ALUMNI NETWORK BUILDER

1. INTRODUCTION

1.1. Scope of The Project

The objective of this application is to develop a system that effectively manages all the data related to the various events that take place in an organization. The purpose is to maintain a centralized database of all events and students related information. The goal is to support various functions and processes necessary to manage the data efficiently.

1.2. Existing System

This existing system is not providing secure registration and profile management of all the users properly. This system is not providing on-line Help. This system doesn't provide tracking of users activities and their progress. This manual system gives us very less security for saving data and some data may be lost due to mismanagement. This system is not providing event management through internet. This system is not providing proper events information. The system is giving manual information through the event management executer.

1.3. Proposed System

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. This system maintains user's personal, a and contact details. This system will provide on line help and search capabilities. User friendliness is provided in the application with various controls provided by system rich user interface. Authentication is provided for this application only registered users can access. Event information files can be stored in centralized database which can be maintained by the system. This system provides the users to manage the events systematically.

2. SYSTEM ANALYSIS

2.1 FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

2.1.1 Operational Feasibility

Question that going to be asked are Will the system be used if it developed and implemented.

If there was sufficient support for the project from the management and from the users.

Have the users been involved in planning and development of the Project.

2.1.2 Technical feasibility

Does the necessary technology exist to do what is been suggested Does the proposed equipment have the technical capacity for using the new system? Are there technical guarantees of accuracy, reliability and data security? The project is developed on Pentium III with 128 MB RAM.
The environment required in the development of system is any windows platform.

The observer pattern along with factory pattern will update the results eventually.

The language used in the development is PHP, Apache Server and database as MySQL.

2.1.2 Economical Feasibility

To decide whether a project is economically feasible, to consider various factors as cost benefit analysis, long-term returns and maintenance costs.

2.2 FUNCTIONAL REQUIREMENTS

Functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases.

NUMBER OF MODULES

The system after careful analysis has been identified to be presented with the following modules:

Alumni Network Builder Module

In ANB project we use PHP and MySQL database. It has two modules.
1. Admin Module
2. User Module

Admin Module

1. **Dashboard:** In this section, admin can see all detail in brief like listed categories, Sponsors, Total Events, Total Registered Users, Total Booking, Total New Booking, Total Confirmed Booking and Total Cancelled Booking.
2. **Category:** In this section, admin manage event category (add and update).
3. **Events:** In this section, admin manages events (add and update).
4. **Manage Users:** In this section, admin can update details of registered users and also block them.
5. **Manage Booking:** In this section, admin can manage booking by cancel and confirm it.
6. **Website Setting:** In this section, admin can update, About us and another general website setting.

Admin can also update his profile, change password and recover password.

User Module

In this module there is two types of user guest user and registered user.
Guest User: In this guest user can see only general information like about us, event details, Contact details and new about events.

Registered users can do following activity

1. Books the events.

2. Update his/her own profile.

3. Change Password.

4. Users can also cancel booking which is not confirmed.

5. Registered user can also recover his/her own password.

2.3 NON-FUNCTIONAL REQUIREMENTS

Performance Requirements:

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:
The system should be able to interface with the existing system The system should
be accurate.
The system should be better than the existing system

Reliability:

In this system reliability means the mail which is send by the source must reach the target user with any modification and accurate.

Security:

The web server and database server should be protected from hacking, virus etc

Portability:

The application will be developed using standard open source software like PHP, Apcahe web server, MySQL database, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability:

This software will be available always.

Maintainability:

In this system the presentation layer is clearly separated from the service layer. So any modification in future will be done with less efforts. The database will be running at the server. Users access these forms by using the user-ids and the passwords.

2.4 HARDWARE REQUIREMENTS:

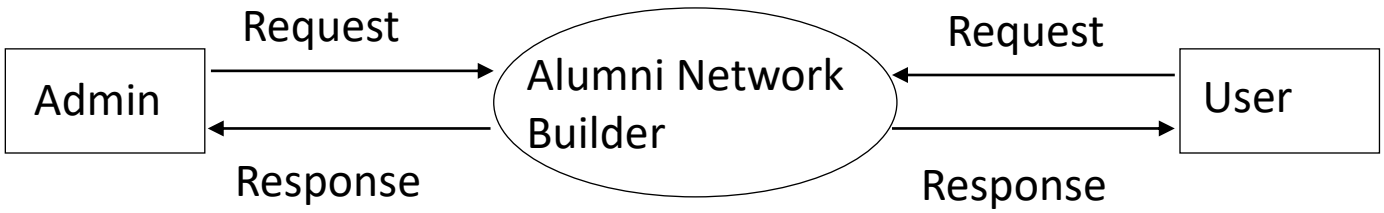
Processor	:	Intel P-IV based system
Processor Speed	:	2.0. GHz
RAM	:	1GB
Hard Disk	:	40GB to 80GB

2.5 SOFTWARE REQUIREMENTS:

Database	:	MySQL
Server	:	Apache
Frontend	:	HTML
Scripting language	:	Java Script
IDE	:	Sublime
Technology	:	PHP

3. SYSTEM DESIGN

3.1 DFD's (Data Flow Diagrams)



Data Flow Diagram for Level-1

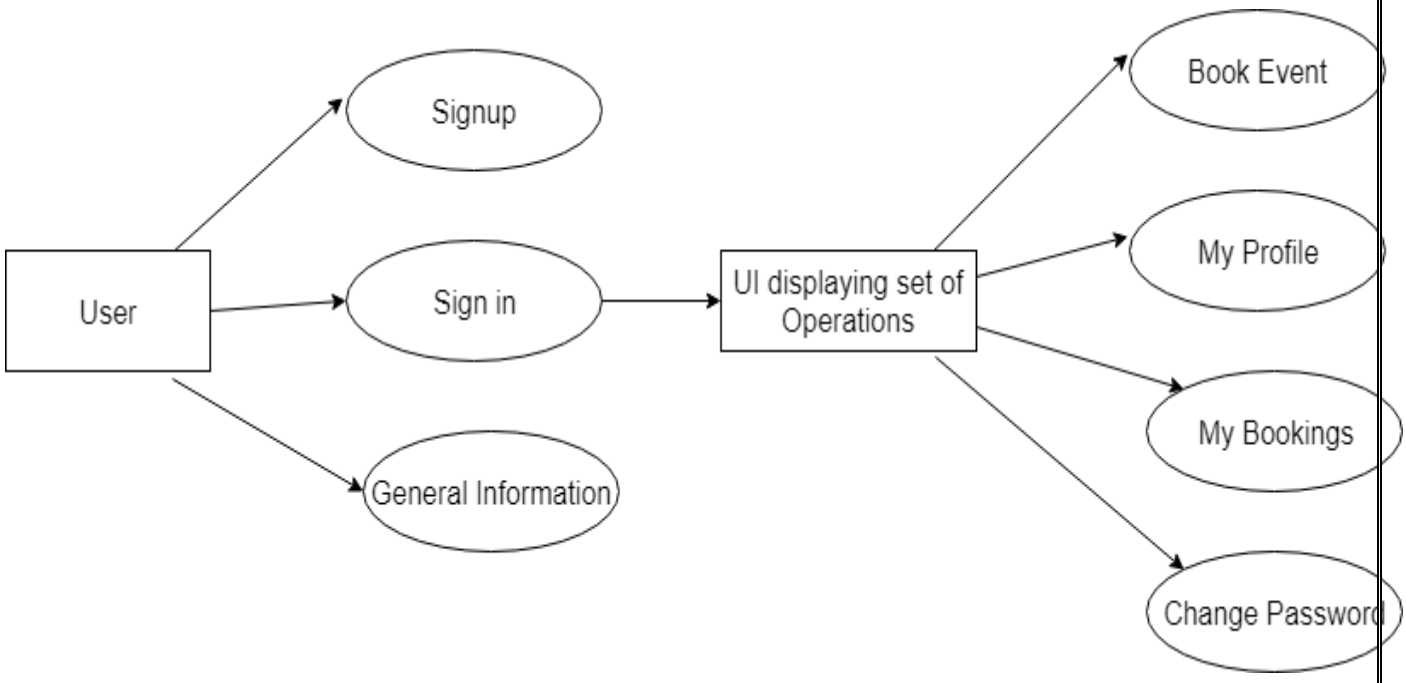
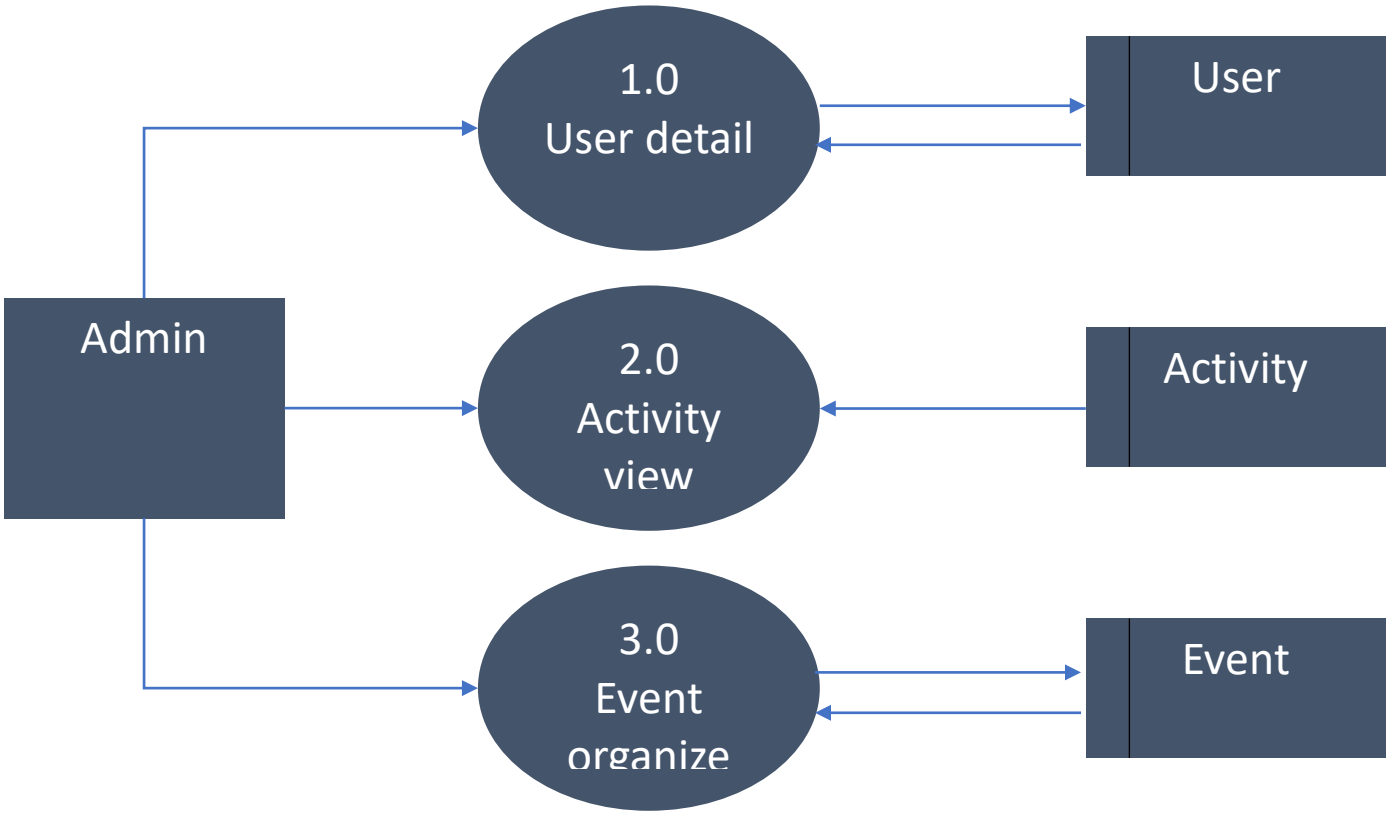


Fig: 3.1.2 Data Flow Diagram for Level-1

Data Flow Diagram for Admin Level-1



3.1.2 Use case diagrams:

Use case diagram consists of actors, use cases and their relationships. These diagrams are especially important in organizing and modelling the behaviours of a system.

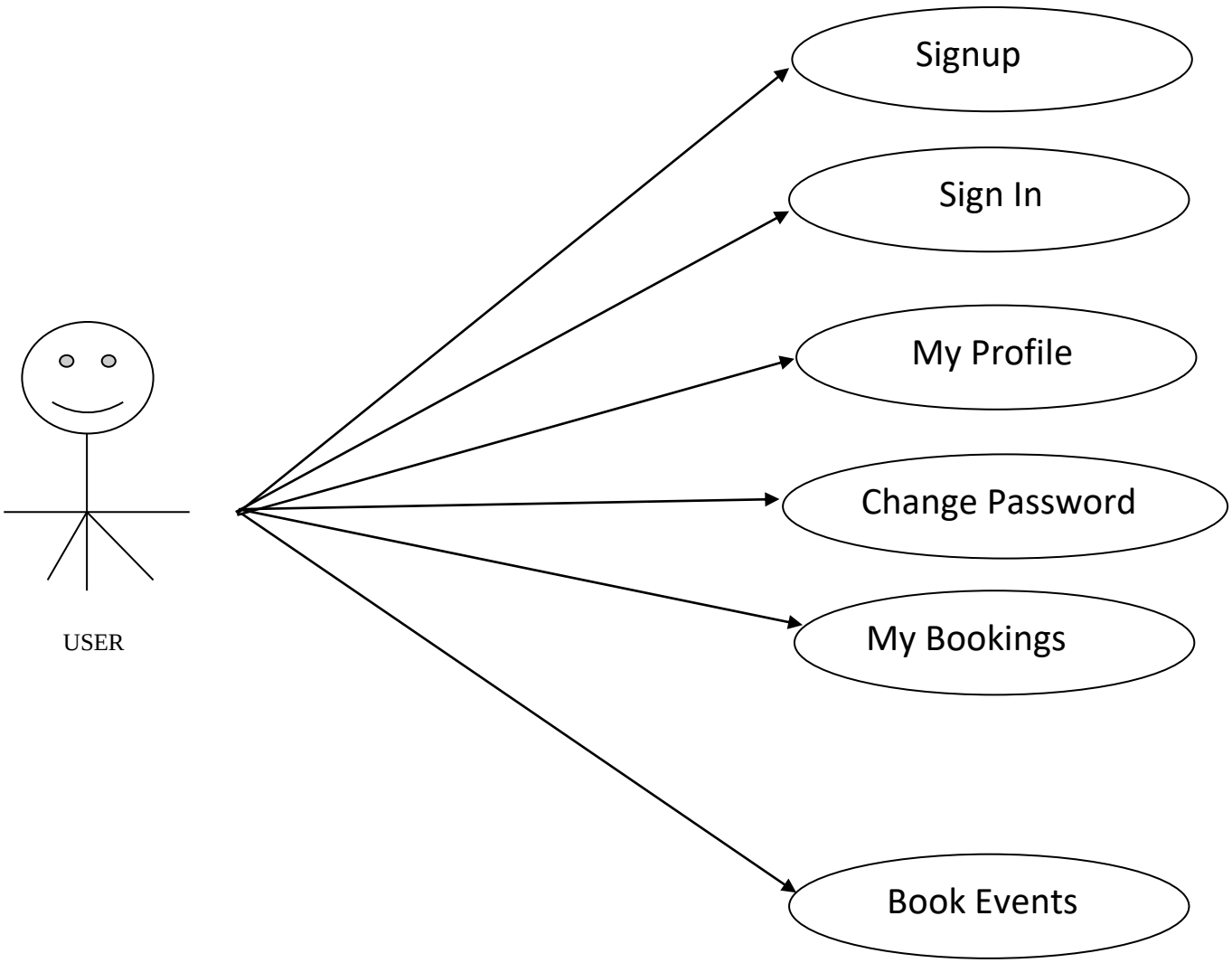


Fig: 3.1.2.1 Use case diagram for user

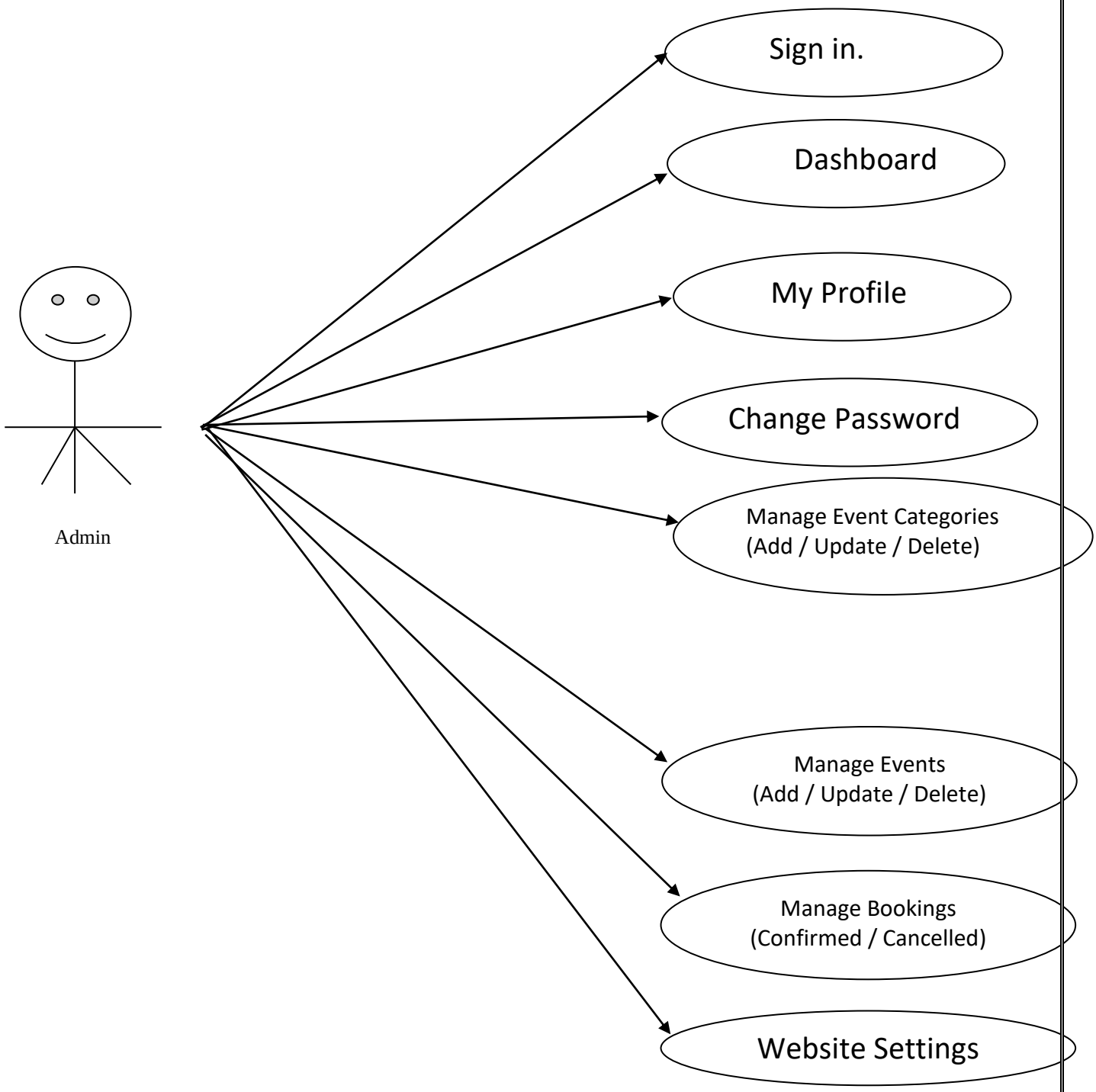


Fig: 3.1.2.2 Use case diagram for Administration

3.3 Tables

The data in the system must be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive, and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies, and optimizing for updates. The MySQL database has been chosen for developing the relevant databases.

ALUMNI NETWORK BUILDER (ANB) contains 10 MySQL tables:

tbladmin : This table store the admin login details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	FullName	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
3	AdminEmail	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
4	UserName	varchar(100)	latin1_swedish_ci		No	*None*		
5	Password	varchar(100)	latin1_swedish_ci		No	*None*		
6	updationDate	timestamp			No	0000-00-00 00:00:00		ON UPDATE CURRENT_TIMESTAMP()

tblusers: This table store the user personal and login details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Userid	int(11)			No	*None*		AUTO_INCREMENT
2	FullName	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
3	UserName	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
4	Emailid	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	PhoneNumber	bigint(12)			Yes	*NULL*		
6	UserGender	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
7	UserPassword	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
8	RegDate	timestamp			Yes	current_timestamp()		
9	LastUpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()
10	IsActive	int(1)			Yes	*NULL*		

tblcategory: This table store the event category details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	CategoryName	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	CategoryDescription	mediumtext	latin1_swedish_ci		Yes	*NULL*		
4	CreationDate	timestamp			Yes	current_timestamp()		
5	UpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()
6	IsActive	char(1)	latin1_swedish_ci		Yes	*NULL*		

tblevents: This table store the event full details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	CategoryId	char(10)	latin1_swedish_ci		Yes	*NULL*		
3	SponserId	char(10)	latin1_swedish_ci		Yes	*NULL*		
4	EventName	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	EventDescription	mediumtext	latin1_swedish_ci		Yes	*NULL*		
6	EventStartDate	date			Yes	*NULL*		
7	EventEndDate	date			Yes	*NULL*		
8	EventLocation	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
9	EventImage	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
10	PostingDate	timestamp			Yes	current_timestamp()		
11	LastUpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()
12	IsActive	int(1)			Yes	*NULL*		

tblbookings: This table store the event booking details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	BookingId	bigint(12)			Yes	*NULL*		
3	UserId	int(11)			Yes	*NULL*		
4	EventId	int(11)			Yes	*NULL*		
5	NumberOfMembers	int(11)			Yes	*NULL*		
6	UserRemark	mediumtext	latin1_swedish_ci		Yes	*NULL*		
7	AdminRemark	mediumtext	latin1_swedish_ci		Yes	*NULL*		
8	UserCancelRemark	mediumtext	latin1_swedish_ci		Yes	*NULL*		
9	BookingDate	timestamp			Yes	current_timestamp()		
10	BookingStatus	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
11	LastUpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

tblgenralsettings : This table store the website general settings.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	SiteName	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	PhoneNumber	bigint(12)			Yes	*NULL*		
4	EmailId	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	address	mediumtext	latin1_swedish_ci		Yes	*NULL*		
6	footercontent	mediumtext	latin1_swedish_ci		Yes	*NULL*		
7	LastUpdationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

4.IMPLEMENTATION

INTRODUCTION:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

4.1 TECHNOLOGIES USED

Programming Language

PHP

- ✓ PHP stands for PHP: Hypertext Preprocessor
- ✓ PHP is a server-side scripting language, like ASP.
- ✓ PHP scripts are executed on the server.
- ✓ PHP supports many databases (MYSQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.)
- ✓ PHP is an open-source software.
- ✓ PHP is free to download and use.

MYSQL

- ✓ MYSQL is a database server.
- ✓ MYSQL is ideal for both small and large applications.
- ✓ MYSQL supports standard SQL.
- ✓ MYSQL compiles on a few platforms.
- ✓ MYSQL is free to download and use.

CSS

- ✓ Cascading Style Sheets (CSS)
- ✓ Simple mechanism
- ✓ Easy for adding style (e.g., fonts, colours, spacing) to Web documents.

5. TESTING

Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionalities of components, sub-assemblies, and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

5.1 Types of Testing

5.1.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel.

5.1.2. Integration Testing

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the system works. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested, and individual subsystems are exercised through their input interface.

Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing.

5.1.5 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

This type of testing ensures that.

All independent paths have been exercised at least once.

All logical decisions have been exercised on their true and false sides.

All loops are executed at their boundaries and within their operational bounds All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form. we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

Basic Path Testing

Established technique of flow graph with Cyclometer complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Conditional Testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on condition is traced to uncover any possible errors.

Data Flow Testing

This type of testing selects the path of the program according to the location of

definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

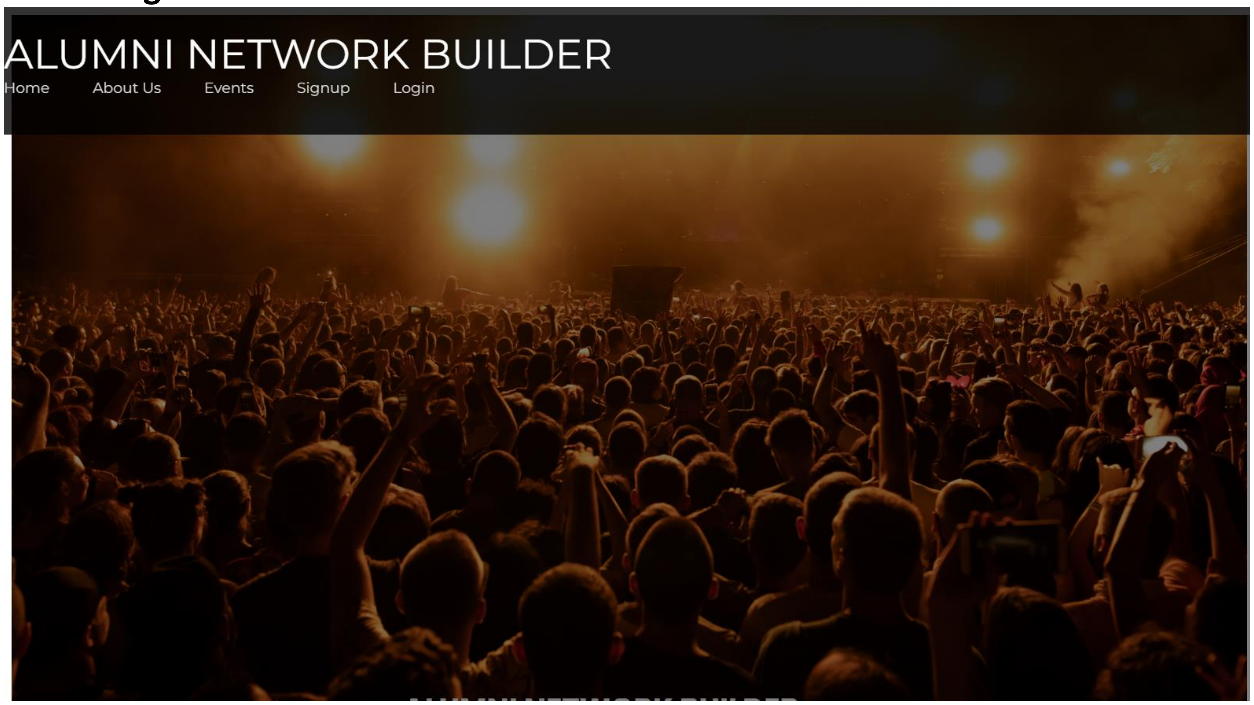
Loop Testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops: All the loops were tested at their limits, just above them and just below them. All the loops were skipped at least once. For nested loops test the inner most loop first and then work outwards. For concatenated loops the values of dependent loops were set with the help of connected loop. Unstructured loops were resolved into nested loops or concatenated loops and tested as above. Each unit has been separately tested by the development team itself and all the input have been validated.

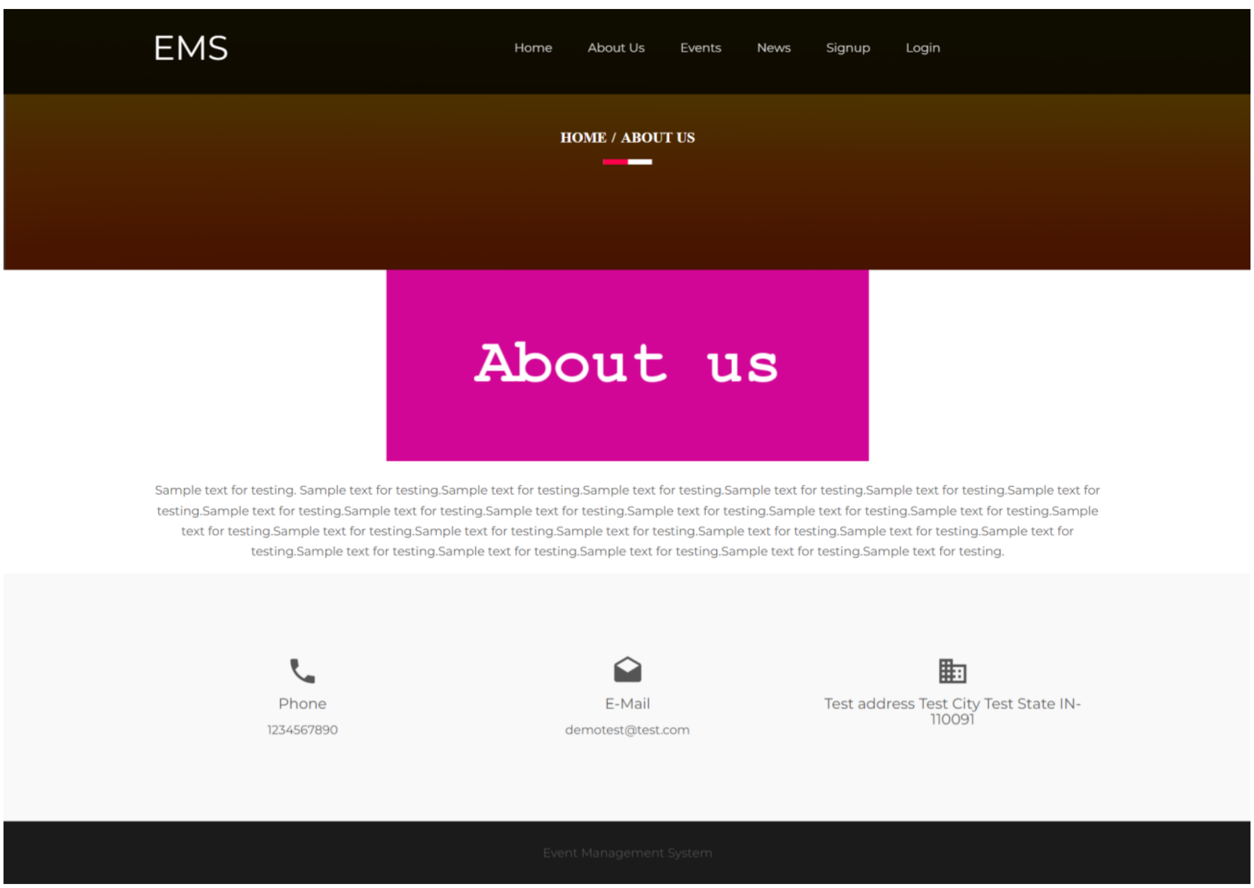
6.Output Screen of Project

User Module Screens

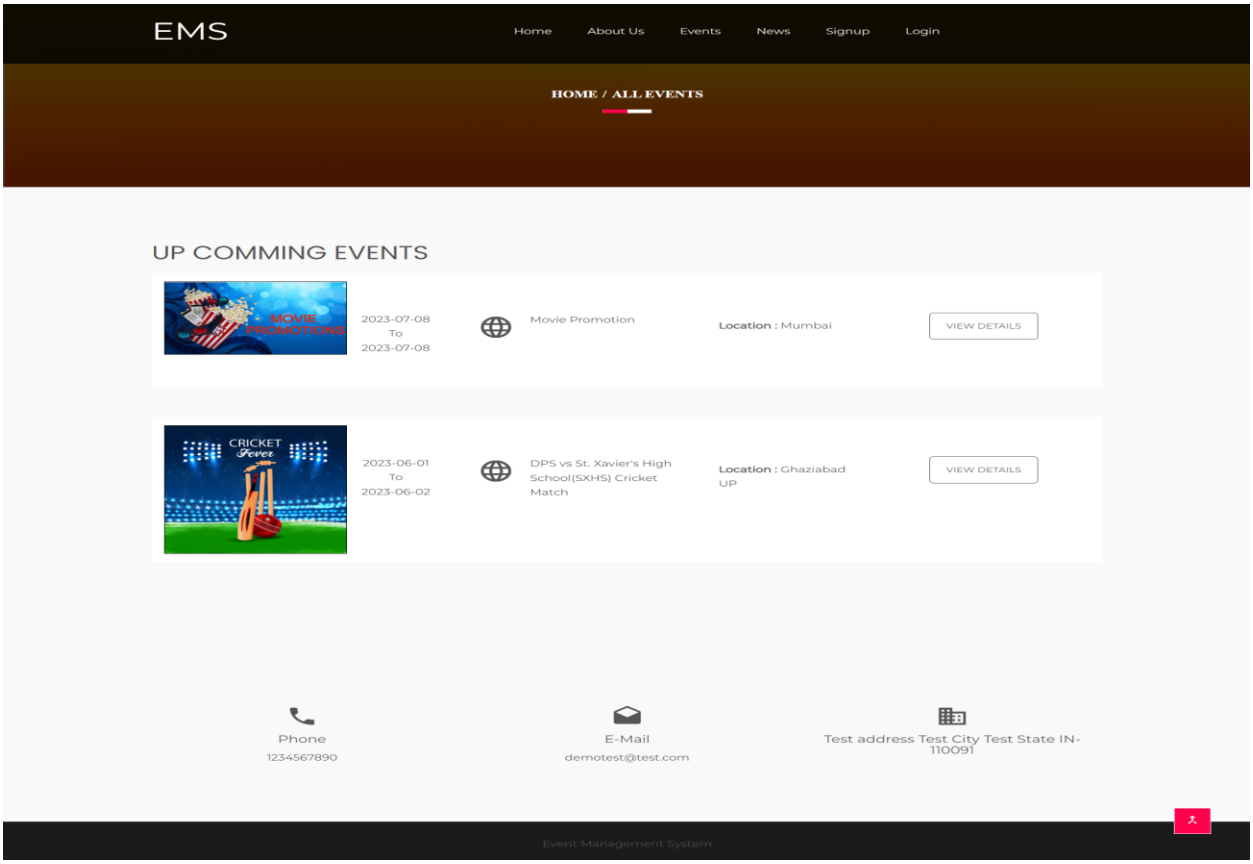
Home Page



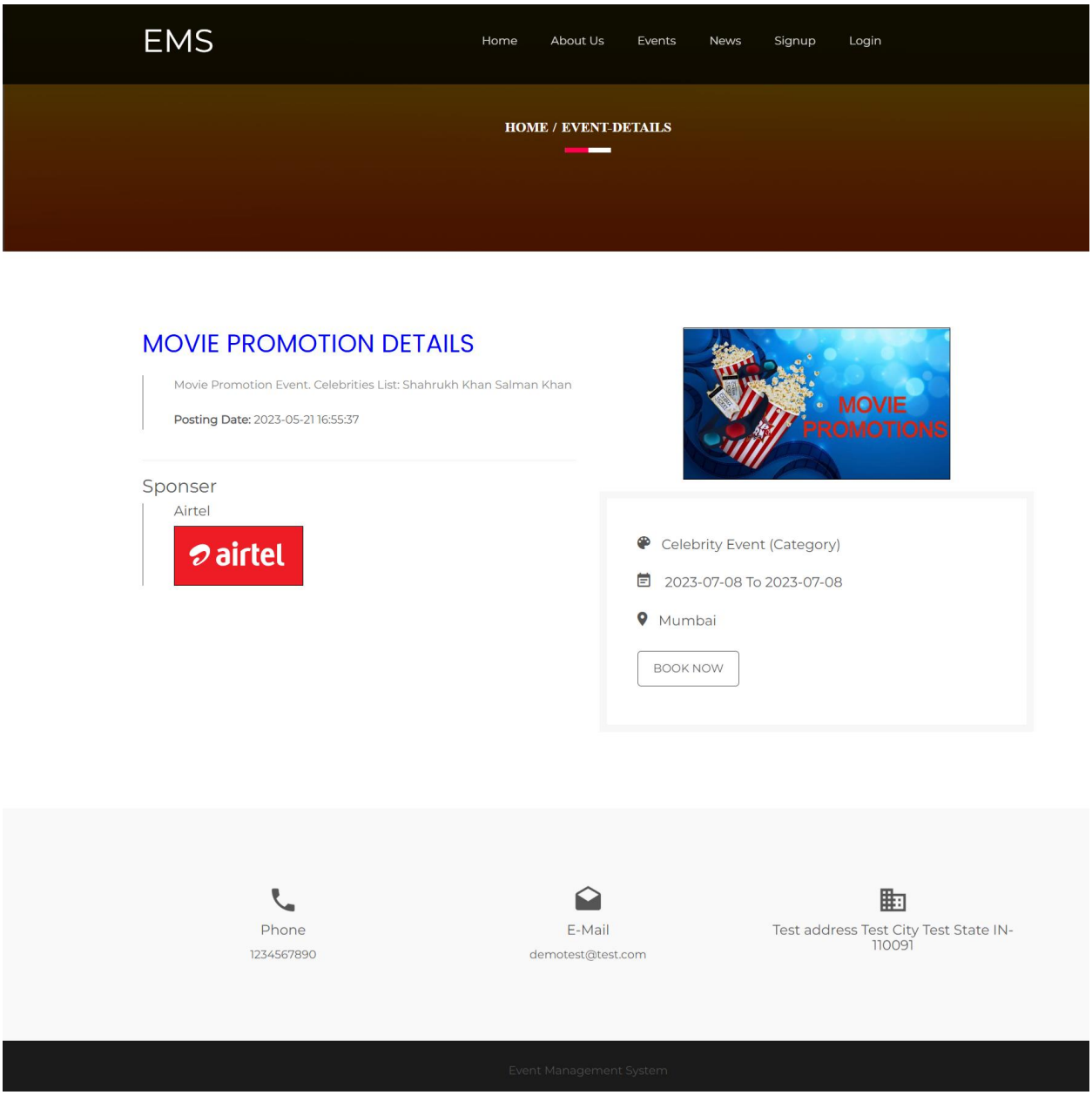
About Us



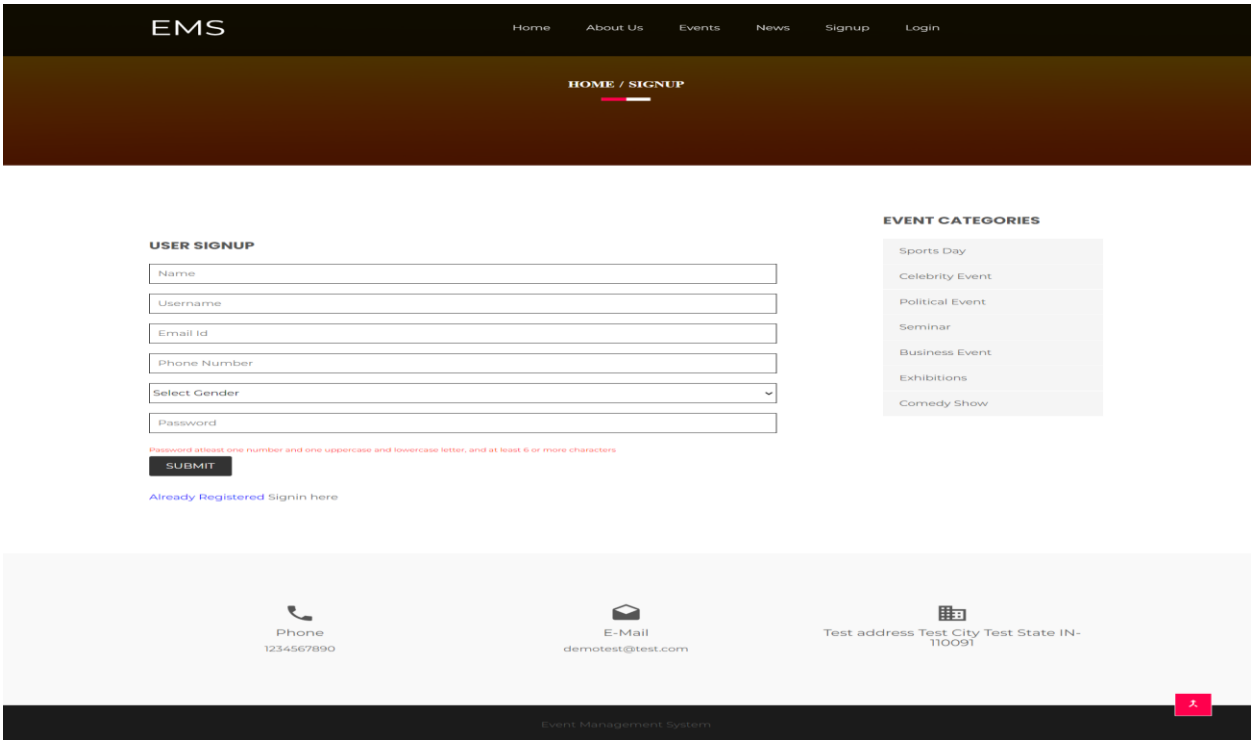
Events Page


EMS
Home
About Us
Events
News
Signup
Login
HOME / ALL EVENTS
UP COMMING EVENTS
MOVIE PROMOTIONS
2023-07-08
To
2023-07-08
Movie Promotion
Location : Mumbai
VIEW DETAILS
CRICKET Fever
2023-06-01
To
2023-06-02
DPS vs St. Xavier's High School(SXHS) Cricket Match
Location : Ghaziabad UP
VIEW DETAILS
Phone
1234567890
E-Mail
demotest@test.com
Test address Test City Test State IN-110091
Event Management System

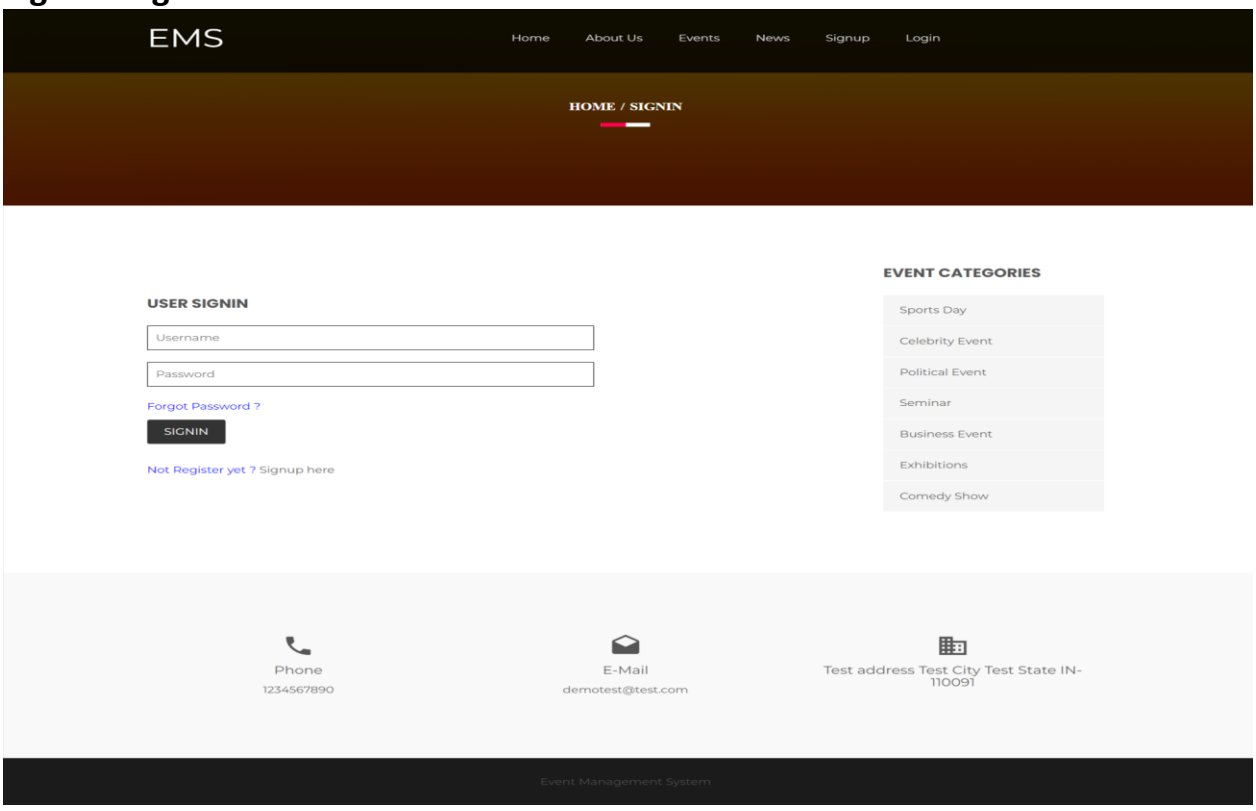
Events Details



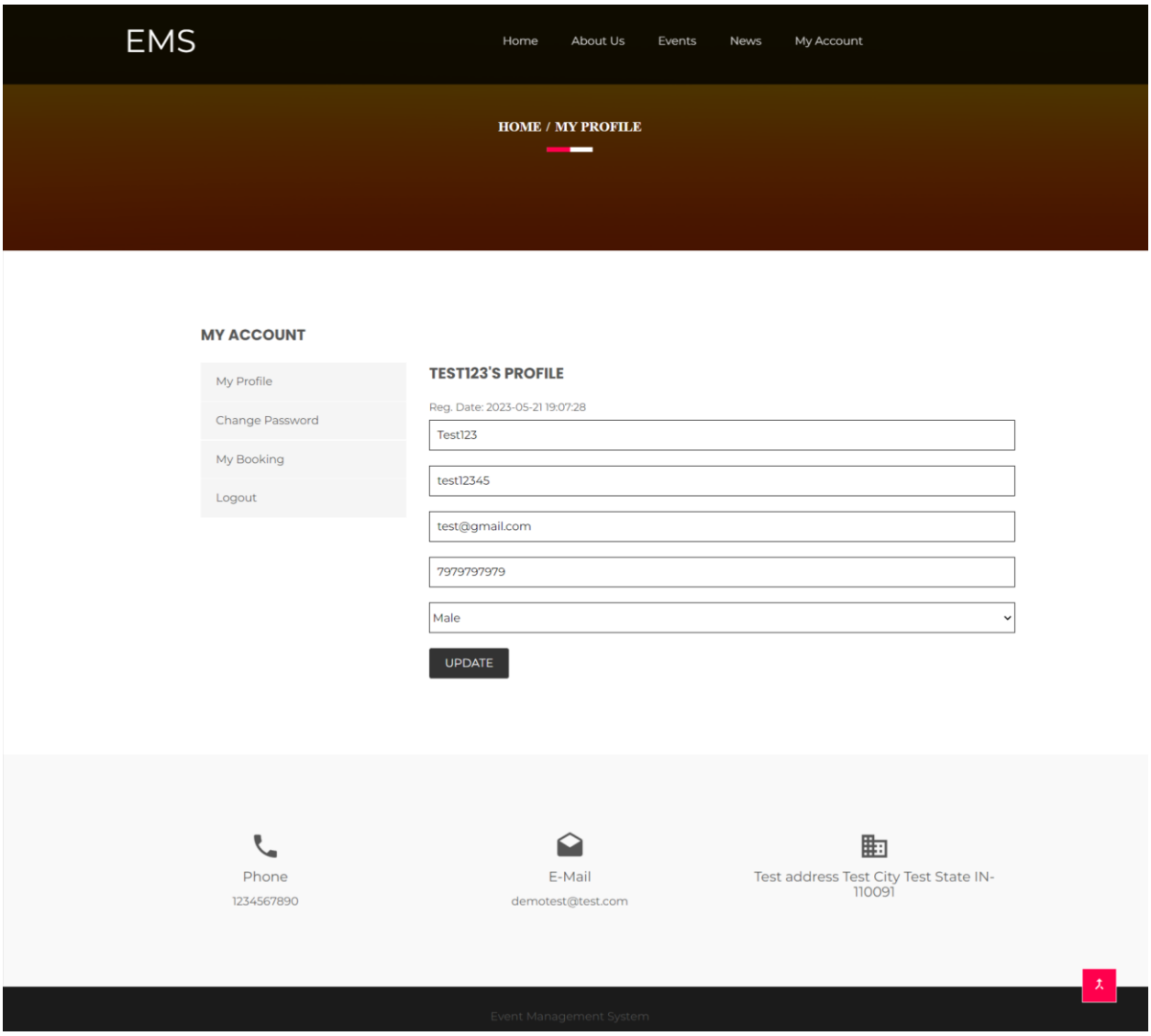
Signup Page



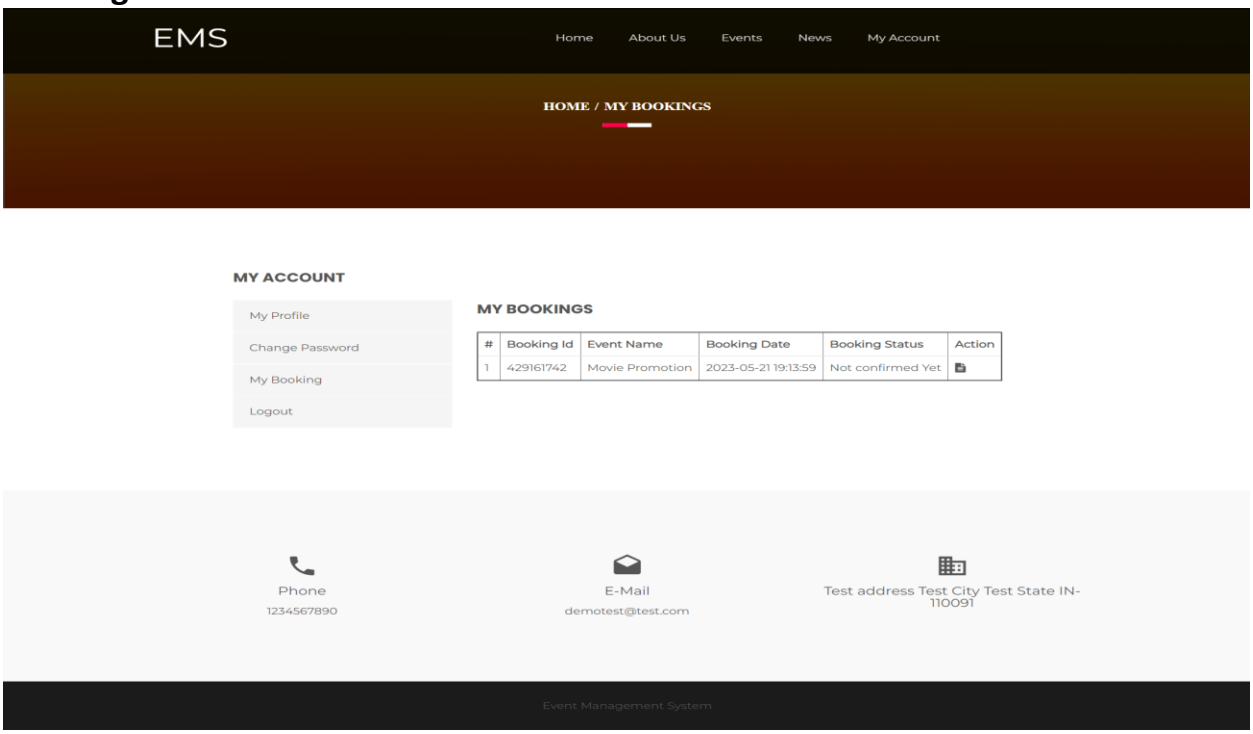
Sign in Page



User Profile

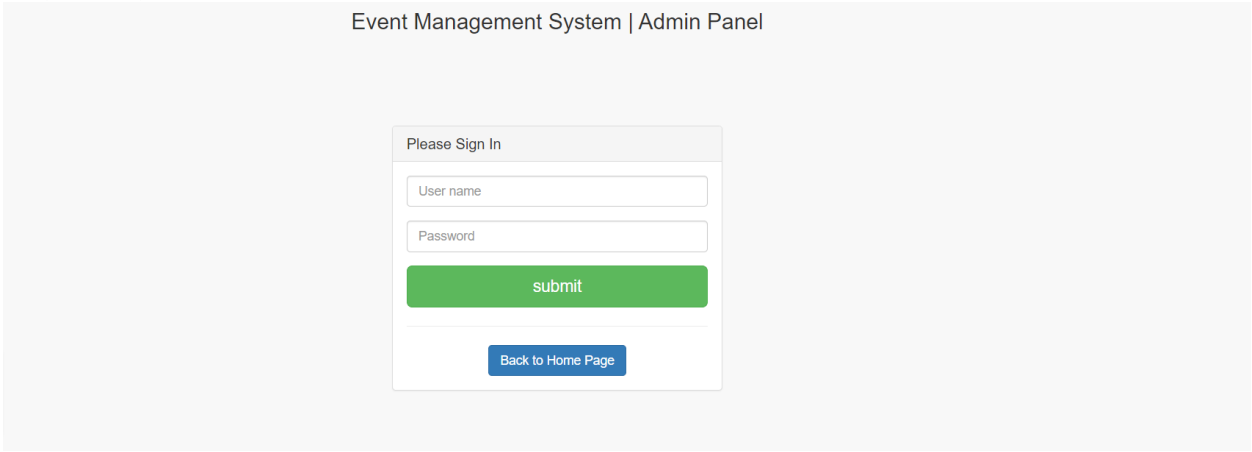


Booking Details

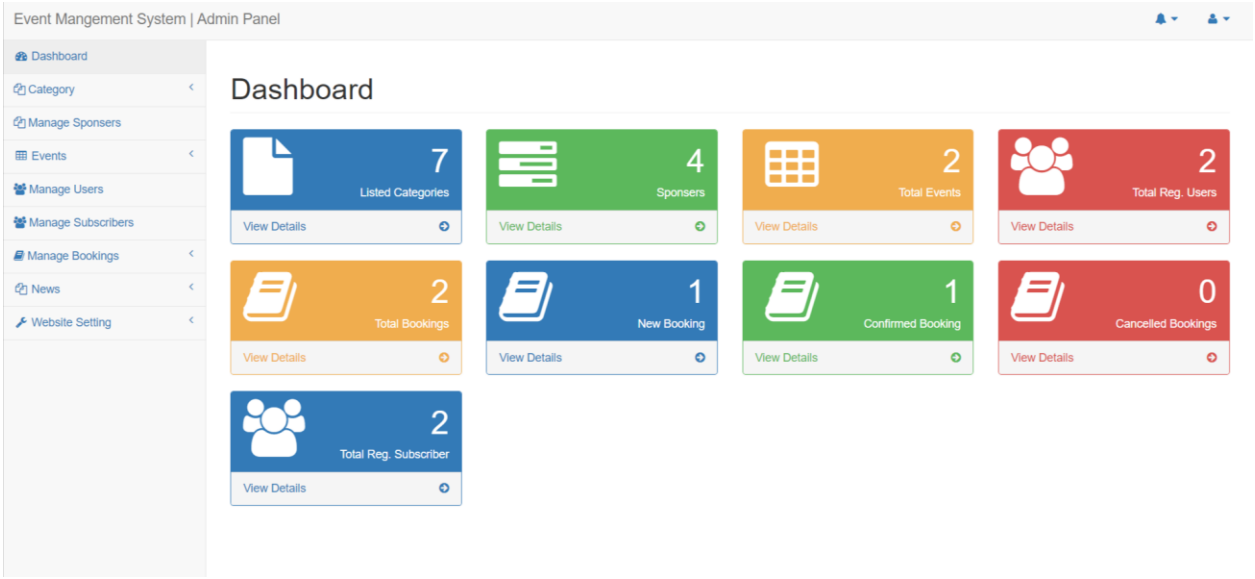


Admin Panel

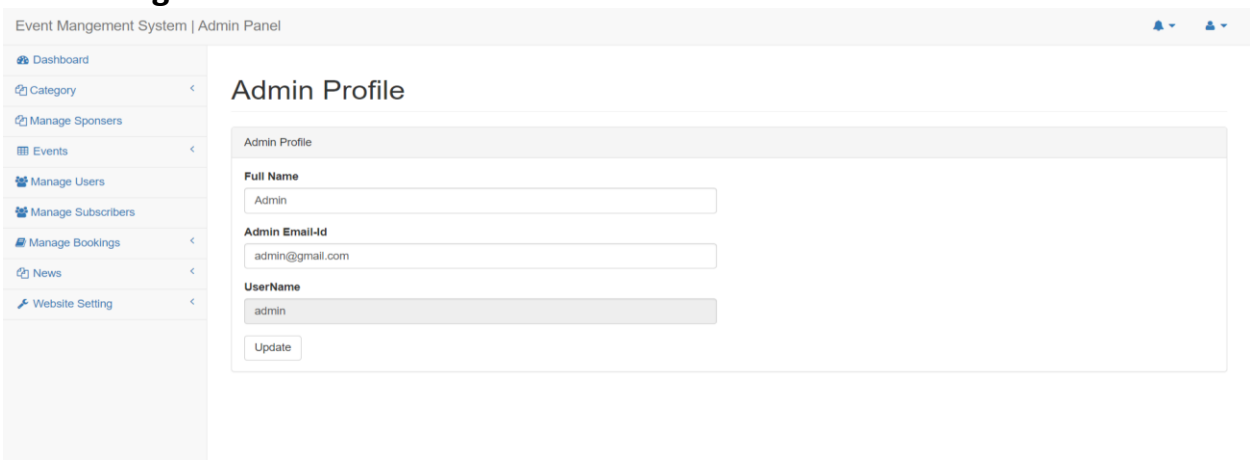
Login Page



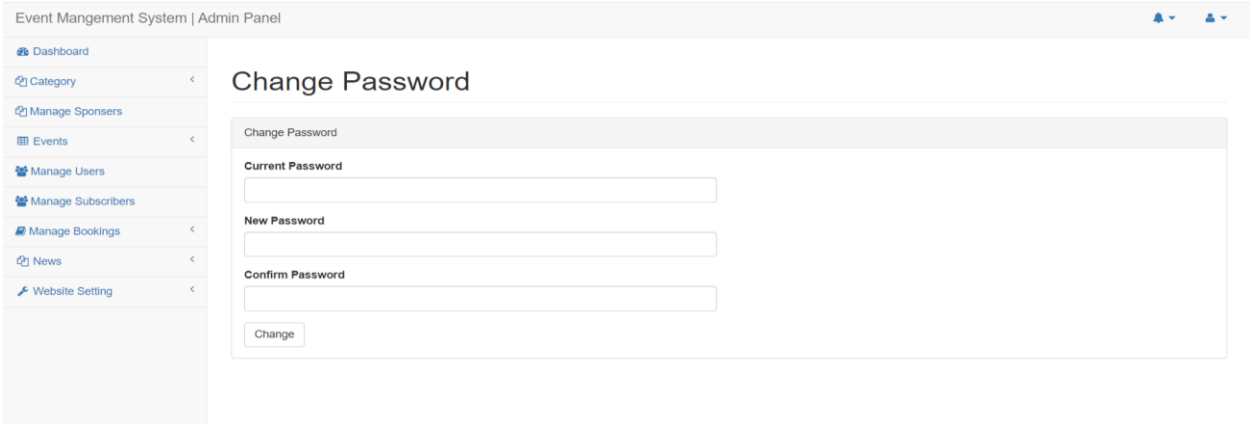
Dashboard



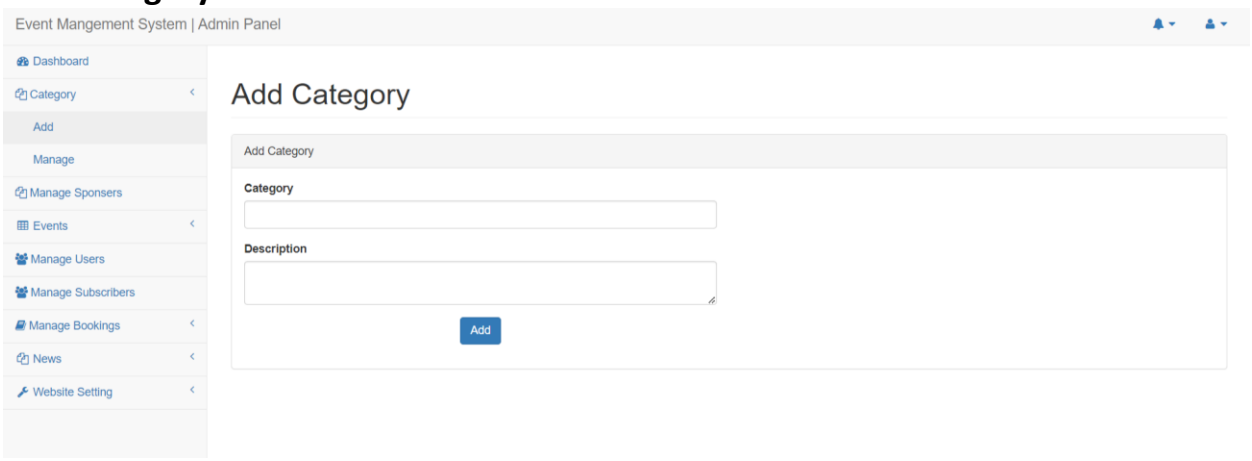
Profile Page



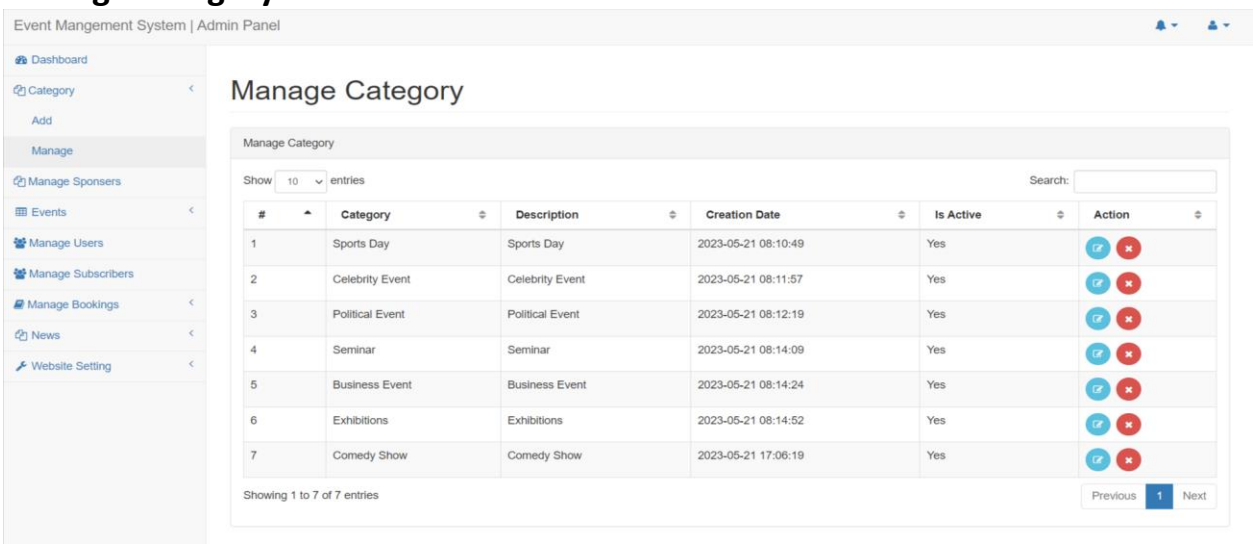
Change Password



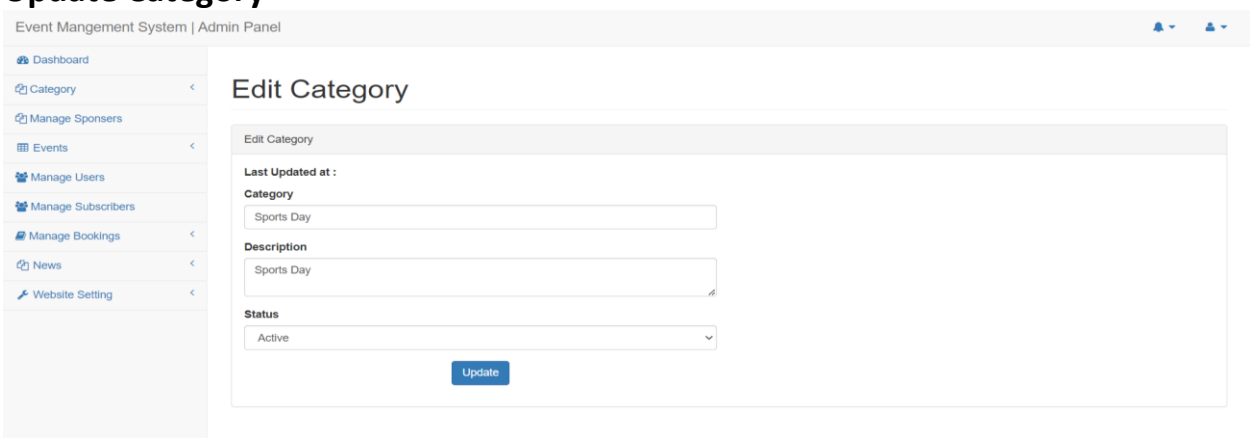
Add Category



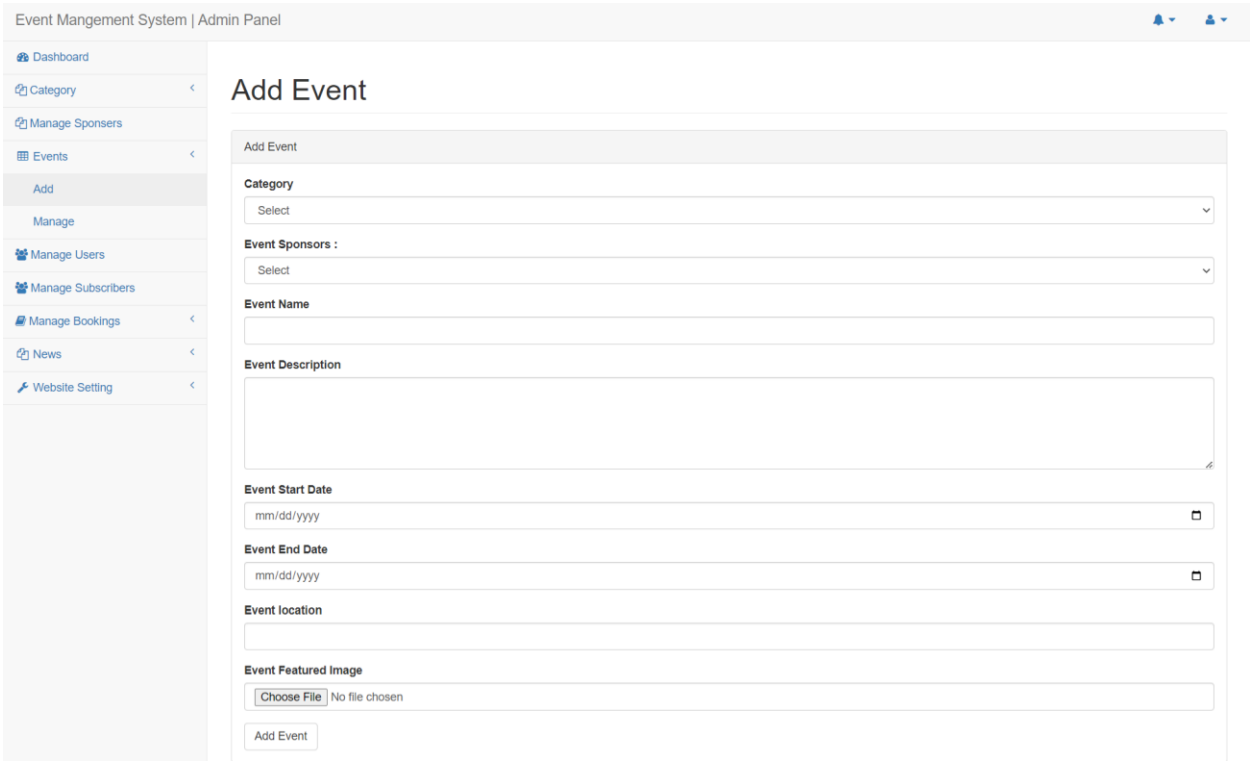
Manage Category



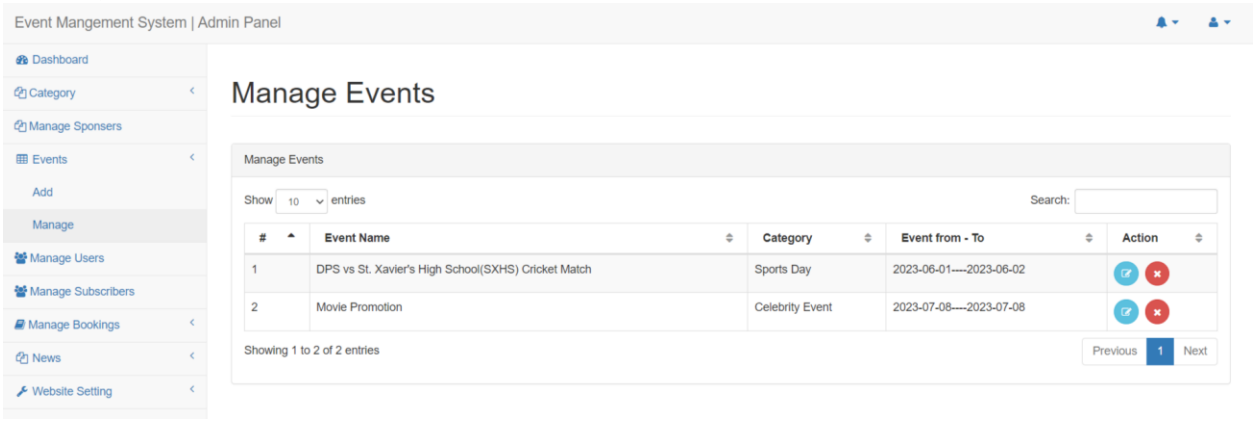
Update Category



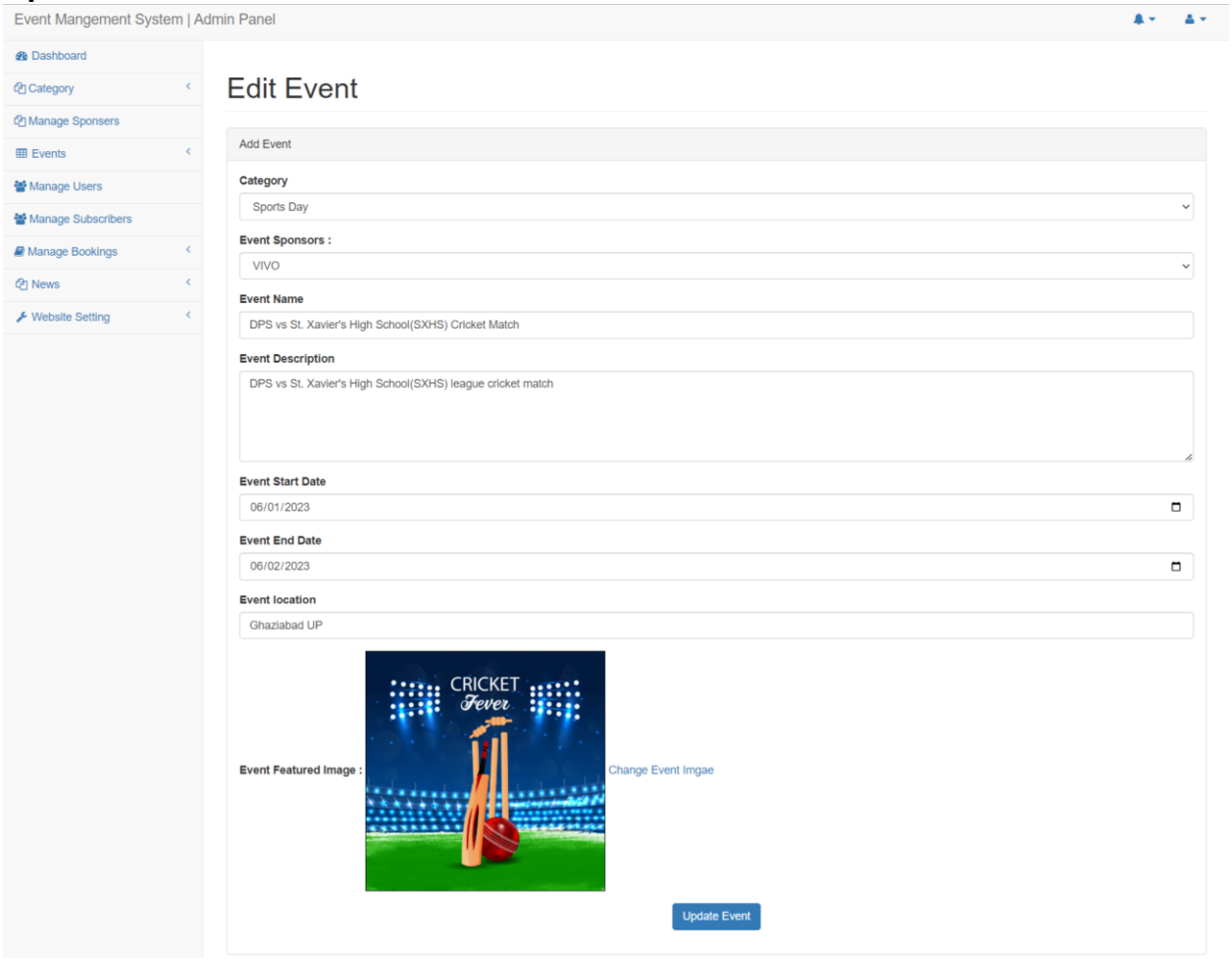
Add Events



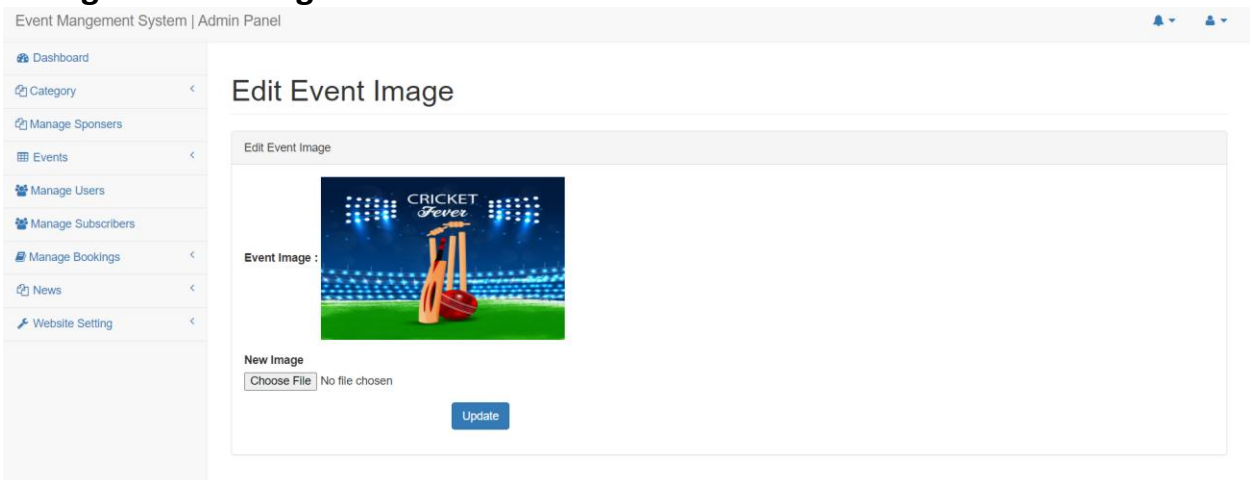
Manage Events



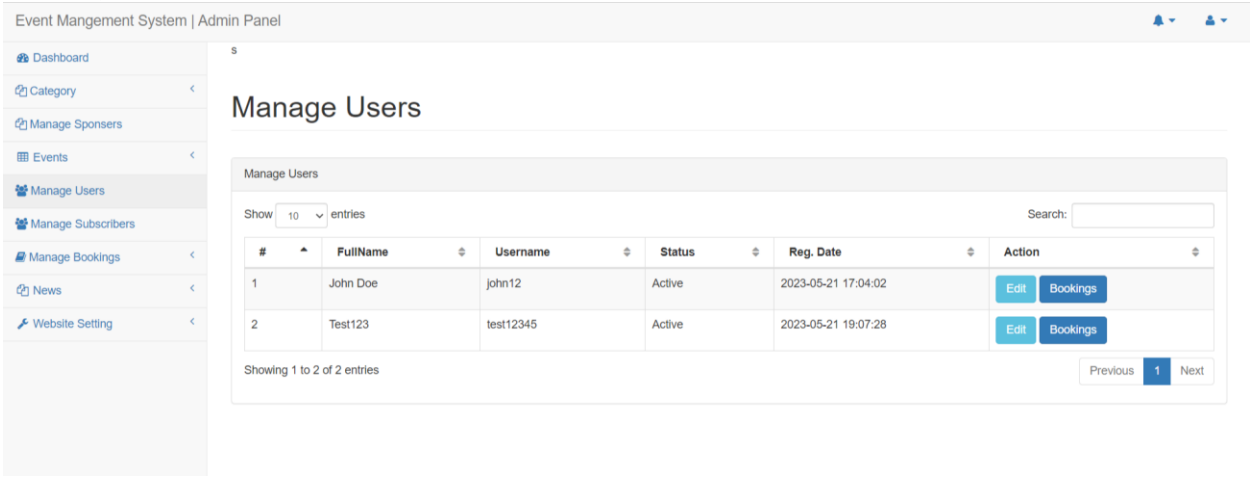
Update Events



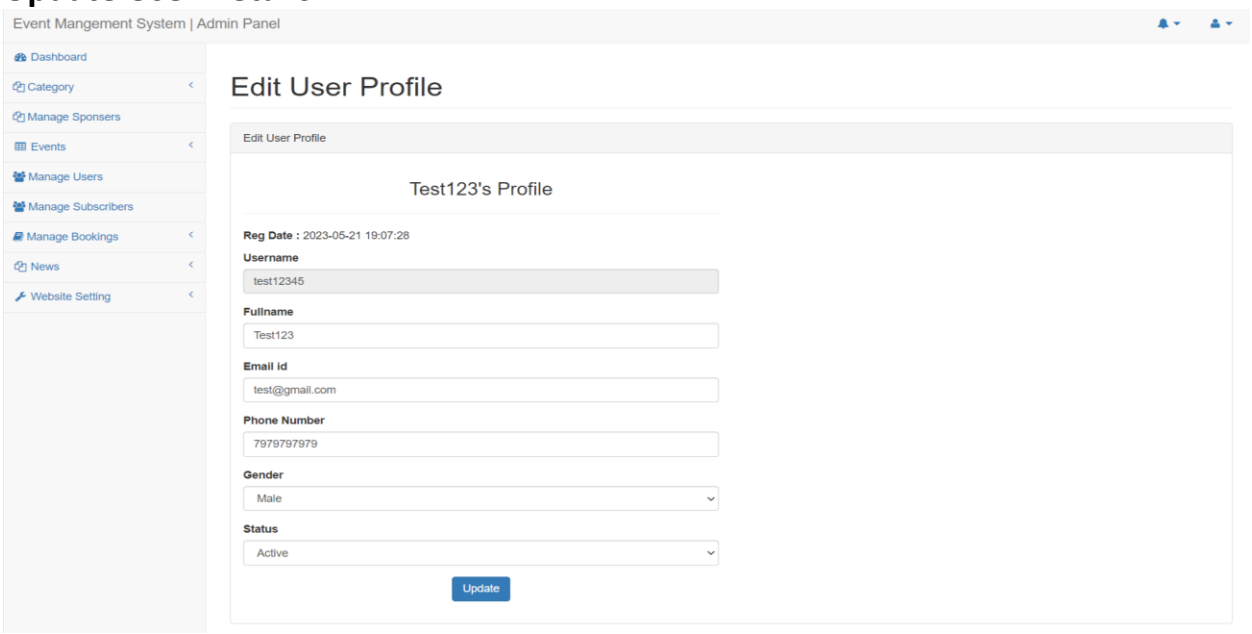
Change Events Image



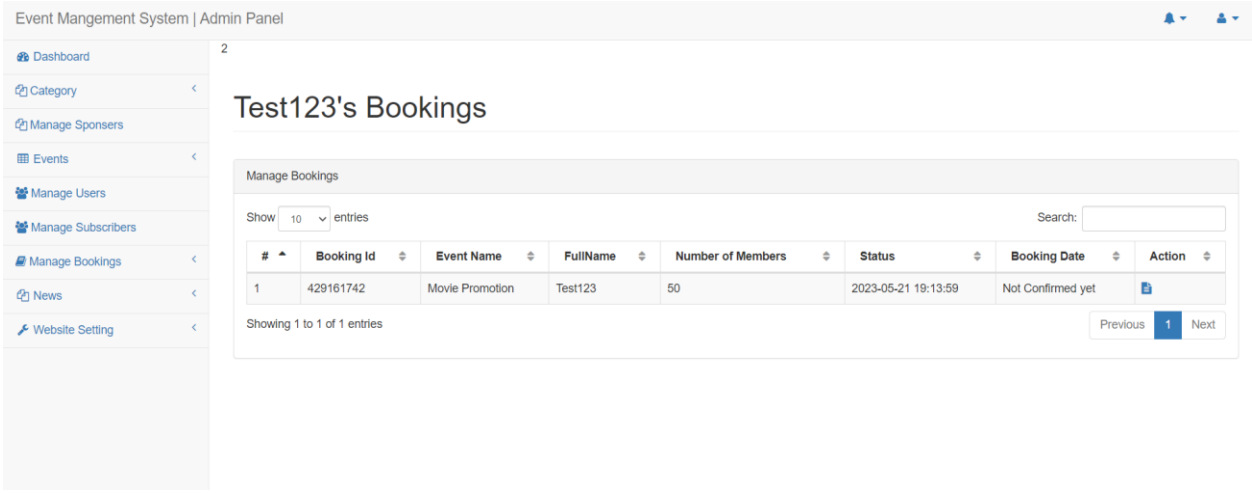
Manage Users



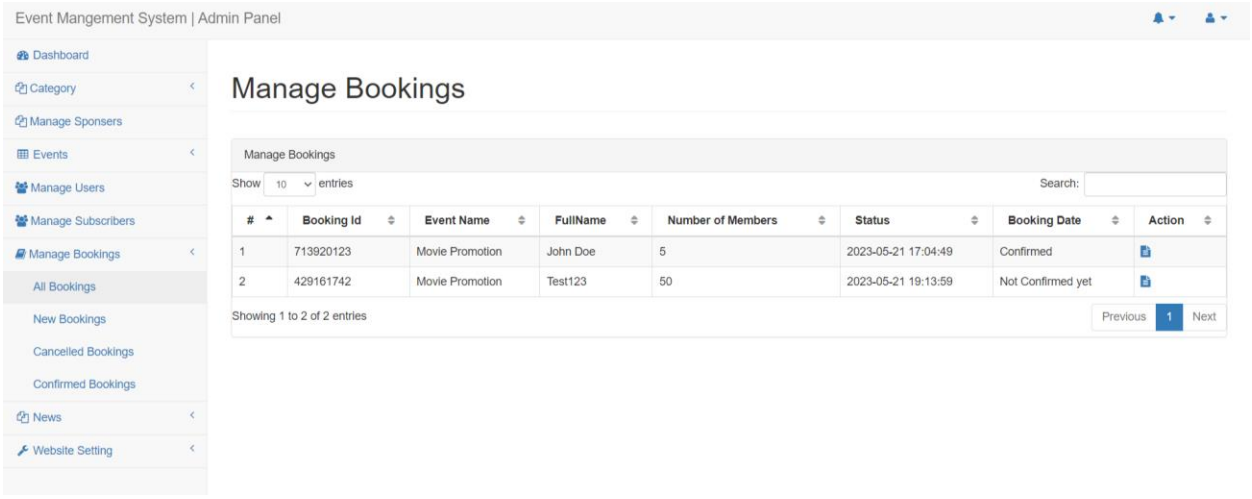
Update User Details



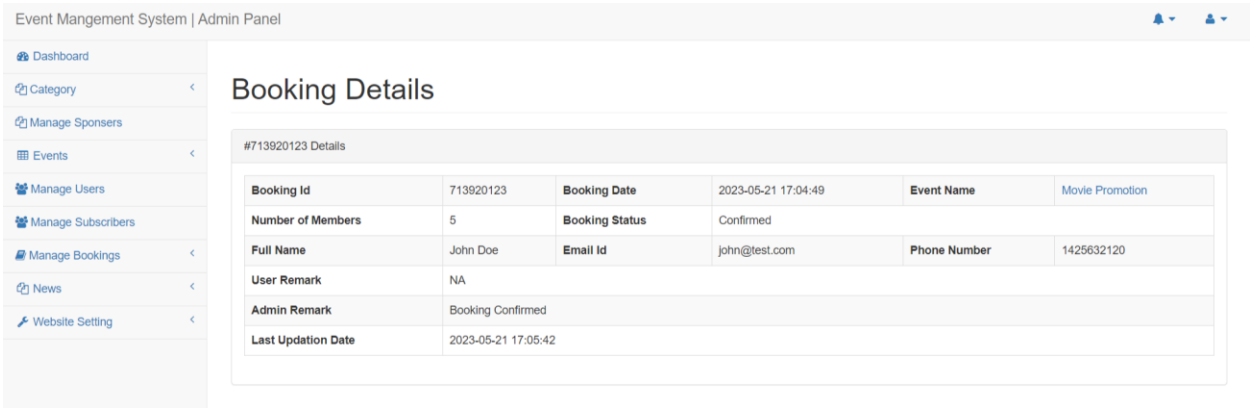
User Bookings



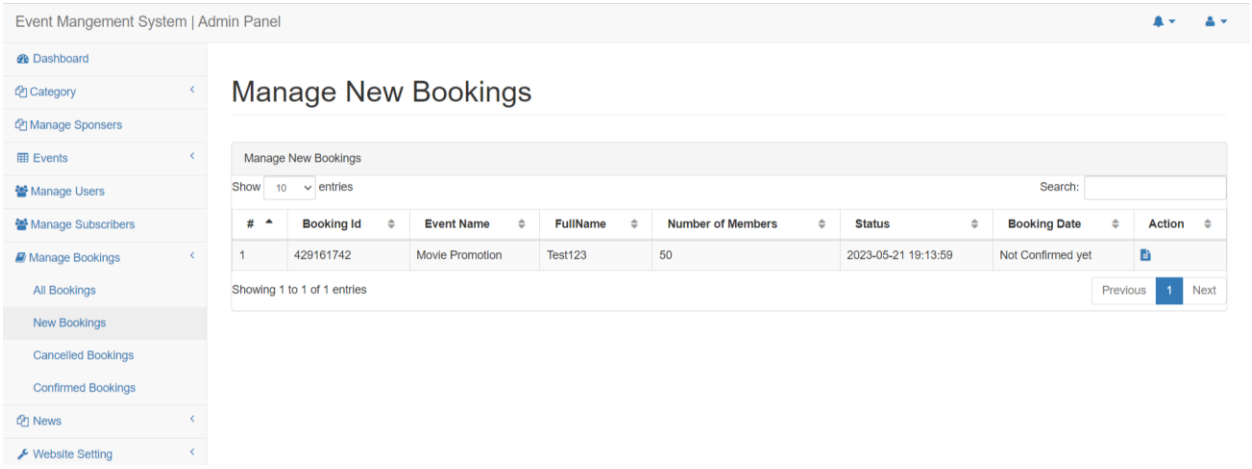
All Booking Events



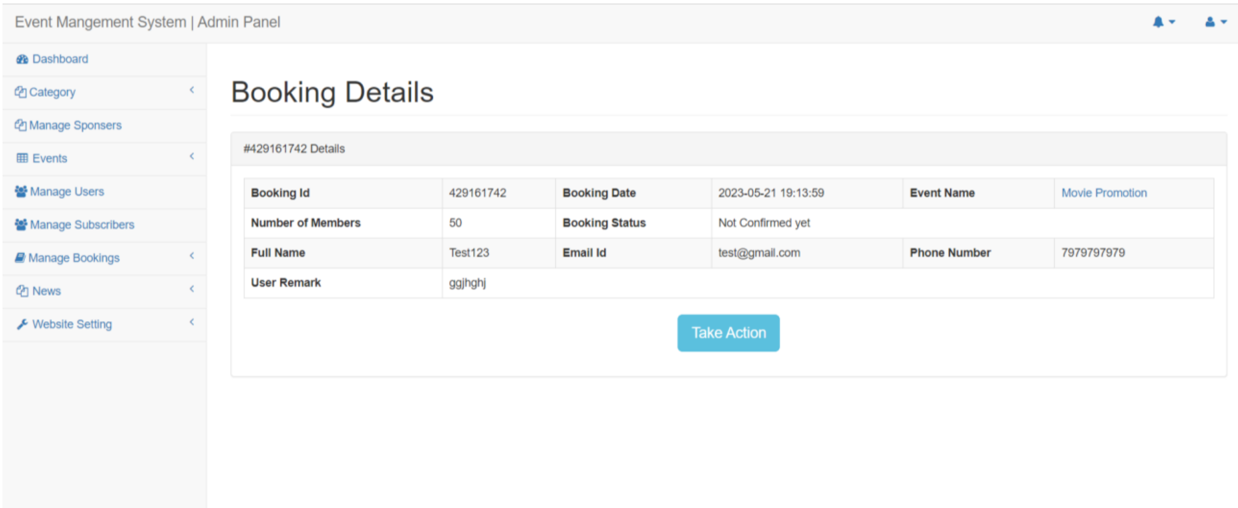
Booking Details



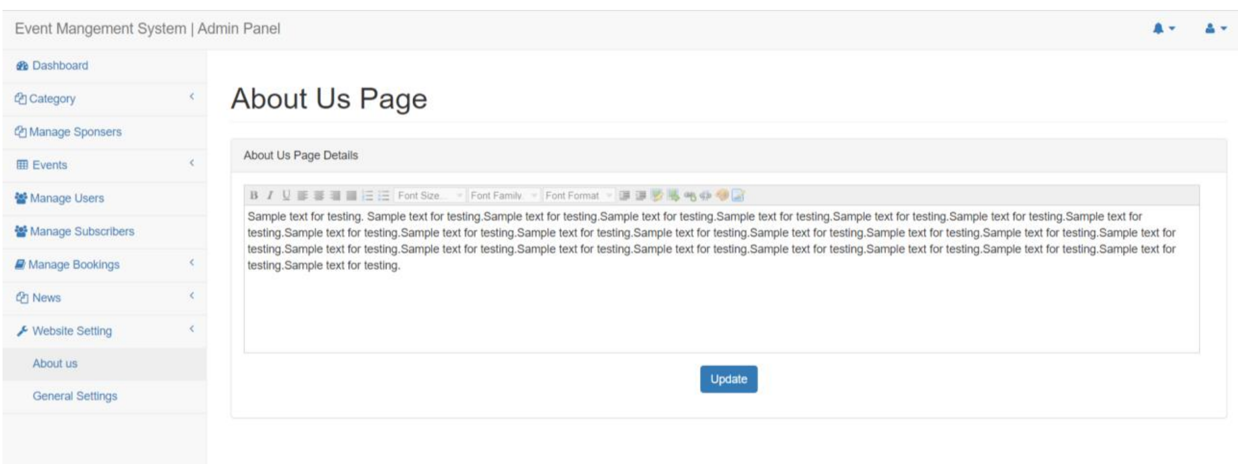
Manage New Events Booking



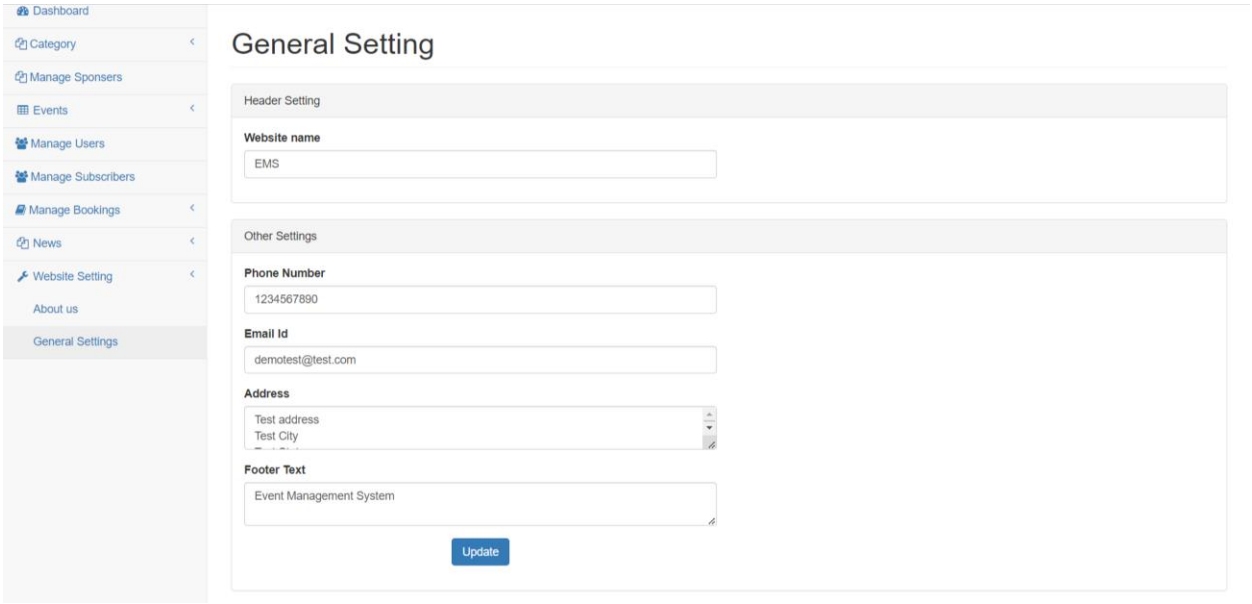
New Booking Details



About Us Page



General Setting



7. CONCLUSION

The **"ALUMNI NETWORK BUILDER"** was successfully designed and is tested for accuracy and quality. During this project we have accomplished all the objectives and this project Meets the needs of the organization. The developed will be used in searching, retrieving. And generating information for the concerned requests.

GOALS ACHIVIED

- ✓ Reduced entry work
- ✓ Easy retrieval of information
- ✓ Reduced errors due to human intervention
- ✓ User friendly screens to enter the data.
- ✓ Portable and flexible for further enhancement
- ✓ Web enabled.
- ✓ Fast finding of information request

8. REFERENCES

- ✓ www.w3schools.com
- ✓ ***php**.net*
- ✓ *en.wikipedia.org/wiki/**PHP***
- ✓ www.hotscripts.com/category/**php**/
- ✓ www.**apache**.org
- ✓ www.**mysql**.com/click.php?e=35050

9.APPENDIX

9.1 Sample code

Code for all events

```
<?php
session_start();
error_reporting(0);
include('includes/config.php');

?>

<!doctype html>
<html class="no-js" lang="en">
   <head>

      <title>ALUMNI NETWORK BUILDER | user signin </title>
      <!-- bootstrap v3.3.6 css -->
      <link rel="stylesheet" href="css/bootstrap.min.css">
      <!-- animate css -->
      <link rel="stylesheet" href="css/animate.css">
      <!-- meanmenu css -->
      <link rel="stylesheet" href="css/meanmenu.min.css">
      <!-- owl.carousel css -->
      <link rel="stylesheet" href="css/owl.carousel.css">
      <!-- icofont css -->
      <link rel="stylesheet" href="css/icofont.css">
      <!-- Nivo css -->
      <link rel="stylesheet" href="css/nivo-slider.css">
      <!-- animaton text css -->
      <link rel="stylesheet" href="css/animate-text.css">
      <!-- Metrial iconic fonts css -->
      <link rel="stylesheet" href="css/material-design-iconic-font.min.css">
      <!-- style css -->
      <link rel="stylesheet" href="style.css">
      <!-- responsive css -->
      <link rel="stylesheet" href="css/responsive.css">
      <!-- color css -->
      <link href="css/color/skin-default.css" rel="stylesheet">

      <!-- modernizr css -->
      <script src="js/vendor/modernizr-2.8.3.min.js"></script>
   </head>
   <body>
      <!--body-wraper-are-start-->
```

```
<div class="wrapper single-blog">

  <!--slider header area are start-->
  <div id="home" class="header-slider-area">
      <!--header start-->
        <?php include_once('includes/header.php');?>
      <!-- header End-->
   </div>
  <!--slider header area are end-->

   <!--  breadcumb-area start-->
   <div class="breadcumb-area bg-overlay">
      <div class="container">
        <ol class="breadcrumb">
          <li><a href="index.php">Home</a></li>
          <li class="active">All Events</li>
        </ol>
      </div>
   </div>
   <!--  breadcumb-area end-->
<div class="upcomming-events-area off-white ptb100">
      <div class="container">
        <div class="row">
          <div class="col-xs-12">
           <h1 class="section-title">Up Comming Events</h1>
          </div>
          <div class="total-upcomming-event col-md-12 col-sm-12 col-xs-12">

<?php
// Fetching Upcomong events
$isactive=1;
$sql = "SELECT EventName,EventLocation,EventStartDate,EventEndDate,EventImage,id from
tblevents where IsActive=:isactive order by id desc ";
$query = $dbh -> prepare($sql);
$query->bindParam(':isactive',$isactive,PDO::PARAM_STR);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $row)
{
   ?>

                <div class="single-upcomming shadow-box">
                  <div class="col-md-4 hidden-sm col-xs-12">
                    <div class="sue-pic">
```

```
                        <img src="admin/eventimages/<?php echo htmlentities($row->EventImage);?>" alt="<?php echo htmlentities($row->EventName);?>" style="border:#000 1px solid">
                    </div>
                    <div class="sue-date-time text-center" style="padding-left:5%;padding-top:15%;">
                        <span><?php echo htmlentities($row->EventStartDate);?></span>To
                        <span><?php echo htmlentities($row->EventEndDate);?></span>
                    </div>
                </div>
                <div class="col-md-3 col-sm-5 col-xs-12">
                  <div class="uc-event-title">
                    <div class="uc-icon"><i class="zmdi zmdi-globe-alt"></i></div>
                    <a href="#"><?php echo htmlentities($row->EventName);?></a>
                  </div>
                </div>
                <div class="col-md-2 col-sm-3 col-xs-12">
                  <div class="venu-no">

                    <p><b>Location :</b> <?php echo htmlentities($row->EventLocation);?></p>
                  </div>
                </div>
                <div class="col-md-3 col-sm-4 col-xs-12">
                  <div class="upcomming-ticket text-center">
<a href="event-details.php?evntid=<?php echo htmlentities($row->id);?>" class="btn-def bnt-2 small">View Details</a>
                  </div>
                </div>
              </div>
 <?php } }  else { ?>
 <p>No Record Found</p>
 <?php } ?>

              <hr />

            </div>
          </div>
        </div>
      </div>
      <!--up comming events area-->
      <!--main blog area start-->

      <!--information area are start-->
        <?php include_once('includes/footer.php');?>
      <!--footer area are start-->
    </div>
    <!--body-wraper-are-end-->
```

```
        <!--==== all js here====-->
        <!-- jquery latest version -->
        <script src="js/vendor/jquery-3.1.1.min.js"></script>
        <!-- bootstrap js -->
        <script src="js/bootstrap.min.js"></script>
        <!-- owl.carousel js -->
        <script src="js/owl.carousel.min.js"></script>
        <!-- meanmenu js -->
        <script src="js/jquery.meanmenu.js"></script>
        <!-- Nivo js -->
        <script src="js/nivo-slider/jquery.nivo.slider.pack.js"></script>
        <script src="js/nivo-slider/nivo-active.js"></script>
        <!-- wow js -->
        <script src="js/wow.min.js"></script>
        <!-- Youtube Background JS -->
        <script src="js/jquery.mb.YTPlayer.min.js"></script>
        <!-- datepicker js -->
        <script src="js/bootstrap-datepicker.js"></script>
        <!-- waypoint js -->
        <script src="js/waypoints.min.js"></script>
        <!-- onepage nav js -->
        <script src="js/jquery.nav.js"></script>
        <!-- animate text JS -->
        <script src="js/animate-text.js"></script>
        <!-- plugins js -->
        <script src="js/plugins.js"></script>
        <!-- main js -->
        <script src="js/main.js"></script>
    </body>
</
```