# A PROJECT ON

**“EVENT MANAGEMENT SYSTEM”**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF

DIPLOMA IN ADVANCED COMPUTING FROM CDAC



#### SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

**Hinjawadi**

**SUBMITTED BY:**

GARIMA JADAUN

MOHIT DHAWALE

SWAPNIL PATIL

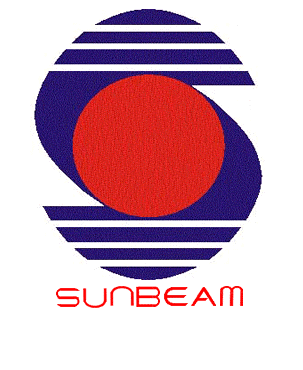
MINAL ZADE

**UNDER THE GUIDENCE OF:**

MRS LALITA SHINDE

Faculty Member

Sunbeam Institute of Information Technology, PUNE.



**CERTIFICATE**

This is to certify that the project work under the title ‘Event Management System’ is done by Garima Jadaun in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

|  |  |
| --- | --- |
| **Project Guide** | **Mr. Yogesh Kolhe** |
| **Mrs Lalita Shinde** | **Course Co-Coordinator** |

Date:01/09/2023

# A PROJECT ON

**EVENT MANAGEMENT SYSTEM**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

****

#### SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

**Hinjawadi**

**SUBMITTED BY:**

GARIMA JADAUN

MOHIT DHAWALE

SWAPNIL PATIL

MINAL ZADE

**UNDER THE GUIDENCE OF:**

MRS LALITA SHINDE

(Faculty Member)

Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

GARIMA JADAUN

MOHIT DHAWALE

SWAPNIL PATIL

MINAL ZADE

DAC March 2023 Batch,

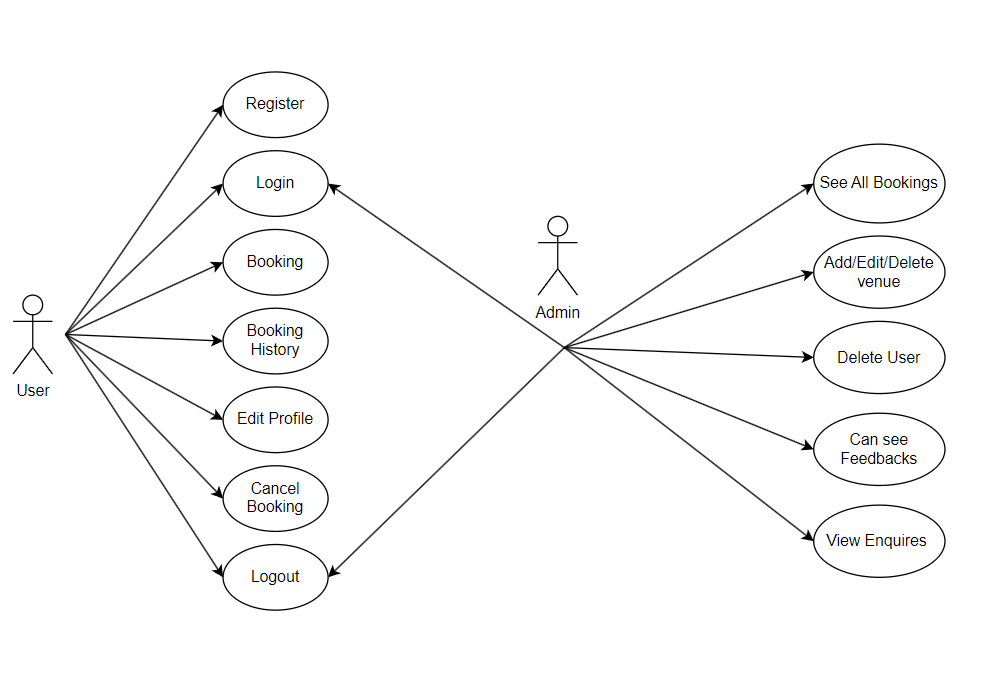
SIIT Pune

# INTRODUCTION TO PROJECT

Online Event management system was web-based application that enhanced project management to the creation and development of large scale events such as festivals, ceremonies, weddings, blood donation. It involved studying the brand, identifying its target audience, devising the event concept, and coordinating the technical aspects before actually launching the event. The Online Event management system (OEMS) enabled customers/ clients view various packages/products about the event and make booking through the online platform. The process of planing and coordinating the event is usually referred to as event planing and it included budgeting, scheduling, site selection, acquiring necessary permits, coordinating Transportation and parking, arranging for speakers or entertainers, arranging decor, event security, catering, coordinating with third party vendors, and emergency plans. Each event was different in its nature so process of planning & execution of each event differed on basis of type of event.

**2. REQUIREMENTS**

**2.1 FUNCTIONAL REQUIREMENTS**

****

**User Account**

The customer, who will henceforth be called the ‘user’, will be presented with two choices. A user can choose one of these and his choice would be governed by whether he is a guest or a registered user and whether he wants to check the availability of dates or also book them. The terms ‘registered user’ and ‘guest’ are described below.

A user who has booked the event earlier would have been given a user id and a password. This ‘personal information’ would be henceforth referred to as ‘profile’. Such a user with a profile in DB-user shall be called a ‘registered user’. A registered user will be able to check the availability of dates as well as book a event by logging into the system.

A new user, on the other hand, would either have to

1. register himself with the system by providing personal information or
2. log into the system as a guest.

In case of ‘a’, the new user becomes a registered user.

In case of ‘b’, the new user would remain a guest.

A guest can only check the availability of datesand images of events but cannot book events.

But a registered user can also act as a guest if he only wants to check the availability of Dates but registered user can book events.

The system shall present the user with an option to exit from the system at any time during the following processes.

**2.1.1Guest User:**

**1. Event Overview:**

- A guest user can browse events as a guest, viewing event images and brief descriptions of events here guest user cannot book events.

**2.1.2 User Registration:**

**1. Registration:**

- Decide to book an event and register by providing personal details: first name, last name, email, age, etc.

- Receive a confirmation email with login credentials after registration.

**2. Login:**

- Log in using the provided email and password.

**Booking an Event:**

**1. Event Selection:**

- Browse events after logging in.

- Choose from different event categories: festival (e.g., Ganesh Utsav, Navratri), ceremony (e.g., ring ceremony, name ceremony), blood donation, etc.

**2. Venue Selection:**

- Select an event from the chosen category.

- Choose a venue based on available options.

- Specify the event date and duration of the booking.

**3. Event Customization:**

- Customize the event further by selecting decoration themes and instrument setups.

- Choose a meal plan (e.g., royal, deluxe, vegetarian, non-vegetarian).

**4. Booking Confirmation:**

- Review the selected options.

- Click "Book Now" to proceed.

**Payment and Bill:**

**1. Billing Information:**

- Enter billing information securely.

**2. Bill Summary:**

-View a summary of the booking details, including event, date, venue, hotel, decoration, instruments, and meal choices.

**3. Print Bill:**

- Option to print a bill copy for record.

**Post-Booking:**

**1. Booking Confirmation:**

-Receive a confirmation email with event details, customization choices, and a reference code.

**2. Event Management:**

- Access the user account to manage booked events.

- View event details and make changes if within the cancellation period.

**3. Cancellation:**

- Initiate the cancellation process if within the allowed period (e.g., 15 days before the event).

- Apply refund policies as per the timing of cancellation.

**4. Feedback:**

- After the event, provide feedback to help organizers improve their system.



**2.1.3 Admin Management:**

**Event Management:**

1. Add Event:

- Admin can add new event details such as event name, type (festival, ceremony, blood donation), date options, and descriptions.

- Events can be categorized for easy navigation.

2. Update Event:

- Admin can modify event details, update dates, and make changes to event descriptions.

3. Delete Event:

- Admin can remove events that are no longer relevant or available.

**Instrument Management:**

1. Add Instrument:

- Admin can add new musical instruments available for event customization.

2. Update Instrument:

- Admin can update instrument details, descriptions, and availability.

3. Delete Instrument:

- Admin can remove instruments from the system if needed.

**Food Item Management:**

1. Add Food Item:

- Admin can add new food items for event meal customization (e.g., royal, deluxe, vegetarian, non-vegetarian).

2. Update Food Item:

- Admin can update food item details, descriptions, and options.

3. Delete Food Item:

- Admin can remove food items from the available choices.

* 1. **NON FUNCTIONAL REQUIREMENTS**

**2.2.1 Interface**

Go to Appendix B for user interfaces

**2.2.2 Performance**

**Number of Concurrent Users:**

ARS shall be able to handle at least 1000 transactions/inquiries per second

**Booking of Tickets:**

The system is susceptible to any temporary server failure since it uses the strong feature of Struts 2 and Hibernate. Hence the examination will be continued even if the sever gets disconnected in between the examination.

**2.2.3 Constraint**

ARS shall be able to handle at least 1000 transactions/inquiries per second

**2.2.4 Other Requirements:**

**Hardware Interfaces**

The SPMS is expected to function on Intel PIII 900 MHz Processor equivalent or above, 128 MB RAM, 10 GB HDD.

**Software Interface The SPMS shall work on MS Windows operating systems family (MS Windows 98, MS Windows NT Workstation, MS Windows 2000, MS Windows XP). It configures to work with MySQL database. This System works on Apache Tomcat server. It uses browser IE 5.0 & above. It uses IIS 5.0 server.**

**3. DESIGN**

**3.1 Database Design**

The following table structures depict the database design.

# Table1: FeedBack User List Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| 3 | Feedback\_UserLidt\_id | BigInt | 8 | 0 |
| 0 | User\_list\_id | BigInt | 8 | 0 |

## **Table2: Feedback Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | BigInt | 8 | 0 |
| 0 | Email | Varchar | 20 | 1 |
| 0 | Subject | Varchar | 255 | 1 |
| 0 | Message | Varchar | 255 | 1 |
| 3 | Venue\_Id | BigInt | 8 | 0 |
| 0 | Rating | BigInt | 8 | 0 |

# Table3: User Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | BigInt | 8 | 0 |
| 0 | RoleId | BigInt | 8 | 1 |
| 0 | FirstName | Varchar | 20 | 1 |
| 0 | LastName | Varchar | 20 | 1 |
| 0 | Gender | Varchar | 10 | 1 |
| 0 | Dob | Date | 8 | 1 |
| 0 | Email | Varchar | 20 | 1 |
| 0 | MobNo | Varchar | 15 | 1 |
| 0 | Password | Varchar | 255 | 1 |
| 0 | State | Varchar | 20 | 1 |
| 0 | City | Varchar | 20 | 1 |

## **Table4: Venue Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | BigInt | 8 | 0 |
| 0 | Costing | Double | 8 | 1 |
| 0 | City | Varchar | 25 | 1 |
| 0 | State | Varchar | 25 | 1 |
| 0 | Description | Varchar | 200 | 1 |
| 0 | Name | Varchar | 25 | 1 |

### Table5: Bill Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | BigInt | 8 | 0 |
| 0 | Amount | Double | 8 | 0 |
| 0 | BillDate | Date | 8 | 0 |
| 0 | Booking\_Id | Int | 4 | 0 |
| 0 | Payment\_Type | Varchar | 10 | 0 |
| 0 | Total\_Amount | Double | 8 | 0 |
| 0 | User\_billId | BigInt | 8 | 0 |

# Table6: Booking Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | Int | 4 | 0 |
| 0 | Equipment | Varchar | 30 | 0 |
| 0 | Food\_type | Varchar | 30 | 0 |
| 0 | Booking\_date | Date | 8 | 0 |
| 0 | End\_Date | Date | 8 | 0 |
| 0 | Event | Varchar | 30 | 0 |
| 0 | No\_Of\_Guest | Int | 4 | 0 |
| 0 | Start\_Date | Date | 8 | 0 |
| 0 | Status | Varchar | 30 | 0 |
| 0 | User\_booking\_Id | BigInt | 8 | 0 |
| 0 | Venue\_Id | BigInt | 8 | 0 |

### Table7: Cancellation Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | BigInt | 8 | 0 |
| 0 | Reason | Varchar | 255 | 1 |
| 0 | Description | Varchar | 255 | 1 |
| 0 | Refundable\_Amount | Double | 8 | 0 |
| 0 | Cnc\_date | Date | 8 | 0 |
| 3 | Booking\_Id | BigInt | 8 | 0 |
| 3 | Bill\_Id | BigInt | 8 | 0 |

## **Table8: Contact\_Us**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Id | BigInt | 8 | 0 |
| 0 | Email | Varchar | 20 | 0 |
| 0 | Message | Varchar | 25 | 1 |
| 0 | Name | Varchar | 20 | 1 |

**E-R Diagram,Dataflow diagram and Class Diagram:**

Go to Appendix A

**4. CODING STANDARDS IMPLEMENTED**

### Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | Person, BankVault, SMSMessage, Dept | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type prefixes like ‘C’ for class. |
| Method | Camel | getDetails, updateStore | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | personName, bankCode | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios. |
| Interface | Pascal with "I" prefix | Disposable | Do not use the ‘\_’ sign |
| Property | Pascal | ForeColor, BackColor | Use a noun or noun phrase to name properties. |
| Associated private member variable | \_camelCase | \_foreColor, \_backColor | Use underscore camel casing for the private member variables |
| Exception Class | Pascal with "Exception" suffix | WebException, |  |

### Comments

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

**5. TEST REPORT**

**Another group called Linux did the testing and the report of the testing is given hereunder.**

**GENERAL TESTING:**

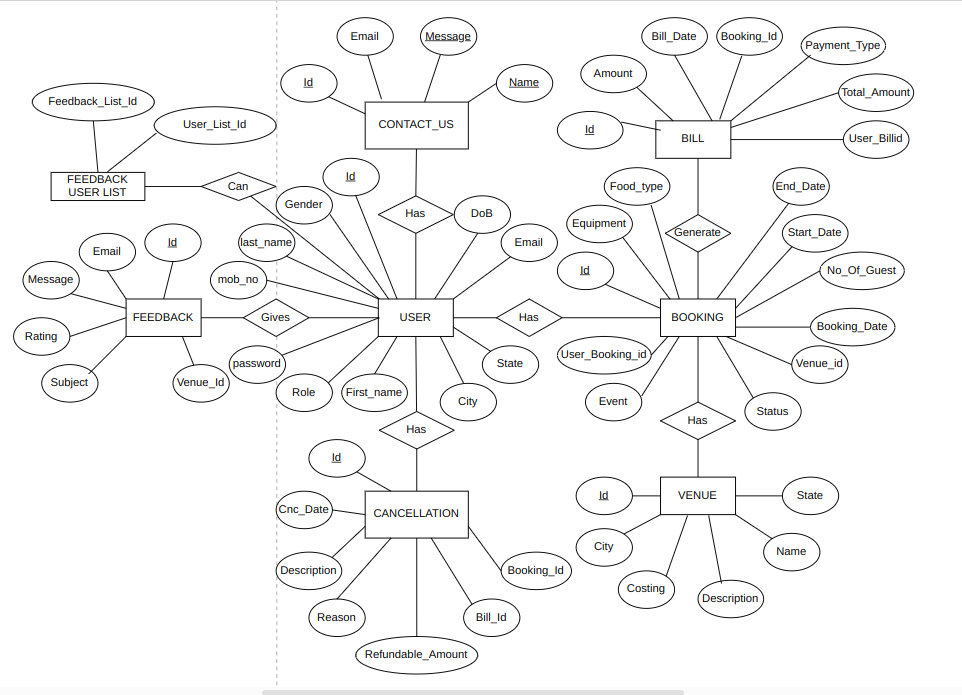
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | Register Page | Redirected to Next page | OK | Nothing |
| 2 | Login Page | Pop-up will come | Ok | Please enter username and password again . |
| 4 | Quick search Venue | Gives all venue details | Ok | Nothing |
| 5 | Booking Ticket | All the fields should be filled for submission | Ok | Nothing |
| 6 | Checking login or not | User is logged in or not | Ok | Nothing |
| 7 | Add booking details for booking event | Add informations according to venue and booking | Ok | Nothing |
| 8 | Goto Profile page | Set added information about person | Ok | Nothing |
| 9 | Add information in booking table | Save this all data into booking table | Ok | Nothing |
| 10 | Transaction | On back it should be reverted to previous page | Ok | Nothing |
| 11 | View transaction done | It shows you all transactions done previously | Ok | Nothing |
| 12 | Logout | It will logout from user profile. | Ok | Nothing |
|  | **STATIC TESTING** |  |  |  |
| **SR-NO** | **Deviation** | **Program** |  |  |
| 1 | Commenting not followed | All Web Application |  |  |

**6. PROJECT MANAGEMENT RELATED STATISTICS**

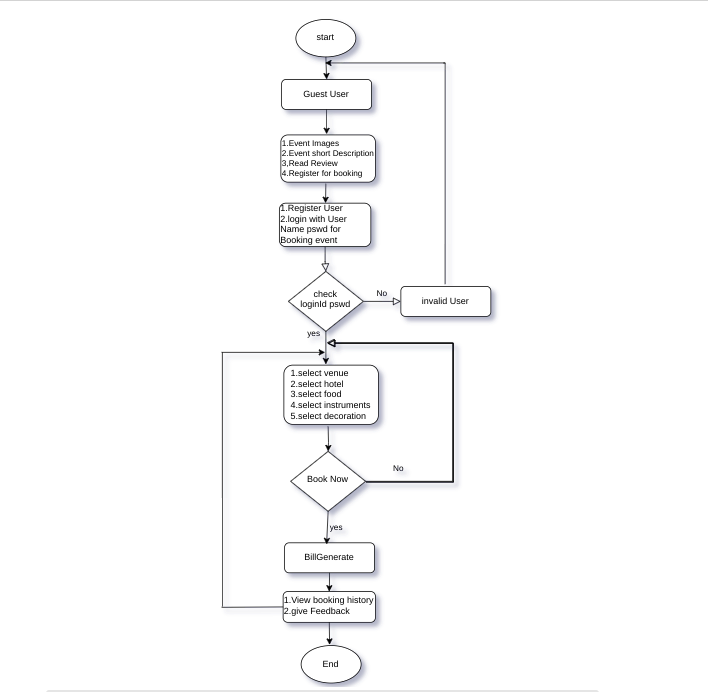
|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **WORK PERFORMED** | ****SLC Phase**** | **Additional Notes** |
| APRIL 12,2023 | Project Allotment and User Requirements Gathering | Feasibility Study | Our team met the client Mr. Nitinkudale (CEO, SIIT Pune) to know his requirements. |
| APRIL 17,2023 | Initial SRS Document Validation  And Team Structure Decided | Requirement Analysis  (Elicitation) | The initial SRS was presented to the client to understand his requirements better |
| MAY 18,2023 | Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces | Requirement Analysis &  Design Phase | Database Design completed |
| JUNE 25,2023 | Business Logic Component design Started | Design Phase | ---------------------- |
| AUG 16-18,2023 | Coding Phase Started | Coding Phase | 70% of Class Library implemented. |
| AUG 19,2023 | Implementation of Web Application and Window Application Started | Coding Phase | Class Library Development going on. |
| AUG 20, 2023 | Off | Off | Off |
| AUG 21, 2015 | Implementation of Web Application and Window Application Continued | Coding Phase and Unit Testing | Class Library Modified as per the need. |
| AUG 24, 2023 | Implementation of Web Application and Window Application Continued | Coding Phase and Unit Testing | -- |
| AUG 26, 2023 | After Ensuring Proper Functioning the Required Validations were Implemented | Coding Phase and Unit Testing | Module Integration was done by the Project Manager |
| AUG 26, 2023 | The Project was Tested by the respective Team Leaders and the Project Manager | Testing Phase (Module Testing) | -- |
| AUG 27,2023 | Off | Off | Off |
| AUG 26, 2023 | The Project was Submitted to Other Project Leader of Other Project Group For Testing | Testing Phase (Acceptance Testing) | The Project of Other Team was Taken up by the Team for Testing |
| AUG 28-29, 2023 | The Errors Found were Removed | Debugging | The Project was complete for submission |
| AUG 30, 2023 | Final Submission of Project |  |  |

Appendix A:

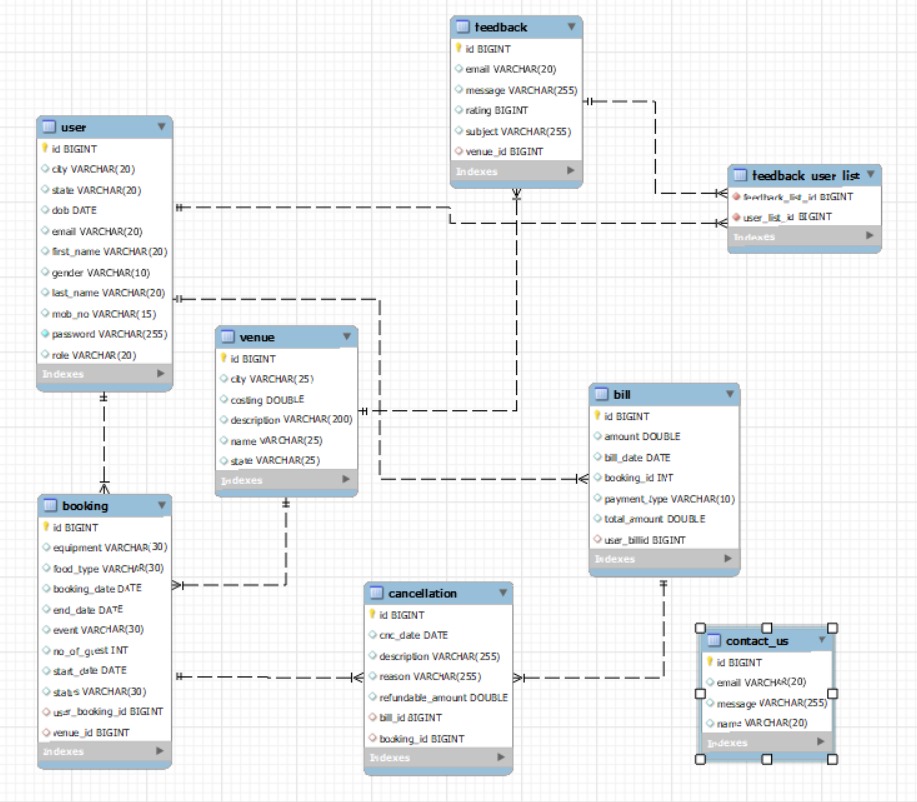
Entity Relationship Diagram:



**Data Flow Diagram:**

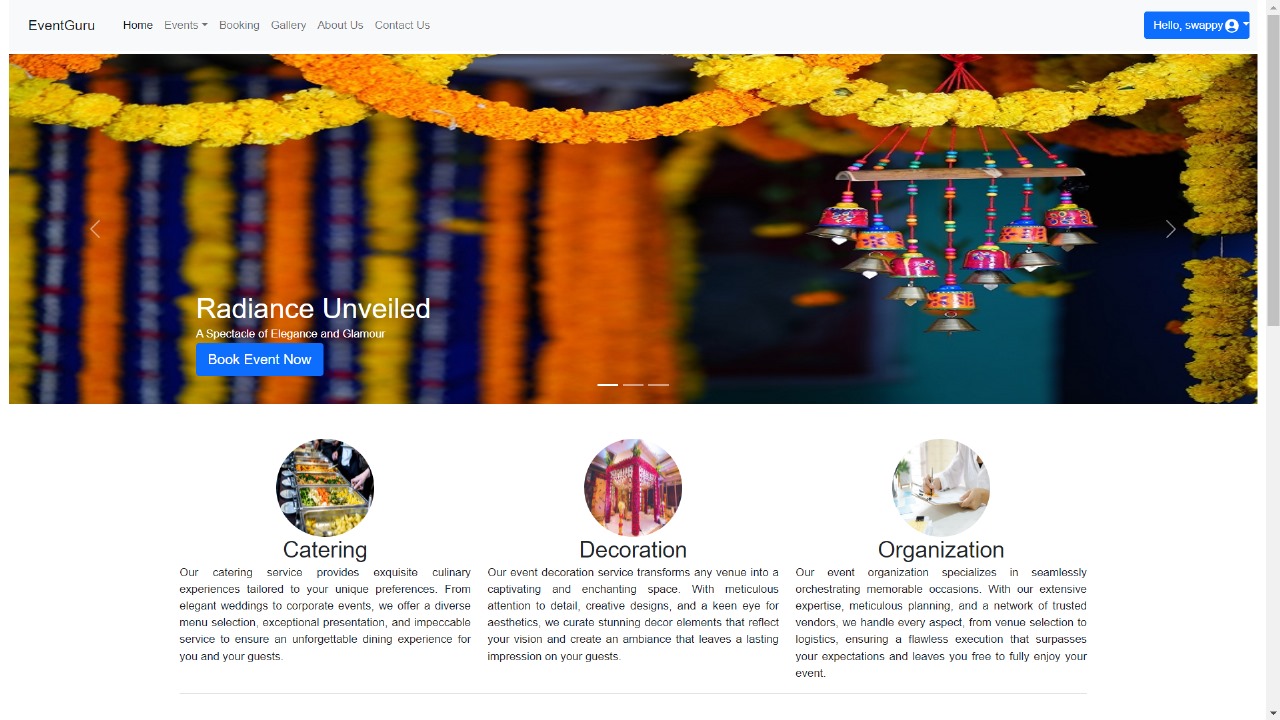
****

**Class Diagram :**

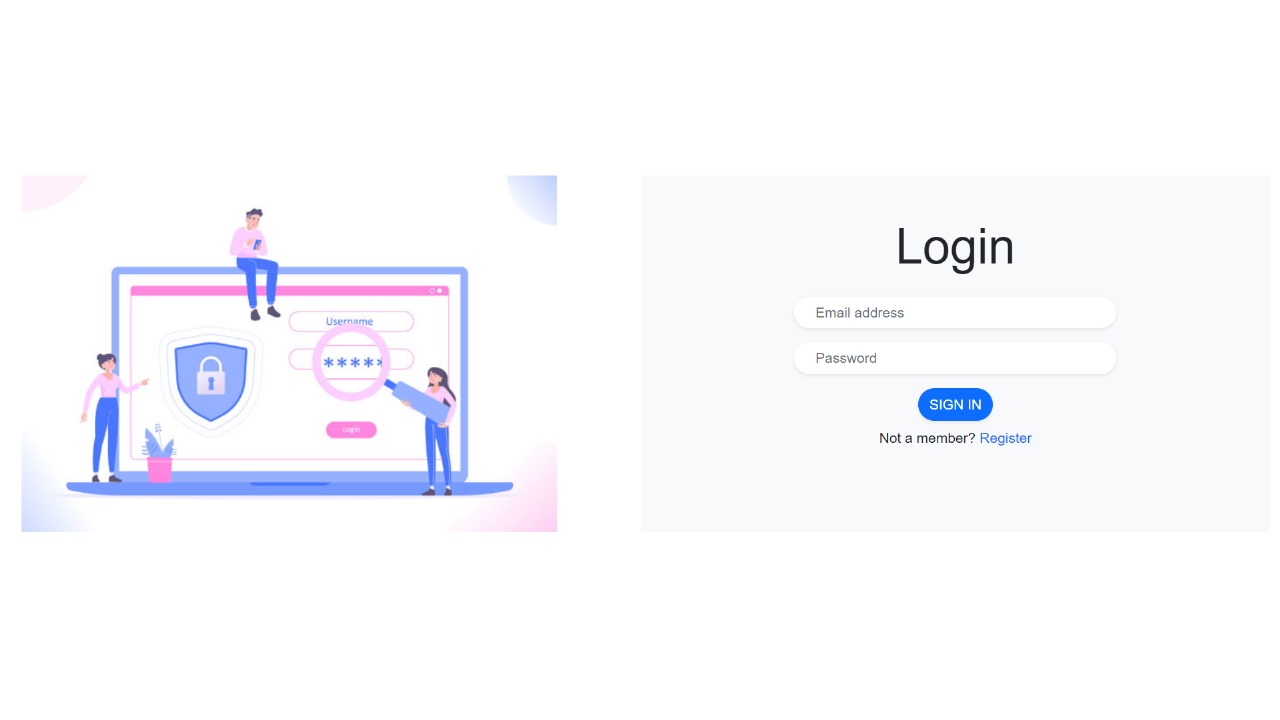
****

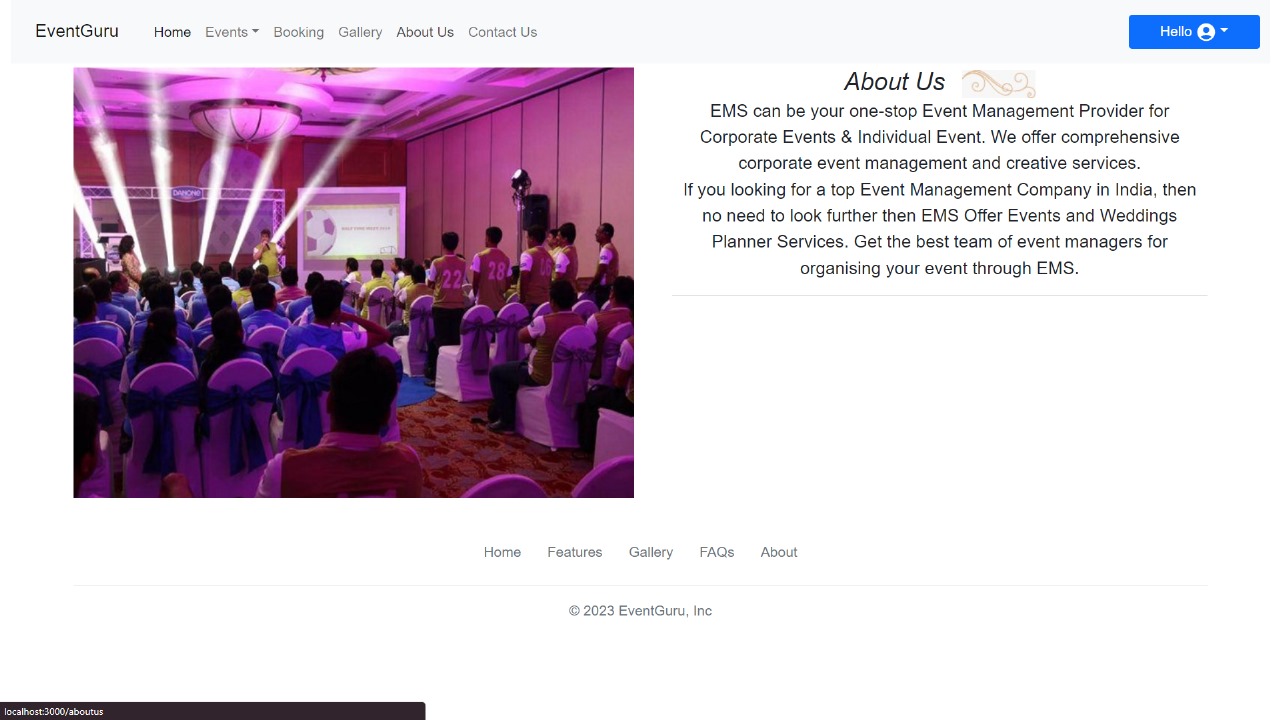
Appendix B

Home Page:

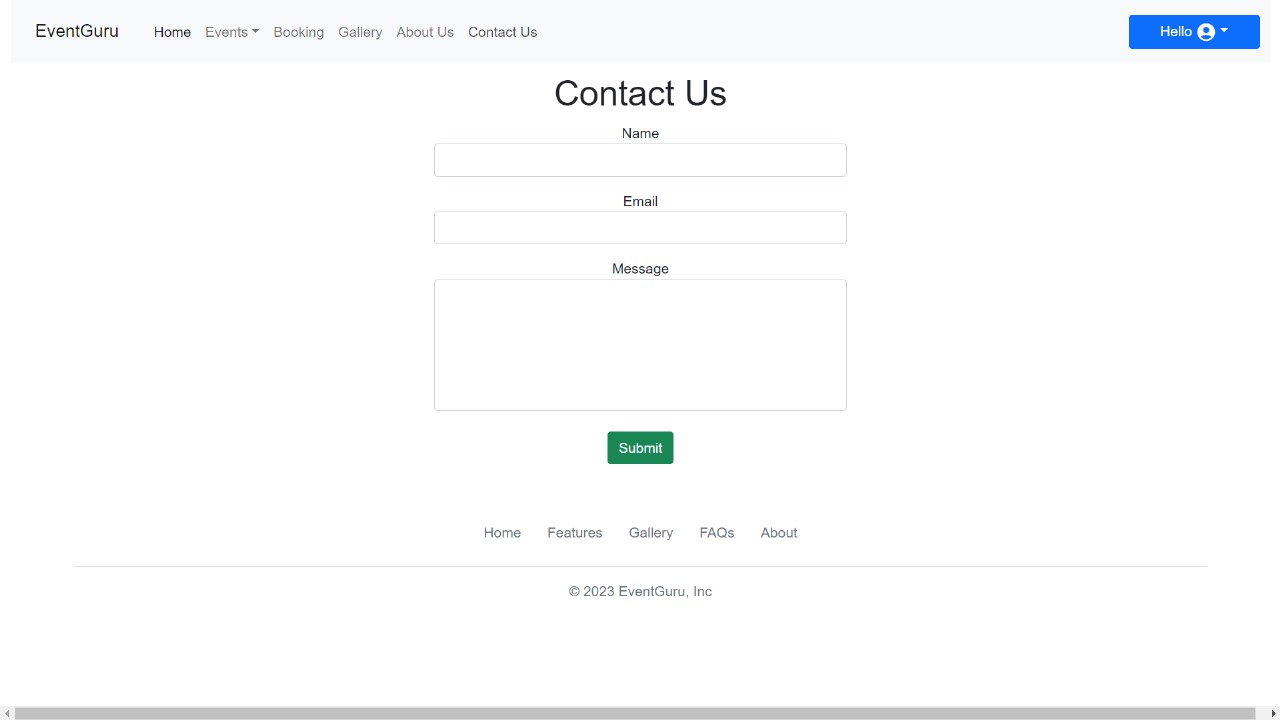


Login Page:

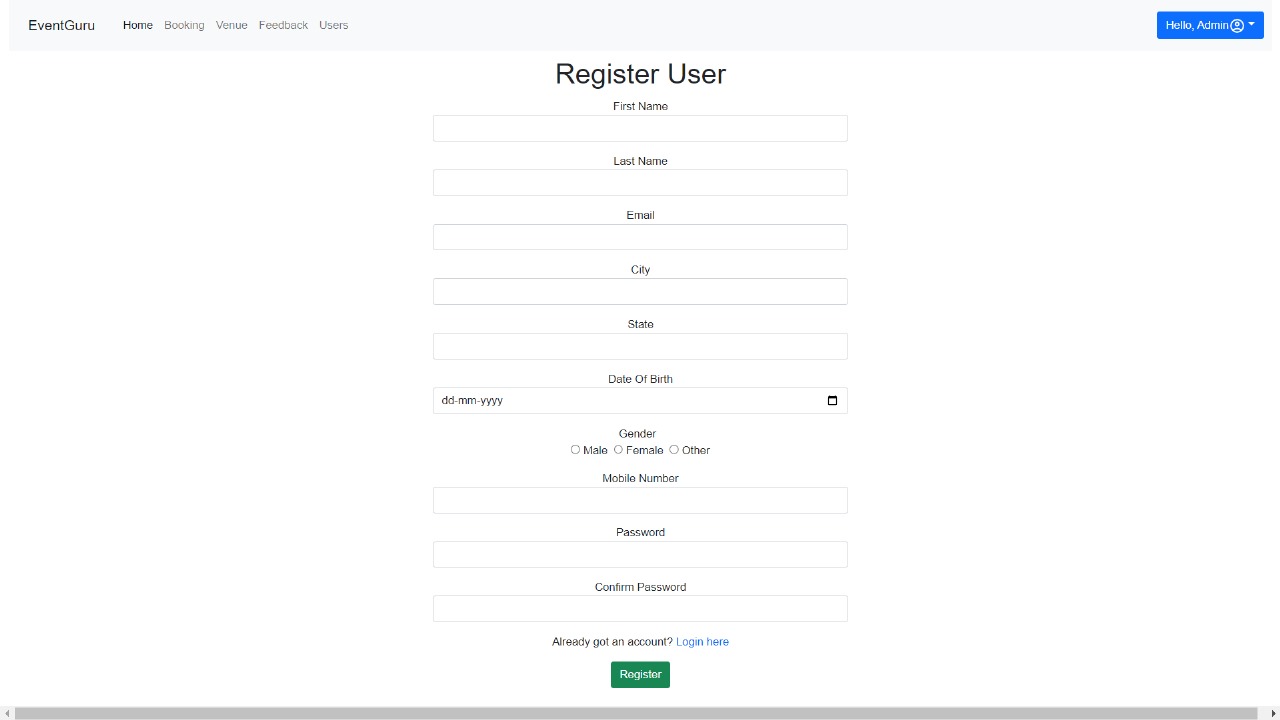


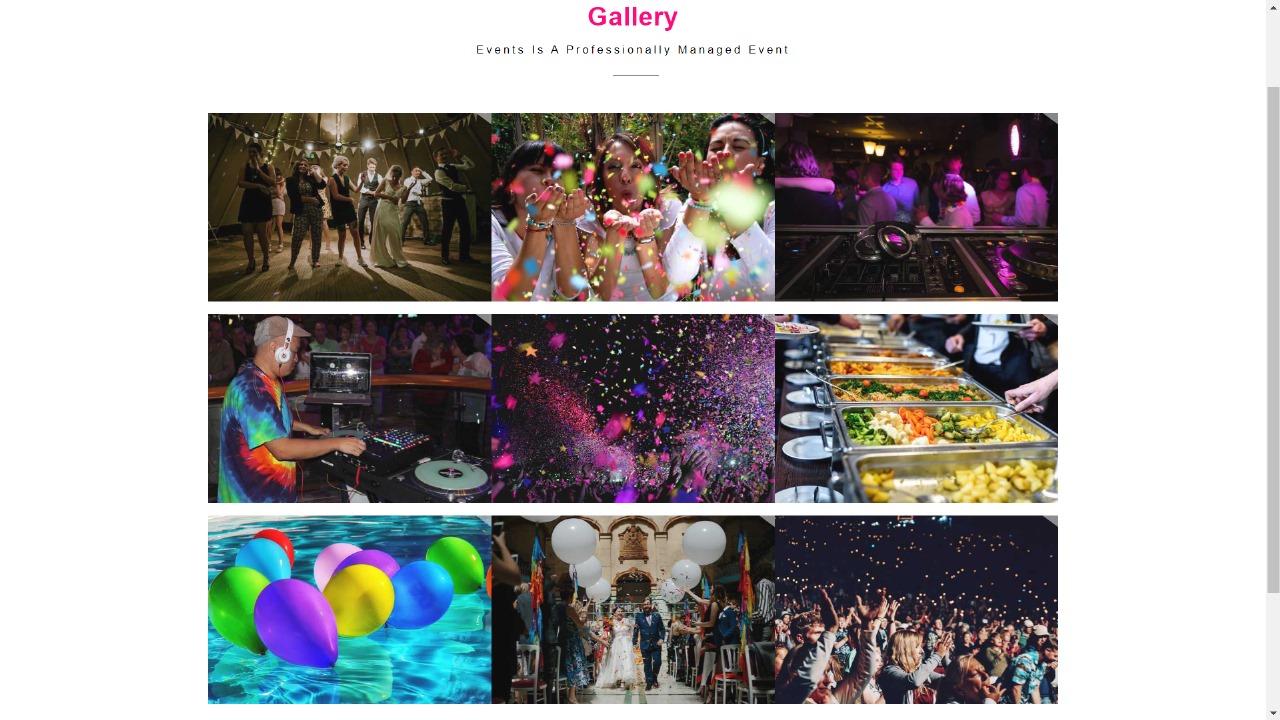
AboutUs:

ContactUs:

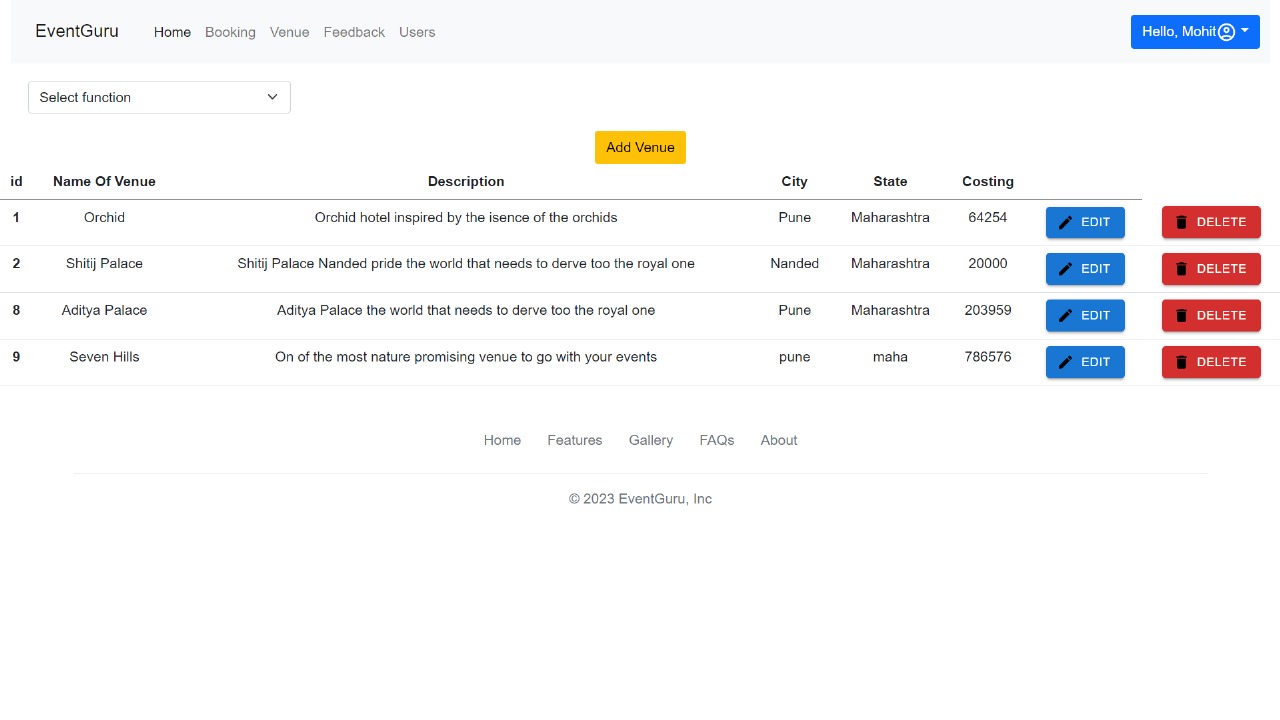


Registration:

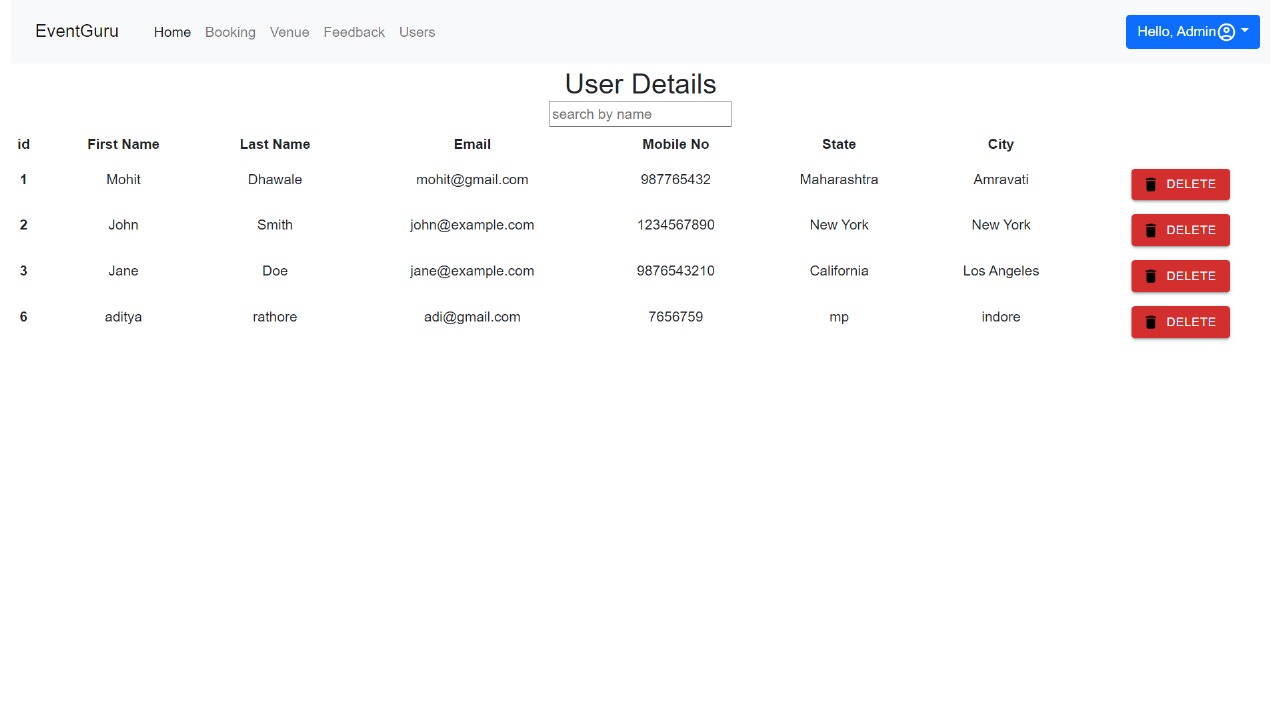
Gallery:



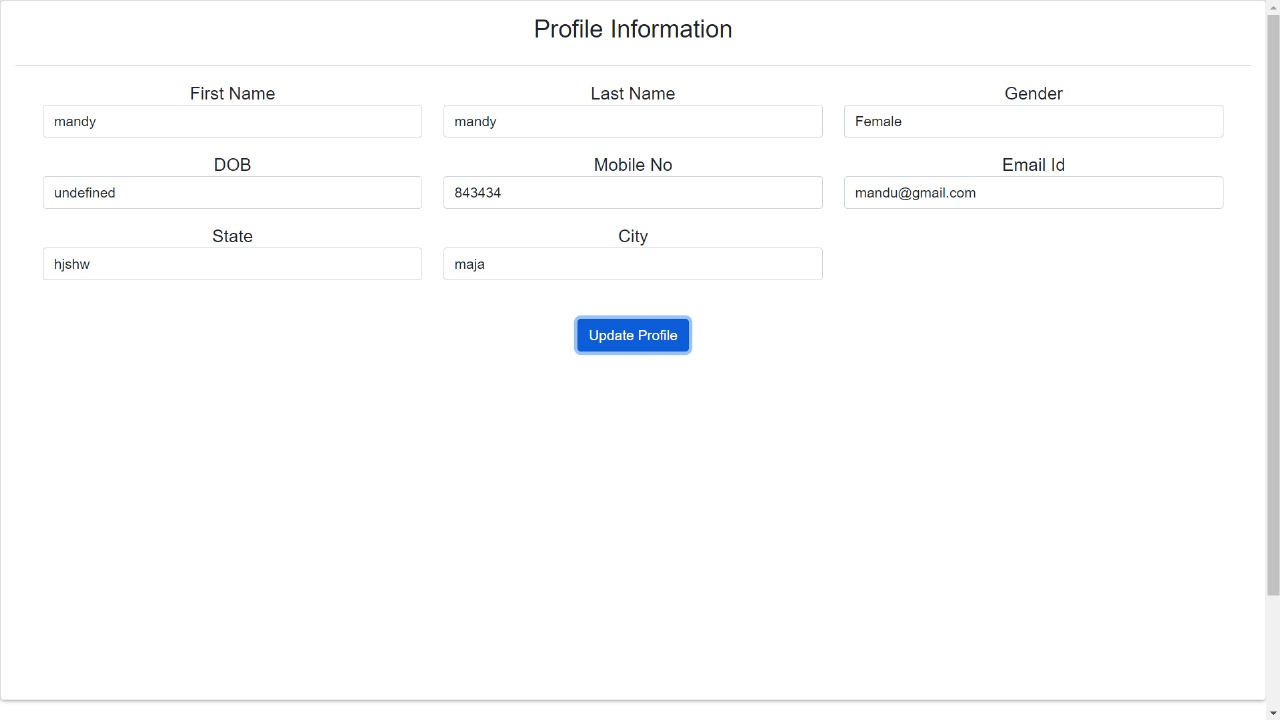
Add/Delete Venue:



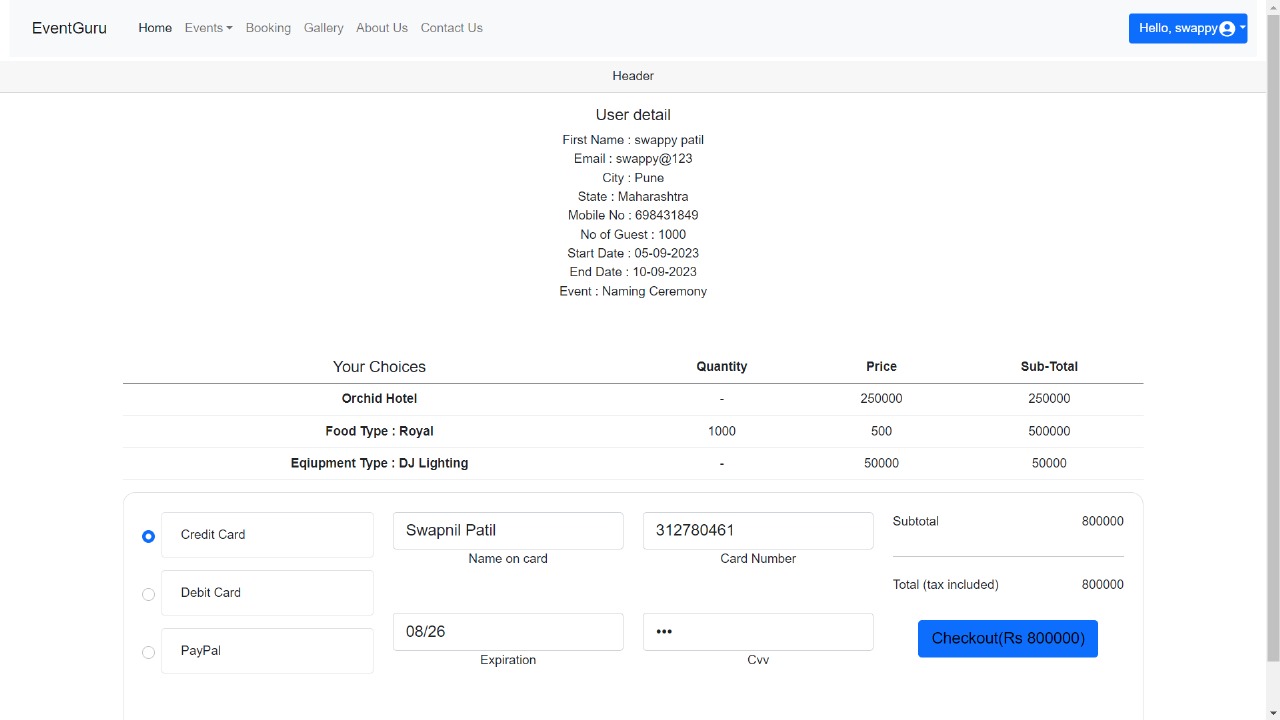
Get/Delete User Details:



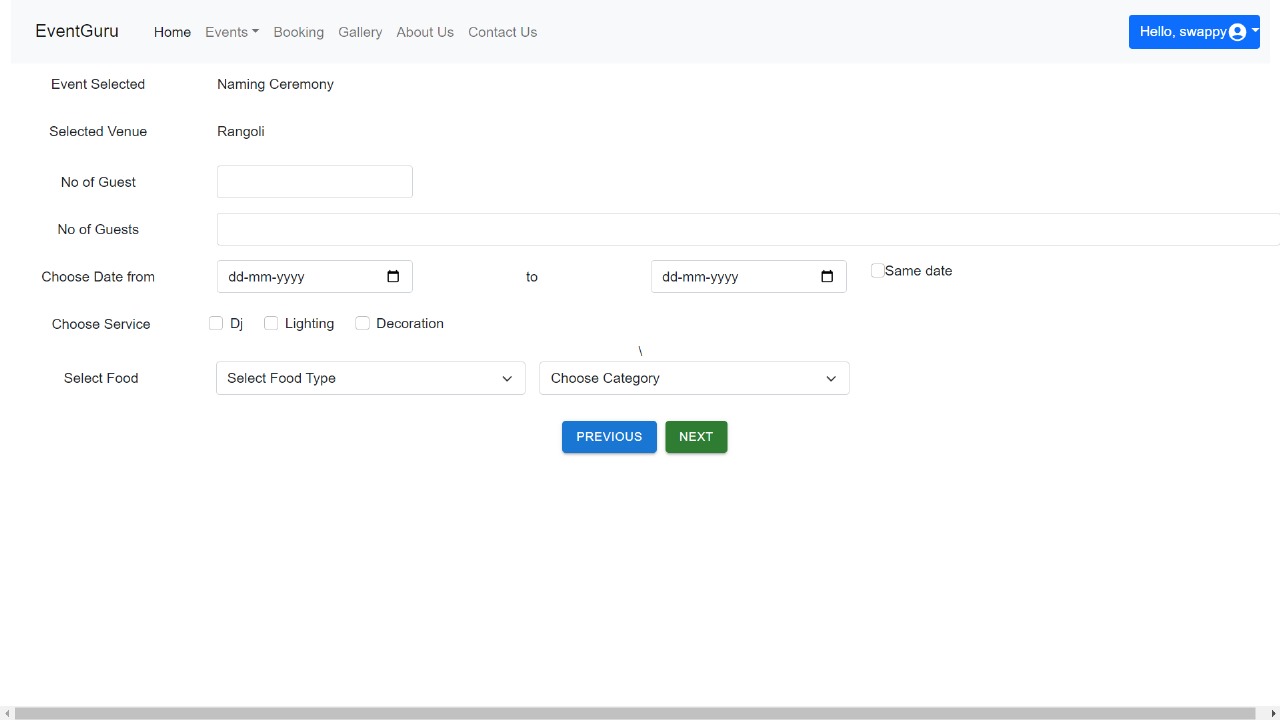
User Profile:



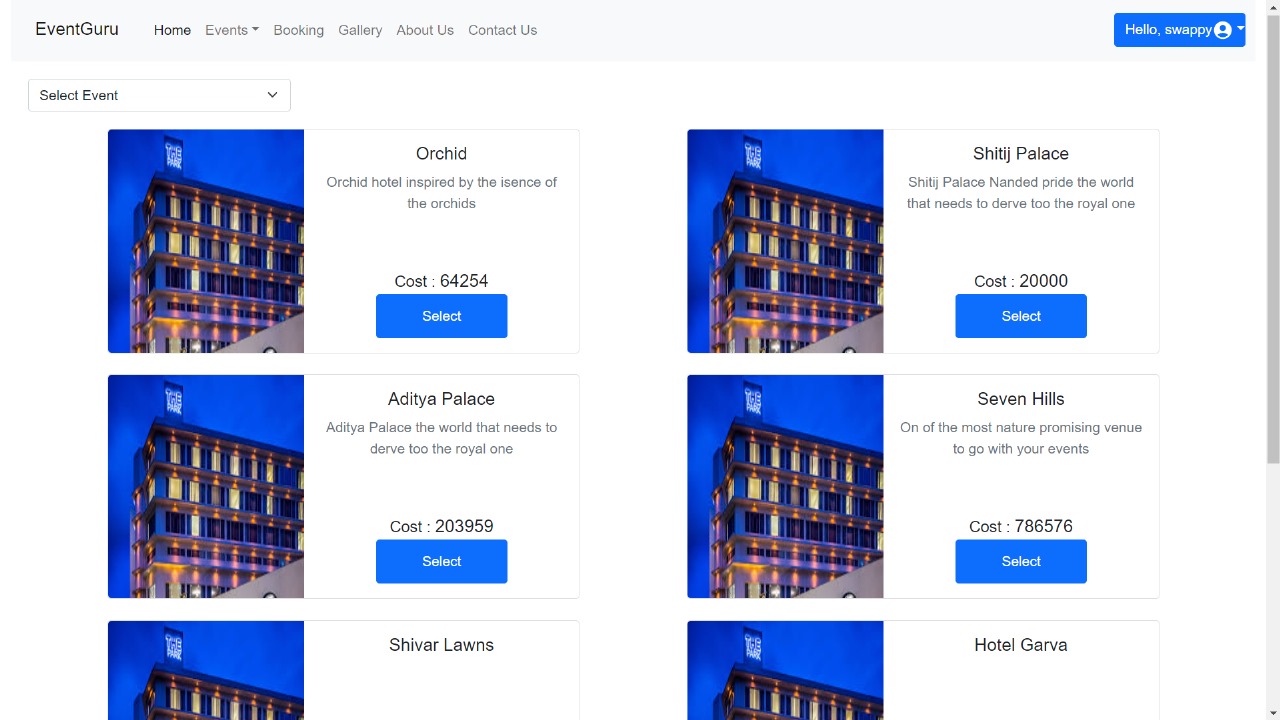
Bill:



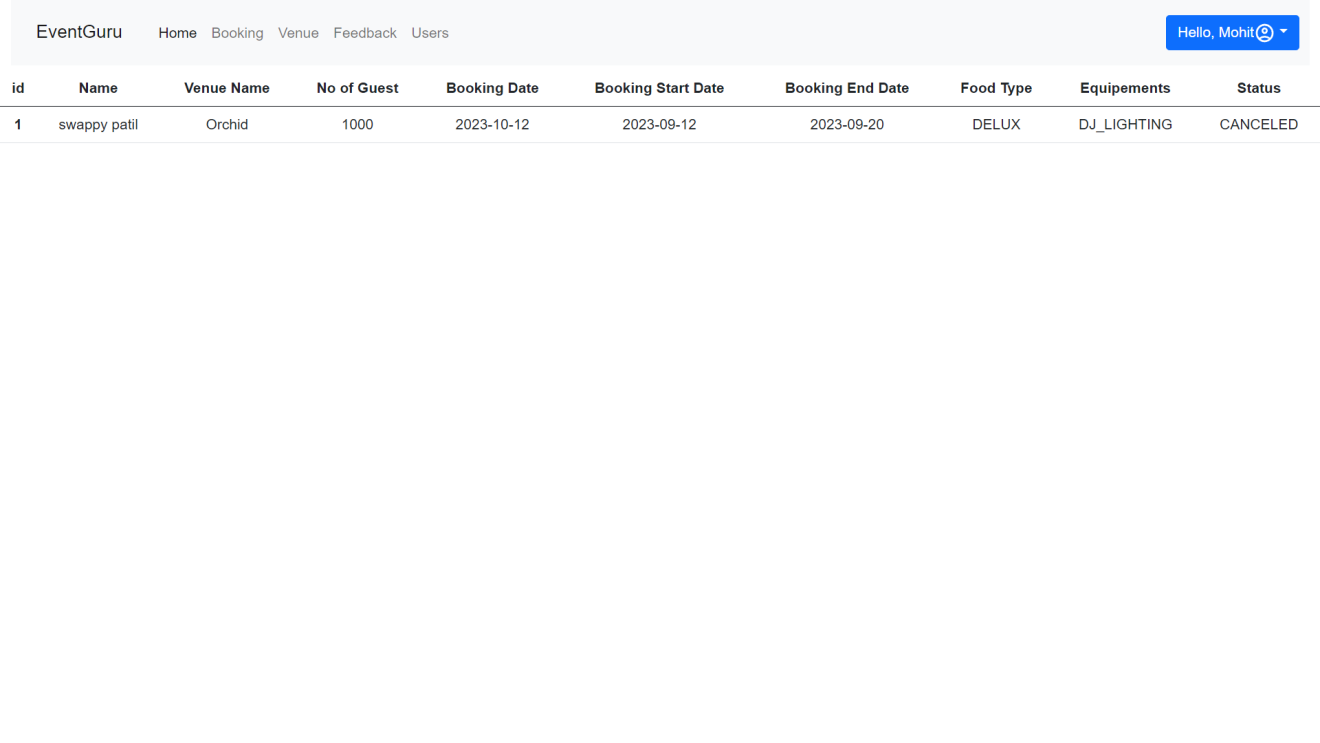
Booking:



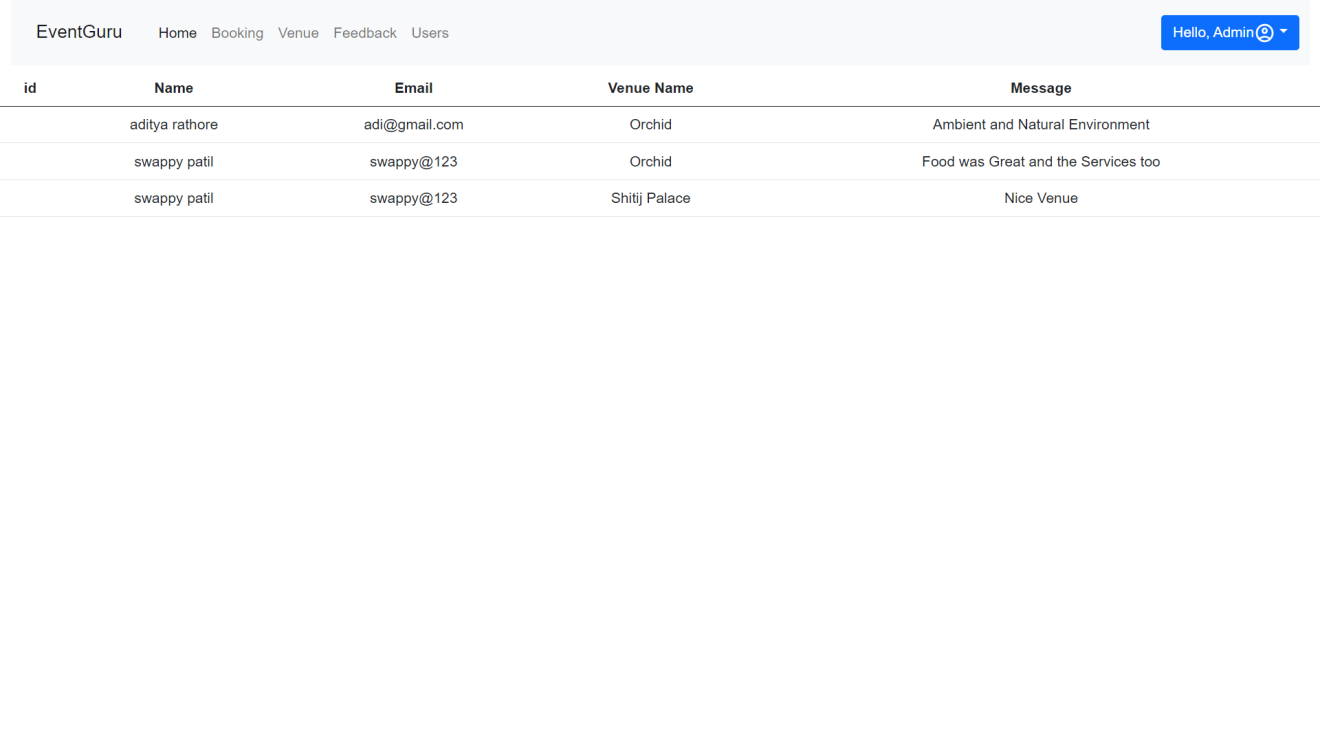
HotelDetails:



CustomerAllDetails:



All FeedBack:



**7.REFERENCES**

[http://www.google.com](http://www.google.com/)

[http://](http://www.xml101.com:8081/xml/)www.getbootstrap.com

http://www.webdevelopersjournal

http://www.github.com

http://www.w3.org

http://www.wikipedia.org

https://legacy.reactjs.org/tutorial/tutorial.html

https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/