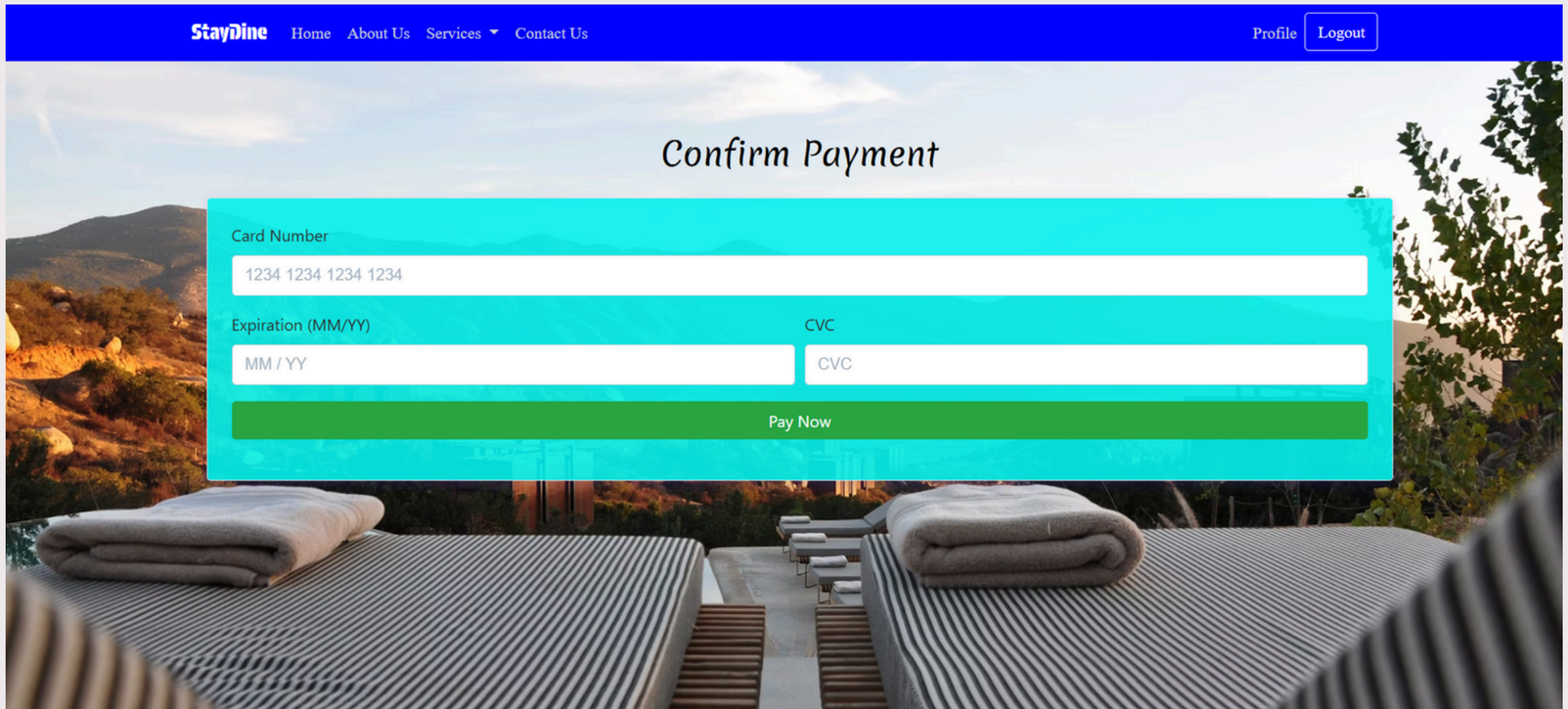
Biryani - ₹400.00

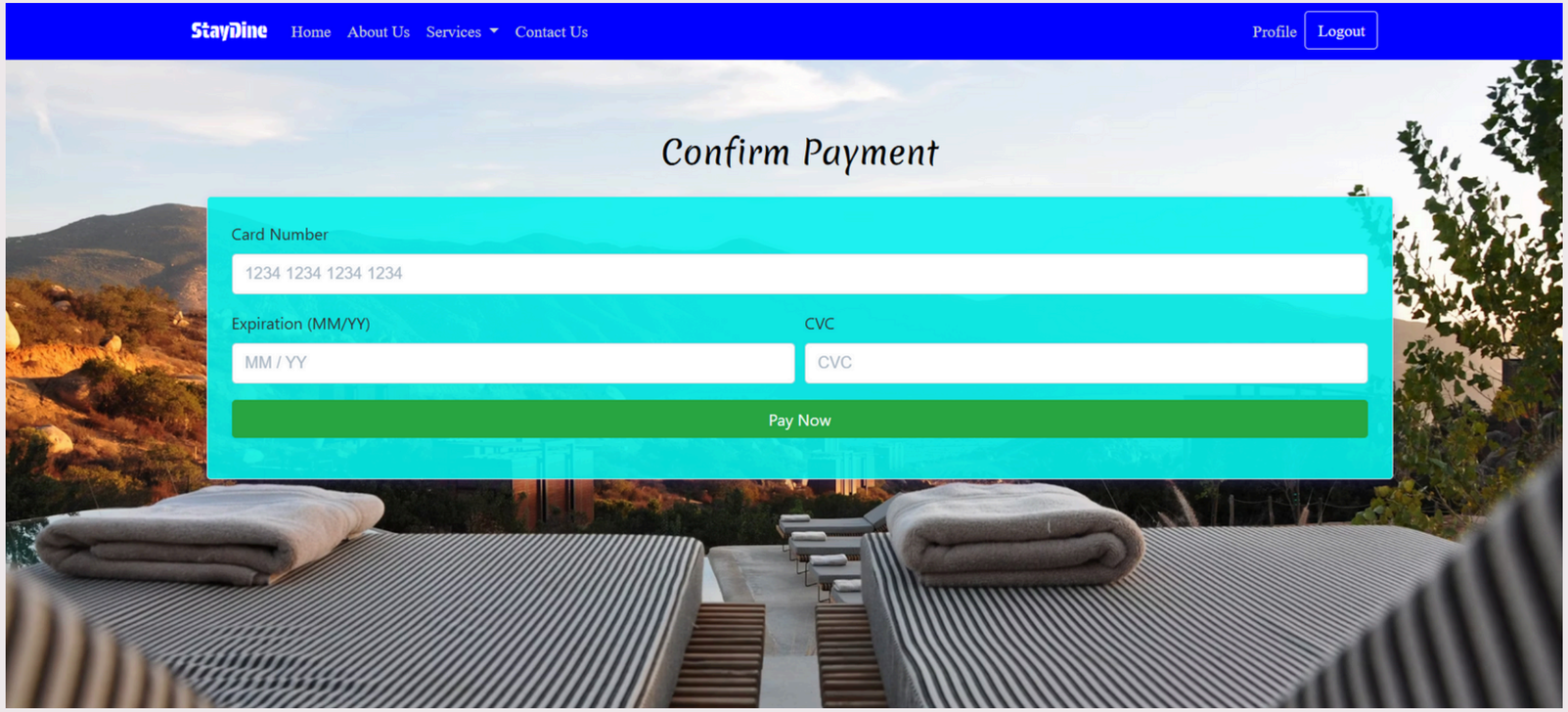
Quantity

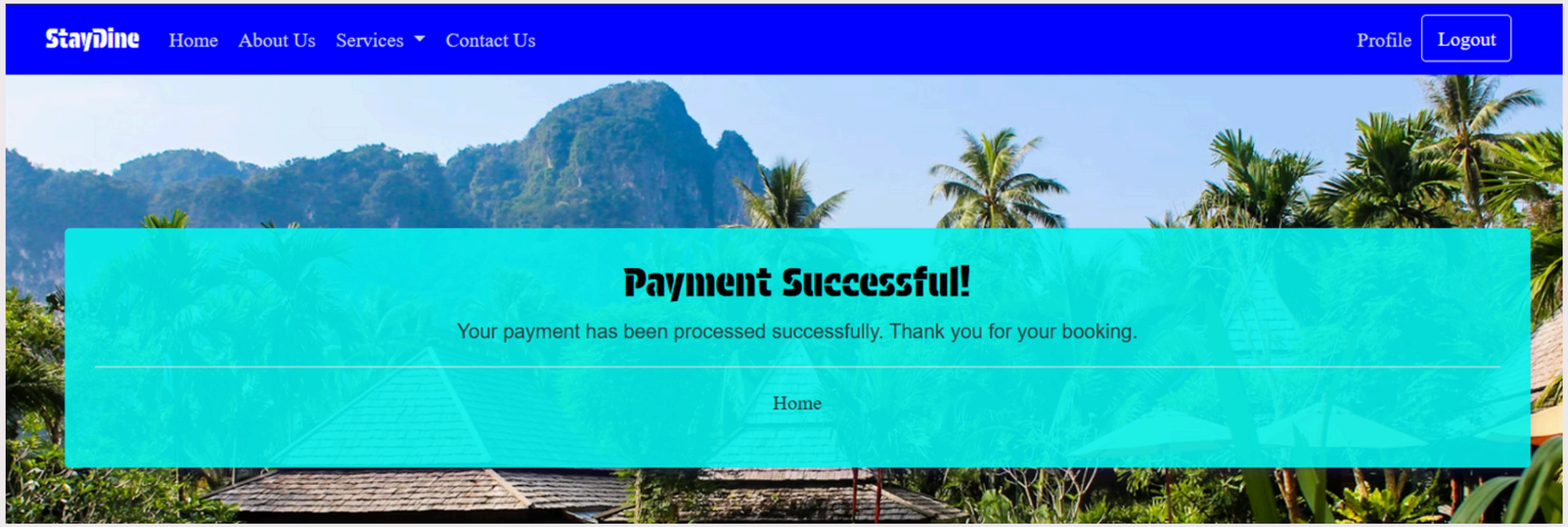
Enter quantity for Biryani

Place Order


StayDine
Home
About Us
Services
Contact Us
Your room booking has been placed successfully.
Payment
Total Amount: ₹ 15000.00
Amount: 15000.00
Proceed to Payment


StayDine
Home
About Us
Services
Contact Us
Profile
Logout
Confirm Payment
Card Number
1234 1234 1234 1234
Expiration (MM/YY)
MM / YY
CVC
CVC
Pay Now


StayDine
Home
About Us
Services
Contact Us
Profile
Logout
Confirm Payment
Card Number
1234 1234 1234 1234
Expiration (MM/YY)
MM / YY
CVC
CVC
Pay Now


StayDine
Home
About Us
Services
Contact Us
Profile
Logout
Payment Successful!
Your payment has been processed successfully. Thank you for your booking.
Home

Django administration

Site administration

ACCOUNTS		
Email addresses	+ Add	Change

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

SOCIAL ACCOUNTS		
Social accounts	+ Add	Change
Social application tokens	+ Add	Change
Social applications	+ Add	Change

STAYDINE		
Accommodations	+ Add	Change
Bed types	+ Add	Change
Contacts	+ Add	Change
Dinings	+ Add	Change
Highlightss	+ Add	Change
Menu items	+ Add	Change
Order items	+ Add	Change
Orders	+ Add	Change
Room types	+ Add	Change

USERS		
Payments	+ Add	Change
Profiles	+ Add	Change

Recent actions

My actions

- ✕ shaik_masthan_jilani
  User
- Google OAuth
  Social application
- \+ Biryani
  Menu item
- \+ Paneer Tikka
  Menu item
- \+ Butter Chicken
  Menu item
- Google OAuth
  Social application
- \+ Mango Mousse
  Menu item
- \+ Hot Chocolate Lava Brownie
  Menu item
- \+ Gulab Jamun
  Menu item
- \+ Chicken Pakora
  Menu item

Event Bookings

Christmas Event

Event ID: 1

Event Name: Christmas

Price: ₹50,000

Max Guests: 100

Easter Event

Event ID: 2

Event Name: Easter

Price: ₹40,000

Max Guests: 80

Good Friday Event

Event ID: 3

Event Name: Good Friday

Price: ₹30,000

Frontend Design

- ***Events Page***: A more detailed and clear event page, showcasing grand celebrations from past years.
- ***Staff Page***: For the staff, to find out the average income, recruitment and work assigned, tips earned and aura metre.
- ***Combining Frontend with Django***: Integrating the frontend with Django by creating templates and forms.
- ***Enhancing User Experience***: Using HTML, CSS for basic skeletal structure and styling, and JavaScript for interactivity.
- ***Adding Advanced Features***: Implementing user authentication, file handling, and other advanced features.

Django Plugins:

Crispy Forms: This plugin was used for better login, register and log out pages, which store the registered user's information in the Users table provided by the Django admin page.

Django Allauth: This plugin was added, to add features like auto sign in with google, using API provided by google cloud console. This was also used to change and modify the user’s password, by sending an email to recover the password.

Stripe: This plugin helps create a smooth payment gateway, with RuPay, MasterCard

Future Work (Completed)

- **Complete Frontend Integration**: To fully integrate the frontend with user authentication, reservation features, and event booking forms. Real-time availability checks for hotel rooms and restaurant tables.
- **Staff Management Interface**: Enhancing the admin panel where staff can manage bookings, assign rooms, and update restaurant tables.
- **Reporting & Analytics**: Implementing dashboards for hotel/restaurant managers to view analytics (e.g., total bookings, popular menu items, customer feedback).
- **Responsive Design**: Optimizing for mobile and tablet views and ensuring accessibility for visually impaired users.

Future Work (Completed)

- **Auto Signing in with Google**: Streamlining user experience with easy sign-in options.

- **Email Verification**: Secure user registration and login.
- **Live Chat Support**: Providing real-time support for customer queries.

- **Payment Gateway**: Adding a secure payment option for users to complete reservations and orders. Integrating UPI/Net Banking/Cash payment gateways.
- **Archiving Past Bookings**: To archive old data from the bookings table into auditing them to a new table, to store old data in a new table.

References

- SQL, Client-Server Architecture, from Lecture PPTs by Amrita Chaturvedi Ma’am.
- W3Schools for a bit of HTML, CSS and JavaScript, Online course from FreeCodeCamp.
- MDN Web Docs for in-depth guides and tutorials.
- Django Official Documentation.