**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Jnana Sangama, Belagavi-590018, Karnataka**



**DATABASE MANAGEMENT SYSTEM MINIPROJECT REPORT**

**on**

**“EVENT MANAGEMENT SYSTEM”**

**Submitted by**

**CHAVVA VENKATA PRATHEEK REDDY**

**1BI16CS215**

**For the academic year 2018-19**

****

**Department of Computer Science & Engineering**

**BANGALORE INSTITUTE OF TECHNOLOGY**

K.R. Road, V.V.Pura, Bengaluru-560 004

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnanasangama”, Belagavi-590018, Karnataka**

**BANGALORE INSTITUTE OF TECHNOLOGY**

**K.R. Road, V.V.Puram, Bengalure-560 004**

****

**Department of Computer Science & Engineering**

***Certificate***

This is to certify that the implementation of **DBMS MINI PROJECT** (15CSL58) entitled “EVENT MANAGEMENT SYSTEM” has been successfully completed by

**USN: 1BI16CS215 NAME: CHAVVA VENKATA PRATHEEK REDDY**

of V semester B.E. for the partial fulfillment of the requirements for the Bachelor's degree in Computer Science & Engineering of the Visvesvaraya Technological University during the academic year 2018-2019.

**Lab Incharges :**

1. Tejasvini P S **Dr. Asha T**

Asst. Professor Professor and Head

Dept. of CS&E Department of CS&E

Bangalore Institute of Technology Bangalore Institute of Technology

Bangalore Bangalore

1. Kavitha K

Asst. Professor

Dept. of CS&E

Bangalore Institute of Technology

Bangalore

Examiners: 1) 2)

**ACKNOWLEDGEMENT**

While presenting my project on “**EVENT MANAGEMENT SYSTEM”**, it is important to acknowledge the help rendered by various persons.

I would like to convey my thanks to the HOD **Dr. Asha T** for being kind enough to provide an opportunity to do a project in this institution.

I am most humbled to mention the enthusiastic influence provided by **Prof. Tejasvini P S** and  **Prof. Kavitha K** , on the project for their ideas, time to time suggestions for being a constant guide and co-operation showed during the venture and making this project a great success.

I would also take this opportunity to thank my friends and family for their constant support and help. I’m very much pleasured to express my sincere gratitude to the friendly co-operation showed by all the **staff members**  of Computer Science Department BIT.

CHAVVA VENKATA PRATHEEK REDDY

1BI16CS215

**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** |  | **Introduction** | 1 |
| **2.** |  | **Back end Design** | 3 |
| **2.1** | **Conceptual Database design** | 4 |
| **2.2** | **Logical Database design** | 5 |
| **2.3** | **Normalization** | 6 |
| **3.** |  | **Front End Design** | 11 |
| **3.1** | **Screen Layout** | 12 |
| **3.2** | **Connectivity** | 14 |
| **4.** |  | **Major Modules** | 15 |
| **5.** |  | **Implementation using PHP,HTML and MYSQL** | 18 |
| **6.** |  | **Snapshots** | 30 |
| **7.** |  | **Applications** | 38 |
| **8.** |  | **Conclusion** | 40 |

**CHAPTER-1**

**INTRODUCTION**

**1.1 INTRODUCTION**

Event booking and management system is a convenient method to book and view events in and around the user’s location in the desired time. These events may be of any type including musicals, sports, concerts, shows, fests etc. This project enables the user’s to access all details regarding the event and make bookings

Event booking and management system software project is a project that serves the functionality of an event manager. The system allows only registered users to login and new users are allowed to resister through the admin in the database. This application is developed in PHP and MYSQLi. The project provides most of the basic functionality required for booking of an event. It allows the user to select a duration and select an event from the list of events happening in this duration. Once the user selects the event among the list of events eg(Concerts, Shows, Musicalsetc), the system then allows the user to view the date of the event, venue and the types of tickets. The user then selects the desired type of tickets and books it. All this data is logged in the database and the user is given a booking number for his booking. All the details regarding the user’s transactions and the user’s details are stored in the database.

The event management system has various advantages. The event management system is useful as it allows the user to book for the event at a click of a button.The user gets all the resources at a single place instead of wandering around for bookings.This system is effective and saves time and cost of the users.The event management system is disadvantageous only because it reduces employment opportunities.

**1.2 PROBLEM STATEMENT**

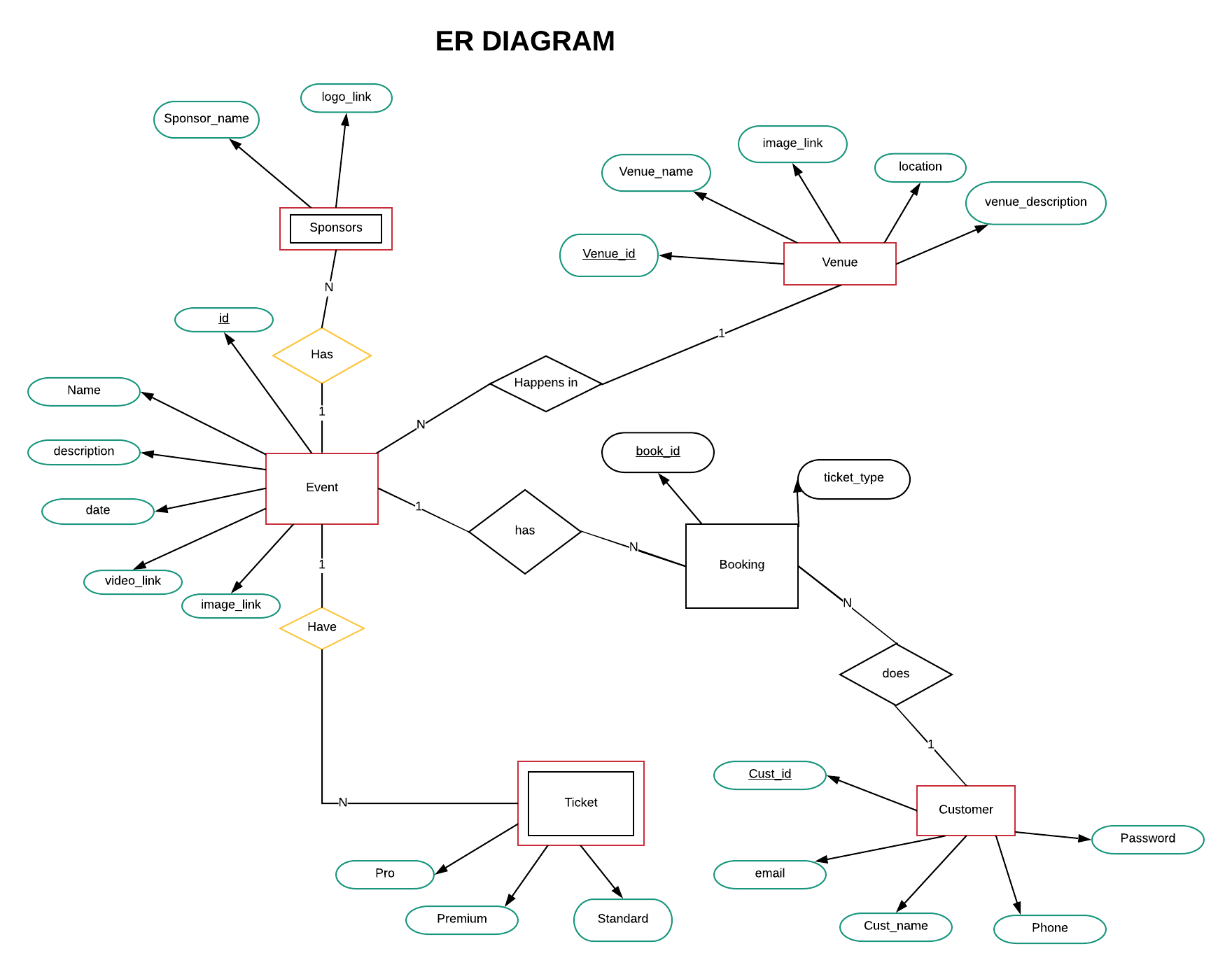
This project develops a database management system, which would serve the basic functionalities. It would accept database administration and SQL commands. The SQL Interpreter then interprets these commands. The result of interpretation is then transformed into an understandable form. The changed form is evaluated to give the result. The result could be modifying or retrieving from the database. It would prompt the user with error messages if the queries were not correct.

‘

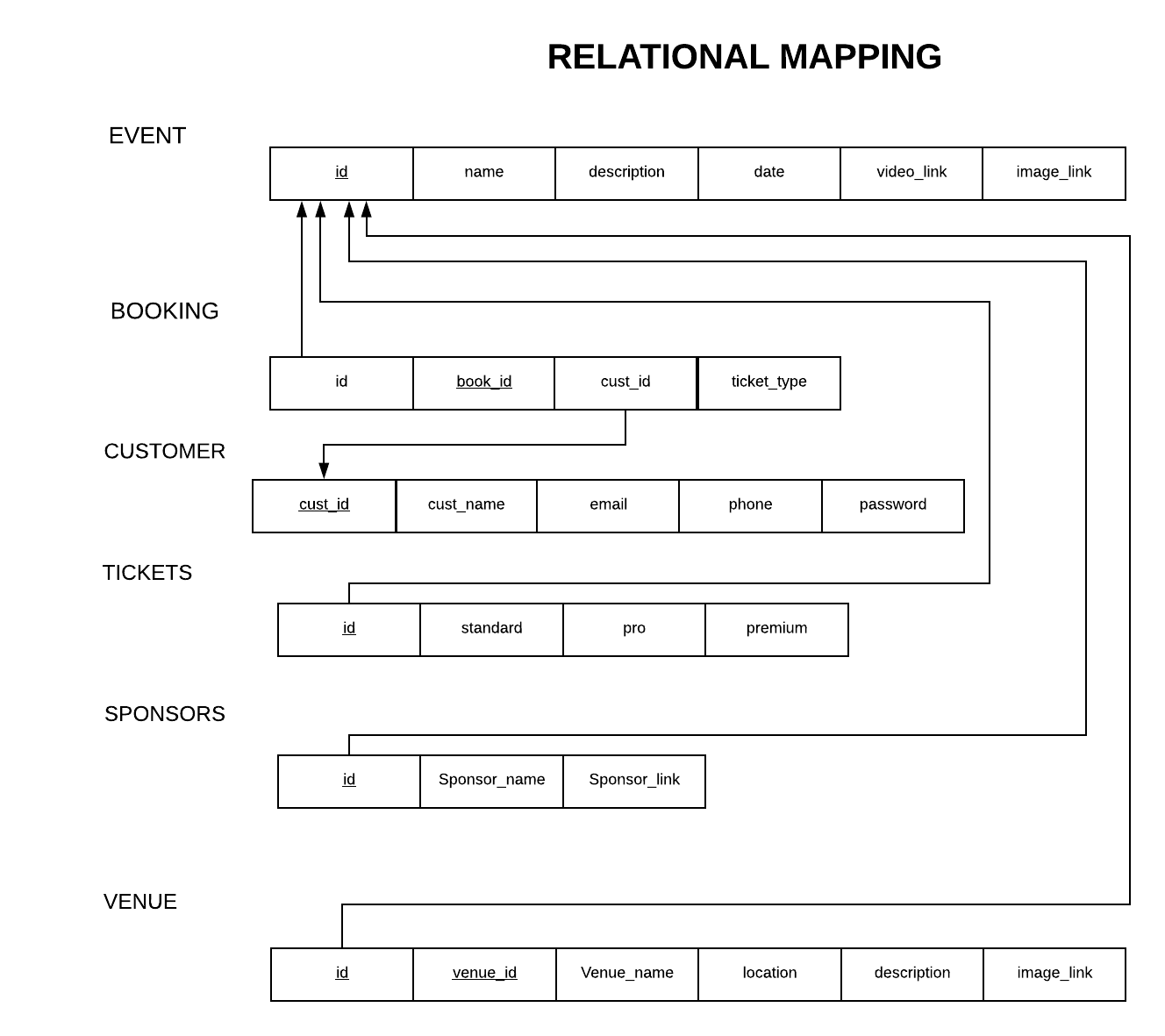
**CHAPTER-2**

**BACK END DESIGN**

**2.1 CONCEPTUAL DATABASE DESIGN:**



**2.2 LOGICAL DATABASE DESIGN:**



**2.3 NORMALIZATION:**

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

* Eliminating redundant(useless) data.
* Ensuring data dependencies make sense i.e. data id logically stored.

**FIRST NORMAL FORM (1NF):**

As per First Normal Form

1. There are no duplicated rows in the table.
2. Each cell is single valued or atomic.

**SECOND NORMAL FORM (2NF):**

As per Second Normal Form, a table is in 2NF if every non prime attribute is not partially dependent on any key of the table.

**THIRD NORMAL FORM (3NF):**

Third Normal Form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this *transitive functional dependency* should be removed from the table and also the table must be in the Second Normal Form.

**2.3.1 NORMALISATION OF EVENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Name | Description | Date | Video\_link | Image\_link |

ID name

ID description

ID date

ID video\_link

ID image\_link

**First Normal Form**

In EVENT, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**2.3.2 NORMALISATION OF BOOKING**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Book\_id | Cust\_id | Ticket\_type |

Book\_id id

Book\_id cust\_id

Book\_id ticket\_type

**First Normal Form**

In BOOKING, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**2.3.3 NORMALISATION OF CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cust\_id | Cust\_name | Email | Phone | password |

cust\_id cust\_name

cust\_id email

cust\_id phone

cust\_id password

**First Normal Form**

In CUSTOMER, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**2.3.4 NORMALISATION OF TICKETS**

|  |  |  |  |
| --- | --- | --- | --- |
| id | Standard | Pro | Premium |

id standard

id pro

id premium

**First Normal Form**

In TICKETS, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**2.3.4 NORMALISATION OF TICKETS**

|  |  |  |  |
| --- | --- | --- | --- |
| id | Standard | Pro | Premium |

id standard

id pro

id premium

**First Normal Form**

In TICKETS, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**2.3.5 NORMALISATION OF SPONSORS**

|  |  |  |
| --- | --- | --- |
| id | Sponsor\_name | Sponspor\_link |

id Sponsor\_name

id Sponsor\_link

**First Normal Form**

In SPONSORS, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**2.3.1 NORMALISATION OF EVENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Venue\_id | Venue\_name | Location | description | Image\_link |

ID, Venue\_id Venue\_name

ID ,Venue\_id location

ID ,Venue\_id description

ID ,Venue\_id image\_link

**First Normal Form**

In EVENT, all the attributes are atomic and there cannot be duplicate rows.

Hence, it is in 1NF.

**Second Normal Form**

This is already in 2NF since every non key attribute is fully dependent on primary key.

**Third Normal Form**

Since there is no transitive functional dependency, therefore table is already in 3NF.

**CHAPTER-3**

**FRONT END DESIGN**

**3.1 SCREEN LAYOUT DESIGN FOR FORMS**

**HTML <p> TAG:**

The HTML <p> element represents a paragraph of text. Paragraphs are usually

represented in visual media as blocks of text that are separated from adjacent blocks by

vertical blank space and/or first-line indentation. Paragraphs are block level elements.

**HTML <div> TAG:**

The HTML <div> element is the generic container for flow content and does not

inherently represent anything. Use it to group elements for purposes such as styling

(using the class or id attributes), marking a section of a document in a different language

(using the lang attribute), and so on.

**HTML <br> TAG:**

The HTML <br> element produces a line break in text (carriage-return). It is

useful for writing a poem or an address, where the division of lines is significant.

**HTML <input> TAG:**

The HTML <input> element is used to create interactive controls for web-based

forms in order to accept data from the user. An <input> works varies considerably depending on the value of its type attribute, hence the different types are covered in their own separate reference pages. If this attributes is not specified, the default type adopted type is text.

<input> elements of type text create basic, single-line inputs. You should use them

anywhere you want the user to enter a single-line value and there isn't a more specific

input type available for collecting that value (for example, if it's a date, URL, email,

or search term, you've got better options available).You can provide a useful placeholder inside your text input that can provide a hint as to what to enter by including using the placeholder attribute. The available types are as follows:

password: A single-line text field whose value is obscured. Use the maxlength and minlength attributes to specify the maximum length of the value that can be entered.

**HTML <table> TAG:**

The **HTML <table> element** represents tabular data — that is, information expressed via a two-dimensional data table.

The **HTML <tr> element** defines a row of cells in a table. Those can be a mix of <td> and <th> elements.

The **HTML <td> element** defines a cell of a table that contains data. It participates in the table model.

The **HTML <head> element** defines a set of rows defining the head of the columns of

the table.

The **HTML <body> element** groups one or more <tr> elements as the body of

a <table> element.

**HTML <form> TAG:**

The HTML <form> element represents a document section that contains interactive controls to submit information to a web server. It is possible to use the :valid and :invalid CSS pseudo-classes to style a <form>element. The HTTP method that the browser uses to submit the form. Possible values are:

**post:** Corresponds to the HTTP POST method ; form data are included in the body of the form and sent to the server.

**get:** Corresponds to the HTTP GET method; form data are appended to the action attribute URI with a '?' as separator, and the resulting URI is sent to the server. Use this method when the form has no side-effects and contains only ASCII characters. This value can be overridden by a form method attribute on a <button> or <input>element.

**action:** The URI of a program that processes the form information. This value can be overridden by a form action attribute on a <button> or <input> element.

**REGISTER FORM:** The user cannot view the flight details and book a ticket if he has not registered. So to login he has to register himself first. The register form consists of six text fields and one register button. The text fields consist of username, password, password again, first name, last name and email .The register button posts the data to the servlet.

**LOGIN FORM:** The login form consists of two text fields and one login button. The text fields consist of username where user enters the username with which he has registered and password where the user enters the password given when he had registered. The login button posts the data to the servlet

**EVENT BOOKING FORM:** Once the user logins himself, he can start booking and viewing the event details. This event booking form consists of two test fields which are namely to enter the start date and a end date where these two text fields make use of a date picker and the search button posts the entered data to the servlet and shoes the event details.

**EVENT LIST:** this is to choose between the events in the given date. After selecting an event, it is shows details of individual event such as venue, date and ticket prices.

**3.2 Connection between Front End and Back End**

## **3.2.1 Connecting to a MySQL database**

You need your MySQL server address (if the database is on the same server as the web server it will most likely be **localhost** or **127.0.0.1**), username*,*password and database name. Create a **filenamehere.php** file and open and close the phpcode with tags before the html, you can put regular html after it. Open the file in a browser and you should see nothing apart from the title tag, if you see the error the username/password or database name may be wrong.

// Create connection

**<?php**

**$connect\_error = 'sorry, server is down';**

**mysql\_connect('localhost', 'root', '') or die($connect\_error);**

**mysql\_select\_db('lr') or die($connect\_error);**

**?>**

Here the local host is the server name, root is the username and since I have not given any password that field is empty.

**CHAPTER – 4**

**MAJOR MODULES**

The major modules of the Event management system are:

* Login
* Search events
* Event list
* Event details
* Select tickets and booking
* Customer details
* Edit details
* Logout

1. **Login**

In the login module there are two labels inside the textbooks namely email-id and password and an arrow button that leads the user to the next event searching page.the user has to enter the email-id and the password to login.

2. **Search events**

If the customer clicks on “Log in”, with correct authentication, the page will drive them to next page which is the search page. In the search page, the customer needs to enter a time period. After selecting the time period of interest the customer has to press the go button to enter the next page.

3. **Event list**

. On entering the time period, all the events happening within this time period will be displayed in this page which is the results page of the event details. In this page , the customer can choose the desired event by clicking on the ID of the event. This will lead to the details page.

4. Event **details**

The event details page displays

* A video about the event
* The description of the event
* The event venue
* The sponsors of the event
* The ticket types

5. **Select tickets and booking**

On viewing the ticket types, the customer can select a ticket type to buy. A dropdown appears requesting the customer to enter their email ID and the preferred ticket type. The customer can then click on the “buy now” button to confirm their booking.

After buying the tickets, the customer can either

* Edit their account details

The customer can update email ,password and phone number.

* Delete their account

The customer can delete the account.

* Log out

A “log out” button is placed in the search page to help the customer log out in between the transaction.

**CHAPTER – 5**

**IMPLEMENTATION IN PHP, HTML & MYSQL**

**DATABASE CODE**:

**TABLES CREATION**

Booking table

CREATE TABLE IF NOT EXISTS `booking` (

`book\_id` int(11) NOT NULL,

`id` int(2) NOT NULL,

`cust\_idl` varchar(2) NOT NULL,

`ticket\_type` text NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

Customer table

CREATE TABLE IF NOT EXISTS `customer` (

`cust\_id` varchar(2) NOT NULL,

`cust\_name` text NOT NULL,

`email` varchar(30) NOT NULL,

`password` text NOT NULL,

‘phone` int(10) NOT NULL,

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

Events table

CREATE TABLE IF NOT EXISTS `events` (

`id` int(2) NOT NULL,

`name` text NOT NULL,

`description` text NOT NULL,

`video\_link` varchar(55) NOT NULL,

‘venue\_id` int(5) NOT NULL,

`image\_link` varchar(55) NOT NULL,

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

Sponsors table

CREATE TABLE IF NOT EXISTS `sponsor` (

`id` int(2) NOT NULL,

`sponsor\_name` text NOT NULL,

`logo\_link` varchar(55) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

Ticket table

CREATE TABLE IF NOT EXISTS `ticket` (

`id` int(3) NOT NULL,

`standard` int(3) NOT NULL,

`pro` int(3) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

Venue table

CREATE TABLE IF NOT EXISTS `venue` (

`id` int(3) NOT NULL,

`venue\_name` text NOT NULL,

`location` text NOT NULL,

`venue\_description` text NOT NULL,

`photo\_link` varchar(55) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=latin1;

ALTER TABLE `booking`

  ADD PRIMARY KEY (`book\_id`),

  ADD KEY `id` (`id`),

  ADD KEY `cust\_id` (`cust\_id`);

ALTER TABLE `booking\_backup`

  ADD PRIMARY KEY (`book\_id`),

  ADD KEY `id` (`id`),

  ADD KEY `cust\_id` (`cust\_id`);

ALTER TABLE `customer`

  ADD PRIMARY KEY (`cust\_id`);

ALTER TABLE `event`

  ADD PRIMARY KEY (`id`),

  ADD KEY `venue\_id` (`venue\_id`);

ALTER TABLE `sponsors`

  ADD KEY `id` (`id`);

ALTER TABLE `ticket`

  ADD PRIMARY KEY (`id`);

ALTER TABLE `venue`

  ADD PRIMARY KEY (`venue\_id`);

ALTER TABLE `booking`

  MODIFY `book\_id` int(3) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=85;

ALTER TABLE `booking\_backup`

  MODIFY `book\_id` int(3) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=85;

ALTER TABLE `sponsors`

  MODIFY `id` int(3) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

ALTER TABLE `booking`

  ADD CONSTRAINT `booking\_ibfk\_1` FOREIGN KEY (`id`) REFERENCES `event` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,

  ADD CONSTRAINT `booking\_ibfk\_2` FOREIGN KEY (`cust\_id`) REFERENCES `customer` (`cust\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `event`

  ADD CONSTRAINT `event\_ibfk\_1` FOREIGN KEY (`venue\_id`) REFERENCES `venue` (`venue\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `sponsors`

  ADD CONSTRAINT `sponsors\_ibfk\_1` FOREIGN KEY (`id`) REFERENCES `event` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `ticket`

  ADD CONSTRAINT `ticket\_ibfk\_1` FOREIGN KEY (`id`) REFERENCES `event` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;

**INSERTION OF VALUES**

INSERT INTO `booking` (`book\_id`, `id`, `cust\_id`, `ticket\_type`) VALUES

(72, 1, 53, 'pro'),

(73, 1, 53, 'pro'),

(74, 4, 53, 'pro'),

(75, 4, 53, 'pro'),

(76, 3, 53, 'pro'),

(77, 1, 53, 'pro'),

(78, 1, 53, 'standard'),

(79, 1, 53, 'standard'),

(80, 1, 53, 'pro'),

(81, 1, 53, 'premium'),

(82, 1, 53, 'pro'),

(83, 4, 53, 'pro'),

(84, 4, 53, 'pro');

INSERT INTO `customer` (`cust\_id`, `cust\_name`, `email`, `password`, `phone`) VALUES

(53, 'Dave', '[dave@gmail.com](mailto:dave@gmail.com)', '123', '8095208192'),

(54, 'Clinton', '[clinton@gmail.com](mailto:clinton@gmail.com)', '123', '9964062018'),

(55, 'John', '[john@gmail.com](mailto:john@gmail.com)', '123', '9964062018');

INSERT INTO `event` (`id`, `name`, `description`, `date`, `video\_link`, `venue\_id`, `image\_link`) VALUES

(1, 'Sunburn Festival', 'Sunburn festival is a commercial EDM fest held in India every year. This year India\'s most loved DJ Nucleya is back!', '2018-09-01', '<https://www.youtube.com/watch?v=7yzXmLz-Q14>', 12, 'img/venue-info-bg.jpg'),

(2, 'Comedy Fest', 'A special showcase of the country\'s best comics. Will surely tickle your funny bone!', '2018-10-04', '[https://www.youtube.com/watch?v=W6Vpf9zmIQE\r\n](https://www.youtube.com/watch?v=W6Vpf9zmIQE%5Cr%5Cn)', 11, 'img/slide.jpg'),

(3, 'Life In disaster', 'Funniest interview ever!!', '2018-12-30', 'https://www.youtube.com/watch?v=beVSHiyog4o', 14, 'img/subscribe-bg.jpg'),

(4, 'Holi Party', 'Indulge in a splash of Colors this Holi!', '2018-09-20', 'https://www.youtube.com/watch?v=Hh-o5g4tLVE', 12, 'img/holi.jpg');

INSERT INTO `sponsors` (`id`, `sponsor\_name`, `logo\_link`) VALUES

(1, 'Strider', 'img/sponsors/1.png'),

(1, 'runtastic', 'img/sponsors/2.png'),

(1, 'EditShare', 'img/sponsors/3.png'),

(3, 'cadent', 'img/sponsors/6.png'),

(3, 'gategroup', 'img/sponsors/5.png'),

(1, 'editshare', 'img/sponsors/3.png'),

(4, 'Strider', 'img/sponsors/1.png'),

(2, 'editshare', 'img/sponsors/3.png');

5

INSERT INTO `ticket` (`id`, `standard`, `pro`, `premium`) VALUES

(1, 200, 300, 400),

(2, 100, 200, 300),

(3, 150, 300, 500);

INSERT INTO `venue` (`venue\_id`, `venue\_name`, `location`, `Venue\_description`, `photo\_link`) VALUES

(11, 'Hard Rock Cafe', 'Bangalore', 'It cannot get harder than this!\r\nRock and roll-themed chain with a high-energy vibe serving burgers & American classics.', 'img/venue-gallery/1.jpg'),

(12, 'Palace Grounds', 'Bangalore', 'The sprawling grounds surrounding the palace used for holding public events including music concerts, weddings and much more.', 'img/hotels/2.jpg'),

(13, 'Elite grounds', 'Mumbai', 'Only elites and legends allowed!\r\nGreat ambiance, great memories. ', 'img/hotels/1.jpg'),

(14, 'Paramount studios', 'Delhi', 'Paramount studios - Live shows, Theater and Enriched experiences', 'img/gallery/8.jpg'),

(15, 'The Leela Palace ', 'Chennai', 'The Leela Palace Chennai is a sea facing 5-star luxury business hotel. Enjoy serenity and comfort all in one.', 'img/venue-gallery/3.jpg');

]

**FRONT END IMPLEMENTATION IN PHP AND HTML**

**Login Page**

<?php

require 'connection.php';

require 'valid.php';;

if(isset($\_POST['email'])&&isset($\_POST['password'])){

    $username=$\_POST['email'];

    $password=$\_POST['password'];

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

        $sql="select cust\_id from customer where email='".$username."' and password='".$password."' limit 1";

        if($query=mysqli\_query($conn,$sql)){

            $query\_run=mysqli\_num\_rows($query);

            if($query\_run==0)

            {

                echo "<script>alert('Invalid Username or Password')</script>";

            }

        elseif($query\_run==1)

        {

                echo $user\_id=mysqli\_fetch\_array($query,MYSQLI\_NUM);

                $\_SESSION['user\_id']=$user\_id[0];//setting up session variables.

                Header('Location: search.php');

        }

    }

}

}

**Code for event results page**

<?php

require 'connection.php';

require 'valid.php';

$output='';

if(loggedin())

{

  echo 'Welcome '.$\_SESSION['user\_id'];

  }

else

{

  Header('Location: login.php');

}

if(isset($\_POST['from']))

{

  $from = $\_POST['from'];

$to = $\_POST['to'];

$query = mysqli\_query($conn,"SELECTid,name,description,image\_link FROM event WHERE date BETWEEN '$from' AND '$to' ") or die("could not search!");

$count = mysqli\_num\_rows($query);

if($count !== 0)

    {

    while($row = mysqli\_fetch\_array($query)){

$event\_id = $row['id'];

$event\_name = $row['name'];

$event\_desc = $row['description'];

$event\_image = $row['image\_link'];

$output .= '<div class="row">

<div class="col s12 m7">

<div class="card">

<div class="card-image">

<imgsrc="'.$event\_image.'">

<span class="card-title">'.$event\_name.'</span>

</div>

<div class="card-content">

<p>'.$event\_desc.'</p>

</div>

<div class="card-action">

<form action="detail.php" method="post">

<div class="btn-group btn-group-justified" role="group" aria-label="...">

<div class="btn-group" role="group">

<input type="submit" class="btnbtn-default" name="navbtnplace" value="'.$event\_id.'">

</div>

</div>

</form>

</div>

</div>

</div>

</div>';

}

}

else if($count ==0){

$output = 'There was no search result!';

}

}

**Code for Event Details Page**

<?php

require 'valid.php';

require 'connection.php';

$output = '';

if(loggedin())

{

  echo 'Welcome '.$\_SESSION['user\_id'];

  }

else

{

  Header('Location: login.php');

}

if(isset($\_POST['navbtnplace'])) {

$\_SESSION['event\_id'] = $\_POST['navbtnplace'];

echo $\_SESSION['event\_id'];

$user\_id = $\_SESSION['user\_id'];

$event\_id = $\_SESSION['event\_id'];

$event\_query = mysqli\_query($conn,"SELECT \* FROM event WHERE id='$event\_id'") or die("could not search!");

$event\_result = mysqli\_fetch\_array($event\_query);

$venue\_id = $event\_result['venue\_id'];

echo $venue\_id;

$venue\_query = mysqli\_query($conn,"SELECT \* FROM venue WHERE venue\_id='$venue\_id'") or die("could not search!");

$venue\_result = mysqli\_fetch\_array($venue\_query);

$sponsor\_query = mysqli\_query($conn,"SELECT \* FROM sponsors WHERE id='$event\_id'") or die("could not search!");

$count = mysqli\_num\_rows($sponsor\_query);

if($count !== 0)

    {

    while($row = mysqli\_fetch\_array($sponsor\_query)){

$logo\_link = $row['logo\_link'];

$output .= '<div class="col-lg-3 col-md-4 col-xs-6">

<div class="sponsor-logo">

<imgsrc="'.$logo\_link.'" class="img-fluid" alt="">

</div>

</div>';

}

}

else if($count ==0){

$output .= 'There are no sponsors for this event';

}

**Ticket Booking Query**

$ticket\_query = mysqli\_query($conn,"SELECT \* FROM ticket WHERE id='$event\_id'") or die("could not search!");

$ticket\_result = mysqli\_fetch\_array($ticket\_query);

**Code for Update Account Details Page**

<?php

require 'valid.php';

require 'connection.php';

if(isset($\_POST['submit'])){

$phone = $\_POST['phone'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$update = "UPDATE customer SET email = '$email', password = '$password', phone = '$phone' WHERE cust\_id = '".$\_SESSION['user\_id']."' ";

$query = mysqli\_query($conn,$update);

if($query){

echo '<br>UPDATE data is successful';

} else {

    echo '<br>UPDATE data is not valid';

    session\_destroy();

    Header("Location: login.php");

}

}

**Code for Delete Page**

<?php

require('valid.php');

require('connection.php');

$id=$\_SESSION['user\_id'];

$result = mysqli\_query($conn,"DELETE FROM customer WHERE cust\_id='$id'") or die ( mysqli\_error());

session\_destroy();

echo "Account deleted successfully.";

header("Location: login.php");

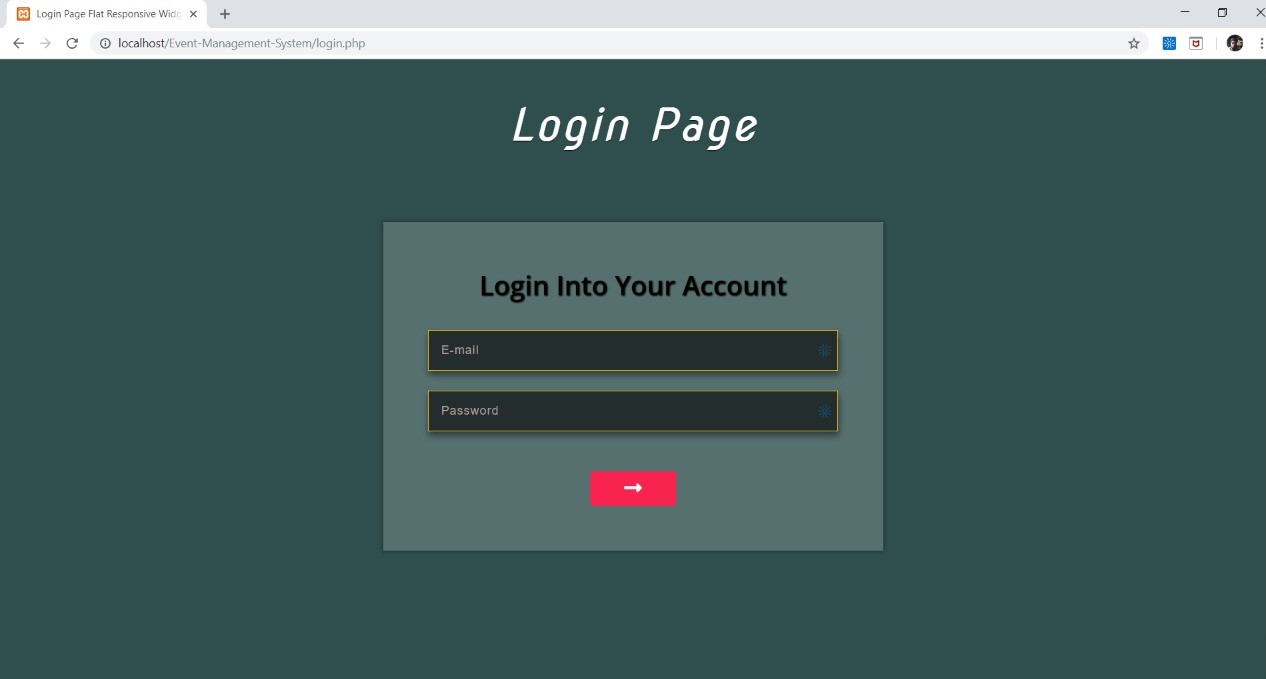
?>

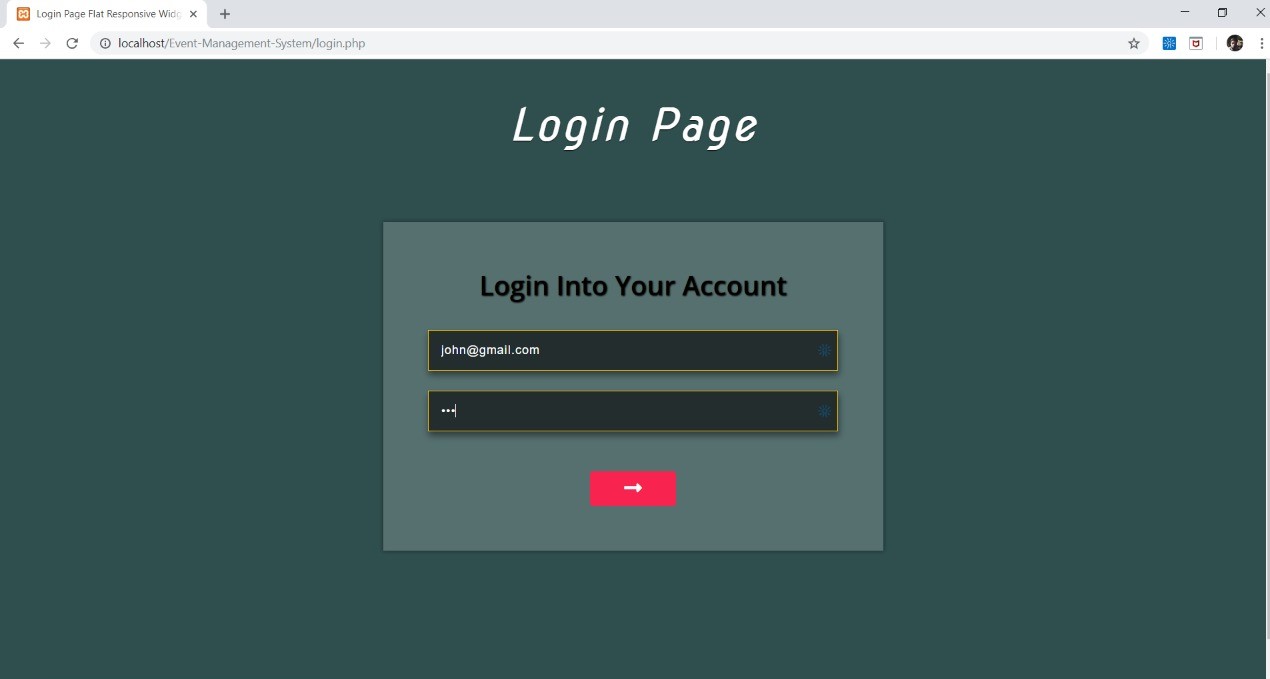
**CHAPTER – 6**

**SNAPSHOTS**

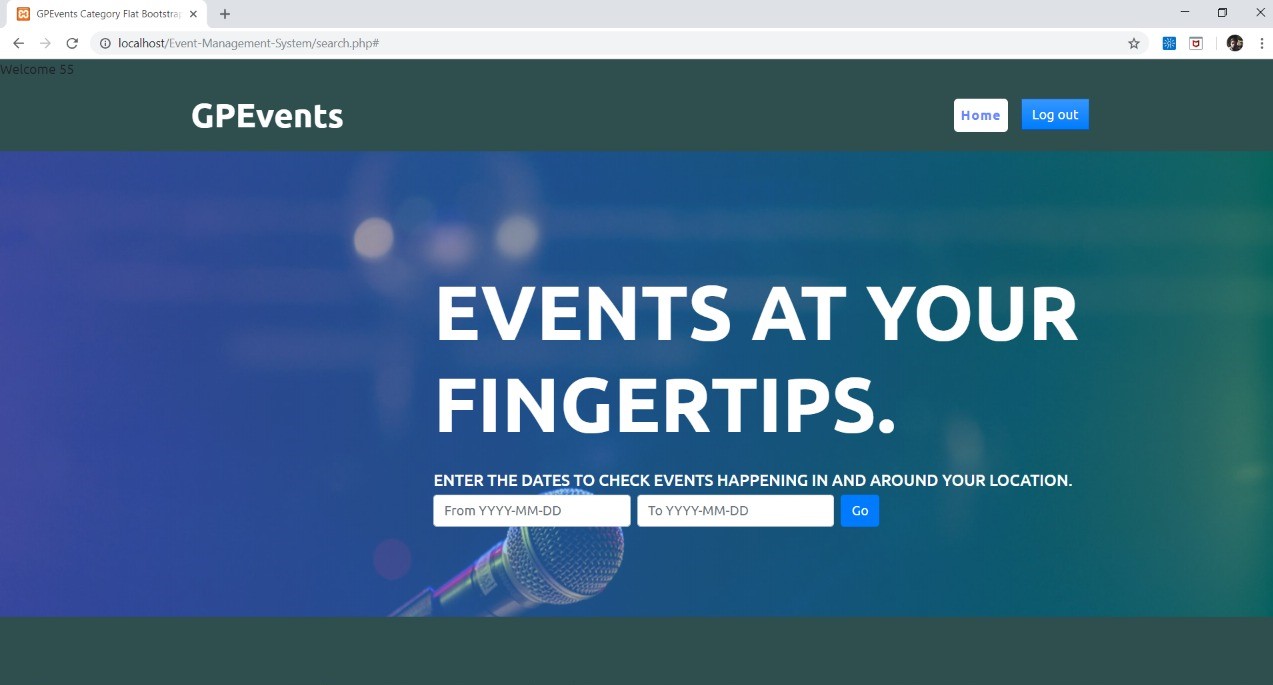
**SNAPSHOTS**

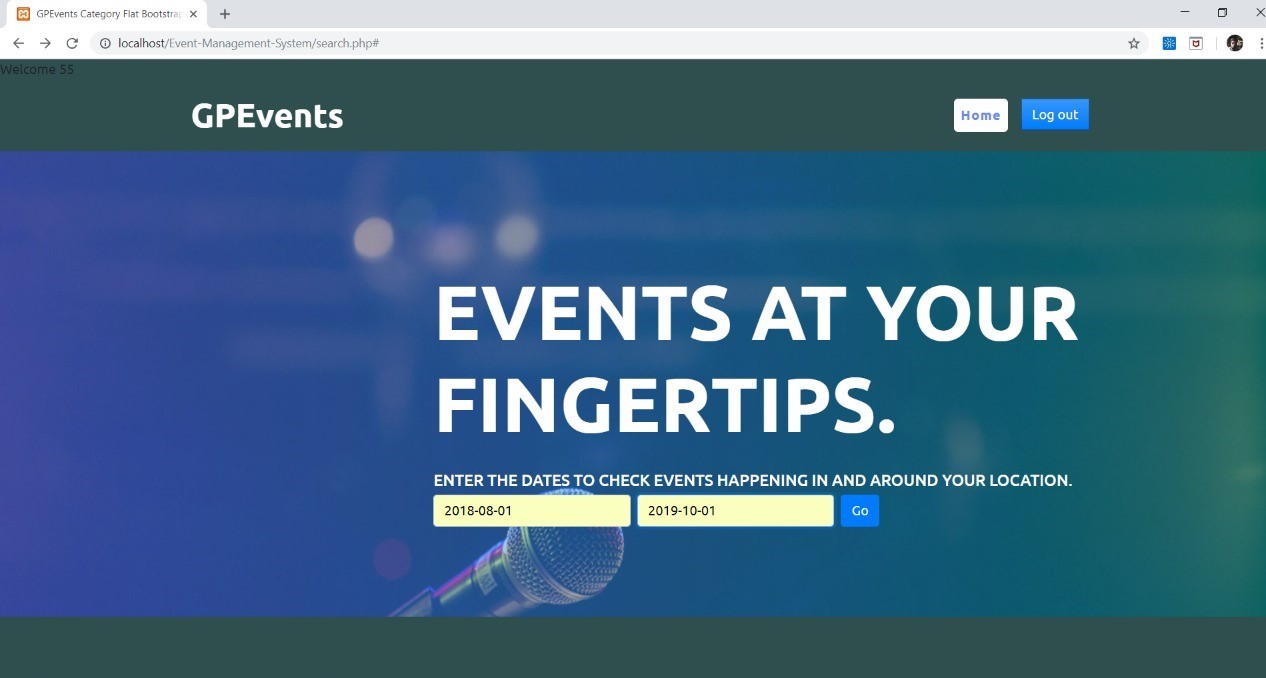
1. The user enters the mail-id and password and logins

****

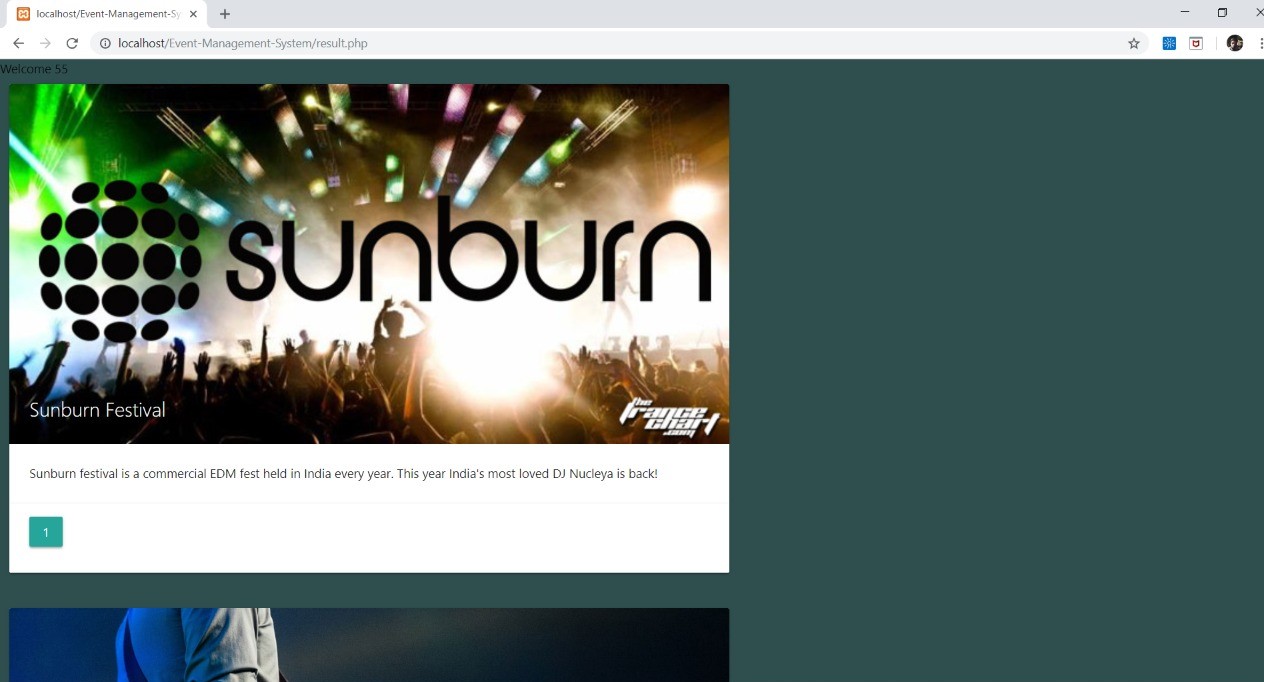
****

The user enters the time period

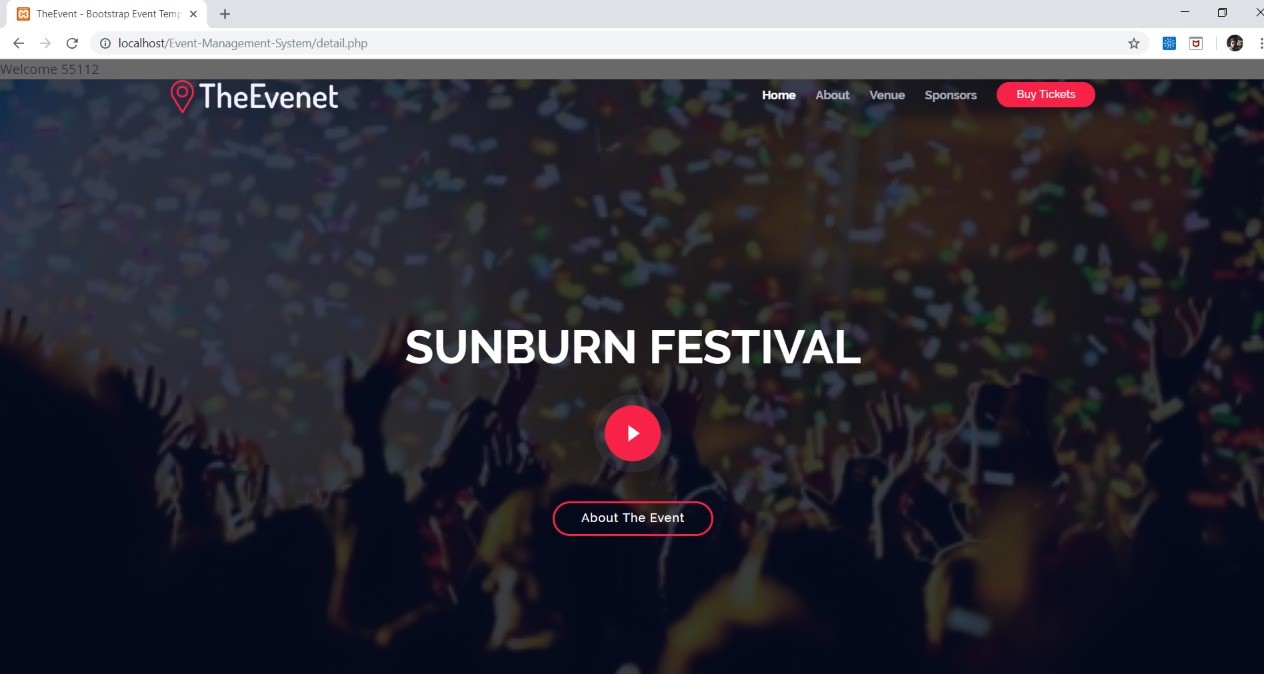
****

****

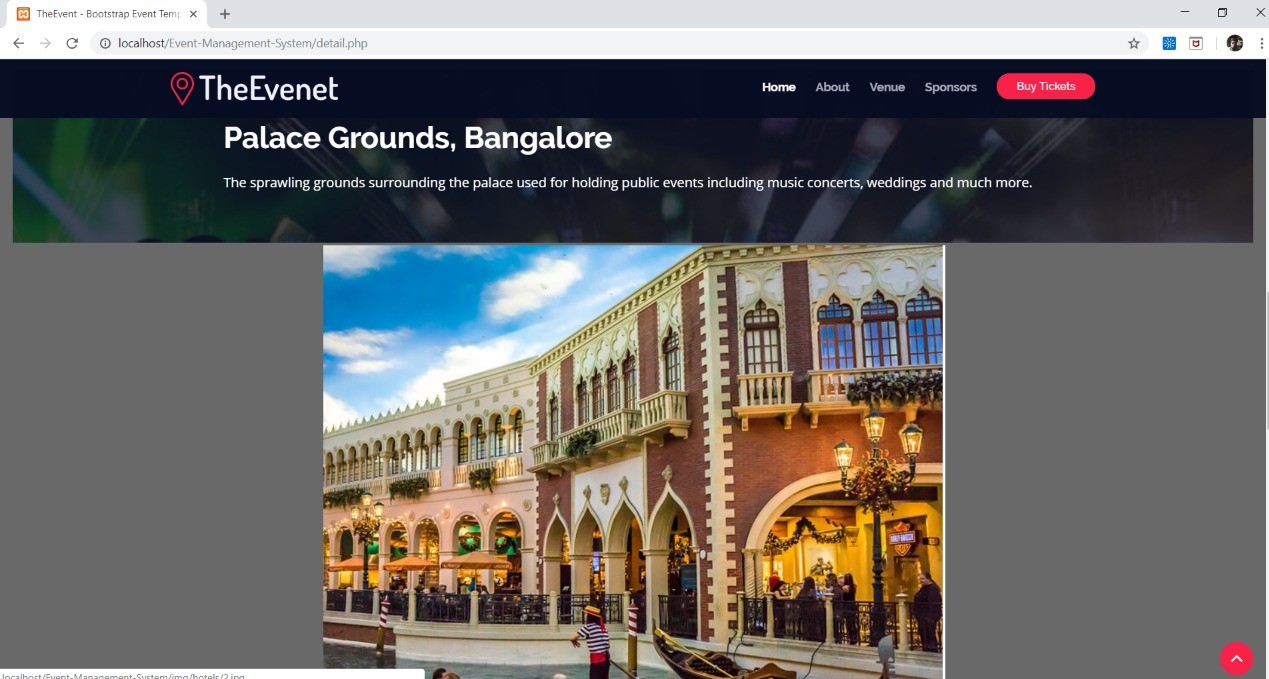
The list of event appears on the screen based on the entered time period

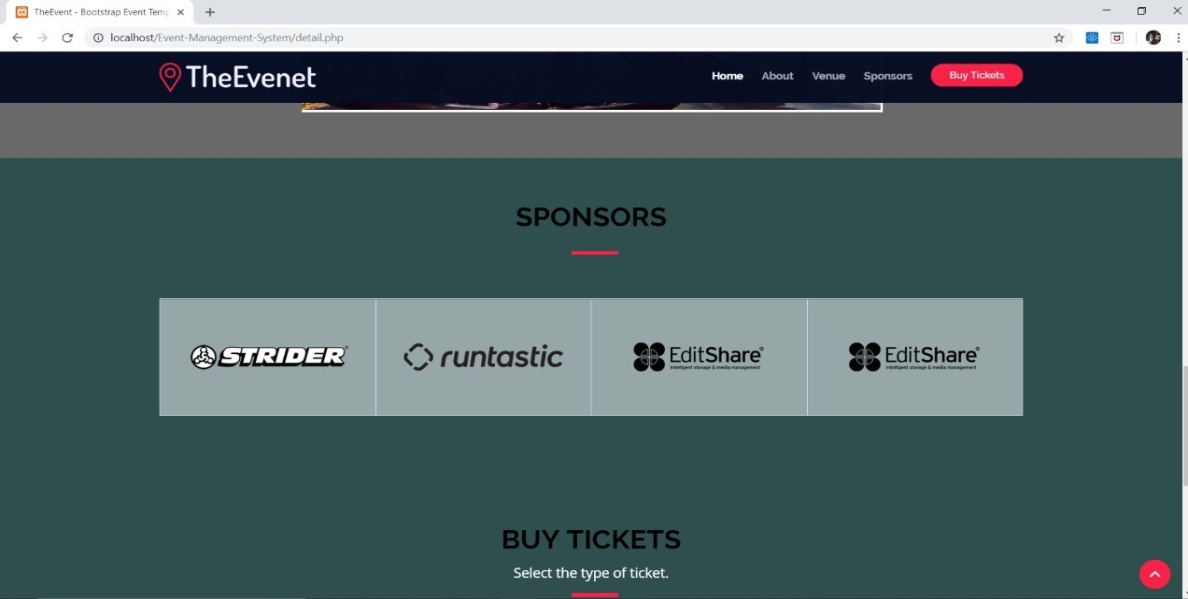


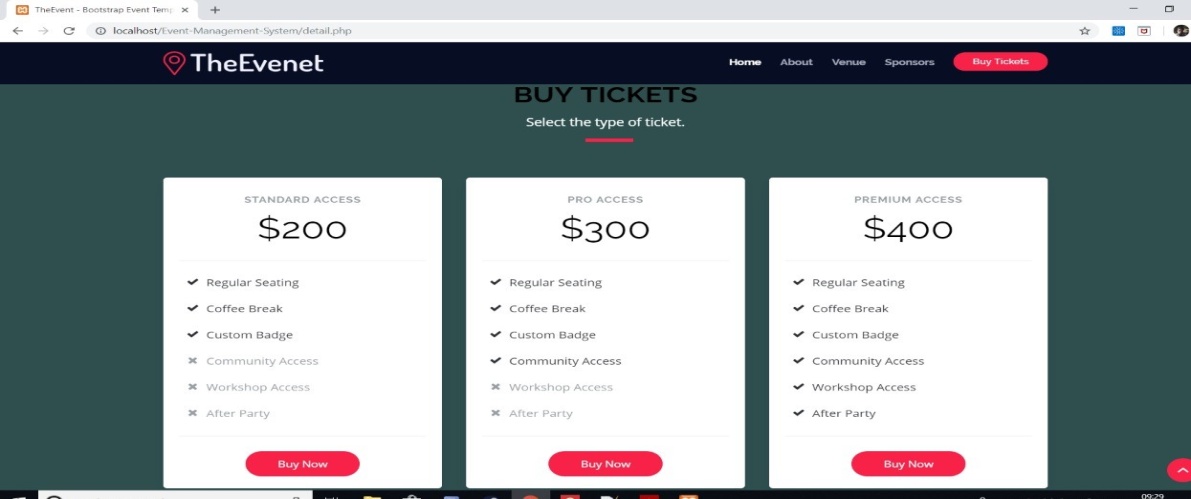
4.The user selects the event by clicking on the event id



5. The details of the selected event like venue ,sponsors and tickets are displayed.

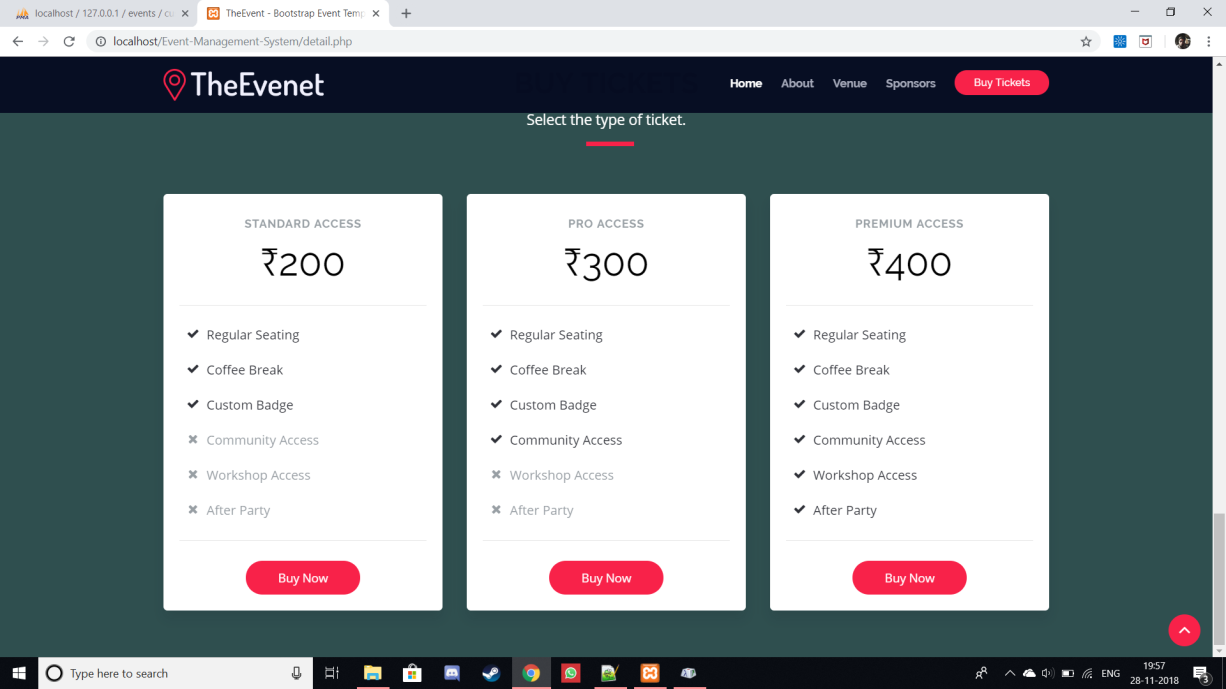


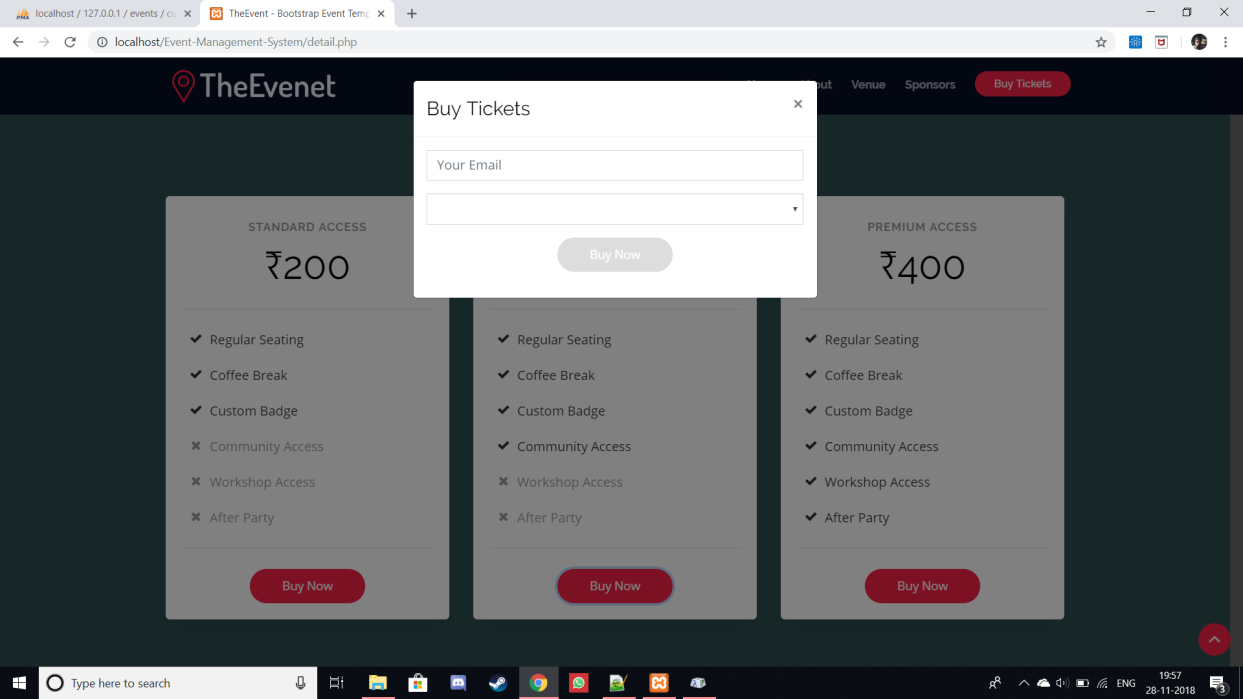


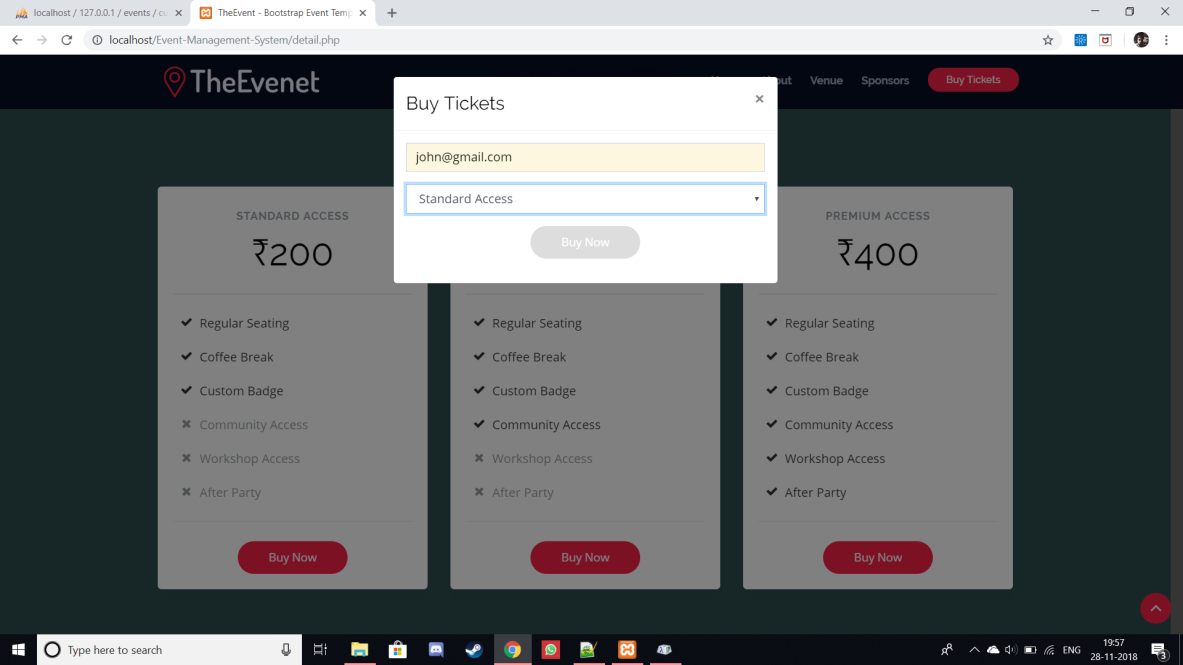


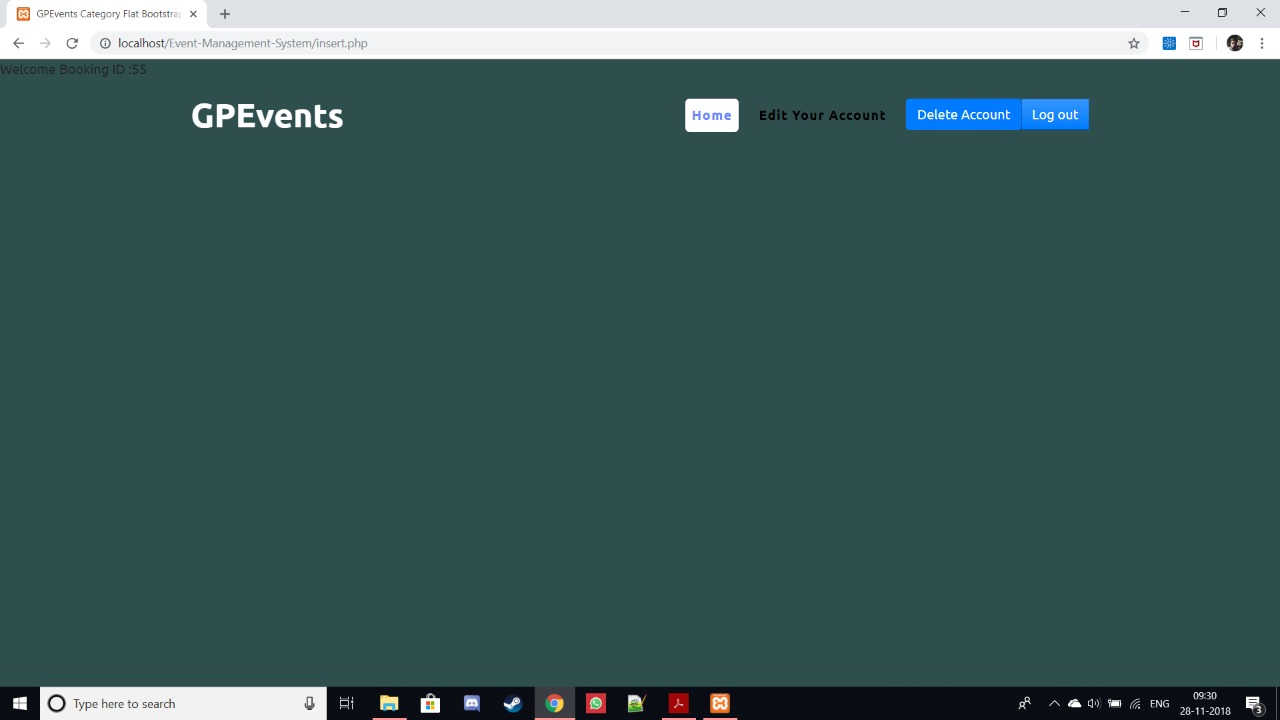
6.The user has to buy the ticket by entering the mail-id and the type of the ticket.

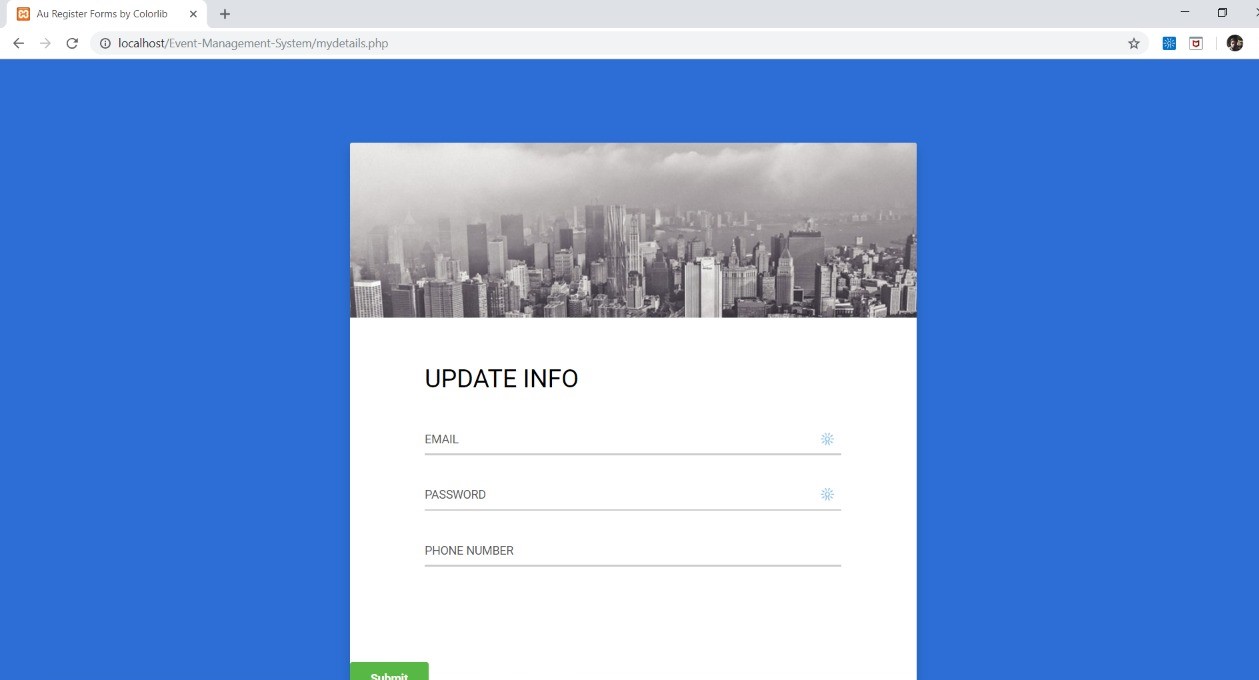
A notification appears stating that the ticket has been booked successfully .



****

****

****

****

**CHAPTER– 7**

**APPLICATIONS**

**APPLICATIONS**

* The event management system has various advantages. The event management system is useful as it allows the user to book for the event at a click of a button. The user gets all the resources at a single place instead of wandering around for bookings. This system is effective and saves time and cost of the users.
* It also reduces the burden on the organizers side as the services offered will be known to the customer.
* Customers need not go to the office for discussing about the event

Specifications and services.

* Event management system is used to reduce the stress of customer and provide various variety of options present on the same platform.

**CHAPTER-8**

**CONCLUSION**

**CONCLUSION**

It was a wonderful learning experience working on this project where designing took me through various phases of project development and gave me real insight into the world of software engineering. The joy of working and the skill involved while tackling the various problems and challenges gave me a feel of developer’s industry.

This project is simulated using the software called XAMPP. The project

uses PHP and HTML functions to show the demonstration of EVENT MANAGEMENT SYSTEM

Finally I would like to conclude that, with the use of event management System we all can save precious time, energy of the customer to buy a ticket. Customers can book ticket easily sitting at their home or office or any place. It helps the customer by reducing the stress and the options are available at the finger tip of the customer.

It was due to project that I came to know how professional software’s are designed. I enjoyed each and every bit of work In order to complete this project successfully.