

Chapter 1

INTRODUCTION

1.1 Overview of database management system

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated. Any piece of information can be data, for example the name of your school. Database is actually a place where related pieces of information are stored, and various operations can be performed on it. A DBMS is a software that allows creation, definition and manipulation of databases. Dbms is actually a tool used to perform any kind of operation on data in a database. Dbms also provides protection and security to databases. It maintains data consistency in case of multiple users. Here are some examples of popular dbms, MySql, Oracle, Sybase, Microsoft Access and IBM DB2 etc.

1.2 Problem statement

Traditionally, planning a wedding has always been a long process which involves a lot of time, money, effort, stress and high risks. In modern society due to competitive vendors the wedding has become a fashion for both the couple and their parents [2]. Many brides do not want to take on the stress of planning and stage managing their own wedding. Wedding planners help by providing planning services up to the wedding and coordinating the event on the big day. A wedding planner is a professional who assists with the design, planning and management of a client's wedding. But their cost is very high, and the couple cannot afford it. They provide a key role in making sure that you enjoy your wedding day as you have dreamt of, and they will make sure that all things will go smoothly according to the plan [2]. But wedding planners give their own suggestions, and they control the couple's ideas with their own planning. They are ready to introduce their own dealers who provide commissions for them once the wedding planner will arrange clients. A wedding planner can charge a flat fee for services, tact a percentage on to the key wedding expenditures. Therefore, there is always a chance the couple will go over budget and not even realize it. Arranging bridal materials, making guest list and send the invitations are very challenging tasks but wedding planning applications make it too easy. On the wedding day there are a series of activities such as speeches, dances, family

pictures etc. to be done involving many guests but the couple does not have time to waste on finding the right guest for each activity. The proposed system has ‘Task Management’ function that assigns ownership of each task to the couple’s family members and keeps track of their progress. It will help the wedding couple to enjoy the wedding day without any hassle.

1.3 Objectives

The “wedding event management” is a web-based platform that aims to help organize successful wedding events. The system assists the couples in the decision making and planning processes associated with all aspects of a wedding organization. The system offers features that the couples can retrieve information for wedding products and services as well as information of vendors in the shortest possible time. Also, vendors can gain benefit of getting more recognition from clients and generating more revenue. The proposed system has the ability to explore wedding goods/service suppliers (Vendors) across thirteen relevant categories and allow vendors to enter and manage all relevant information such as price, client list, models, previous work details, locations, exhibition details and more. Wedding couples have the ability to register themselves in the web application by entering the name of the bride, name of the groom, and date of wedding. Managing the guest list is another important task provided by the proposed system. Check who has already confirmed and who has not. Balancing the budget is one of the hardest aspects of planning a wedding. The wedding couple has no experience of dealing with paying vendor in the countdown to a wedding. Most couples have never thrown a celebration like a wedding before. Therefore, they do not know how to budget for a wedding [2]. The proposed system will help the wedding couple to figure out a budget for the event that is just right for them. It will also help them to stay within budget when planning their big day. So many things need to be done on the wedding and prior to this day. So planning is very important to make the wedding day special and memorable. The wedding checklist is generated for that. It helps to prepare for the big day without missing any of the bits and pieces that involved in planning

1.4 Dataset description

Venue Management System is an Online Venue management software project that serves the functionality of a Venue manager. The system allows only registered users to login and new users are allowed to register on the application. This is a web application, but desktop application of the

same application is also available. The project provides most of the basic functionality required for a Service. It allows the user to select from a list of event types. Once the user enters an event type e.g.(Marriage, Stage Show etc), the system then allows the user to select the date and time of event, place and the event equipment's. All this data is logged in the database and the user is setting up his username and password while registering. The data is then sent to the administrator (website owner) and they may interact with the client as per his requirements and his contact data stored in the database.

1. USER authentication

This module is mainly based on the user. The system will check the admin user ID and password for authentication. After the verification for authorization the user can be able to proceed with the process.

2. User registration

This module covers the details about the registration of users which they can register by itself by adding data like name, password, email id and further details. After registration they can sign in with their user ID and password.

3. Service booking

The User can book a service. The details to be entered are username (unique), service content, User ID and Event category. Service ID and Booking Date are automatically taken by the system.

4. Booking searching

In this module, the user can search for a service with its title and get its details.

5. User Searching and filtering users by Customer by Event

A User can be searched for with their username.

6. Service filter

A User can know the number of services given by an admin by category as expensive and affordable.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Software and Hardware

Software Configuration:

Operating system: Windows 11, 64 bits

Front end: Html, CSS, JS

Server-side language: XAMPP

Back end: MySQL

Web server: Apache

Browser: Chrome, Brave

Application software: XAMPP

Hardware Configuration:

Processor: Ryzen 7

RAM: 8 GB

Hard disk: 1TB

SYSTEM DESIGN

3.1 ER Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An Entity Relationship Diagram contains different symbols and connectors that visualize two important information: The major entities within the system scope and the interrelationships among these entities.

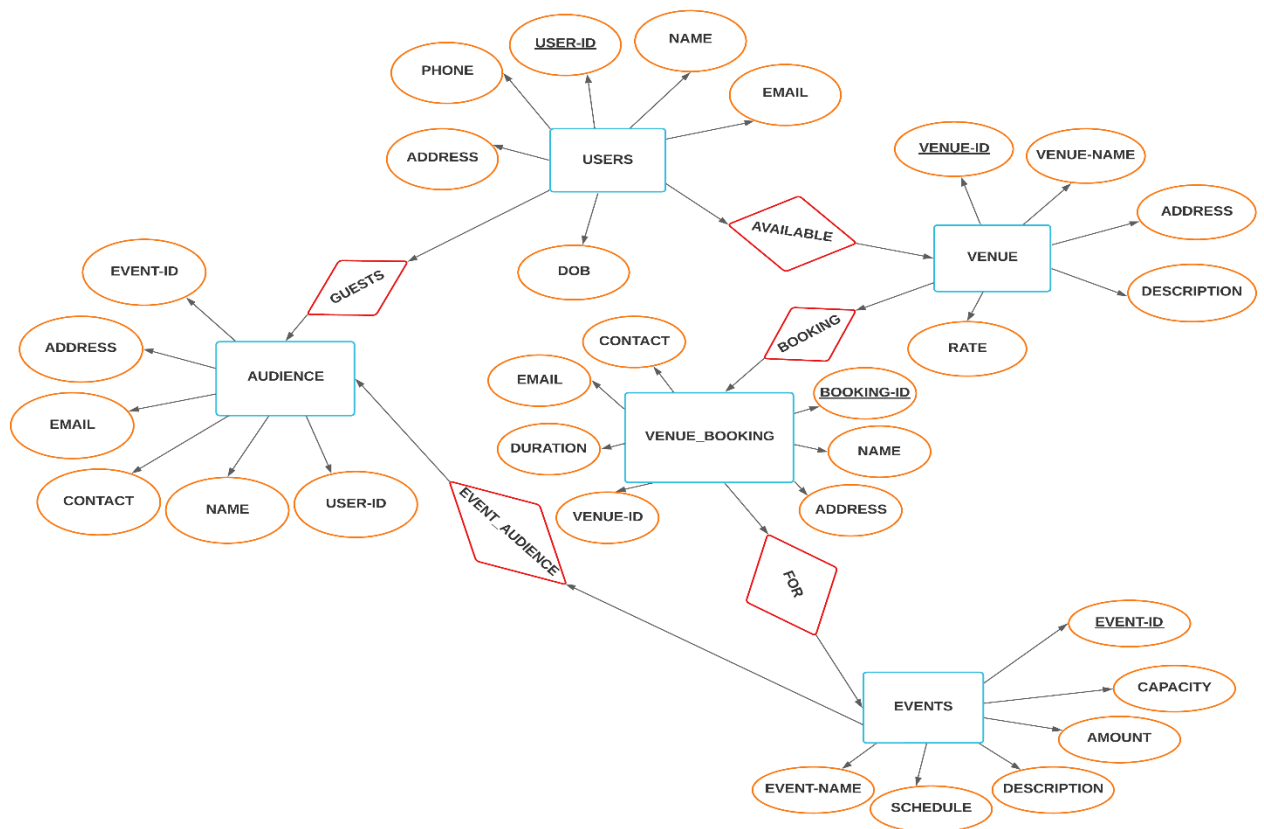


Fig 3.1 ER-Diagram

3.2 Schema Diagram:

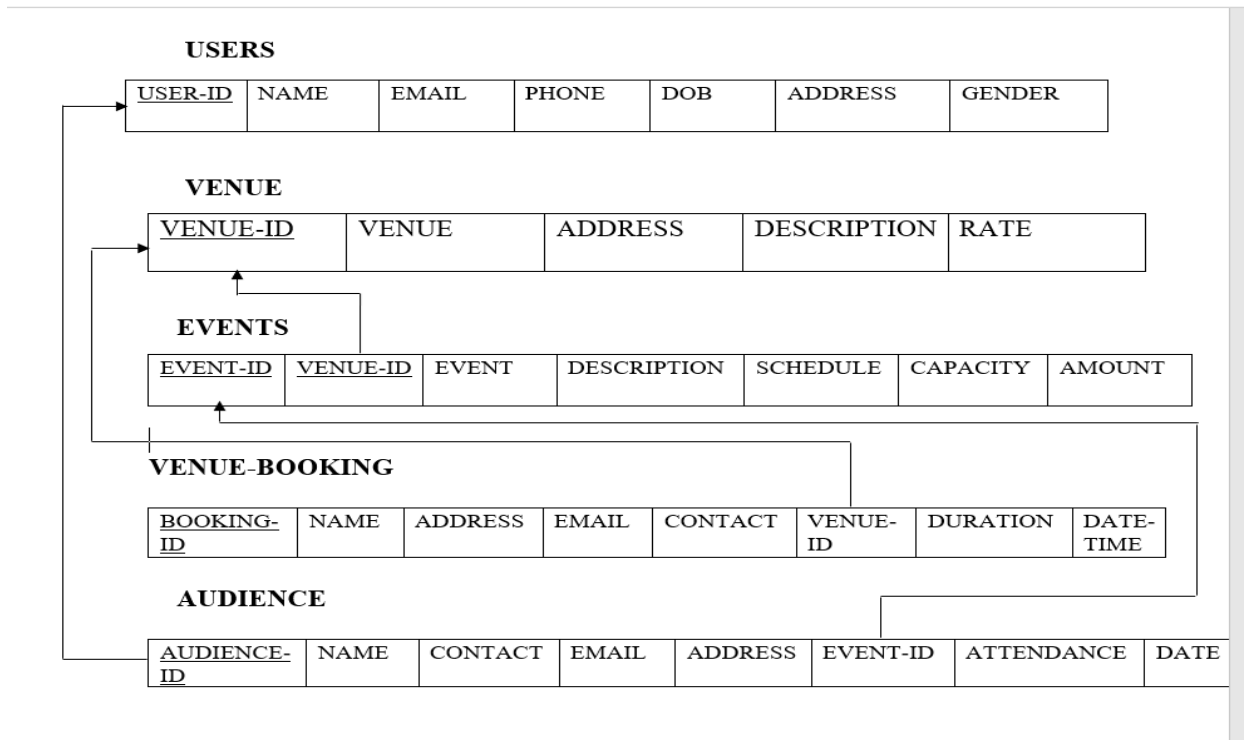


Fig 3.2 Schema Diagram

3.3 ER To Relational Mapping

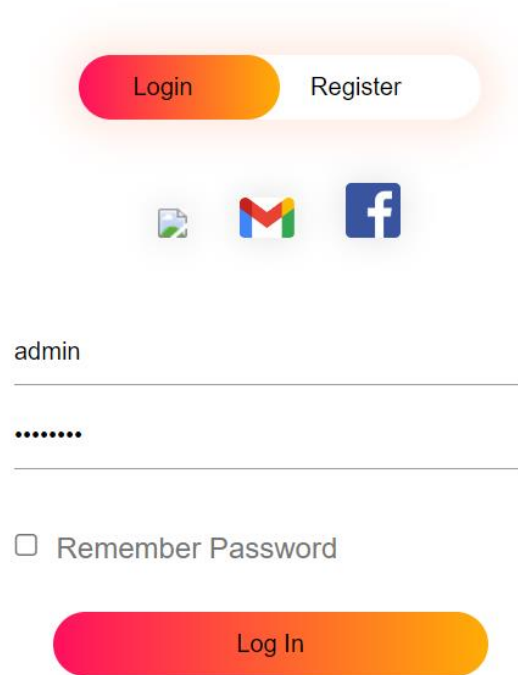
Here is the process of mapping the given CREATE TABLE statements to an ER diagram using the seven-step ER mapping method:

1. Mapping of the entities: The entities in the given tables are "audience," "events," "users," "venue," and "venue_booking."
2. Mapping of weak entity: My Database doesn't have any weak entity types.
3. users->audience in (1:1) Only users can be guests in the event.
4. Mapping of 1:N relationships: I have used the foreign key approach. Foreign Key of user_id
 - users-> venue by (1: N) Many users can select one venue.
 - venue->venue-booking for (1: N) Many persons can book for one venue.
 - venue-booking->events (N:1) Many venues can be booked for one event.
 - events ->audience of (N:1) One events can attend many events.

5. Mapping of binary M:N relationship: My database doesn't have any weak entity types so no mapping required.
6. Mapping of multivalued attributes: My database doesn't have any weak entity types so no mapping required.
7. Mapping of N-ary relationships: My database doesn't have any weak entity types so no mapping required.

3.4 Overview of GUI:

This is the LOGIN PAGE



The login page features a central white card with rounded corners. At the top of the card are two buttons: a red 'Login' button and a white 'Register' button. Below these are three social media icons: a generic image icon, a Gmail icon, and a Facebook icon. Under the icons are two input fields; the first contains the text 'admin' and the second contains seven dots for a password. Below the password field is a checkbox labeled 'Remember Password'. At the bottom of the card is a large red button labeled 'Log In'.

Fig 3.4.1 Login Page

This is the REGISTER PAGE

Login

Register



admin

siddhantpriyadarshi18581@gmail.com

09798951701

wqeadj

2001/12/01

Select whether you are Male or Female
☒Male ☐Female

.....

☐ I Agree to the Terms & Conditions

Register

Fig 3.4.2 Register Page

The HOME PAGE

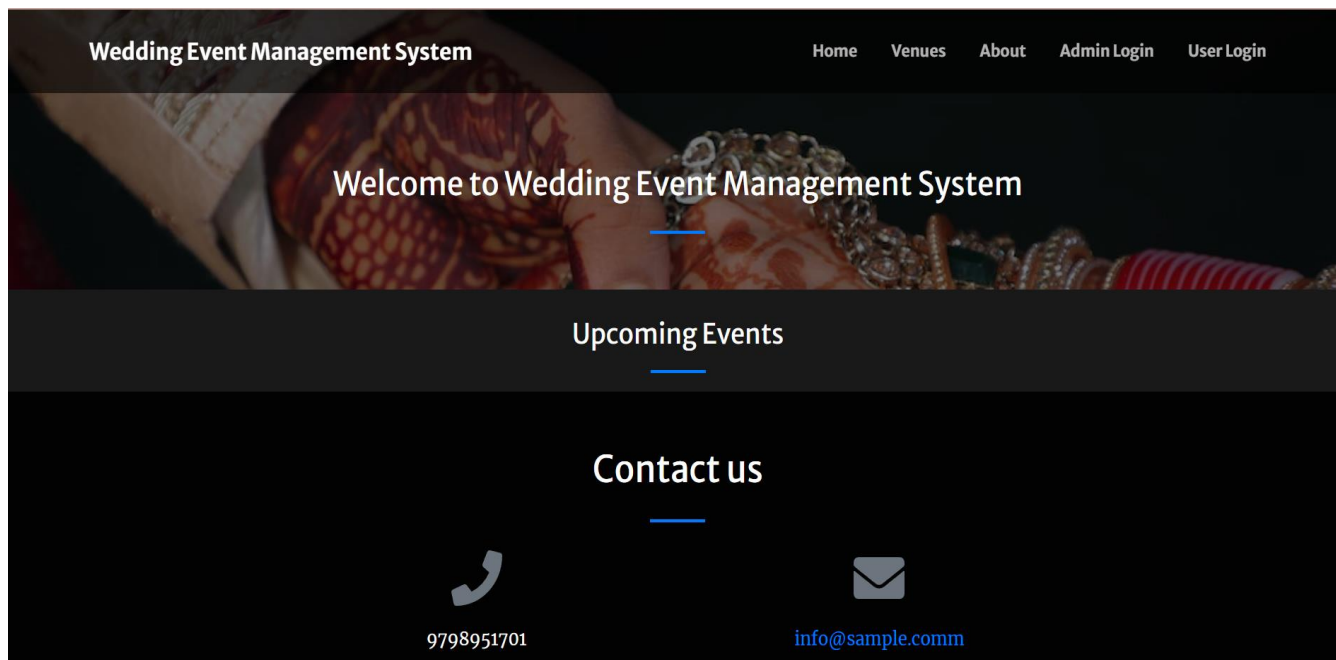


Fig 3.4.3 Home Page

The VENUE PAGE

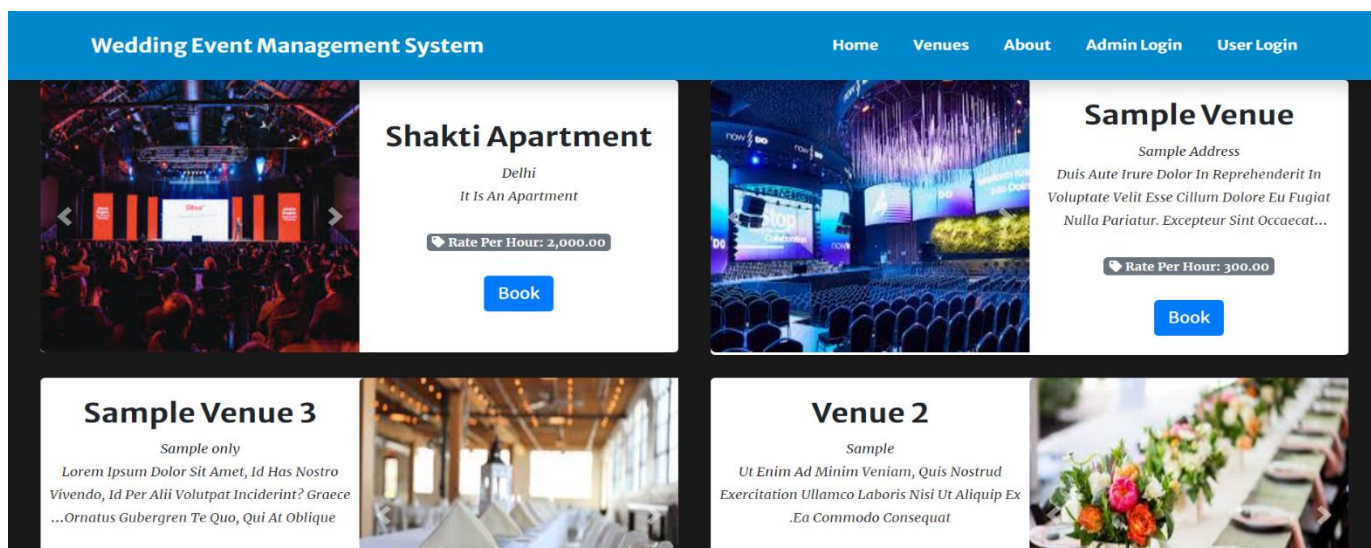


Fig 3.4.4 Venue Page

The ADMIN PAGE

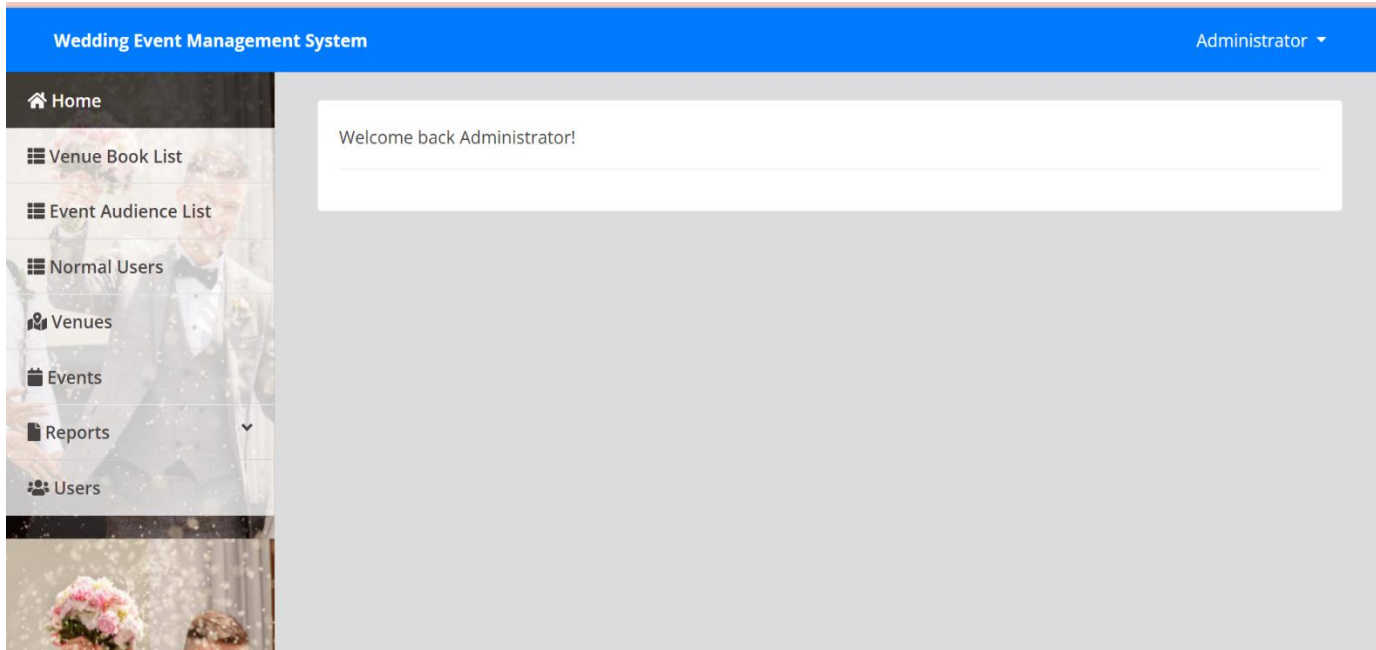


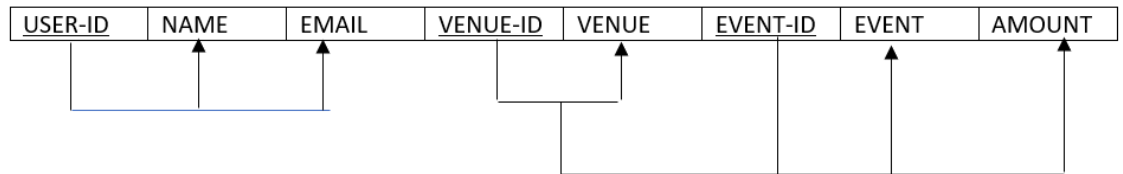
Fig 3.4.5 Admin Page

3.5 Normalization

Normalization is the process of analyzing the given relation schema based on their functional dependencies and primary key to achieve the desirable properties of minimizing redundancy and minimizing insert, delete and update anomalies. The normal form of a relation refers to the highest normal condition that it meets, and hence the degree to which it has been normalized.

Normalization rules are divided into the following forms:

1. First Normal Form(1NF)
2. Second Normal Form(2NF)
3. Third Normal Form(3NF)
4. Boyce-Codd Normal Form

INITIAL STEP:**Fig 3.5.1 Normalizing Table****1NF:**

Audience:(id (Primary Key), name, contact, email, address, event_id (Foreign Key), payment_status, attendance_status, status, date_created)

Events:(id (Primary Key), venue_id (Foreign Key), event, description, schedule, type, audience_capacity, payment_type, amount, banner, date_created)

Venue:(id (Primary Key), venue, address, description, rate)

Venue_booking:(id (Primary Key), name, address, email, contact, venue_id (Foreign Key), duration, datetime, status)

Users:(id (Primary Key), name, username, password, type)

2NF:

Audience:(id (Primary Key), name, contact, email, address_id (Foreign Key), event_id (Foreign Key), payment_status, attendance_status, status, date_created)

Events:(id (Primary Key), venue_id (Foreign Key), event_description, schedule, type, audience_capacity, payment_type, amount, banner, date_created)

Venue:(id (Primary Key), venue, address_id (Foreign Key), description, rate)

Address:(id (Primary Key), address, city, state, zip, country)

Venue_booking:(id (Primary Key), name, address_id (Foreign Key), email, contact, venue_id(Foreign Key), duration, datetime, status)

Users:(id (Primary Key), name, username, password, type)

3NF:

Audience:(id (Primary Key), name, contact_id (Foreign Key), event_id (Foreign Key), payment_status, attendance_status, status, date_created)

Events:(id (Primary Key), venue_id (Foreign Key), event_type_id (Foreign Key), schedule, audience_capacity, amount, banner, date_created)

Venue:(id (Primary Key), venue, address_id (Foreign Key), description, rate)

Address:(id (Primary Key), address, city, state, zip, country)

Event_type:(id (Primary Key), event, description, type, payment_type)

Venue_booking:(id (Primary Key), contact_id (Foreign Key), venue_id (Foreign Key), duration, datetime, status)

Contact:(id (Primary Key), name, email, address_id (Foreign Key))

Users:(id (Primary Key), name, username, password, type)

FINAL DECOMPOSITION:

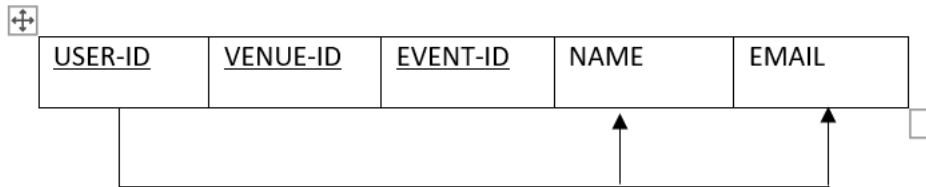


Fig 3.5.2 First Table



Fig 3.5.3 Second Table

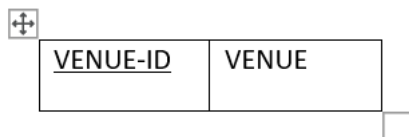


Fig 3.5.4 Third Table

Chapter 4

IMPLEMENTATION

4.1 Table Creation:

AUDIENCE

```
CREATE TABLE `audience` (  
  `id` int(11) NOT NULL PRIMARY KEY,  
  `name` text NOT NULL,  
  `contact` varchar(50) NOT NULL,  
  `email` varchar(100) NOT NULL,  
  `address` text NOT NULL,  
  `event_id` int(30) NOT NULL,  
  `payment_status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0= pending, 1 =Paid',  
  `attendance_status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '1= present',  
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0 = for verification, 1 = confirmed,2= declined',  
  `date_created` datetime NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change D
<input type="checkbox"/>	2 name	text	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/>	3 contact	varchar(50)	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/>	4 email	varchar(100)	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/>	5 address	text	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/>	6 event_id	int(30)			No	None			Change D
<input type="checkbox"/>	7 <u>payment_status</u>	tinyint(1)			No	0	0= pending, 1 =Paid		Change D
<input type="checkbox"/>	8 <u>attendance_status</u>	tinyint(1)			No	0	1= present		Change D

Console

Fig 4.1.1 Audience Table

EVENTS

```
CREATE TABLE `events` (
  `id` int(30) NOT NULL PRIMARY KEY,
  `venue_id` int(30) NOT NULL,
  `event` text NOT NULL,
  `description` text NOT NULL,
  `schedule` datetime NOT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 1 COMMENT '1=Public, 2=Private',
  `audience_capacity` int(30) NOT NULL,
  `payment_type` tinyint(1) NOT NULL DEFAULT 1 COMMENT '1=Free,payable',
  `amount` double NOT NULL DEFAULT 0,
  `banner` text NOT NULL,
  `date_created` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

The screenshot shows a database management interface with the following components:

- Navigation Bar:** Includes links for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.
- Table Structure View:** The 'Table structure' tab is selected, showing a table with 8 columns.
- Table Columns:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
2	venue_id	int(30)			No	None			Change Drop More
3	event	text	utf8mb4_general_ci		No	None			Change Drop More
4	description	text	utf8mb4_general_ci		No	None			Change Drop More
5	schedule	datetime			No	None			Change Drop More
6	type	tinyint(1)			No	1	1=Public, 2=Private		Change Drop More
7	audience_capacity	int(30)			No	None			Change Drop More
8	payment_type	tinyint(1)			No	1	1=Free,payable		Change Drop More

Fig 4.1.2 Events Table

VENUE

```
CREATE TABLE `venue` (
  `id` int(30) NOT NULL PRIMARY KEY,
  `venue` text NOT NULL,
  `address` text NOT NULL,
  `description` text NOT NULL,
  `rate` float NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table structure
Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	venue	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	address	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	description	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	rate	float			No	None			Change Drop More

☐ Check all
 With selected: Browse Change Drop Primary Unique Index Spatial Fulltext
 Add to central columns Remove from central columns

Fig 4.1.3 Venue Table

VENUE-BOOKING

```
CREATE TABLE `venue_booking` (
  `id` int(30) NOT NULL PRIMARY KEY,
  `name` text NOT NULL,
  `address` text NOT NULL,
  `email` varchar(100) NOT NULL,
  `contact` varchar(100) NOT NULL,
  `venue_id` int(30) NOT NULL,
  `duration` varchar(100) NOT NULL,
  `datetime` datetime NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0=for verification,1=confirmed,2=canceled'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(30)			No	None		AUTO_INCREMENT	Change D
<input type="checkbox"/> 2	name	text	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/> 3	address	text	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/> 4	email	varchar(100)	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/> 5	contact	varchar(100)	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/> 6	venue_id	int(30)			No	None			Change D
<input type="checkbox"/> 7	duration	varchar(100)	utf8mb4_general_ci		No	None			Change D
<input type="checkbox"/> 8	datetime	datetime			No	None			Change D

Console

Fig 4.1.4 Venue Booking Table

USERS

```
CREATE TABLE `users` (
  `id` int(30) NOT NULL PRIMARY KEY,
  `name` text NOT NULL,
  `username` varchar(200) NOT NULL,
  `password` text NOT NULL, `type` tinyint(1) NOT NULL DEFAULT 2 COMMENT '1=Admin,2=Staff')
```

<div> <div>Table structure</div> <div>Relation view</div> </div>									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(30)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	name	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	username	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	password	text	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	type	tinyint(1)			No	2	1=Admin,2=Staff		Change Drop More

☐ Check all
 With selected:
 Browse
 Change
 Drop
 Primary
 Unique
 Index
 Spatial
 Fulltext

Add to central columns
 Remove from central columns

Fig 4.1.5 Users Table

4.2 Description of Table:

Audience:

It contains the list of audience who booked for the events or they are invited in the wedding. Which make it easier to keep the track of guests and also keep the counts of guests in the events.

Events:

It contains the description of events, event name, and event venue. It can help to keep track to the guests who are participating in various wedding events and also notify them.

Venue:

It contains the description of the available venues. And venue name so that it is easier to be booked by the users itself.

Venue-Booking:

It contains the users description who has booked which venue at what date and duration of the booking with date to be booked.

Normal Users:

It contains the description of normal users who are logged in and can book venues and can see the upcoming respective events.

4.3 Populated Tables:

Insert into Venue-Booking

	id	name	address	email	contact	venue id	duration	datetime	status <small>0=for verification,1=confirmed,2=canceled</small>
Delete	1	John Smith	Sample	asdasd@gmail.com	+18456-5455-55	2	1 night	2020-10-14 17:00:00	1
Delete	2	Siddhant Priyadarshi	wqeadsj	siddhantpriyadarshi18581@gmail.com	9798951701	4	24hr	2023-01-20 00:01:00	1
Console	Raj	Gujarat	abc		gggg	5	hhh	2023-01-25 14:30:00	1

Fig 4.3.1 Venue Booking Values

Insert into Event Audience List

☐ Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

←↵→

Fig 4.3.2 Event Audience List

4.4 SQL Triggers

Trigger

The screenshot shows a software interface for editing a trigger. The window is titled "Edit" and contains a "Details" section with the following fields:

- Trigger name:** A text input field containing "userexceed".
- Table:** A dropdown menu showing "normalusers".
- Time:** A dropdown menu showing "AFTER".
- Event:** A dropdown menu showing "INSERT".
- Definition:** A text area containing the SQL code:

```
1 INSERT INTO extrausers VALUES(null, NEW.username,
NEW.email,NEW.phone, NEW.address, NOW(),
NEW.gender, NEW.password)
```

At the bottom right of the dialog are two buttons: "Go" and "Close".

Fig 4.4.1 SQL Triggers

Stored Procedure

Edit

Details

Routine name:

Type:

Parameters:

Direction	Name	Type	Length/Values	Options
<input type="button" value="Add parameter"/>				

Definition:

1	SELECT * FROM normalusers			
---	---------------------------	--	--	--

Fig 4.4.2 Stored Procedure

4.5 Database Connectivity:

```
<?php
$servername="localhost";
$username="root";
$password="";
$database_name="event_db";

$conn=mysqli_connect($servername,$username, $password,$database_name);
//now check the connection
if(!$conn){
    die("Connection failed:" . mysqli_connect_error());
}
```

4.6 Modules:

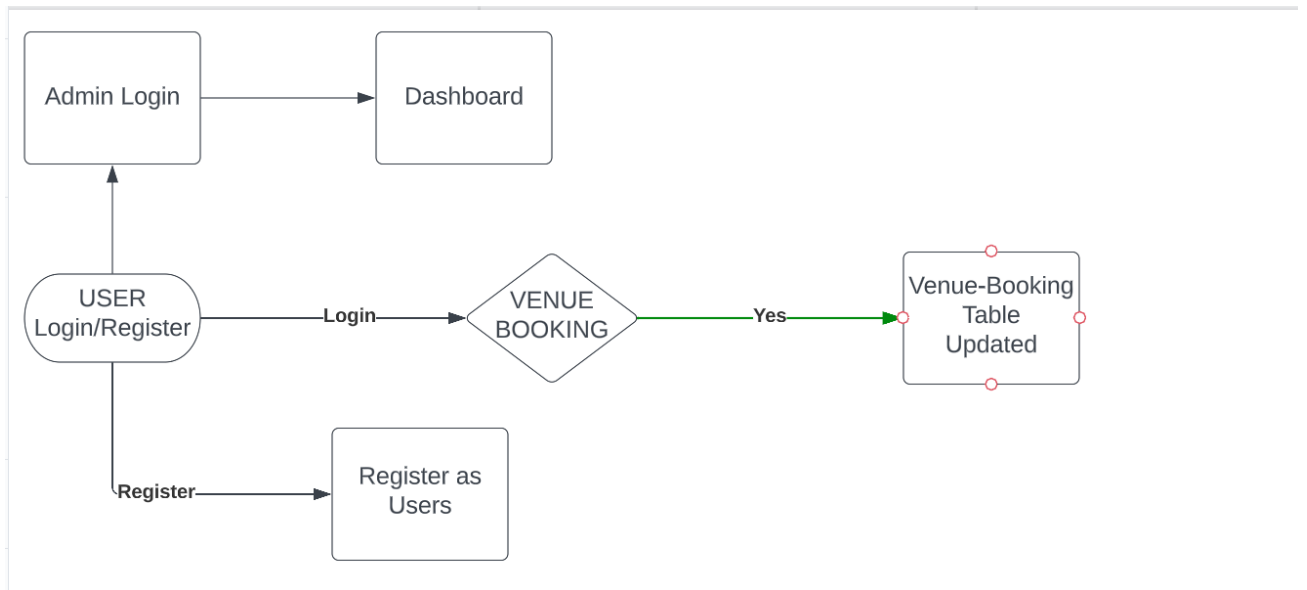
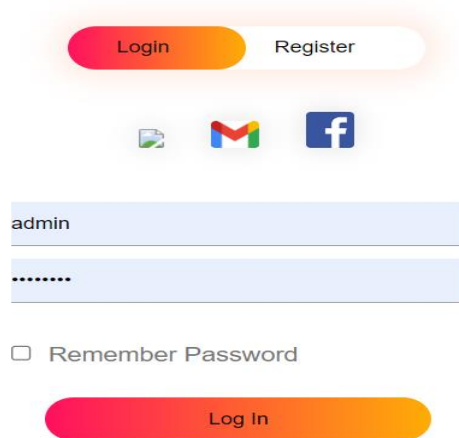


Fig 4.6 Workflow Model

Chapter 5

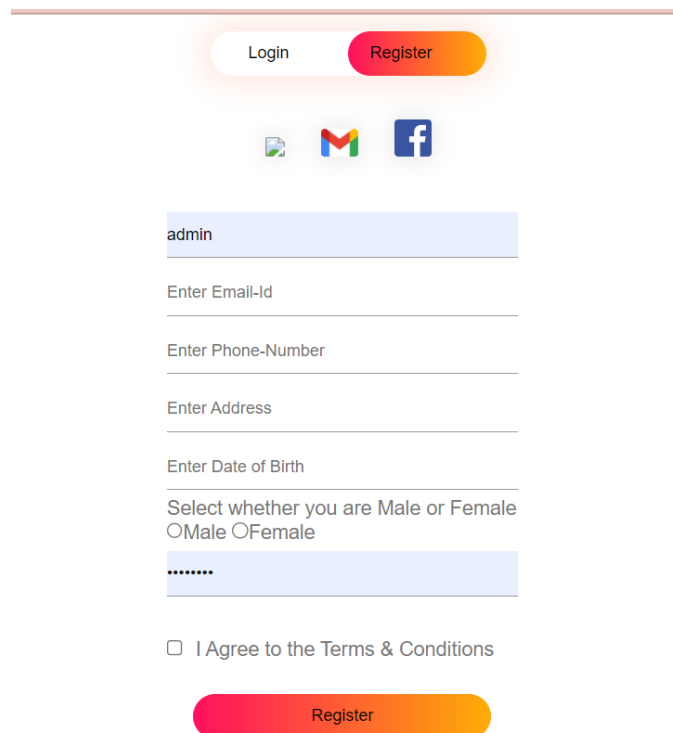
RESULT

This chapter contains GUI built using CSS and HTML. The screenshots contains various html pages. Login page: Figure 5.1 represents page that we get when we run the index.php.



The login page interface features a header with two buttons: 'Login' (orange) and 'Register' (white). Below the header are three social media icons: a generic profile icon, Gmail, and Facebook. The main form consists of two input fields: the first contains the text 'admin', and the second contains a masked password '.....'. Below the password field is a checkbox labeled 'Remember Password'. At the bottom is a large orange button labeled 'Log In'.

Fig 5.1 Login Page Interface



The register page interface features a header with two buttons: 'Login' (white) and 'Register' (orange). Below the header are three social media icons: a generic profile icon, Gmail, and Facebook. The main form consists of several input fields: 'admin' (pre-filled), 'Enter Email-Id', 'Enter Phone-Number', 'Enter Address', and 'Enter Date of Birth'. Below these is a text label 'Select whether you are Male or Female' followed by two radio buttons, 'Male' and 'Female'. The next input field contains a masked password '.....'. Below the password field is a checkbox labeled 'I Agree to the Terms & Conditions'. At the bottom is a large orange button labeled 'Register'.

Fig 5.2 Register Page Interface

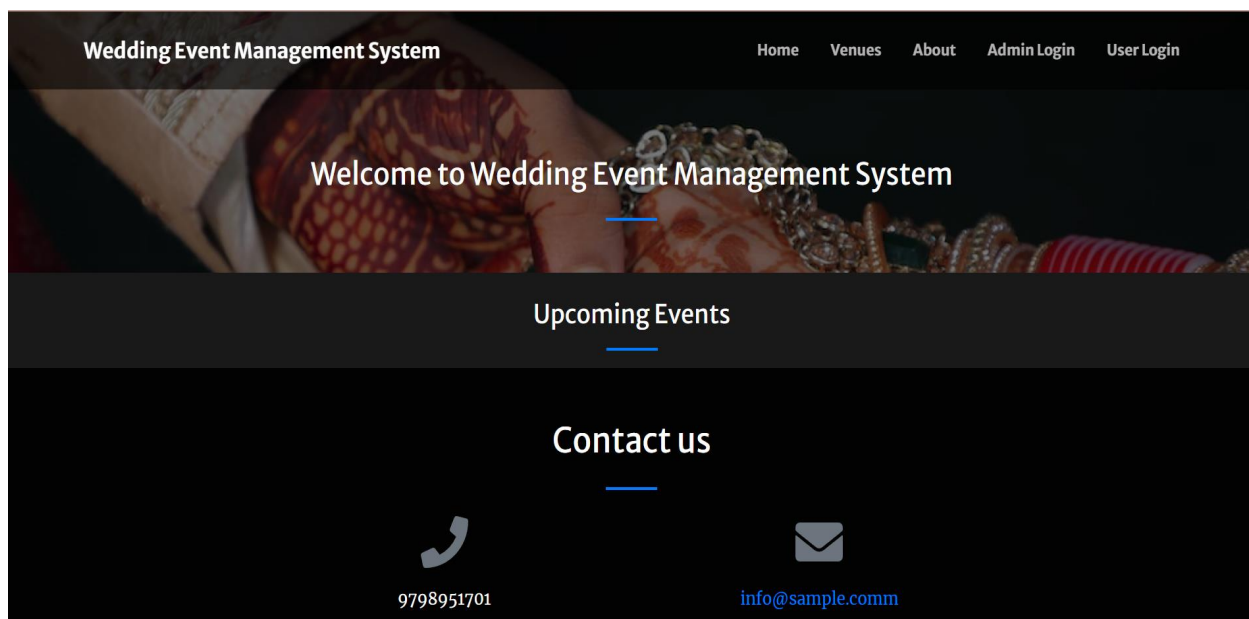


Fig 5.3 Home Page Interface

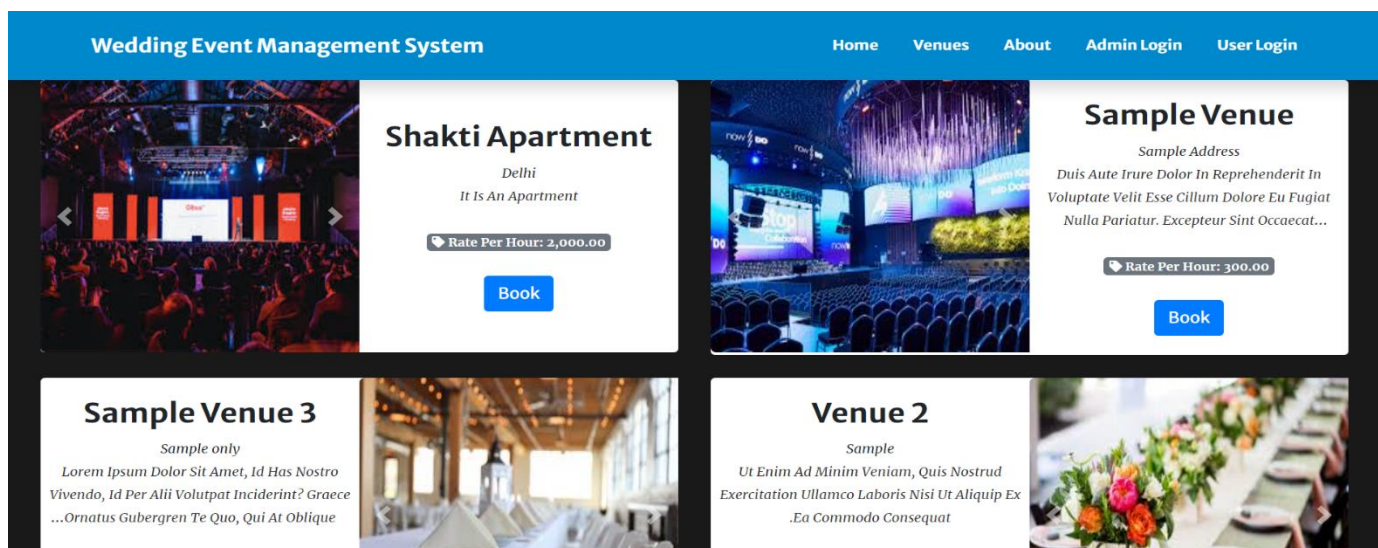


Fig 5.4 Venues Interface

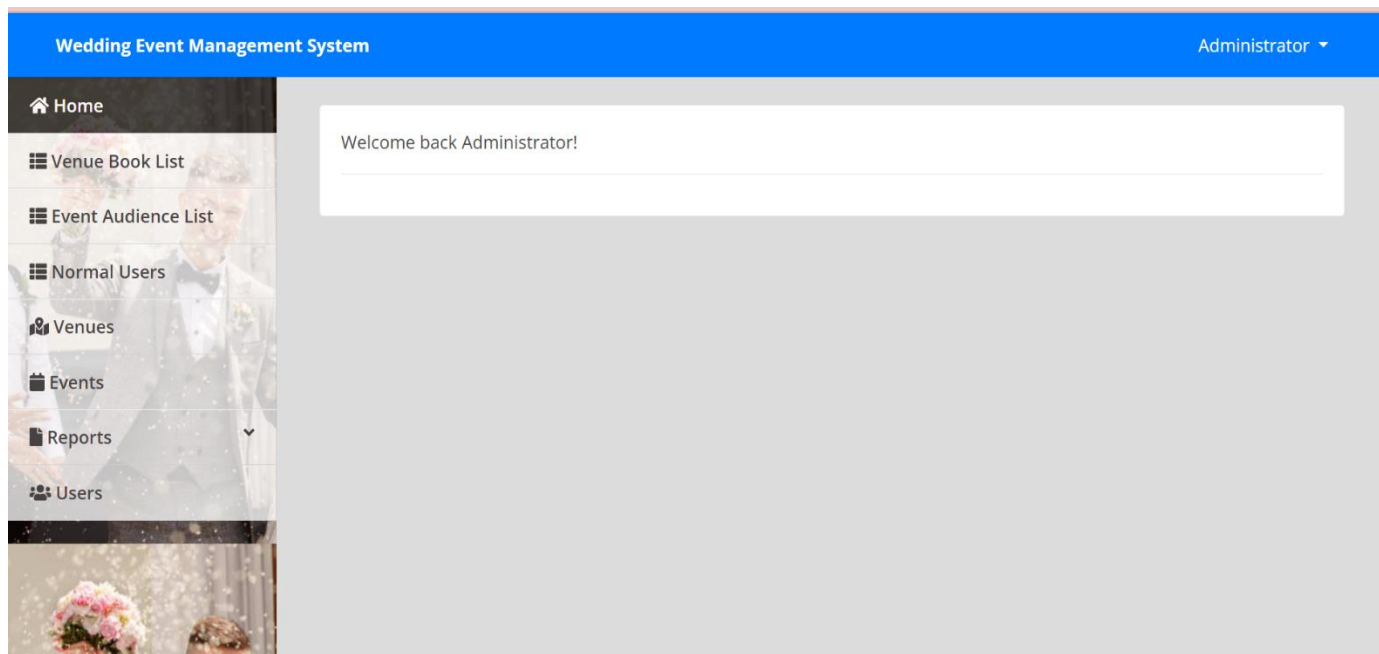
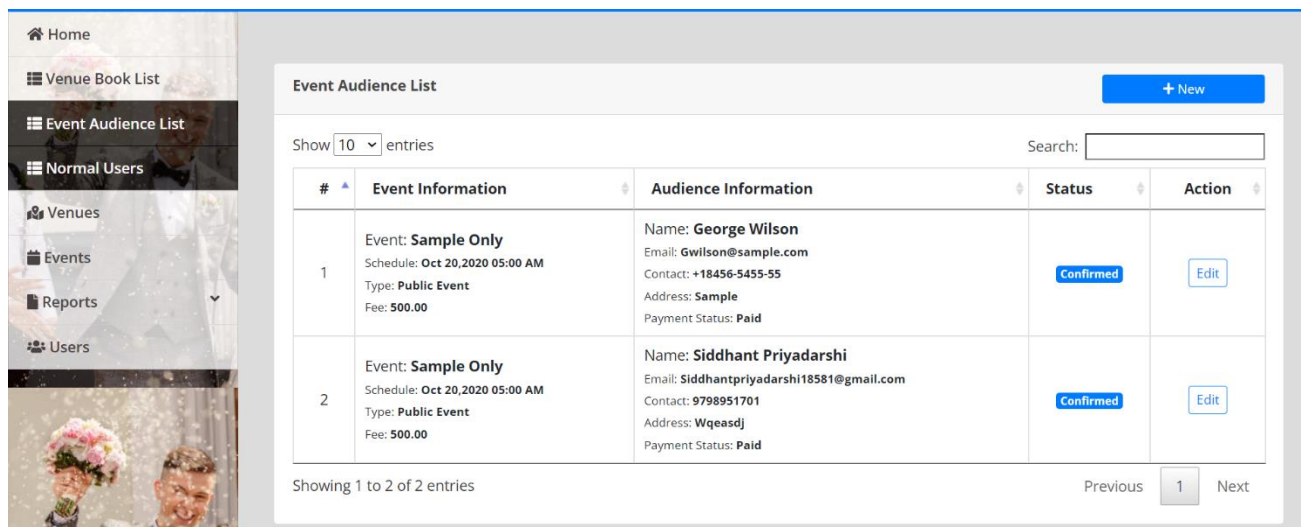


Fig 5.5 Admin Page Interface

Venue Booking List + New				
Show <input type="text" value="10"/> entries		Search: <input type="text"/>		
#	Booking Information	Customer Information	Status	Action
1	Venue: Venue 2 Schedule: Oct 14,2020 05:00 PM Duration: 1 Night	Booked by: John Smith Email: Asdasd@gmail.com Contact: +18456-5455-55 Address: Sample	Confirmed	Edit
2	Venue: Sample Venue 3 Schedule: Jan 20,2023 12:01 AM Duration: 24hr	Booked by: Siddhant Priyadarshi Email: Siddhantpriyadarshi18581@gmail.com Contact: 9798951701 Address: Wqeadsj	Confirmed	Edit
3	Venue: Shakti Apartment Schedule: Jan 25,2023 02:30 PM Duration: Hhh	Booked by: Raj Email: Abc Contact: Gggg Address: Gujarat	Confirmed	Edit

Fig 5.6 Venue Booking List Interface



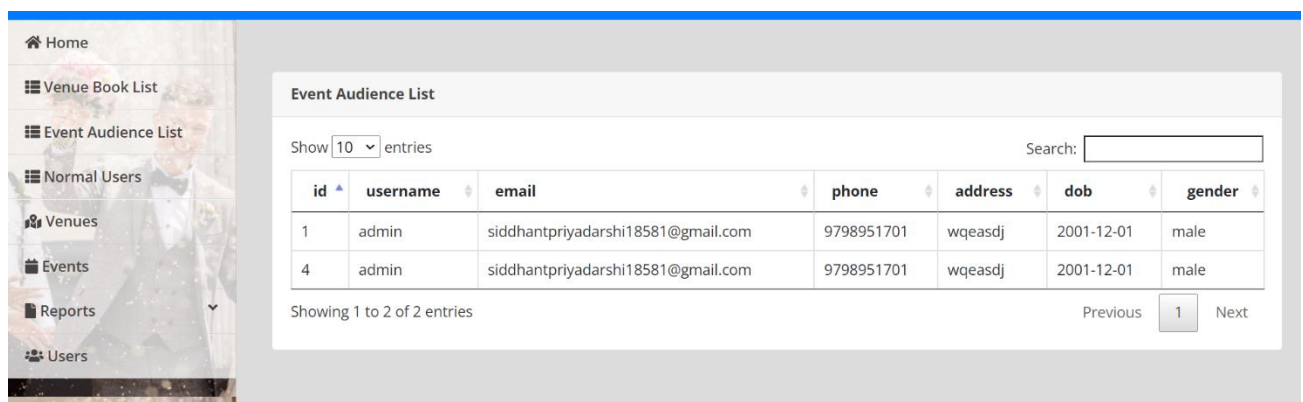
Event Audience List + New

Show entries Search:

#	Event Information	Audience Information	Status	Action
1	Event: Sample Only Schedule: Oct 20,2020 05:00 AM Type: Public Event Fee: 500.00	Name: George Wilson Email: Gwilson@sample.com Contact: +18456-5455-55 Address: Sample Payment Status: Paid	Confirmed	Edit
2	Event: Sample Only Schedule: Oct 20,2020 05:00 AM Type: Public Event Fee: 500.00	Name: Siddhant Priyadarshi Email: Siddhantpriyadarshi18581@gmail.com Contact: 9798951701 Address: Wqeadsj Payment Status: Paid	Confirmed	Edit

Showing 1 to 2 of 2 entries Previous Next

Fig 5.7 Event Audience List Interface



Event Audience List

Show entries Search:

id	username	email	phone	address	dob	gender
1	admin	siddhantpriyadarshi18581@gmail.com	9798951701	wqeadsj	2001-12-01	male
4	admin	siddhantpriyadarshi18581@gmail.com	9798951701	wqeadsj	2001-12-01	male

Showing 1 to 2 of 2 entries Previous Next

Fig 5.8 Users Interface

List of Venue						+ New Entry
Show <input type="text" value="10"/> entries			Search: <input type="text"/>			
#	Venue	Address	Description	Rate	Action	
1	Sample Venue	Sample Address	Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.	300.00	Edit Delete	
2	Venue 2	Sample	Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	250.00	Edit Delete	
3	Sample Venue 2	Sample Address	Lorem ipsum dolor sit amet, id has nostro vivendo, id per alii volutpat inciderint? Graece ornatus gubergren te quo, qui at oblique accusamus, id pro eros etiam conceptam! Ullum clita.	280.00	Edit Delete	
4	Sample Venue 3	Sample only	Lorem ipsum dolor sit amet, id has nostro vivendo, id per alii volutpat inciderint? Graece ornatus gubergren te quo, qui at oblique accusamus, id pro eros etiam conceptam! Ullum clita.	1,000.00	Edit Delete	
5	Shakti	Delhi	It is an apartment	2,000.00	Edit	

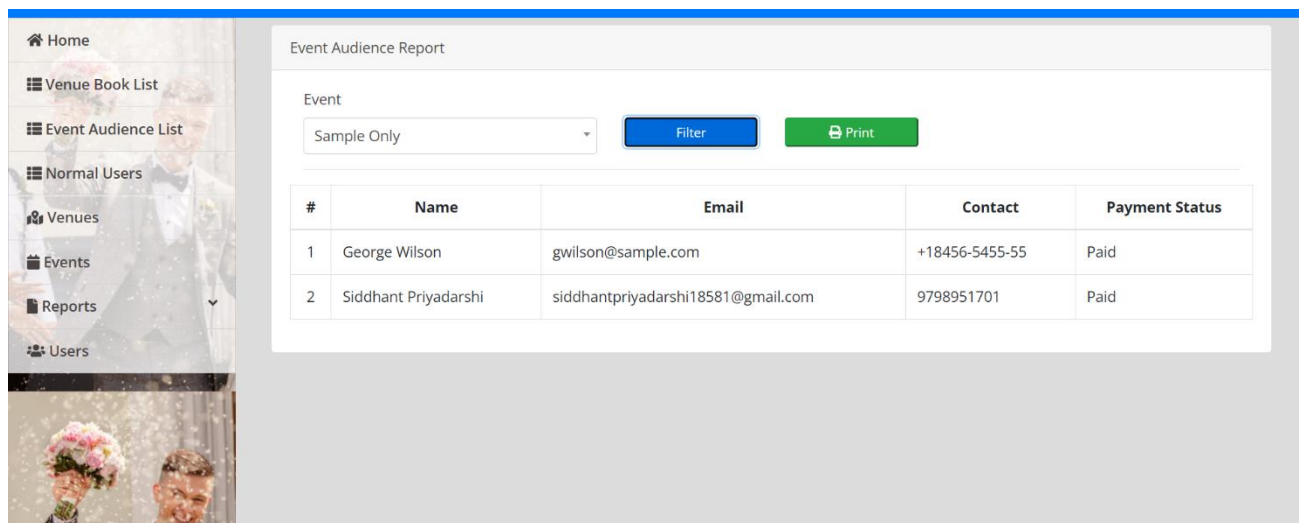
Fig 5.9 Venue List Interface

List of Events						+ New Entry
Show <input type="text" value="10"/> entries			Search: <input type="text"/>			
#	Schedule	Venue	Event Info.	Description	Action	
1	Oct 20, 2020 05:00 AM	Sample Venue	Event: Sample Only Type: Public Event Fee: 500.00	Sample HeaderLorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna...	View Edit Delete	
2	Oct 23, 2020 06:00 PM	Venue 2	Event: Event 2 Type: Public Event Fee: Free	Lorem ipsum dolor sit amet, viris eleifend convenire mei te, ei mea probo cotidieque? Pri gloriatur disputationi vituperatoribus ex...	View Edit Delete	
3	Oct 28, 2020 05:00 PM	Venue 2	Event: Sample Private Type: Private Event Fee: Free	Wedding	View Edit Delete	

Showing 1 to 3 of 3 entries

Previous Next

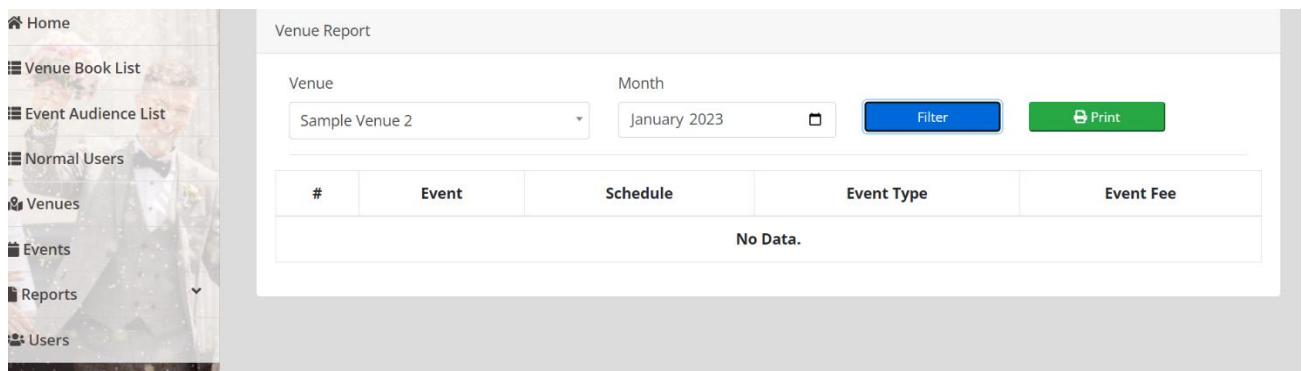
Fig 5.10 Events List Interface



The Event Audience Report interface features a sidebar with navigation links: Home, Venue Book List, Event Audience List, Normal Users, Venues, Events, Reports, and Users. The main content area is titled 'Event Audience Report' and includes a filter section with an 'Event' dropdown set to 'Sample Only', a blue 'Filter' button, and a green 'Print' button. Below this is a table with the following data:

#	Name	Email	Contact	Payment Status
1	George Wilson	gwilson@sample.com	+18456-5455-55	Paid
2	Siddhant Priyadarshi	siddhantpriyadarshi18581@gmail.com	9798951701	Paid

Fig 5.11 Event Audience Report



The Venue Report interface features a sidebar with navigation links: Home, Venue Book List, Event Audience List, Normal Users, Venues, Events, Reports, and Users. The main content area is titled 'Venue Report' and includes a filter section with a 'Venue' dropdown set to 'Sample Venue 2', a 'Month' dropdown set to 'January 2023', a blue 'Filter' button, and a green 'Print' button. Below this is a table with the following data:

#	Event	Schedule	Event Type	Event Fee
No Data.				

Fig 5.12 Venue Report Interface

Chapter 6

CONCLUSION & FUTURE ENHANCEMENTS

It is worth mentioning that this project is open for future enhancement. Additional features like a display of all blogs. More categories to post on, images and videos added to the blogs can be added. Further enhancements can be made to the project, so that the website functions in a very attractive and useful manner than the present one. It is concluded that the application works well and satisfy the needs. The application is tested very well and errors are properly debugged. It also acts as the sharing of files to the valuable resources.

Chapter 6