**CHAPTER 1**

**INTRODUCTION**

College Event-Fest management system is a Database application which is helpful for of the company to book spares. This mini-Project is implemented using HTML and PHP. The project also showcases embedded multimedia and Cascaded Style Sheet.

Operations supported by the application are insert, delete, update and retrieve. Admin has rights to insert, update and delete the event from event list as and when event are available and delete the event which are in the event list.

This application being as a platform to know the events, fest , to apply for the events. Event organizer is an application under project management for managing festivals or social events like gathering, colleges, events, conferences etc. by this application user can register the students, after registering, user can login, after login, event details including name ,contact, address, venue of the event, date, event conducting time, cost of events etc.

In the following sections, a brief introduction about the tools, languages and the databases used to develop the project are discussed.

* 1. **HTML**

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing.

HTML documents are nothing but web pages which contains HTML tags and plain text. The purpose of a web browser is to read HTML documents and display them as web pages. The browser does not display the HTML tags,but uses the tags to interpret the content of the page.

## History

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing. HTML is a simplified version of SGML.

## Tools inorder to use HTML

Tools help us in process of creating HTML document. Some are as follows

* TEXT EDITOR: To create the HTML code we require a text editor or a word processor. Such as, Notepad, WordPad. We are using notepad++ in developing this project.
* WEB BROWSER: The code created by an editor should be executed. This operation can be performed with help of a web browser. Such as Internet Explorer, Netscape navigator, Mozilla Firefox etc.
* GRAPHICS SOFTWARE: To include picture we require a graphic software like Adobe Photoshop.
* Web server: To make the document is to be available on the internet then,we will have to host it on a web server.

## **1.1.1 Significant Language Features**

HTML files are written in ACSII text, so the user can use any text editor to create his/her web page, though a browser of one sort or another is necessary to view the web page. HTML is case insensitive with its language commands. The characters within the document, however, are case sensitive. The language consists of various "tags" which are known as elements. These allow the browser to understand (and put into the desired/specified format) the layout, background, headings, titles, lists, text and/or graphics on the page. The elements are classified according to their function in the HTML document. There are head elements and body elements. The head elements identify properties of the entire document, while body elements actually mark text as content and show a change in the appearance in one way or another. Most elements have a beginning and an ending which encompass the text the user wishes to mark with the tag. All HTML documents must begin with the element and end with the element. Some of the other elements which may be used are tags to create lists--both ordered lists as well as unordered lists. The user may also create larger or smaller, bolder, italicized, or underlined text. Attributes may be used along with the elements. These perform functions such as placement of text, indication of the source files of images, and identification of links to the document or part of the document.

**1.1.2 HTML Code**

Copy and paste the following HTML code into your newly open text file.Which just displays hello world..

<html>  
<header><title>This is title</title></header>  
<body>  
 This is sample text...  
<!-- We use this syntax to write comments -->  
<!-- Page content and rest of the tags here.... -->  
<!-- This is the actual area that gets shown in the browser →

Hello world  
</body>  
</html>

**1.1.3 HTML TAGS**

HTML tags are keywords surrounded by angle brackets like <html>. These are in pair format such that every first tag in pair is start tag where as second tag is end tag. These start and end tags are also called as opening tags and closing tags respectively.

**Tags Used In Project**

The HTML tags are the basis, in order to do this Project. By using some of the important and basically taught tags are used in this Project. Here are some of the tags used in making the Project called AUTOMOBILE MANAGEMENT SYSTEM.

HTML Attributes

Attributes provide additional information about HTML elements.

* HTML elements can have **attributes**
* Attributes provide **additional information** about an element
* Attributes are always specified in **the start tag**
* Attributes come in name/value pairs like: **name="value"**

Some basic text formatting HTML tags are listed:

|  |
| --- |
| **Tag Description** |
| <html> Defines an HTML document |
| <body> Defines the document's body |
| <h1> to <h6> Defines header 1 to header 6 |
| <p> Defines a paragraph |
| <br> Inserts a single line break |
| <b> Defines bold text |
| <!--> Defines a comment |
| <small> Defines small text |

Some of the HTML tags used to create a table are listed:

In an HTML file we can create tables with the Table tags, which in turn will render the browser to display the table in the web page.

|  |
| --- |
| **Tag Description** |
| <table> Defines a table |
| <th> Defines a table header |
| <tr> Defines a table row |
| <td> Defines a table cell |
| <tbody> Defines a table body |
| <tfoot> Defines a table footer |

**A Simple Form**

A form in a web page allows the users to input various data online. In an HTML document; forms can be created with the Form tags. In the following table, some basic Form tags are listed:

|  |
| --- |
| **Tag Description** |
| <form> Defines a form for user input |
| <input> Defines an input field |
| <textarea> Defines a text-area |
| <label> Defines a label to a control |
| <fieldset> Defines a fieldset |
| <legend> Defines a caption for a fieldset |
| <select> Defines a selectable list |
| <optgroup> Defines an option group |
| <option> Defines an option in the drop box |
| <button> Defines a push button |

**Image Tags**

In an HTML document we can insert and display images by using the image tags.

In the following table, some basic Image tags are listed:

**Tag Description**

<IMG SRC attributes> Defines an image

The “src” attribute is used to display an image on a web page. “src” stands for “source”, and its value is the url of the image to be displayed on the page. The url indicates the location where the image is stored. Attributes may be height, width, align so on.

**Background colour**

Using bgcolour attribute this can be done. This is body tag attribute. Six digit hexadecimal code represent the colours.

Syntax: <body text=”text\_color” bgcolor = ”background\_color”>

**Anchor tag**

Anchor tag is used to link two or more different web pages.

Ex: <a href=”next page to be opened”>click here</a> where href stands for hyper link reference.

## **Areas of Application**

HTML only has one area of application at this time and that is the development of web pages. However, not all browsers support all the tags in all versions of HTML. Because of this, it is wise not to design your web page for a specific browser, because what may look fantastic on your browser has no guarantee of looking great on someone else's browser.

**1.2 PHP**

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere.

PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP. Hypertext Pre-processor. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks.

**PHP developer**

PHP developers develop programs, applications, and web sites using the dynamic scripting language PHP. PHP is known for web development and business applications. Depending on job function, PHP developers may be classified as software developers or web developers.

**Tags Description**

 <?php to open PHP section

?>  to close PHP sections

ECHO prints the lines

**1.3 DATABASE**

A database is a collection of [information](http://searchsqlserver.techtarget.com/definition/information) that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.**Database** software systems are programmed in SQL, and examples include Microsoft SQL Server, MySQL, Oracle SAP HANA and FoxPro.

A DBMS system is also required to protect the integrity of data and provide its security.A database management system (**DBMS**) is system software for creating and managing databases. The **DBMS** provides users and programmers with a systematic way to create, retrieve, update and manage data.

**1.4 MYSQL**

MySql is a powerful database. It's very good and free of charge. Many developers in the world selected mysql and php for developing their website.

The MySQL database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used in more than 6 million installations ranging from large corporations to specialized embedded applications on every continent in the world. (Yes, even Antarctica!)

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control.

Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make you successful.

**1.5 XAMP**

The acronym XAMP refers to a set of free ([open source](http://www.webopedia.com/TERM/O/open_source.html)) [applications](http://www.webopedia.com/TERM/A/application.html), combined with Microsoft Windows, which are commonly used in [Web server](http://www.webopedia.com/TERM/W/Web_server.htm) environments. The XAMP stack provides developers with the four key elements of a Web server:  an [operating system](http://www.webopedia.com/TERM/O/operating_system.htm), [database](http://www.webopedia.com/TERM/D/database.html), Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, [Microsoft Windows](http://www.webopedia.com/TERM/M/Microsoft_Windows.html) is the operating system (OS), [Apache](http://www.webopedia.com/TERM/A/Apache_Web_server.html) is the Web server,  [MySQL](http://www.webopedia.com/TERM/M/MySQL.html) handles the database components, while [PHP](http://www.webopedia.com/TERM/P/PHP.html), [Python](http://www.webopedia.com/TERM/P/Python.htm), or [PERL](http://www.webopedia.com/TERM/P/Perl.html) represents the dynamic scripting languages.

**1.6 Notepad++**

Notepad**++** is a [text editor](https://en.wikipedia.org/wiki/Text_editor) and [source code editor](https://en.wikipedia.org/wiki/Source_code_editor) for use with [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). Unlike [Microsoft Notepad](https://en.wikipedia.org/wiki/Microsoft_Notepad), the built-in Windows text editor, it supports [tabbed](https://en.wikipedia.org/wiki/Tab_(GUI)) editing, which allows working with multiple open files in a single window. The project's name comes from the [C](https://en.wikipedia.org/wiki/C_(programming_language)) [increment operator](https://en.wikipedia.org/wiki/Increment_operator).

Notepad++ is distributed as [free software](https://en.wikipedia.org/wiki/Free_software). At first the project was hosted on [SourceForge.net](https://en.wikipedia.org/wiki/SourceForge.net), from where it has been downloaded over 28 million times.

**1.7 Web Browser**

**Google Chrome** is a free **webbrowser** from Google which we are using here. With its clean design and advanced features, Chrome has quickly become one of the most popular web browsers worldwide. In this lesson, we'll talk about the **features of Google Chrome**, how to **download and install Chrome** to your computer, and how to **sign in to Chrome** using a Google account.

**CHAPTER 2**

**System Analysis and Design**

In this chapter, a complete description of the project development is discussed. The requirements of the project identified are showcased. The database design is done Using High-Level Conceptual Data Models

###### **2.1 Requirement Analysis**

Following requirements were identified during the requirement collection and analysis.

Software Requirements:

1. Operating System- Windows 1
2. Internet Browser (Mozilla Firefox/Google Chrome)
3. Notepad++
4. MySql (database)

Hardware Requirements:

1. Dual Core GHZ or later CPU.

2. 4GB RAM.

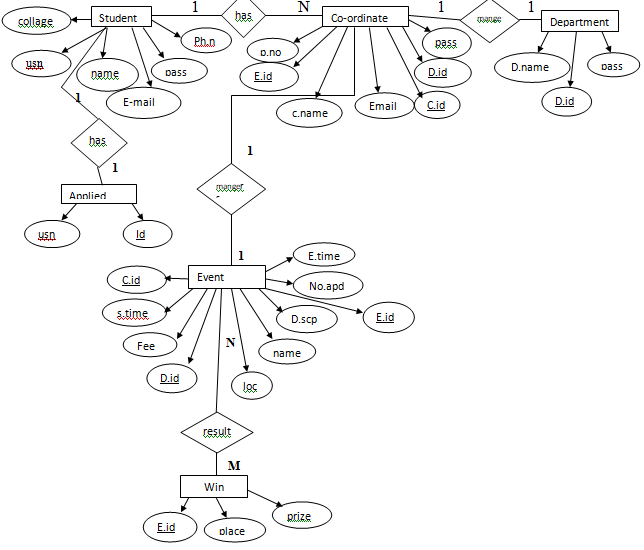
3. 100gb storage minimum.

### Feasibility Study

### The feasibility study carried out showed that the requirements that were to be included could be provided by the use of RDBMS software such as MySQL which is available as an open source and for the front end HTML pages with processing capability provided by the Scripting language php.

**2.2ER-Diagram**

Following is the conceptual representation of the requirements identified as an ER-Diagram.



**Figure 2.2 ER Diagram for College Event, Fest Management**

**2.3Relational Schema**

The relational schema diagram has been derived from the ER-Diagram in Figure 2.1 using the ER-Relational mapping algorithm

**Student**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Usn | Pass | Phno | Email | Clg |

**Applied**

|  |  |
| --- | --- |
| Usn | Eid |

**Co-Ordinater**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cname | Cid | Pass | Did | Phno | Email | Event\_id |

**Department**

|  |  |  |
| --- | --- | --- |
| Dname | Did | Pass |

**Event**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Eid | Descp | Cid | Did | Loc | Stime | Etime | Fee | No Applied |

**Win**

|  |  |  |  |
| --- | --- | --- | --- |
| Eid | Usn | Place | Price |

**Figure 2.3 Relational Schema Diagram Showing the Primary key and Foreign key relationships**

**2.4 Functional Requirements**

Functional requirements of a software project interpret the function of a part. It defines its functions, input and output. The typical functional requirements include:

Application contains 4 modules:

* Admin module.
* Co-ordinates module.
* Department module.
* Student module.

Admin module:

* Admin can able to view all the event information and remove the event.
* Admin can able to view all Students information and remove the student.
* Admin can able to view all department information.
* Admin can able to view all co-ordinates’ information.
* Admin can able to view all wins information.
* Admin can able to add a department.
* Admin can able to Host an event.
* Admin can able to login.

Co-ordinates module:

* Co-ordinates able to view the event information.
* Co-ordinates able to view Students information.
* Co-ordinates able to view all win information.
* Co-ordinates able to search Students information.
* Co-ordinates able to login.

Department module:

* Department can view all the event information and remove the event.
* Department can view all wins information..
* Department can able to view all co-ordinates’ information.
* Department can able to update co-ordinates’ information.
* Department able to view all wins information.
* Department can able to remove event.
* Department can able to Host an event, update event.
* Department can able to login.

Student module:

* Student able to view the event information.
* Student able to login.
* Student able to sign up.

**2.5 Non- Functional Requirements**

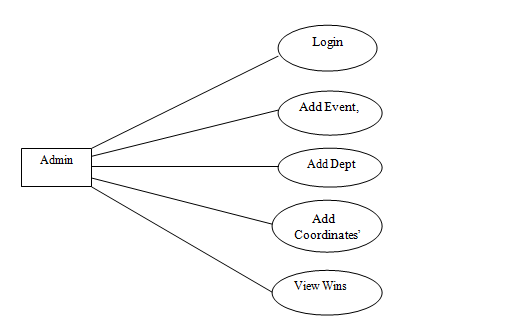
A non-functional requirement specifies the canon of the articular process not the particular judgment of the system and particular behavior of the process. Non-functional requirements define how the system works.

* Usability: Reduce time for finding location and be user friendly GUI so that users can navigate easily through it.
* Accuracy: To understand use of this application consider the flow of actions happening, initially by opening the application window, it will ask for the user authorization where the user needs to login.
* Reliability: The software architectural pattern of the proposed selective event optimizer invokes the major use of the user interfaces for the selection of the events.
* Performance: Once the application starts, the user application should complete all tasks without errors. The system should capable to enhance with further technology in future to improve its features compared to the existing system
* The system should be reliable and it should be related in all the condition and it should be recoverable in all the situation or condition if error occurs.

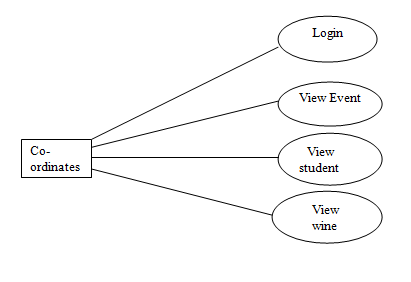
**2.6 Use Case Diagram**

The use case diagrams usually refer to behavioral diagrams helps people to understand the interaction between user and system. Use case diagram identify different users of the system. It is used to define some set of actions, which is called as use cases.Actors are the result of some valuable use cases. Use case figures are also called as unified modeling language.

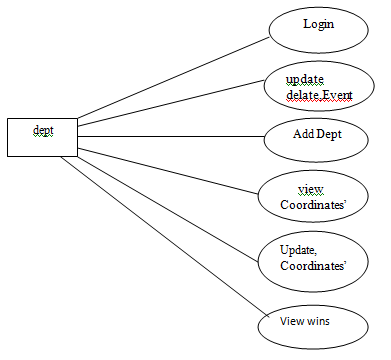
**Admin module :**

****

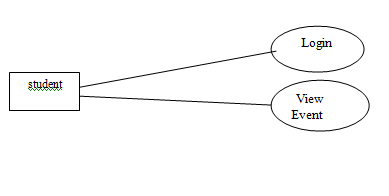
**Co-ordinate Module:**



**Department module:**



**Student module:**



**CHAPTER 3**

**SYSTEM IMPLEMENTATION**

**3.1 Database Design**

Table structure for table `admin`

CREATE TABLE IF NOT EXISTS `admin` (

`name` varchar(20) NOT NULL,

`email` varchar(40) NOT NULL,

`pass` varchar(20) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1

Table structure for table `applied`

CREATE TABLE IF NOT EXISTS `applied` (

`usn` varchar(20) NOT NULL,

`eid` int(10) NOT NULL,

PRIMARY KEY (`usn`,`eid`),

KEY `eid` (`eid`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Table structure for table `coordinator`

CREATE TABLE IF NOT EXISTS `coordinator` (

`cname` varchar(20) DEFAULT NULL,

`cid` int(10) NOT NULL,

`pass` varchar(20) NOT NULL DEFAULT 'co123',

`did` int(10) DEFAULT NULL,

`phno` bigint(15) DEFAULT NULL,

`email` varchar(20) DEFAULT NULL,

`eventid` int(10) DEFAULT NULL,

PRIMARY KEY (`cid`),

KEY `did` (`did`),

KEY `eventid` (`eventid`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Table structure for table `dept

CREATE TABLE IF NOT EXISTS `dept` (

`dname` varchar(20) DEFAULT NULL,

`did` int(10) NOT NULL,

`pass` varchar(20) NOT NULL DEFAULT 'dedpt123',

PRIMARY KEY (`did`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Table structure for table `eventt`

CREATE TABLE IF NOT EXISTS `eventt` (

`name` varchar(20) DEFAULT NULL,

`eid` int(10) NOT NULL,

`descp` varchar(50) DEFAULT NULL,

`cid` int(10) DEFAULT NULL,

`did` int(10) DEFAULT NULL,

`loc` varchar(20) DEFAULT NULL,

`stime` time DEFAULT NULL,

`etime` time DEFAULT NULL,

`fee` int(10) DEFAULT NULL,

`noapplied` int(10) DEFAULT '0',

PRIMARY KEY (`eid`),

KEY `cid` (`cid`),

KEY `did` (`did`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Table structure for table `student`

CREATE TABLE IF NOT EXISTS `student` (

`name` varchar(20) DEFAULT NULL,

`usn` varchar(20) NOT NULL,

`pass` varchar(20) DEFAULT NULL,

`phno` bigint(15) DEFAULT NULL,

`email` varchar(50) DEFAULT NULL,

`clg` varchar(20) DEFAULT NULL,

PRIMARY KEY (`usn`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Table structure for table `win`

CREATE TABLE IF NOT EXISTS `win` (

`eid` int(10) NOT NULL,

`usn` varchar(20) NOT NULL,

`place` varchar(20) NOT NULL,

`price` int(10) DEFAULT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

**3.2 Database Connectivity**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>connection</title>

</head>

<body>

<?php

$conn = mysqli\_connect("localhost","root","","fest");

// Check connection

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

}

else{

echo "successfully connected!";

}

?>

</body>

</html>

**3.3 Implementation of Database Operations**

**Source Code for Insert:**

<?php

$eid = $\_POST["eid"];

$ename = $\_POST["ename"];

$desc = $\_POST["desc"];

$did = $\_POST["did"];

$cid = $\_POST["cid"];

$cname = $\_POST["cname"];

$phno = $\_POST["phno"];

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

$loc = $\_POST["loc"];

$stime = $\_POST["stime"];

$etime = $\_POST["etime"];

$fee = $\_POST["fee"];

$first = $\_POST["first"];

$second = $\_POST["second"];

$third = $\_POST["third"];

$f="first";

$s="second";

$t="third";

$u=0;

$link = mysqli\_connect("localhost", "root", "", "fest");

if($link === false){

die("ERROR: Could not connect. " . mysqli\_connect\_error());

}

$sql = "INSERT INTO coordinator (cname,cid,did,phno,email, eventid) VALUES (('$cname'),('$cid'),('$did'),('$phno'),('$email'),('$eid'))";

if(mysqli\_query($link, $sql)){

echo "Co=ordinator Records inserted successfully.\n";

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

$sql = "INSERT INTO eventt (name,eid,descp,cid,did,loc,stime ,etime,fee) VALUES (('$ename'), ('$eid'),('$desc'),('$cid'),('$did'), ('$loc'),('$stime'),('$etime'),('$fee') )";

if(mysqli\_query($link, $sql)){

echo "Event Records inserted successfully.\n";

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

$sql = "INSERT INTO win (eid,usn,place,price) VALUES (('$eid'), ('$u') , ('$f') ,('$first'))";

if(mysqli\_query($link, $sql)){

echo "WIN \n";

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

$sql = "INSERT INTO win (eid,usn,place,price) VALUES (('$eid'), ('$u') , ('$s') ,('$second'))";

if(mysqli\_query($link, $sql)){

echo "Records";

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

$sql = "INSERT INTO win (eid,usn,place,price) VALUES (('$eid'), ('$u') , ('$t') ,('$third'))";

if(mysqli\_query($link, $sql)){

echo " inserted successfully. ";

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

mysqli\_close($link)

**Source Code for Update:**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href=css/main.css rel="stylesheet" type="text/css">

<link href=css/home.css rel="stylesheet" type="text/css">

<title>PESITM</title>

</head>

<body>

<div class="body">

<form action="" method="POST">

<table align="center" style="padding-top:20px" cellpadding="10px" cellspacing="5px" >

<tr>

<th colspan="4" align="center" >

<h2><u>Student Edit Form<u></h2>

</th></tr><tr><td align="center"><label>Enter USN:</label></td><td >

<input type="text" name="USN" placeholder=" Enter USN" required ></td>

<td ><button value="search" name="search"> Search</button></td></tr>

<?php

include ("connection.php");

if ($conn-> connect\_error)

{

die("connection failed:".$conn-> connect\_error);

}

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

{

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

$sql = "SELECT \* from student where usn=$USN";

$result = $conn-> query($sql);

while($row= mysqli\_fetch\_array($result))

{

?>

<form action="" method="POST">

<table align="center" style="padding-top:20px" cellpadding="10px" cellspacing="5px" >

<tr hidden><td align="center"><label>Enter USN:</label></td><td >

<input type="text" name="USN"value="<?php echo $row['usn'] ?>" hidden /></td>

</tr><tr><td><label>Student Name:</label></td><td>

<input type="text" name="sname" value="<?php echo $row['name'] ?>" />

</td></tr><tr><td><label>Phone Number</label></td><td>

<input type="text" name="phoneno" value="<?php echo $row['phno'] ?>"/>

</td><td><label>Email:</label></td><td>

<input type="text" name="email" value="<?php echo $row['email'] ?>"/></td></tr>

<tr><td><label>College:</label></td><td>

<input type="text" name="college" value="<?php echo $row['clg'] ?>"/></td>

</tr><tr style="height:20px"></tr><tr align="center"><td colspan="4">

<button class="btn-sub" value="submit" name="save">Save</button> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<button class="btn-can" name="cancel">cancle</button></td></tr>

<?php

}

}

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

{

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

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

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

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

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

include ("connection.php");

if ($conn-> connect\_error)

{

die("connection failed:".$conn-> connect\_error);

}

$sql = "Update student set usn=$USN,name='$sname',phno='$phoneno',email='$email',clg='$college', where usn= '$USN'";

$result = $conn-> query($sql);

echo "Record updated";

header("Location: home.html");

}

?>

</table>

</form>

</div>

</div>

</body>

</html>

**Source Code for Delete:**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href=css/main.css rel="stylesheet" type="text/css">

<link href=css/home.css rel="stylesheet" type="text/css">

<title>PESITM</title>

</head>

<body>

<div class="body">

<form action="" method="POST">

<table align="center" style="padding-top:20px" cellpadding="10px" cellspacing="5px" >

<tr>

<th colspan="4" align="center" >

<h2><u>Remove Event<u></h2></th></tr><tr>

<td align="center"><label>Enter Event Id:</label></td>

<td ><input type="text" name="USN" placeholder=" Enter USN" required >

</td><td ><button value="search" name="search"> delete</button>

</td></tr></table>

<?php

include ("connection.php");

if ($conn-> connect\_error)

{

die("connection failed:".$conn-> connect\_error);

}

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

{

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

$sql = "delete from eventt where eid=$USN";

$result = $conn-> query($sql);

echo "Record updated";

header("Location: home.html");

}

?></form></div>

</body>

</html>