## Presentup

# **Find Your Ideal Vendors At Best Price**

**Submitted by:**

**Salmanul Farisy k**

**MCA 2nd Semester**

**Roll no : 49**

**Theme of the project**

The project entitled to develop an “Event Management System” that manages the event registered by the user. The main objective of the project to handle each events section for the event console and staff members for the particular event. The system need the registration for the user to create an account for the user. Based on this registration the user can get register event. The most important data that the system receives from the user is “the time and the category of the event” register.

**OBJECTIVE OF THE PROJECT**

The system automatically generate certificate and issue it, also system very efficiently store, maintain, and retrieve data from its database and can be used for further analysis. This system provides latest notification to do its user. Time saving activity. The data on a centralized way which is available to all the event managers. Easy to manage historical data in database. Participants can register for any happening event from anywhere. Event manager can keep records for participants.

**USERS OF THE PROJECT**

* **Administrator (Admin)**

Admin who act as the administrator of the system, who handles the different events registered by the users and also managing the customer’s needs. Admin have the permission to access the entire systems, so anytime he/she can add an event, delete an event, update an event, view the previous and current booked events, also provide confirmation process for booking through email to the customer.

* **Customer**

Customers are end-users, who act as customer, then visit the website and register on it. on. Once registered, they can login to the system by the using username and password which they have entered at the time of registration. If the login process succeeded, the customer can view the different events details provided by the system, book an event and update the information related to the booking events by their primary email-id.

* **Staff**

Staffs are the members who plays main role in conducting of an event. It is the responsibility of the Admin to allocate the staff members to an instant of the event with different works. The staffs can be allocated based up on their act of presents.

**MODULES OF THE SYSYEM**

1. **CUSTOMER MANAGEMENT**

In this module the details of the customer is stored like customer name, email id, username etc. Customer can view different events created by the admin. Customer can book the event which is desired by providing the event details. Customer update personal details and change password too. Customer can also access the gallery and see all the pictures.

1. **EVENT MANAGEMENT**

In this module the admin creates events and stores information like event name, description, rate etc. Admin can add a new event, delete an existing event.

1. **BOOKING MANAGEMENT**

In this module the booking information done by the customer is stored like event name, date, venue etc. Customer can view their booking and can check the status of the booking. Initially the status will be Not Confirmed. When the admin approves the booking done by the customer, the status will change to confirmed. The customer can cancel a booking within the specified number of days.

**DATABASE DESIGN**

Table: User Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Datatype** | **Constraint** | **Description** |
| User \_id | Int(100) | Primary key | Id of user |
| User \_fname | Varchar2(20) | Not Null | First name of user |
| User \_lname | Varchar2(20) | Not Null | Last name of the user |
| User \_gender | Varchar2(10) | Not Null | Gender of the user |
| User \_dob | Varchar2(20) | Not Null | DOB of the user |
| User \_address | Varchar2(300) | Not Null | Address of the user |
| User \_state | Varchar2(20) | Not Null | State of the user |
| User\_district | Varchar2(20) | Not Null | District of the user |
| User \_city | Varchar2(20) | Not Null | City of the user |
| User\_email | Varchar2(20) | Not Null | Email of the user |
| User\_phoneno | Varchar2(20 | Not Null | Phone no: of the user |
| User\_username | Varchar2(20) | Not Null | Username of the user |
| User\_password | Varchar2(20) | Not Null | Password of the user |
| User\_cpassword | Varchar2(20) | Not Null | Confirm password of the user |

Table: Wedding Banquet Booking

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Datatype** | **Constraint** | **Description** |
| Banquet\_id | Int(100) | Primary Key | Id of venue |
| User\_id | Int(100) | Primary Key | Id of venue |
| fname | Vachar2(20) | Not Null | Requester’s first name |
| lname | Vachar2(20) | Not Null | Requester’s last name |
| email | Vachar2(20) | Not Null | Requester’s email |
| username | Vachar2(20) | Not Null | Requester’s username |
| mob | Vachar2(20) | Not Null | Requester’s mobile number |
| altmob | Varchar2(20) | Not Null | Requester’s alternate mobile number |
| bfrom | Vachar2(20) | Not Null | Banquet booking date |
| bto | Vachar2(20) | Not Null | Banquet booking end date |
| blocation | Vachar2(20) | Not Null | Banquet location |
| btype | Varchar2(20) | Not Null | Banquet’s type |
| amt | Varchar2(20) | Not Null | Selected banquet’s type amount |
| veg | Varchar2(10) | Not Null | Is vegitable needed |
| non\_veg | Varchar2(10) | Not Null | Isnon- vegitable needed |
| veg\_imfl | Varchar2(10) | Not Null | Is vegetable+IMFL needed |
| nonveg\_imf | Varchar2(10) | Not Null | Is non-vegetable+IMFL needed |
| ex\_liquor | Varchar2(10) | Not Null | Is External liquor is needed |
| ex\_cool\_drinks | Varchar2(10) | Not Null | Is External cool drinks is needed |
| audio\_visual\_sys | Varchar2(10) | Not Null | Is Audio and visual system is needed |
| Live\_cast | Varchar2(10) | Not Null | Is live casting system is needed |
| message | Varchar2(1000) | Not Null | Additional information or messages |

Table: Wedding Event Booking

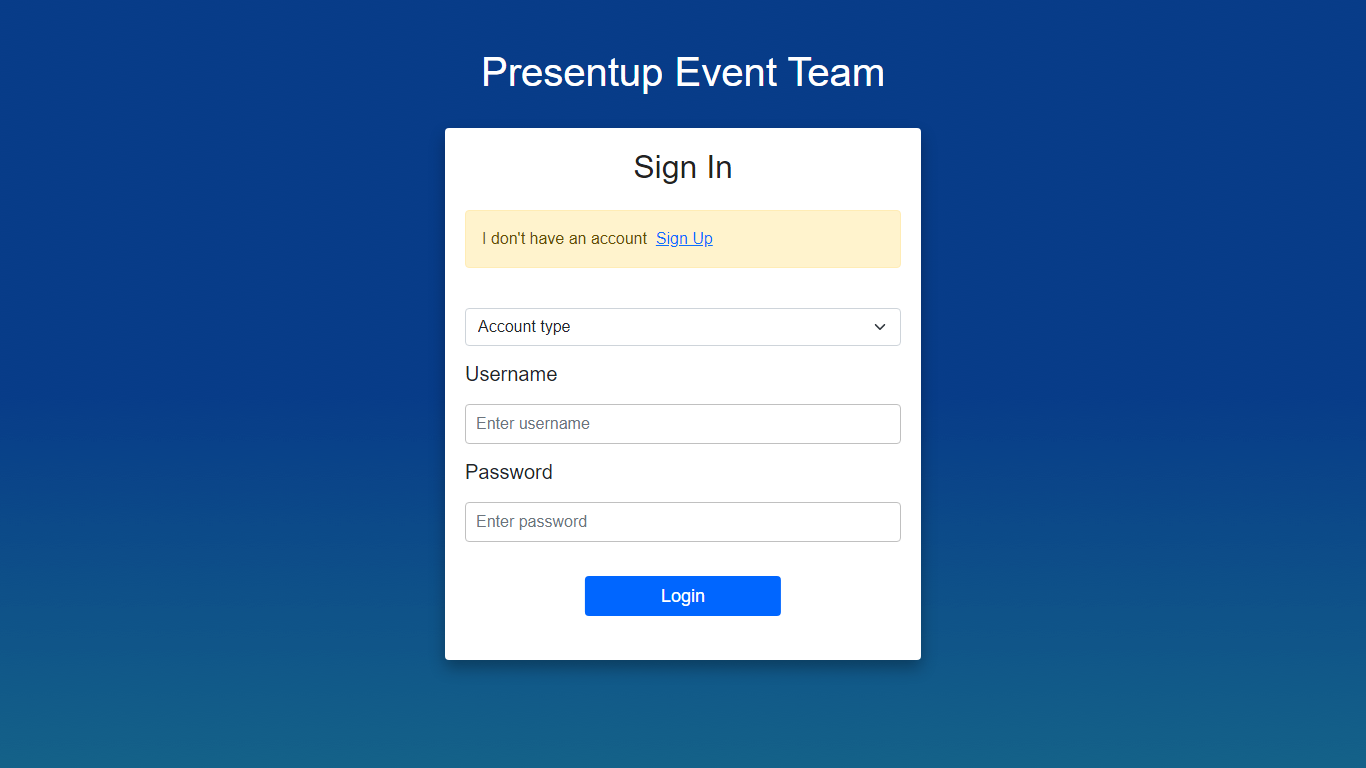
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Datatype** | **Constraint** | **Description** |
| EBook\_id | Int(100) | Primary Key | Id of booking |
| User\_id | Int(100) | Foreign Key | Id of the user |
| event\_start\_date | Date(20) | Not Null | Start date of the event |
| event\_start\_time | Date(20) | Not Null | Start time of the event |
| event\_end\_date | Varchar(10) | Not Null | end date of the event |
| event\_end\_time | Varchar(10) | Not Null | end time of the event |
| partno | Varchar2(1000) | Not Null | Participants no: |
| message | Varchar2(1000) | Not Null | Additional information |

Table: Registration Fee Payment table

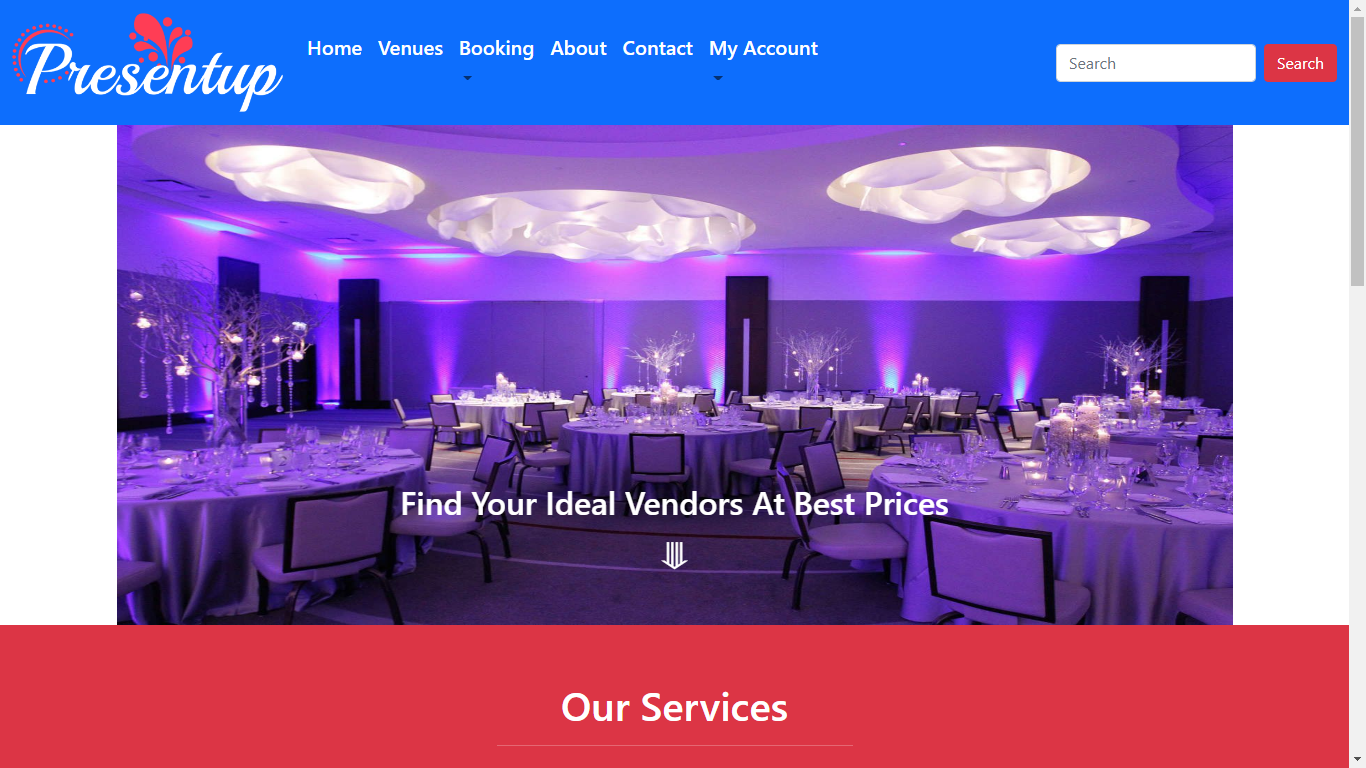
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Datatype** | **Constraint** | **Description** |
| Pay\_id | Int(100) | Primary Key | Id of payment |
| User\_id | Int(100) | Foreign Key | Id of the user |
| name | Varchar2(40) | Not Null | Name on card |
| cardno | Varchar2(40) | Not Null | Card no: |
| CVC | Varchar2(20) | Not Null | CVC no: on card |
| Expr\_month | Varchar2(20) | Not Null | Expire date |
| Expr\_year | Varchar2(20) | Not Null | Expire month |

**SCREENSHOTS**

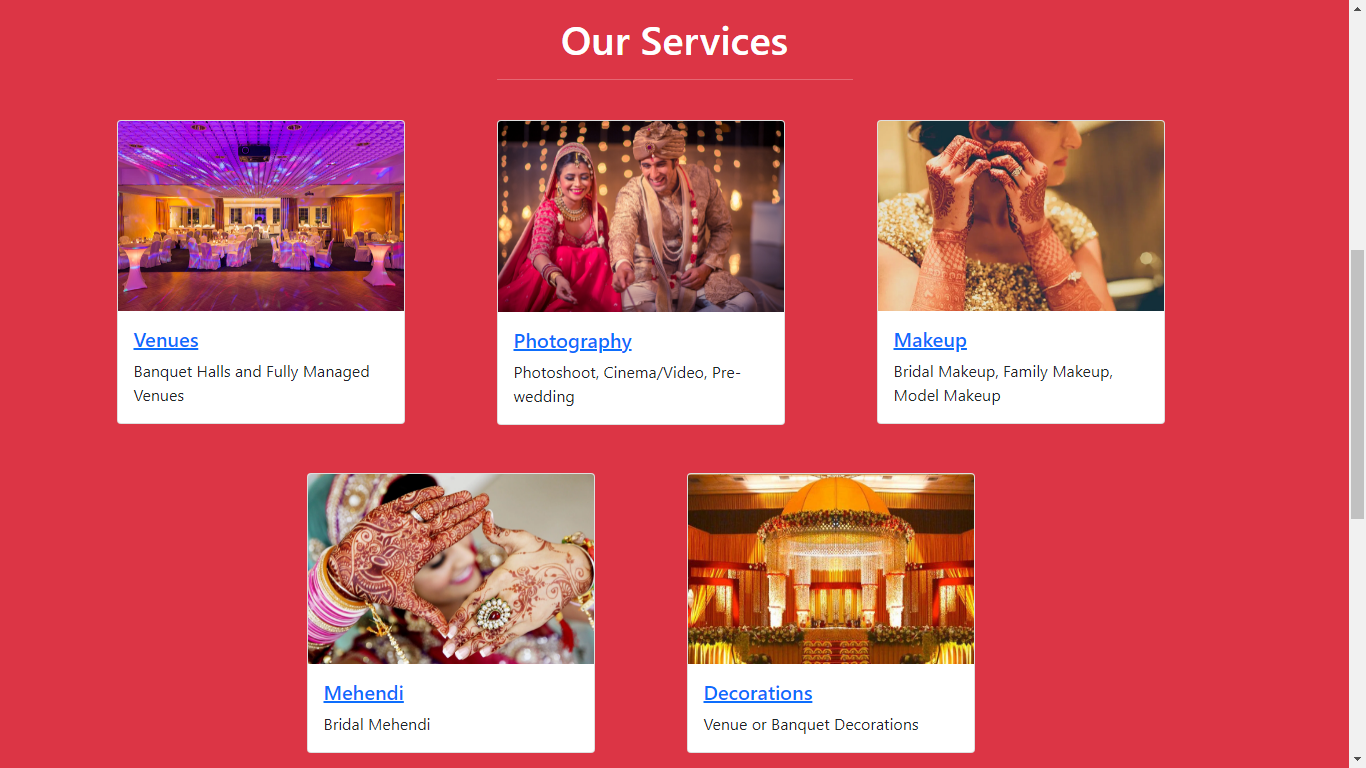
**Login:**



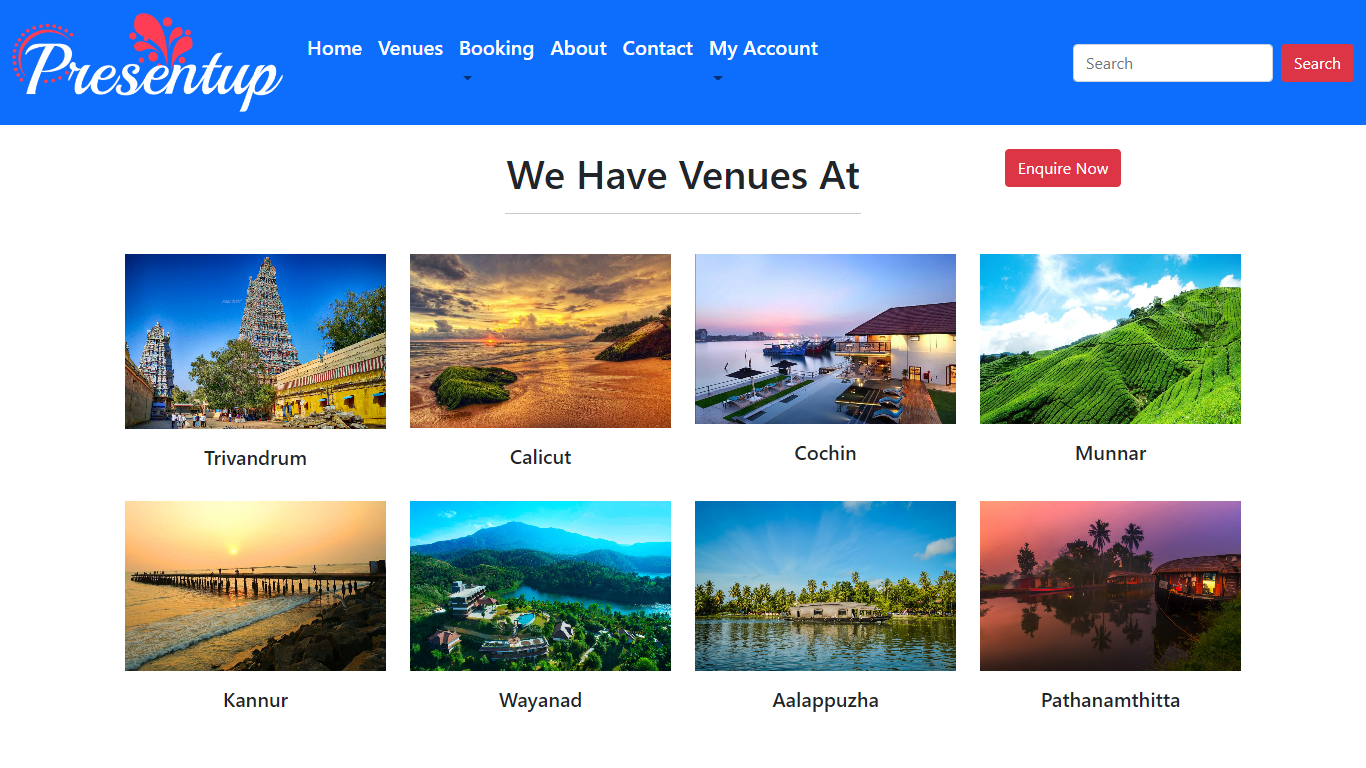
**Home:**



**Our Services:**



**Venues:**



**Sample code:**

**Login check:**

**function check(Request $request)**

**{**

**$request->validate([**

**'acctype'=>'required',**

**'username'=>'required',**

**'password'=>'required|min:4|max:12'**

**]);**

**$acctype=$request->acctype;**

**if($acctype=="Admin")**

**{**

**$admininfo=AdminModel::where('username','=',$request->username)->first();**

**if(!$admininfo)**

**{**

**return back()->with('fail','Invalid username');**

**}**

**else**

**{**

**if(Hash::check($request->password, $admininfo->password))**

**{**

**$request->session()->put('LoggedUser',$admininfo->id);**

**return redirect('/dashboard');**

**}**

**else**

**{**

**return back()->with('fail','Invalid password');**

**}**

**}**

**}**

**else if($acctype=="User")**

**{**

**$userinfo=RegisterModel::where('username','=',$request->username)->first();**

**if(!$userinfo)**

**{**

**return back()->with('fail','Invalid username');**

**}**

**else**

**{**

**if(Hash::check($request->password, $userinfo->password))**

**{**

**$request->session()->put('LoggedUser',$userinfo->id);**

**return redirect('/userhome');**

**}**

**else**

**{**

**return back()->with('fail','Invalid password');**

**}**

**}**

**}**

**else**

**{**

**return back()->with('fail','Select a account type');**

**}**

**}**

**Wedding Banquet Booking:**

**$data=RegisterModel::where('email','=',$request->email)->first();**

**$weddingbookingmodel= new WeddingBookingModel();**

**$weddingbookingmodel->fname=$request->fname;**

**$weddingbookingmodel->lname=$request->lname;**

**$weddingbookingmodel->email=$request->email;**

**$weddingbookingmodel->username=$data->username;**

**$weddingbookingmodel->mob=$request->mob;**

**$weddingbookingmodel->altmob=$request->altmob;**

**$weddingbookingmodel->bfrom=$request->bfrom;**

**$weddingbookingmodel->bto=$request->bto;**

**$weddingbookingmodel->blocation=$request->blocation;**

**$weddingbookingmodel->btype=$request->btype;**

**$btype=$request->btype;**

**$amt=0;**

**if($btype=="Normal")**

**{**

**$amt=$amt+1000;**

**}**

**else if($btype=="Standard")**

**{**

**$amt=$amt+2000;**

**}**

**else**

**{**

**$amt=$amt+3500;**

**}**

**$weddingbookingmodel->amt=$amt;**

**$weddingbookingmodel->veg=$request->veg;**

**$weddingbookingmodel->non\_veg=$request->non\_veg;**

**$weddingbookingmodel->veg\_imfl=$request->veg\_imfl;**

**$weddingbookingmodel->non\_veg\_imfl=$request->non\_veg\_imfl;**

**$weddingbookingmodel->ex\_liquor=$request->ex\_liquor;**

**$weddingbookingmodel->ex\_cooldrinks=$request->ex\_cooldrinks;**

**$weddingbookingmodel->audio\_visual=$request->audio\_visual;**

**$weddingbookingmodel->live\_cast=$request->live\_cast;**

**$weddingbookingmodel->message=$request->message;**

**$save=$weddingbookingmodel->save();**

**if($save)**

**{**

**return view('weddingevent',$data);**

**}**

**else**

**{**

**echo ' ! Something went wrong, try again later';**

**}**

**Wedding Event Booking:**

**$data=WeddingBookingModel::where('id','=',session('LoggedUser'))->first();**

**$weddingeventmodel=new WeddingEventModel();**

**$weddingeventmodel->fname=$data->fname;**

**$weddingeventmodel->lname=$data->lname;**

**$weddingeventmodel->email=$data->email;**

**$weddingeventmodel->username=$data->username;**

**$weddingeventmodel->start\_date=$request->start\_date;**

**$weddingeventmodel->start\_time=$request->start\_time;**

**$weddingeventmodel->end\_date=$request->end\_date;**

**$weddingeventmodel->end\_time=$request->end\_time;**

**$weddingeventmodel->partno=$request->partno;**

**$weddingeventmodel->message=$request->message;**

**$save=$weddingeventmodel->save();**

**if($save)**

**{**

**return view('reg\_payment');**

**}**

**else**

**{**

**echo ' ! Something went wrong, try again later';**

**}**

**REFERENCES**

1. [**https://www.youtube.com/**](https://www.youtube.com/)