ABSTRACT

As the name specifies “STUDENT ATTENDANCE AND LEAVE MANAGEMENT APPLICATION WITH PARENTAL CONTROL SYSTEM” is an online Leave Management System for the students. Hence, Leave Application for those students can reduce burden of staffs in the college. This project allows you to manage the leave of the students and also allows you sanction leave request. This is a web application portal that enhances the organization to manage and control the students’ leave application. Online Leave Management System is a web based application that is developed using PHP and MySQL. Leave Management System is to develop a web based application using which management of leave application and approval is done online. Students of the institute can apply for leave and adjust their load directly from this application.

Front end : php

Back end : MySQL

.

1. INTRODUCTION

In this project, we have designed a Leave Management Application System which helps them to save leave records of the students. This project deals with the problems on managing a leave and avoids the problem which occurs manually. This project helps to overcome those leave application problems for students. In this project, Admin will add student detail and approve the leave application of the students. After applying for a leave, students can view their leave letter. Students after submission of their leave letter, a message regarding their leave is sent to their respective parents. Parents can accept or reject their child's leave application .But the final leave application is approved or rejected only by admin. This helps the students in taking leave unnecessarily as their leave application is taken care by their parents. This application can help in avoiding the leave application that needs to be given to the higher authority with great ease. This application can also help in sending the notifications to the users regarding the leaves that they have availed with great ease.

2 .PROBLEM DEFINITION AND DESCRIPTION

2.1 EXISTING SYSTEM:

In existing system, students can apply leave whenever they need without manually. Student's leave application is sent to the admin. The admin can accept or reject student's leave letter. But this leave letter procedure is done without parent's concern. So, the students unnecessarily take leave without proper concern from their parents. This affects their studies and they miss their proper attendance.

2.2 PROPOSED SYSTEM:

In our proposed system, the main advantage of our project is that student can only take leave only after it is accepted by their parents. This helps in proper regulation in their studies as well as education system. But the admin is the ultimate decision maker. First, the student applies leave, It is viewed by admin. The admin send their leave application letter to their parents. The parents can accept or reject their leave letter. The students can only view their leave letter but they cannot unblock their leave letter. The admin takes final decision to accept or reject them. Even when their parents approve their leave, it is up to the admin who takes final decision. This helps the students in being regular to their studies and also helps their parents to take care of their child.

3. SYSTEM REQUIREMENTS

3.1 Hardware Specification

Processor	:	2.4 GHz processor
Ram	:	1GB
Hard Disk	:	240GB
Monitor	:	LED Monitor
Keyboard	:	Multimedia Keyboard
Mouse	:	Optical mouse

3.2 Software Specification

Operating System	:	Windows 7
Front-End	:	PHP
Web Server	:	Apache
Back End	:	MySQL

4. System Analysis

Module Description:

This standard framework supports the following components

- Admin Module
- StudentModule
- ParentModule

4.1 Admin Module:

Users can access their account through they logged in. The Admin Module is that admin will add student details. The admin will view student leave application. The admin will either accept or reject their leave application.

4.2 Student Module:

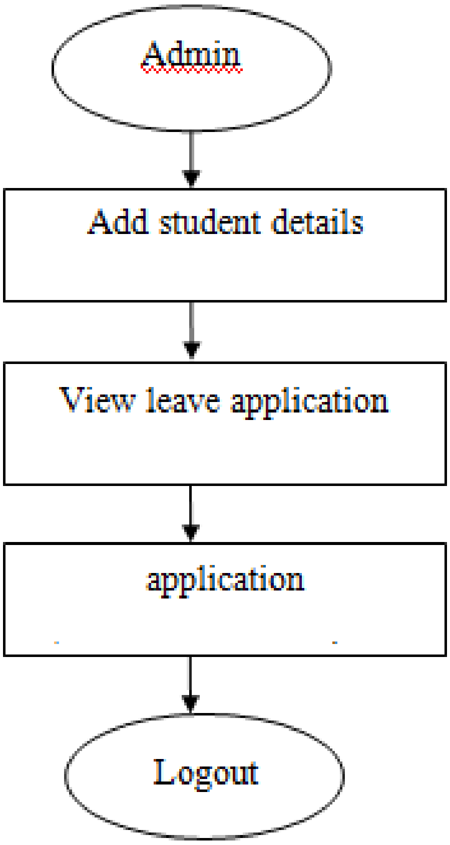
In this module, Students can login to their account. They will apply for leave by logging in their account. After application of leave, students can view their leave application.

4.3 Parent Module:

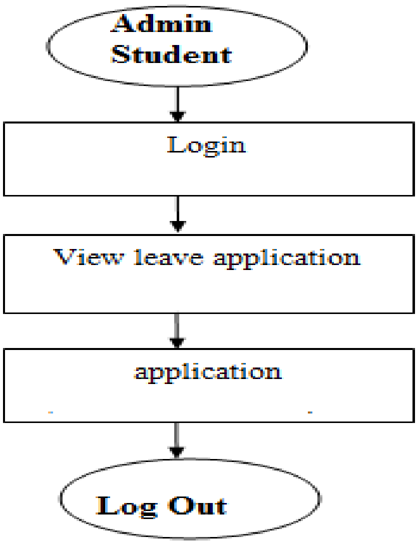
In this module, The admin will send their child's leave application to their parents. The parents can view their child's leave application. They can either accept or reject or leave application.

System Designing

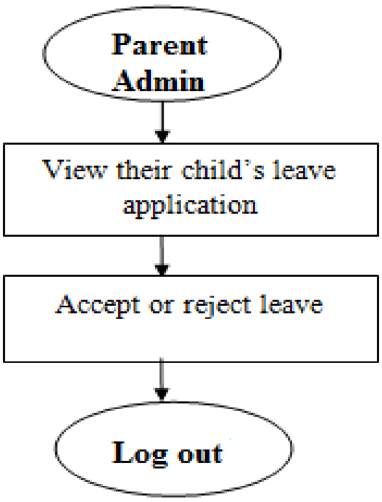
4.1 ADMIN MODULE



4.2 STUDENT MODULE



4.3 PARENT MODULE



5. Database design

Table structure for table student

Field	Type	Null	Default
student_id	int(11)	Yes	NULL
name	varchar(225)	Yes	NULL
register_no	varchar(225)	Yes	NULL
department	varchar(225)	Yes	NULL
year	varchar(225)	Yes	NULL
phone	varchar(225)	Yes	NULL
email	varchar(225)	Yes	NULL
dob	varchar(225)	Yes	NULL
age	varchar(225)	Yes	NULL
parent_name	varchar(225)	Yes	NULL
parent_no	varchar(225)	Yes	NULL
parent_email	varchar(225)	Yes	NULL
address	varchar(225)	Yes	NULL
photo	varchar(225)	Yes	NULL
status	int(11)	Yes	NULL

Table structure for table leave_details

Field	Type	Null	Default
leave_id	int(11)	Yes	NULL
name	varchar(225)	Yes	NULL
register_no	varchar(225)	Yes	NULL
department	varchar(225)	Yes	NULL
year	varchar(225)	Yes	NULL
from	varchar(225)	Yes	NULL
to	varchar(225)	Yes	NULL
reason	varchar(225)	Yes	NULL
status	int(11)	Yes	NULL

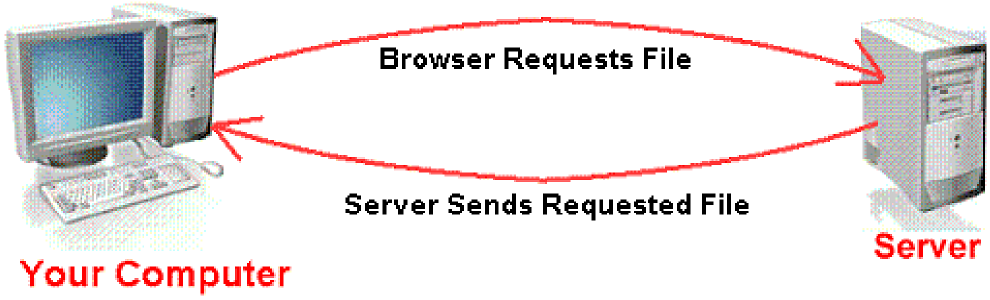
6.Implementation

PHP:

PHP: Hypertext Preprocessor (the name is a recursive acronym) is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on most operating systems and computing platforms.

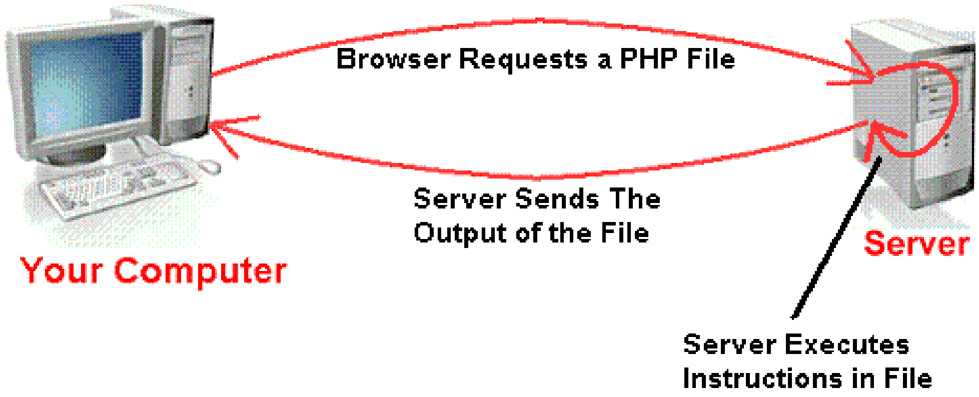
Hypertext refers to files linked together using hyperlinks, such as HTML (Hypertext Markup Language) files. Preprocessing is executing instructions that modify the output. Below is a demonstration of the difference between HTML and PHP files.

Accessing an HTML Page



- Your browser sends a request to that web page's server (computer) for the file (HTML or image) you wish to view.
- The web server (computer) sends the file requested back to your computer.
- Your browser displays the file appropriately.
- If you request a PHP file (ends with ".php"), the server handles it differently.

Accessing a PHP Page


Browser Requests a PHP File
Server Sends The Output of the File
Your Computer
Server
Server Executes Instructions in File

- Your browser sends a request to that web page's server for the PHP file you wish to view.
- The web server calls PHP to interpret and perform the operations called for in the PHP script.
- The web server sends the output of the PHP program back to your computer.
- Your browser displays the output appropriately.

Security

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to control structures described within PHP code). The most common delimiters are <?php to open and ?> to close PHP sections. <script language="php"> and </script> delimiters are also available, as are the shortened forms <? or <?= (which is used to echo back a string or variable) and ?> as well as ASP-style short forms <% or <%= and %>. While short delimiters are used, they make script files less portable as support for them can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

MY SQL:

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 on every continent -- Yes, even Antarctica! -- by individual Web developers as well as many of the world's largest and fastest-growing organizations to save time and money powering their high-volume Web sites, business-critical systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, and Zappos.com.

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, Mac OS, Solaris, HP-UX, IBM AIX, 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.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations.The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

HTML:

HTML, which stands for Hypertext Markup Language, is the predominant markup language for web pages. HTML is the basic building-blocks of webpage. HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like <html>), within the web page content. HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags).

The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpage.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML markup.

There are several types of markup elements used in HTML.

Structural markup describes the purpose of text. For example, <h2>Golf</h2> establishes "Golf" as a second-level heading, which would be rendered in a browser in a manner similar to the "HTML markup" title at the start of this section. Structural markup does not denote any specific rendering, but most web browsers have default styles for element formatting. Text may be further styled with Cascading (CSS).

Presentational markup describes the appearance of the text, regardless of its purpose. For example <b>boldface</b> indicates that visual output devices should render "boldface" in bold text, but gives little indication what devices which are unable to do this (such as aural devices that read the text aloud) should do. In the case of both <b>bold</b> and <i>italic</i>, there are other elements that may have equivalent visual renderings but which are more semantic in nature, such as <strong>strong text</strong> and<em>emphasis text</em> respectively. It is easier to see how an aural user agent should interpret the latter two elements. However, they are not equivalent to their presentational counterparts: it would be undesirable for a screen-reader to emphasize the name of a book, for instance, but on a screen such a name would be italicized. Most presentational markup elements have become deprecated under the HTML 4.0 specification, in favor of CSS based styling.

CSS or Cascading Style Sheets

Cascading Style Sheets is a special style-sheet language that describes the presentation of a document that has been written in a markup language like HTML or XHTML. CSS is commonly used to define the presentation of web pages written in XHTML or HTML. The specifications of CSS are maintained by WWW Consortium or W3C.

To be very precise, CSS came into being as a solution of a problem that arose when it became very difficult to design Web Pages where the content of the HTML documents and its presentation layout was evidently separated. This problem arose when the two most well-known and major web browsers (Internet Explorer and Netscape Navigator) continued adding various new HTML tags to the HTML specification.

JavaScript

JavaScript is a cross-platform, object-oriented scripting language. JavaScript is a small, lightweight language; it is not useful as a standalone language, but is designed for easy embedding in other products and applications, such as web browsers. Inside a host environment, JavaScript can be connected to the objects of its environment to provide programmatic control over them.

Core JavaScript contains a core set of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

Client-side JavaScript extends the core language by supplying objects to control a browser (Navigator or another web browser) and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a relational database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.Through JavaScript's LiveConnect functionality, you can let Java and JavaScript code communicate with each other. From JavaScript, you can instantiate Java objects and access their public methods and fields. From Java, you can access JavaScript objects, properties, and methods.Netscape invented JavaScript, and JavaScript was first used in Netscape browsers.

7. System Testing

UNIT TESTING:

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

SYSTEM TESTING:

Several modules constitute a project. If the project is long-term project, several developers write the modules. Once all the modules are integrated, several errors may arise. The testing done at this stage is called system test. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Testing a specific hardware/software installation. This is typically performed on a COTS (commercial off the shelf) system or any other system comprised of disparate parts where custom configurations and/or unique installations are the norm.

FUNCTIONAL TESTING:

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

8.Codings

Login

```
<?php ob_start();

include "include/dbc.php";


?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="">

<meta name="author" content="">

<title>Hostel Management System</title>


<!-- core CSS -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<link href="css/font-awesome.min.css" rel="stylesheet">

<link href="css/prettyPhoto.css" rel="stylesheet">

<link href="css/animate.min.css" rel="stylesheet">

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

```
<link href="css/responsive.css" rel="stylesheet">



<link rel="shortcut icon" href="images/ico/favicon.ico">
<link rel="apple-touch-icon-precomposed" sizes="144x144" href="images/ico/apple-touch-icon-144-precomposed.png">
<link rel="apple-touch-icon-precomposed" sizes="114x114" href="images/ico/apple-touch-icon-114-precomposed.png">
<link rel="apple-touch-icon-precomposed" sizes="72x72" href="images/ico/apple-touch-icon-72-precomposed.png">
<link rel="apple-touch-icon-precomposed" href="images/ico/apple-touch-icon-57-precomposed.png">
</head><!--/head-->

<body>

<header id="header">
<nav class="navbar navbar-inverse" role="banner">
<div class="container">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
```

```
</button>

<a class="navbar-brand" href="index.php">Hostel Management System</a>

</div>



<div class="collapse navbar-collapse navbar-right">

<ul class="nav navbar-nav">

<li class="active"><a href="student/index.php">Student Login</a></li>

</ul>

</div>

</div><!--/.container-->

</nav><!--/nav-->

</header><!--/header-->

<section id="contact-page">

<div class="container">

<div class="center">

<h2>Admin Login</h2>

</div>

<div class="row contact-wrap">

<div class="status alert alert-success" style="display: none"></div>

<form class="contact-form" name="contact-form" method="post"
enctype="multipart/form-data">

<div class="col-md-offset-4 col-sm-5 col-sm-offset-3">



<div class="form-group">

<label>Admin Name *</label>
```

```
<input type="text" name="aname" class="form-control" required>
</div>
<div class="form-group">
<label> Password </label>
<input type="password" name="pwd" class="form-control" required>
</div>
<div class="form-group">
<button type="submit" name="s1" class="btn btn-primary btn-lg">Login Now</button>
</div>
</div>
</div>
</form>
<?php
if(isset($_POST['s1']))
{
extract($_POST);

if($aname=="admin" && $pwd=="admin")
{
session_start();
$_SESSION['name']="admin";
header("location:student.php");
}
}
```

```
?>

</div><!--/.row-->

</div><!--/.container-->

</section><!--/#contact-page-->

<br><br><br><br><br><br><br><br><br>

<footer id="footer" class="midnight-blue">

<div class="container">

<div class="row">

<div class="col-sm-6">

&copy; 2017. All Rights Reserved.

</div>

<div class="col-sm-6">

<ul class="pull-right">

<li><a href="index.php">Home</a></li>

<li><a href="about-us.php">About Us</a></li>

<li><a href="login.php">Login</a></li>

</ul>

</div>

</div>

</div>

</footer><!--/#footer-->

<script src="js/jquery.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/jquery.prettyPhoto.js"></script>
```

```
<script src="js/jquery.isotope.min.js"></script>

<script src="js/main.js"></script>

<script src="js/wow.min.js"></script>

</body>

</html>
```

Database Connection:

```
<?php

$connect=mysql_connect("localhost","root","")or die("Couldn't Connect");

$db=mysql_select_db("hostel_management",$connect)or die("Couldn't Connect db");

?>
```

Add Student

```
<h2>Student Detail</h2>


<br><br>


<form class="contact-form" name="contact-form" method="post" enctype="multipart/form-data">

<div class="row">

<div class="col-sm-6 ">

<div class="form-group">

<label>Name *</label>

<input type="text" name="name" class="form-control" required>

</div>

</div>
```

```
<div class="col-sm-6 ">
<div class="form-group">
<label>Register No *</label>
<input type="text" name="register_no" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Department *</label>
<input type="text" name="department" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Year *</label>
<input type="text" name="year" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Phone *</label>
<input type="text" name="phone" class="form-control" required>
</div>
</div>
```

```html
<div class="col-sm-6 ">
<div class="form-group">
<label>Email *</label>
<input type="email" name="email" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>DoB *</label>
<input type="date" name="dob" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Age *</label>
<input type="number" name="age" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Student Photo *</label>
<input type="file" name="photo" class="form-control" required>
</div>
</div>
```

```html
<div class="col-sm-6 ">
<div class="form-group">
<label>Parent / Guardian Name *</label>
<input type="text" name="parent_name" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Parent Number *</label>
<input type="text" name="parent_no" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Parent Email*</label>
<input type="email" name="parent_email" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Address *</label>
<textarea class="form-control" rows="5" cols="15" name="address"></textarea>
</div>
</div>
```

```
</div>

<div class="form-group">
<button type="submit" name="s1" class="btn btn-primary btn-lg">Register Now</button>
</div>
</div>
</div>
</form>
<?php
if(isset($_POST['s1']))
{
extract($_POST);
$photo=$_FILES['photo']['name'];
$qry=mysql_query("INSERT INTO `student` (`name` ,`register_no` ,`department` ,`year` ,`phone` ,`email` ,`dob` ,`age` ,`parent_name` ,`parent_no` ,`parent_email` ,`address` ,`photo` )VALUES
('$name' ,'$register_no' ,'$department' ,'$year' ,'$phone' ,'$email' ,'$dob' ,'$age' ,'$parent_name' ,'$parent_no' ,'$parent_email' ,'$address' ,'$photo' )");
if($qry)
{
$move=move_uploaded_file($_FILES['photo']['tmp_name'],"images/student/".$photo);
echo "<script>alert('Added Successfully')
window.location='student.php'</script>";
//header("location:index.php");
}

else

{

echo mysql_error();

}

}    ?>

Student Details

<h2>Student Detail</h2>

<a href="add_student.php" class="btn btn-info" style="float:right">Add Student</a>

<br><br>


<form name="form1" method="post" action="#">


<table border="1" align="center" cellspacing="0" cellpadding="20">

<tr>

<th> Name</th>

<th> Department</th>

<th> Year</th>

<th> Photo</th>

<th> E-mail</th>

<th> Edit </th>

<th> Delete </th>

<th> Decision </th>

</tr>
```

```
<?php ob_start();
include "include/dbc.php";
include "include/session.php";

$qry=mysql_query("select * from student");
$no=mysql_num_rows($qry);
if($no>0)
{


while($row=mysql_fetch_array($qry))
{
$id=$row['student_id'];
?>


<tr>
<td><?php echo $row['name']; ?></td>
<td><?php echo $row['department']; ?></td>
<td><?php echo $row['year']; ?></td>
<td><img src="images/student/<?php echo $row['photo']; ?>" width="100" height="100"></td>
<td><?php echo $row['email']; ?></td>
<td>
<a href="edit_student.php?id=<?php echo
```

```
$id;?>">Edit</a>

</td>

<td>

<a href="delete_student.php?id=<?php echo

$id;?>">Delete</a>

</td>

<td>

<?php

if($row['status']==0)

{

?>

<a href="block_student.php?s=b" class="btn btn-warning">Block</a>

<?php

}

else

{

?>

<a >Blocked</a>

<?php

} ?>

</td>



</tr>

<?php
```

```
}
?>
<?php
}
else
{
echo "Sorry No Records Found";
}

?>
Leave Details
<h2>Leave Details</h2>
<a href="leave_detail.php?status=0" class="btn btn-info">Parent's Approval Pending</a>
<a href="leave_detail.php?status=1" class="btn btn-warning">Admin Approval Pending</a>
<a href="leave_detail.php?status=2" class="btn btn-success"> Approved Applications</a>
<br><br>
<form name="form1" method="post" action="#">

<table border="1" align="center" cellspacing="0" cellpadding="20">
<tr>
<th> Name</th>
<th> Department</th>
```

```
<th> Year</th>

<th> Register No</th>

<th> From</th>

<th> To</th>

<th> Reason</th>

<?php

if($_REQUEST['status']==1)

{

?>

<th> Decision </th>

<?php

}

?>

<th> Delete </th>

</tr>

<?php ob_start();

include "include/dbc.php";

include "include/session.php";

if(empty($_REQUEST['status']))

{

$qry=mysql_query("select * from leave_details where status=0 order by leave_id
desc");

}

else{

$status=$_REQUEST['status'];
```

```
$qry=mysql_query("select * from leave_details where status=$status order by leave_id desc");
}
$no=mysql_num_rows($qry);
if($no>0)
{


while($row=mysql_fetch_array($qry))
{
?>


<tr>
<td><?php echo $row['name']; ?></td>
<td><?php echo $row['department']; ?></td>
<td><?php echo $row['year']; ?></td>
<td><?php echo $row['register_no']; ?></td>
<td><?php echo $row['from']; ?></td>
<td><?php echo $row['to']; ?></td>
<td><?php echo $row['reason']; ?></td>
<?php
if($_REQUEST['status']==1)
{
?>
<td>

<a href="leave_status.php?id=<?php echo

$row['leave_id'];?>&&s=a">Approve</a> /

<a href="leave_status.php?id=<?php echo

$row['leave_id'];?>&&s=r">Reject</a>

</td>

<?php

}

?>

<td>

<a href="delete_leave.php?id=<?php echo

$row['leave_id'];?>">Delete</a>

</td>


</tr>

<?php

}

?>

<?php

}

else

{

echo "Sorry No Records Found";

}
```

```
?>
Student Profile
<?php ob_start();
include "..//include/dbc.php";
include "include/session.php";

$qry=mysql_query("select * from student where register_no='$register_no'");


$row=mysql_fetch_array($qry)

?>
<section id="contact-page">
<div class="container">
<div class="center">
<br>
<h2>Welcome <?php echo ucfirst($row['name']);?></h2>


<form name="form1" method="post" action="#">
<img src="..//images/student/<?php echo $row['photo']; ?>" width="100" height="100" class="img-circle"><br><br>
</div>
<div class="row">
```

```
<div class="col-sm-4 ">

<p><b>Department: </b><?php echo $row['department']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Year: </b><?php echo $row['year']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Phone: </b><?php echo $row['phone']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Email: </b><?php echo $row['email']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>DoB: </b><?php echo $row['dob']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Age: </b><?php echo $row['age']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Parent Name: </b><?php echo $row['parent_name']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Parent Number: </b><?php echo $row['parent_no']; ?></p>

</div>
```

```
<div class="col-sm-4 ">

<p><b>Parent Email: </b><?php echo $row['parent_email']; ?></p>

</div>

<div class="col-sm-4 ">

<p><b>Address: </b><?php echo $row['address']; ?></p>

</div>

</div>
```

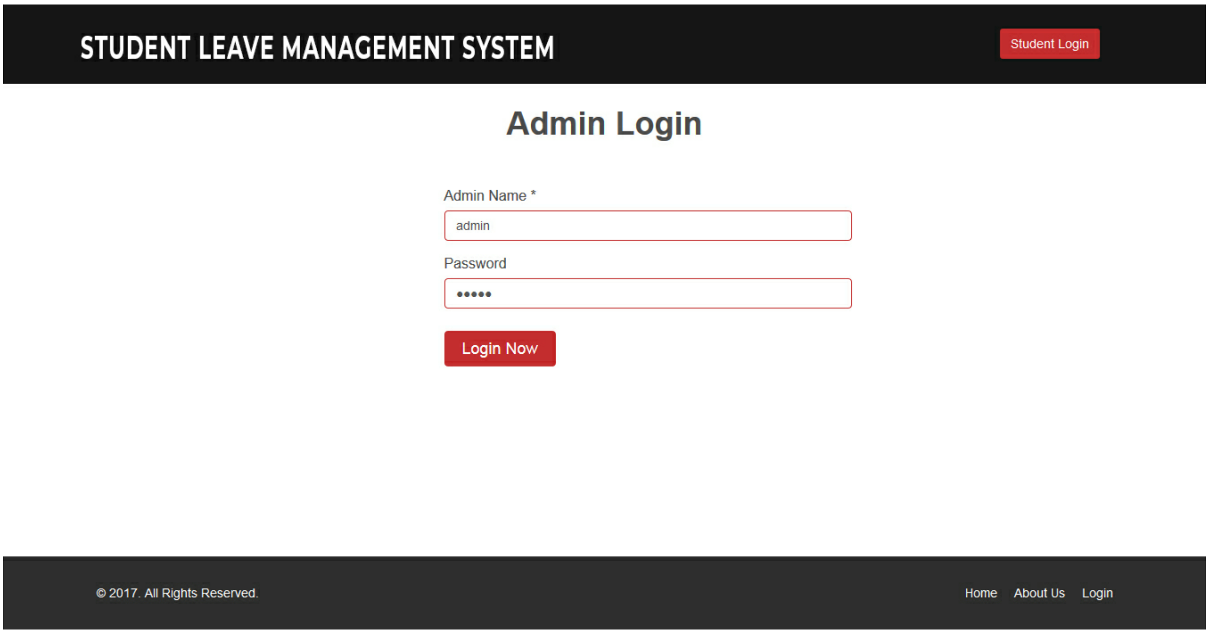
Apply Leave

```
<section id="contact-page">

<div class="container">

<div class="center">

<br>

<h2>Apply Leave </h2>


<br><br>




<form class="contact-form" name="contact-form" method="post" enctype="multipart/form-data">

<div class="row">

<div class="col-sm-6 ">

<div class="form-group">

<label>From *</label>

<input type="date" name="from" class="form-control" required>

</div>
```

```
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>To *</label>
<input type="date" name="to" class="form-control" required>
</div>
</div>
<div class="col-sm-6 ">
<div class="form-group">
<label>Reason *</label>
<input type="text" name="reason" class="form-control" required>
</div>
</div>
</div>
<div class="form-group">
<button type="submit" name="s1" class="btn btn-primary btn-lg">Apply Now</button>
</div>
</div>
</div>
</form>
<?php
if(isset($_POST['s1']))
{
extract($_POST);
$student=mysql_query("select * from student where register_no='$register_no'");

$row=mysql_fetch_array($student);

$name=$row['name'];

$department=$row['department'];

$year=$row['year'];

$qry1=mysql_query("INSERT INTO `leave_details` (`name` ,`register_no`
 ,`department` ,`year` ,`from` ,`to` ,`reason` )VALUES

('$name' ,'$register_no' ,'$department' ,'$year' ,'$from' ,'$to' ,'$reason')");

if($qry1)

{


echo "<script>alert('Your leave request has been sent. Please wait for approval')

window.location='leave.php'</script>";


}

else

{


echo mysql_error();}}?>
```

9.SCREENSHOTS

Home



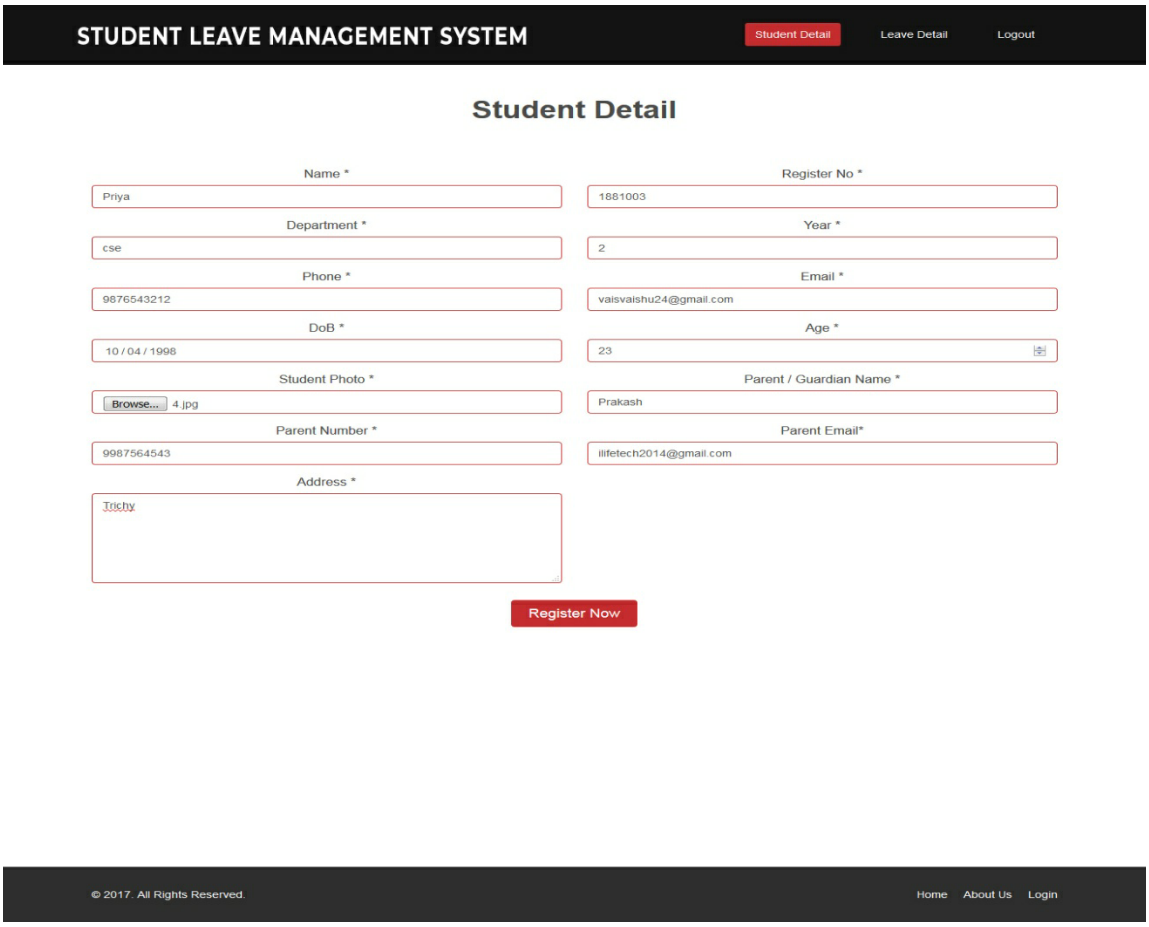
Admin Login

Admin Name *
admin

Password
•••••

© 2017. All Rights Reserved.

Add Student



Student Detail

Name * Priya

Register No * 1881003

Department * cse

Year * 2

Phone * 9876543212

Email * vaisvaishu24@gmail.com

DoB * 10 / 04 / 1998

Age * 23

Student Photo * 4.jpg

Parent / Guardian Name * Prakash

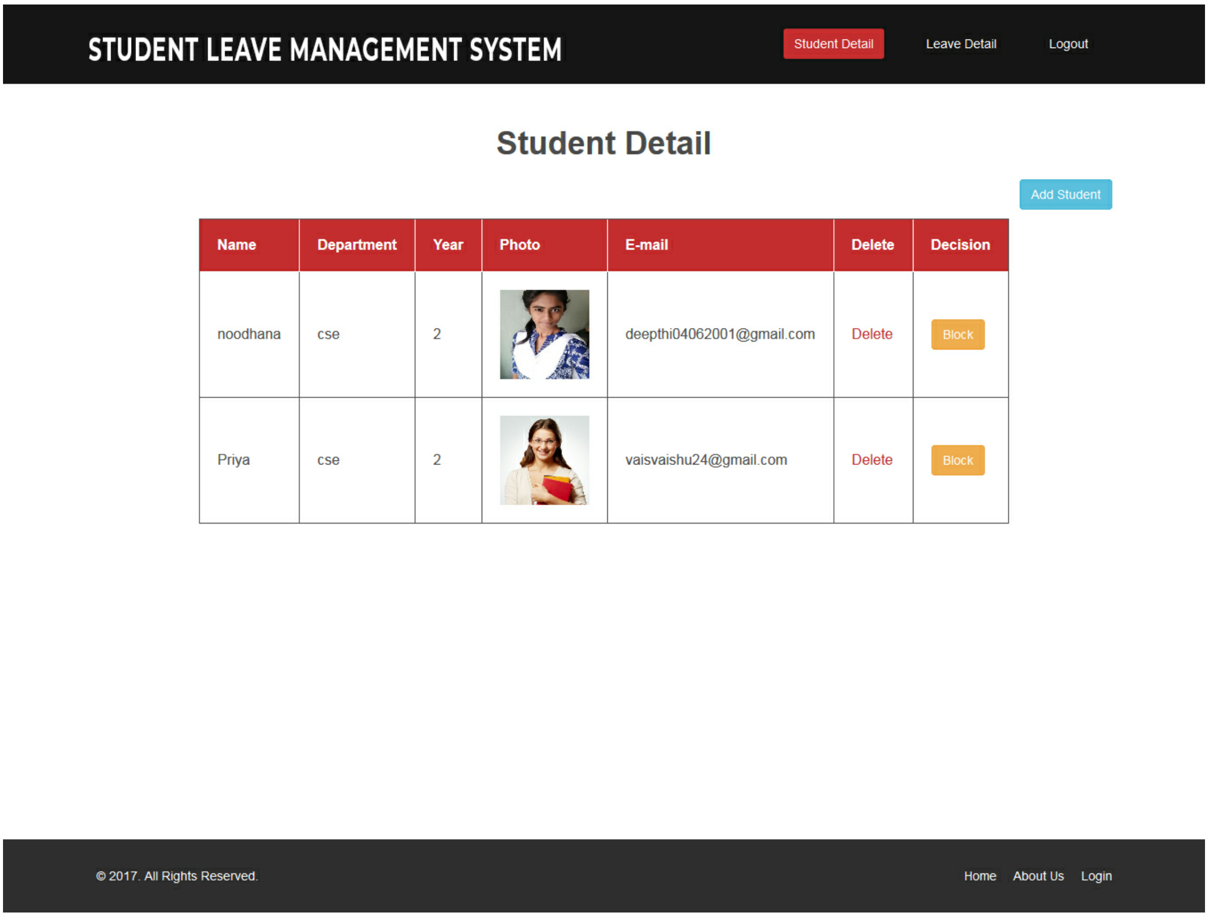
Parent Number * 9987564543

Parent Email* ilifetech2014@gmail.com

Address * Trichy

© 2017. All Rights Reserved.

Student Details



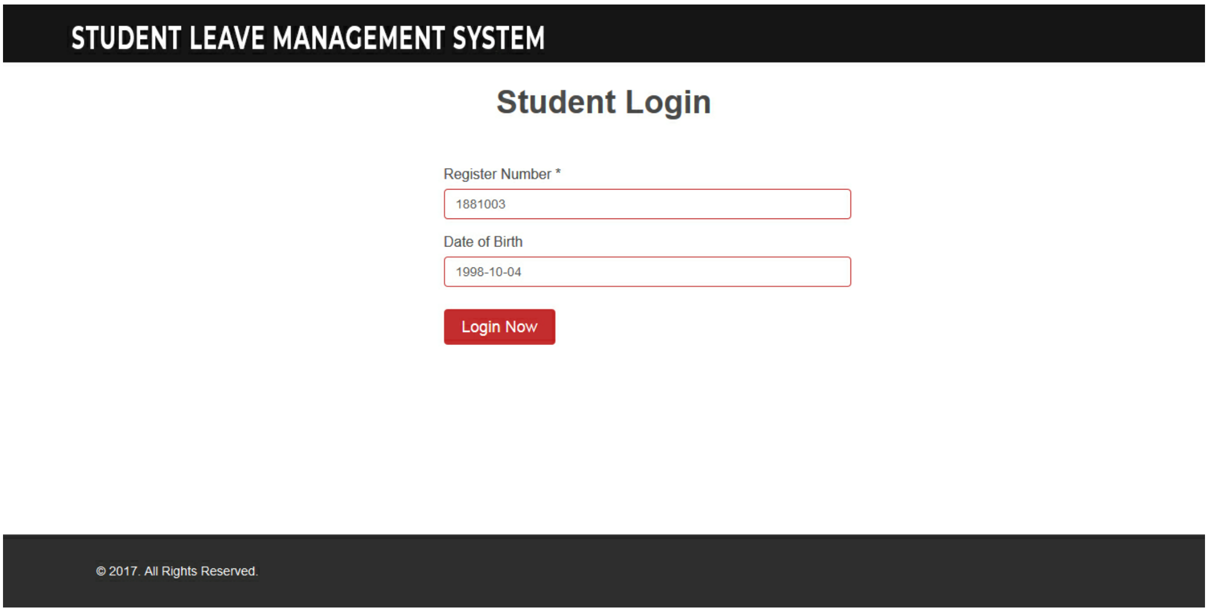
STUDENT LEAVE MANAGEMENT SYSTEM

Student Detail

Name	Department	Year	Photo	E-mail	Delete	Decision
noodhana	cse	2		deepthi04062001@gmail.com	Delete	Block
Priya	cse	2		vaisvaishu24@gmail.com	Delete	Block

© 2017. All Rights Reserved.

Student Login



STUDENT LEAVE MANAGEMENT SYSTEM

Student Login

Register Number *

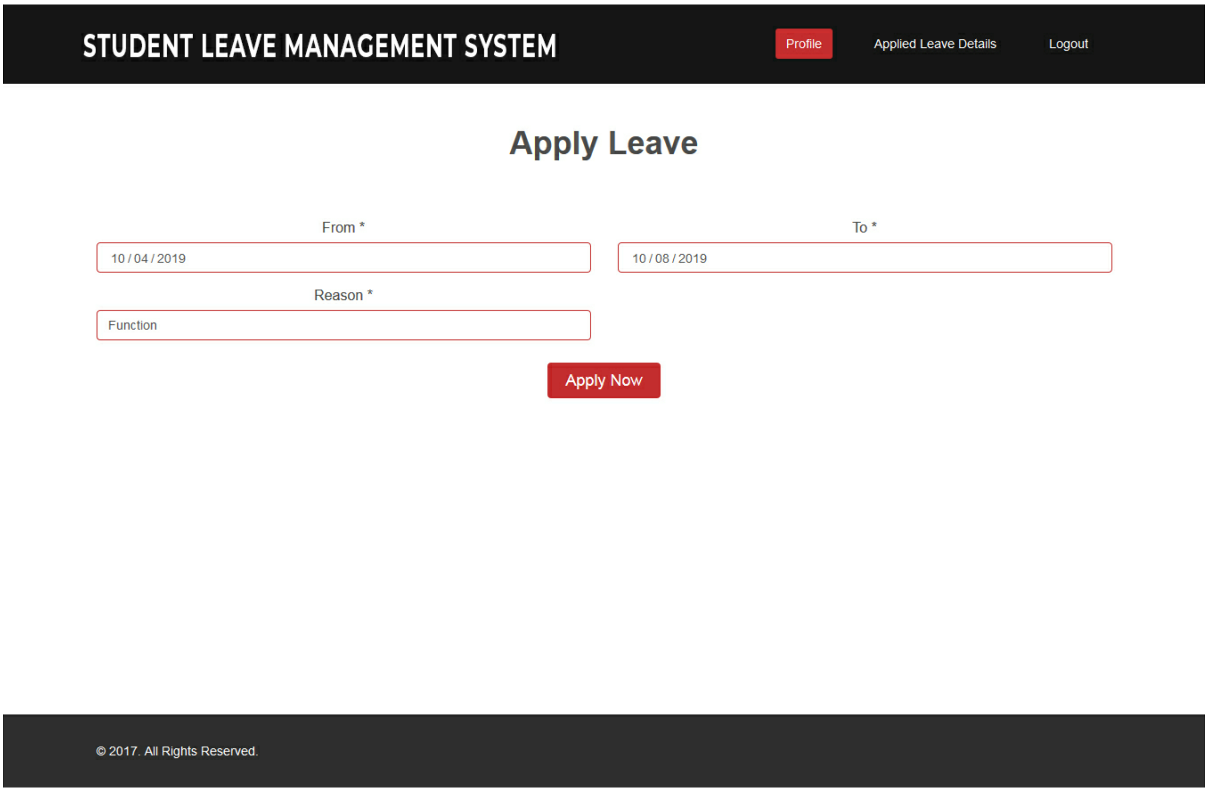
1881003

Date of Birth

1998-10-04

© 2017. All Rights Reserved.

Apply Leave



Apply Leave

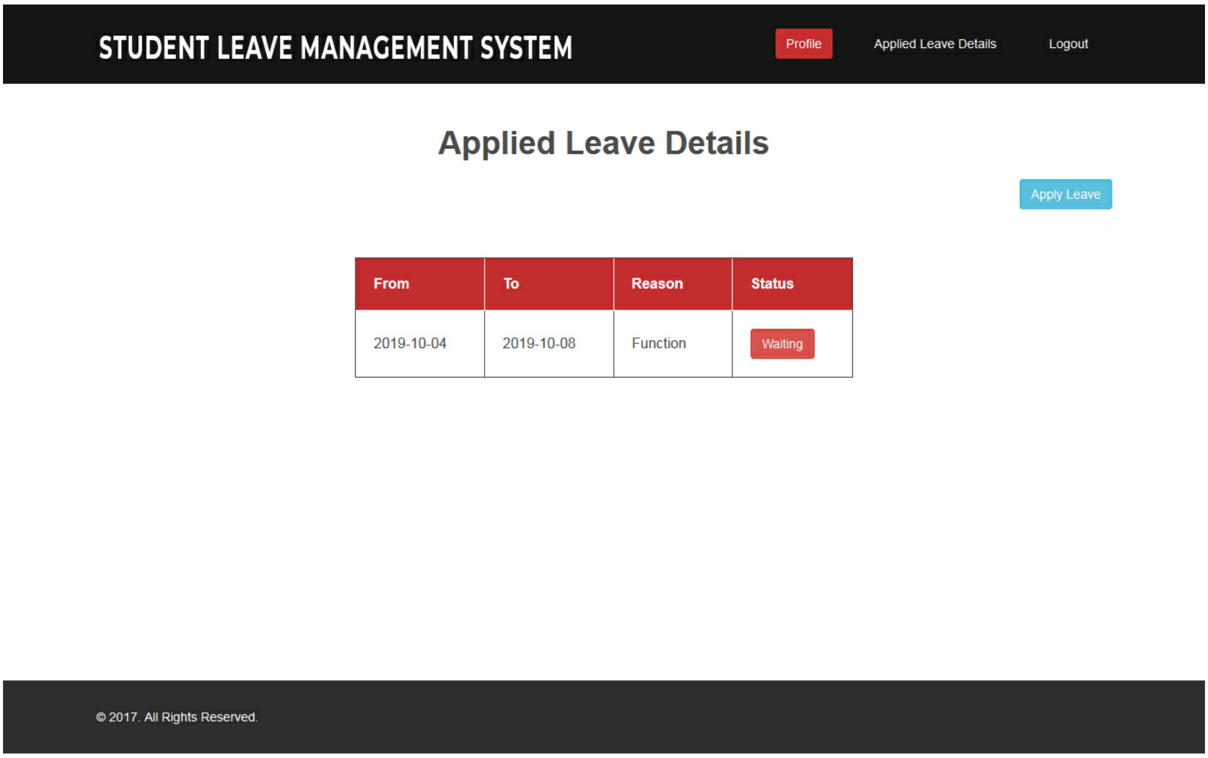
From * 10 / 04 / 2019

To * 10 / 08 / 2019

Reason * Function

© 2017. All Rights Reserved.

Leave Detail

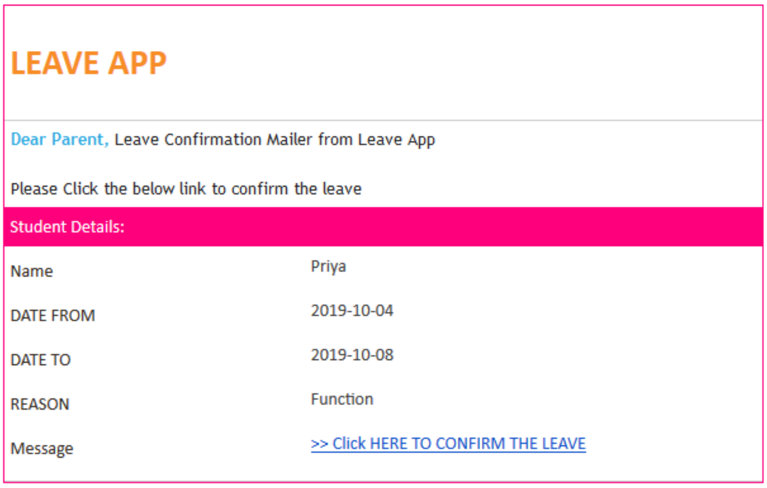


Applied Leave Details

From	To	Reason	Status
2019-10-04	2019-10-08	Function	Waiting

© 2017. All Rights Reserved.

Mail Verification



LEAVE APP

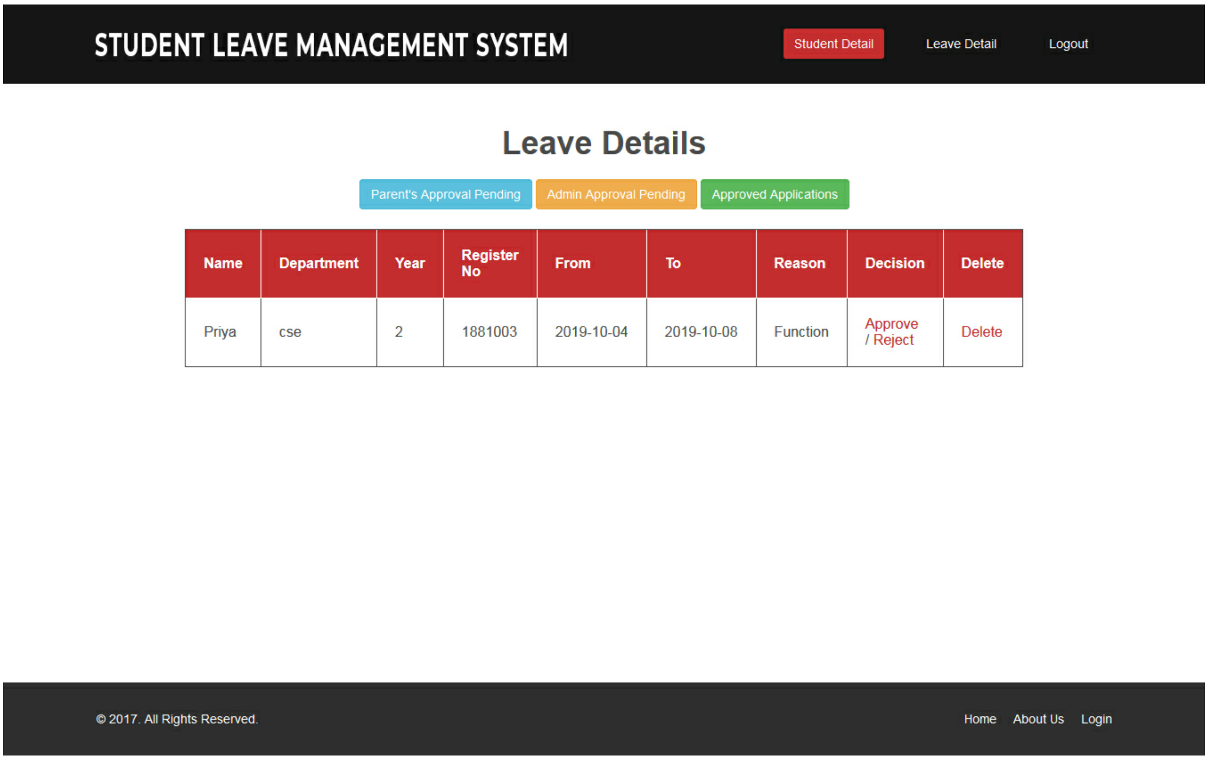
Dear Parent, Leave Confirmation Mailer from Leave App

Please Click the below link to confirm the leave

Student Details:

Name	Priya
DATE FROM	2019-10-04
DATE TO	2019-10-08
REASON	Function
Message	>> Click HERE TO CONFIRM THE LEAVE

After Parent’s Approval



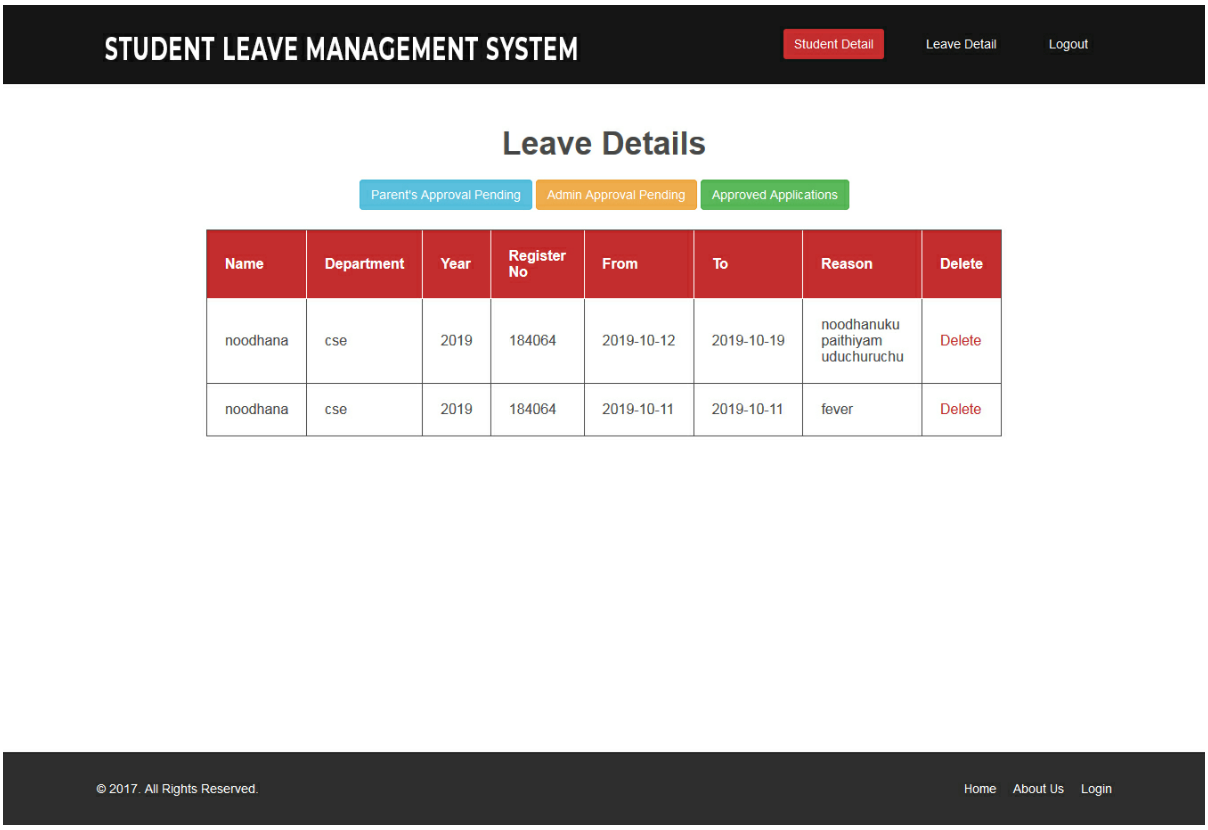
STUDENT LEAVE MANAGEMENT SYSTEM

Leave Details

Name	Department	Year	Register No	From	To	Reason	Decision	Delete
Priya	cse	2	1881003	2019-10-04	2019-10-08	Function	Approve / Reject	Delete

© 2017. All Rights Reserved.

Before Admin Approval

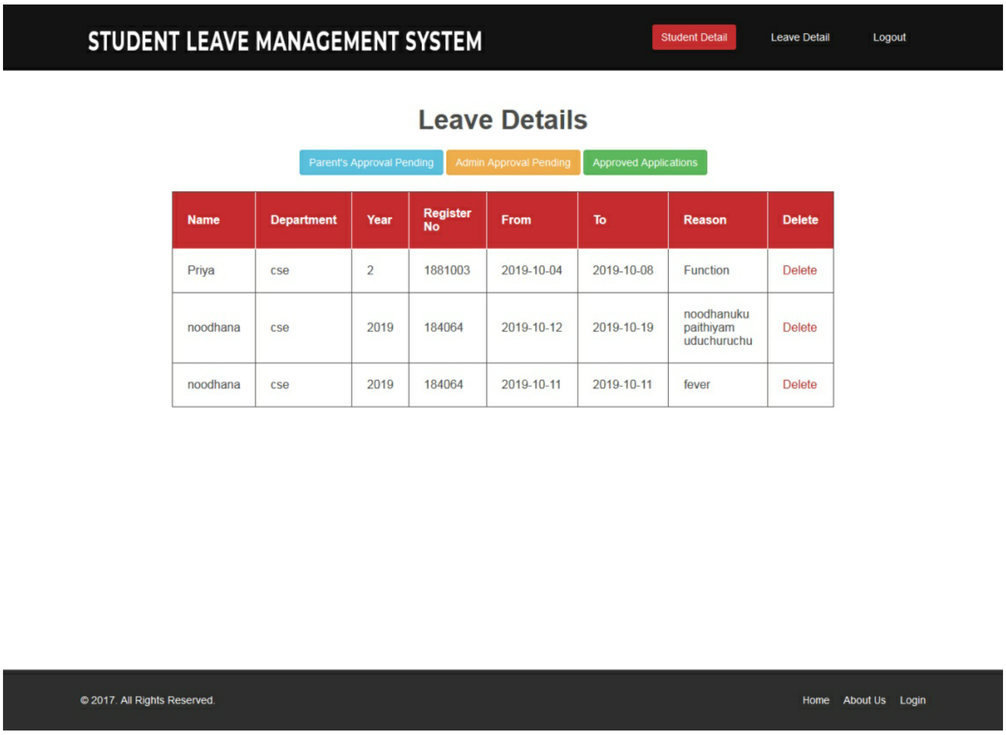


Leave Details

Name	Department	Year	Register No	From	To	Reason	Delete
noodhana	cse	2019	184064	2019-10-12	2019-10-19	noodhanuku paithiyam uduchuruchu	Delete
noodhana	cse	2019	184064	2019-10-11	2019-10-11	fever	Delete

© 2017. All Rights Reserved.

After Admin Approval

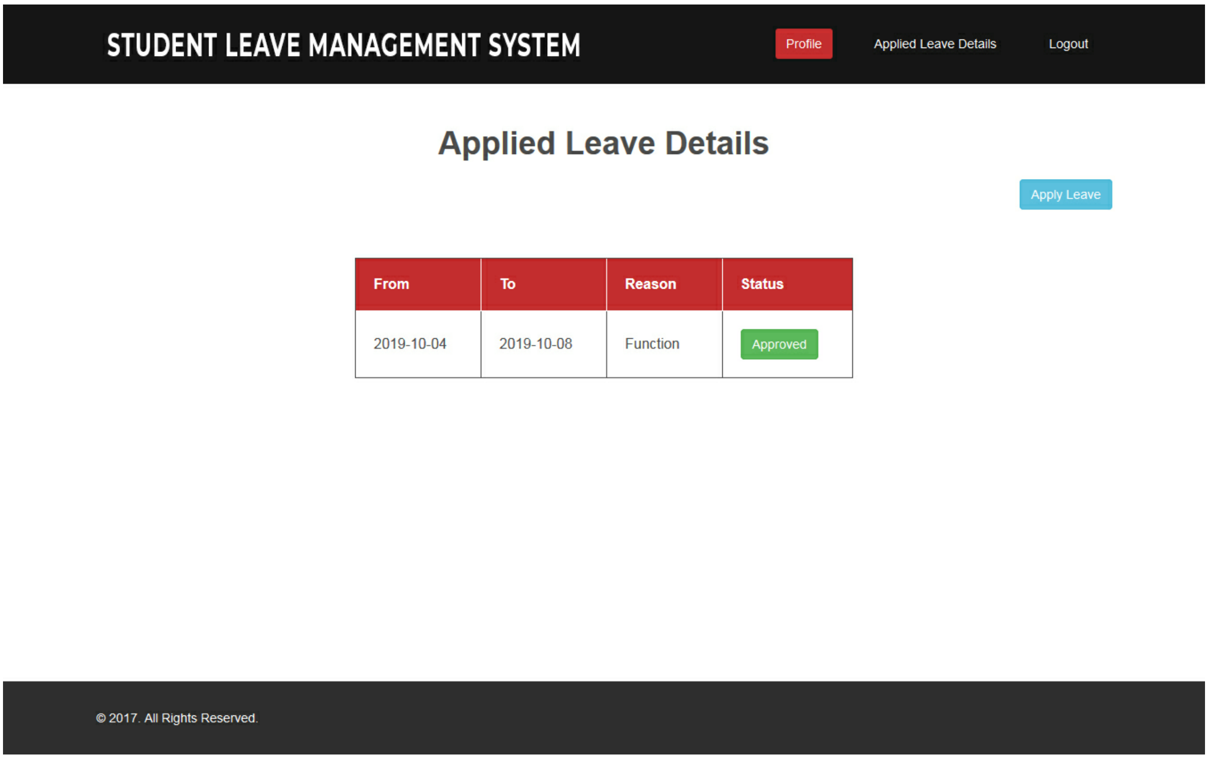


Leave Details

Name	Department	Year	Register No	From	To	Reason	Delete
Priya	cse	2	1881003	2019-10-04	2019-10-08	Function	Delete
noodhana	cse	2019	184064	2019-10-12	2019-10-19	noodhanuku paithiyam uduchuruchu	Delete
noodhana	cse	2019	184064	2019-10-11	2019-10-11	fever	Delete

© 2017. All Rights Reserved.

Student Viewing leave detail after approval

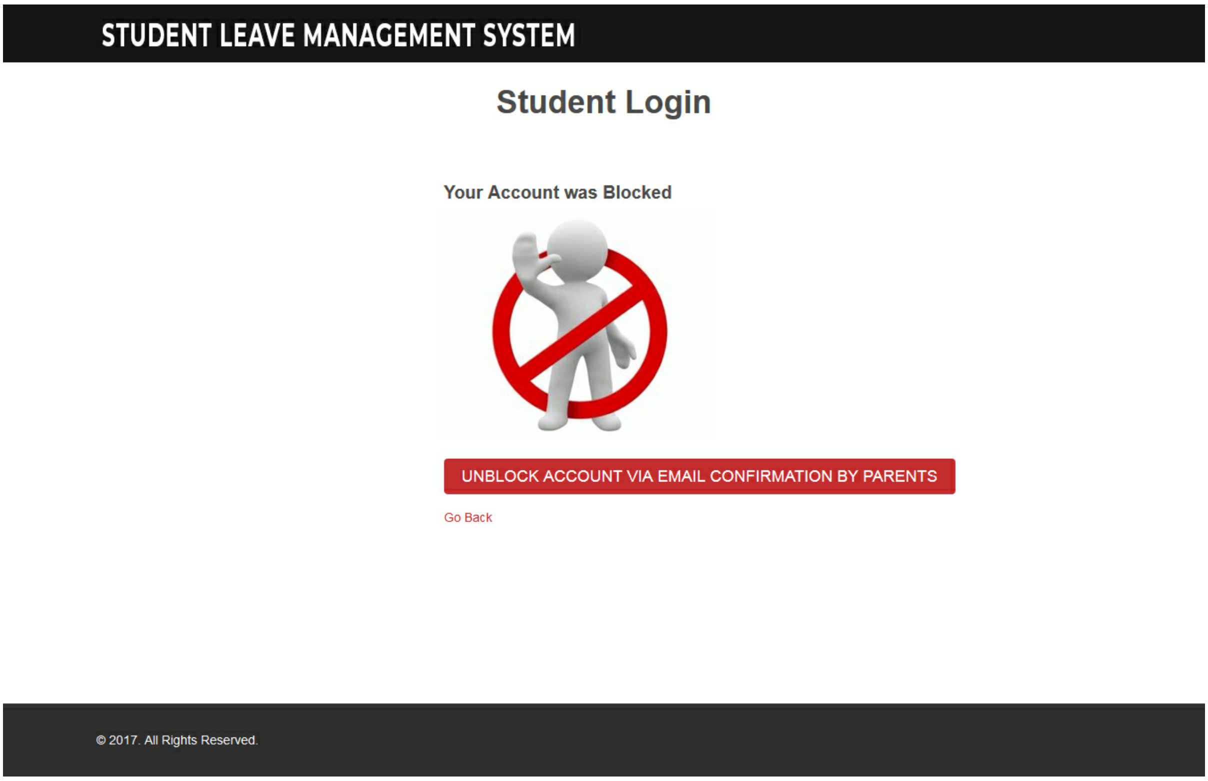


Applied Leave Details

From	To	Reason	Status
2019-10-04	2019-10-08	Function	Approved

© 2017. All Rights Reserved.

Student login after blocking



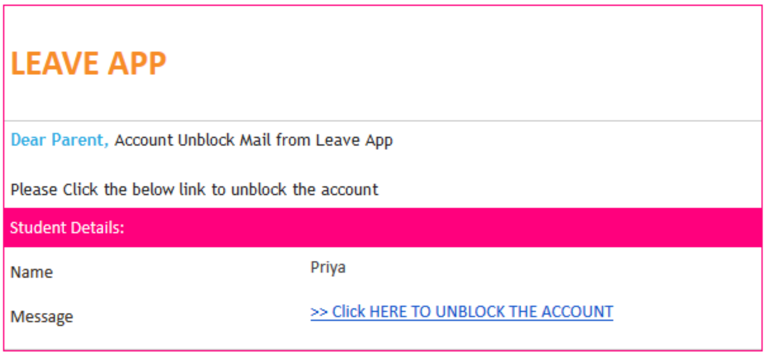
STUDENT LEAVE MANAGEMENT SYSTEM

Student Login

Your Account was Blocked

© 2017. All Rights Reserved.

Unblocking Via email



LEAVE APP

Dear Parent, Account Unblock Mail from Leave App

Please Click the below link to unblock the account

Student Details:

Name	Priya
Message	>> Click HERE TO UNBLOCK THE ACCOUNT

10..CONCLUSION

Online Leave Application for those students in hostel can reduce burden of staffs in the hostel. This projectallows you to manage the leave of the hostel students and also allows you sanction leave request. This is a web application portal that enhances the organization’s hostel to manage and control the students’s leave application. The higher authorities may accept or reject the leave applications requested by the student. And also, student leave details to be known by their parent. Thus this system helps to maintain the leaves taken by the student.

11. BIBLIOGRAPHY

Book Reference

- “Beginning PHP and MySQL”, W.Jason Gilmore, Apress Publication, Third Edition.
- “Learning PHP & MySQL”, Michele E.Davis and John A. Phillips, O’Reilly Publication, Second Edition.
- “PHP and MySQL Web Development”, Luke welling and Laura Thomson, Pearson Education, Fourth Edition.
- Luke Welling & Laura Thompson ,”PHP & MySQL Web Development – by Luke Welling & Laura Thompson” , Developers Library , 4th edition , 2015.
- Robin Nixon,” Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites “ ,O’Reilly, 2nd edition , 2000.
- Davey Shafik , “PHP Master (Paperback)” (Goodreads Author)

published 2011

Website Reference

- www.php.net
- www.wikipedia.com
- www.w3schools.com