**CHAPTER I**

**INTRODUCTION**

**ABOUT THE PROJECT**

The Project “EMPLOYEE MANAGEMENT SYSTEM” is an unofficial try to make a record system for company to store the employee details . The employee management system stores Various details of the employee like Name, Contact details, e-mail id, Address. This project is an eye opener to promote online employee management .

**Need for the Project:**

The following are the essential needs for making this website:

1. During this covid19 pandemic situation few companies are closed and many companies are established the work from home .
2. To make all details of the employees of an company to handle the record easily and also access the record from anywhere and anytime .
3. To promote an automated system to manage the employee details and encourage the user to access the website easily and to manage things more efficiently.
4. To help employee to avoid manual work and reduce workload . This online employee management system is more useful to the work from home employees through online features.

**CHAPTER II**

**SYSTEM ANALYSIS**

**REQUIREMENT ANALYSIS:**

Initially, System Analysis is carried out to find the requirements for this website. Various details were collected from the employee of the company. Lots of areas are covered and data collected regarding employee and works of the employee. The data collected from them are analysed to find essential information for building this project in an excellent manner.

**EXISTING SYSTEM:**

As per the details collected from employee of the company , various records were analysed and the existing system was proposed to have the following drawbacks associated with them. They are:

1. No online employee record management for the company and maintain the records manually .
2. No interface between employee and administrator to know proper information about employees in a company .
3. Lack of awareness of employee of an company to seeing their information are correct or incorrect are not feasible.
4. Employee gives the leave letter by hand written and wait for the administrator and get permission to get the leave.

**PROPOSED SYSTEM:**

The system to be developed must be developed in such a way that it overwhelms the pitfalls identified in the previous system. So, to be more effective, the feasibility study is made.

**FEASIBILITY STUDY**

The system to be developed must be developed in such a way that it overwhelms the pitfalls identified in the previous system. So, to be more effective, the feasibility study is made.

There are various types of feasibility studies carried out for the efficiency of the system. Two of them are very important in the initial stages of the project.

* **Technical Feasibility:**

Technical part of a project is as important as it is going to matter much after automating the system. The software needs to be developed in the right platform with the right software. This technical compatibility alone marks the true success of the project and stands as the difference between the manual and automated system. After careful analysis, the proposal is made to implement the project as a website online.

* **Operational Feasibility**

The Operational part both access and view level decides the output of the project and also the satisfaction of the user who uses the system. The Operational feasibility helps the project to develop user friendly features in the website and can support the development of website in an efficient manner.

**REQUIREMENTS DETERMINATION:**

**HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| HARDWARE | SPECIFICATION |
| MONITOR | 17 “ INCHES |
| KEYBOARD | NUMERIC OR MULTIMEDIA TYPE |
| HARDDISK | MIN 500 MB |
| MOUSE | OPTICAL OR NORMAL |
| RAM | 1 GB OR MORE |

**SOFTWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| COMPONENT | SOFTWARE & TOOLS |
| FRONTEND | HTML5, CSS 3 |
| SERVER-SIDE SCRIPTING | PHP 8.0.2 |
| CLIENT SIDE SCRIPTING | JAVASCRIPT, AJAX |
| BACKEND | MY SQL 5.0.4 |
| EXECUTION SERVER | XAMPP SERVER 8.0.2 |
| IMAGE & VIDEO | PICSART 18.3.2, KINEMASTER 5.2.2 |
| DESIGNING TOOLS | SUBLIME TEXT EDITOR 3.0 |
| BROWSERS | GOOGLE CHROME |

**CHAPTER III**

**SYSTEM DESIGN**

The Mini-Project is used to store the details of the employee and hence it required a user-friendly interface. The design is carefully done and the user interface design is carried out successfully.

**THE USER INTERFACE DESIGN:**

The website is built with template from online websites and manipulated out of it. However, the interface design is created with attractive look so that it captures the customers. The website design is said to have comprised of the following static and dynamic pages:

**STATIC PAGES**

**DYNAMIC PAGES**

**MODULE DESCRIPTION:**

The Project is a unofficial making Employee Record Management system for a company and has the following modules.

1. **Authentication Module**

Authentication Module enables the users to log on to the system. There are two types of users and all of them have a separate login system as selected in the list of values.

1. Administrator
2. Employee
3. **Administrator Module**

Administrator Module enables users to manage all the details of employees and view employees. It also manages various employee project status and salary table and leave status that might be useful for the users.

1. **View Employee Module**

This module contains all the details about Employees, Salary, Contact details and their Address details of the Employee. The Employee maintenance is carried out by the administrator.

1. **My Profile Module**

This module contains all details about Employees Created By the Administrator and the Employee can update their profile( password , contact , address ) Easily. They can also check the other Details ( name, email, department, salary ) Provided by the Administrator .

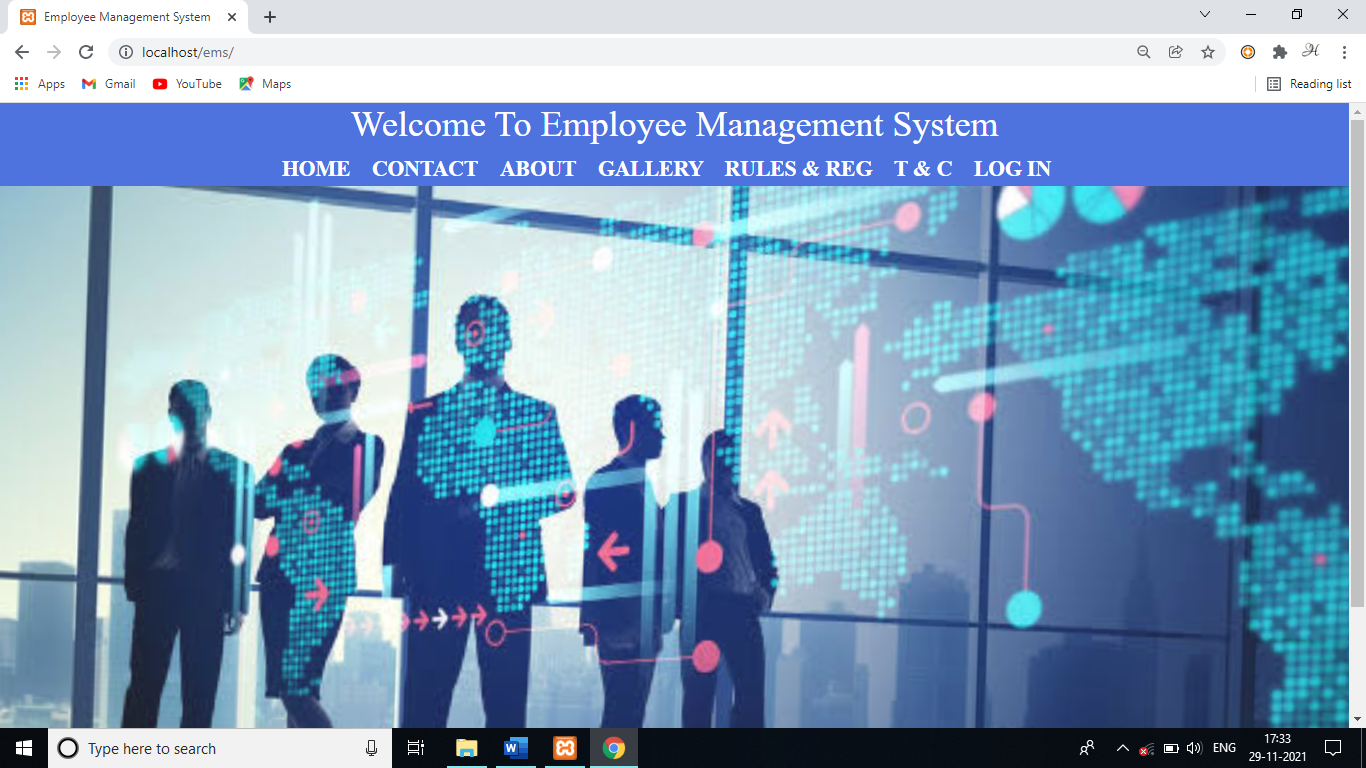
1. **Assign Project Module**

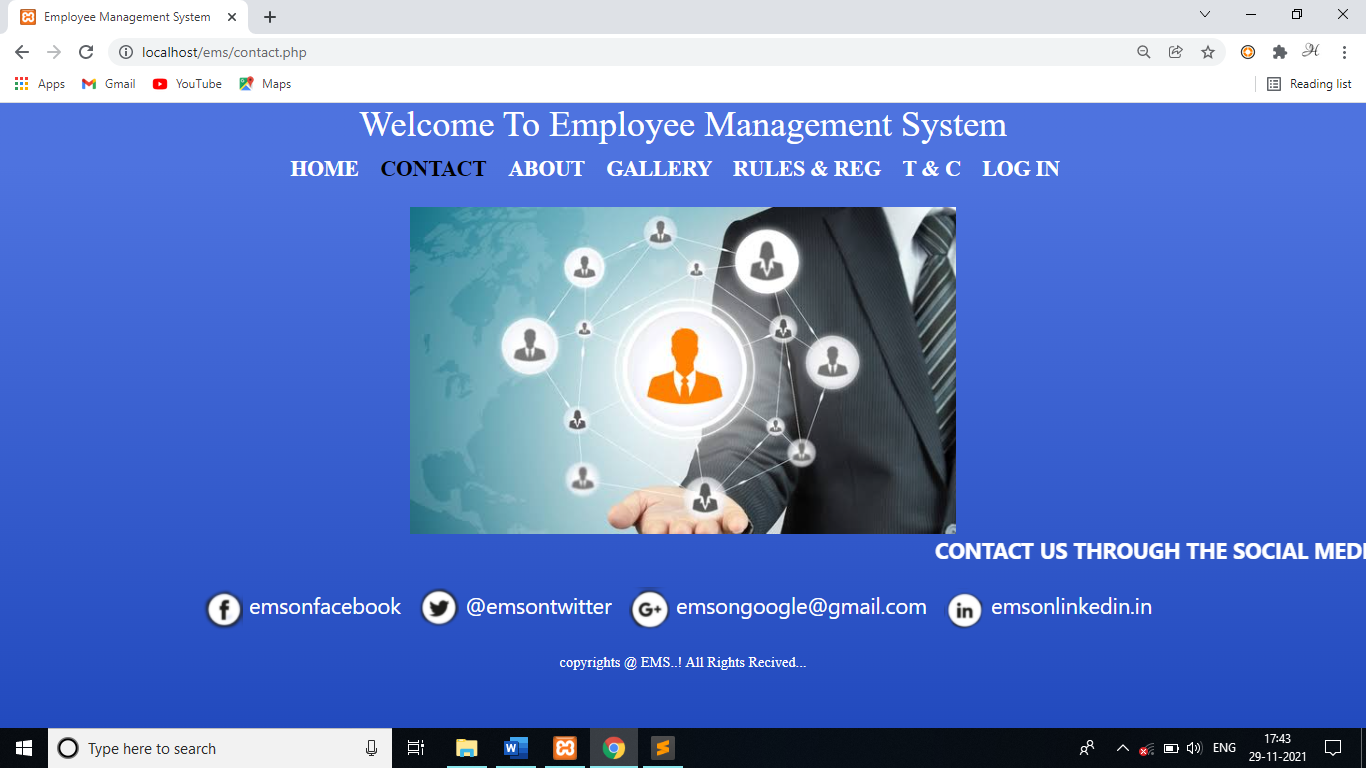
This module used to Admin for assign the project to employee by entering employee id and project name and due date . Admin can check the assigned project status on the project status module . After the employee submitted the project , Admin can add the salary to the particular employee the employee can view the salary status .

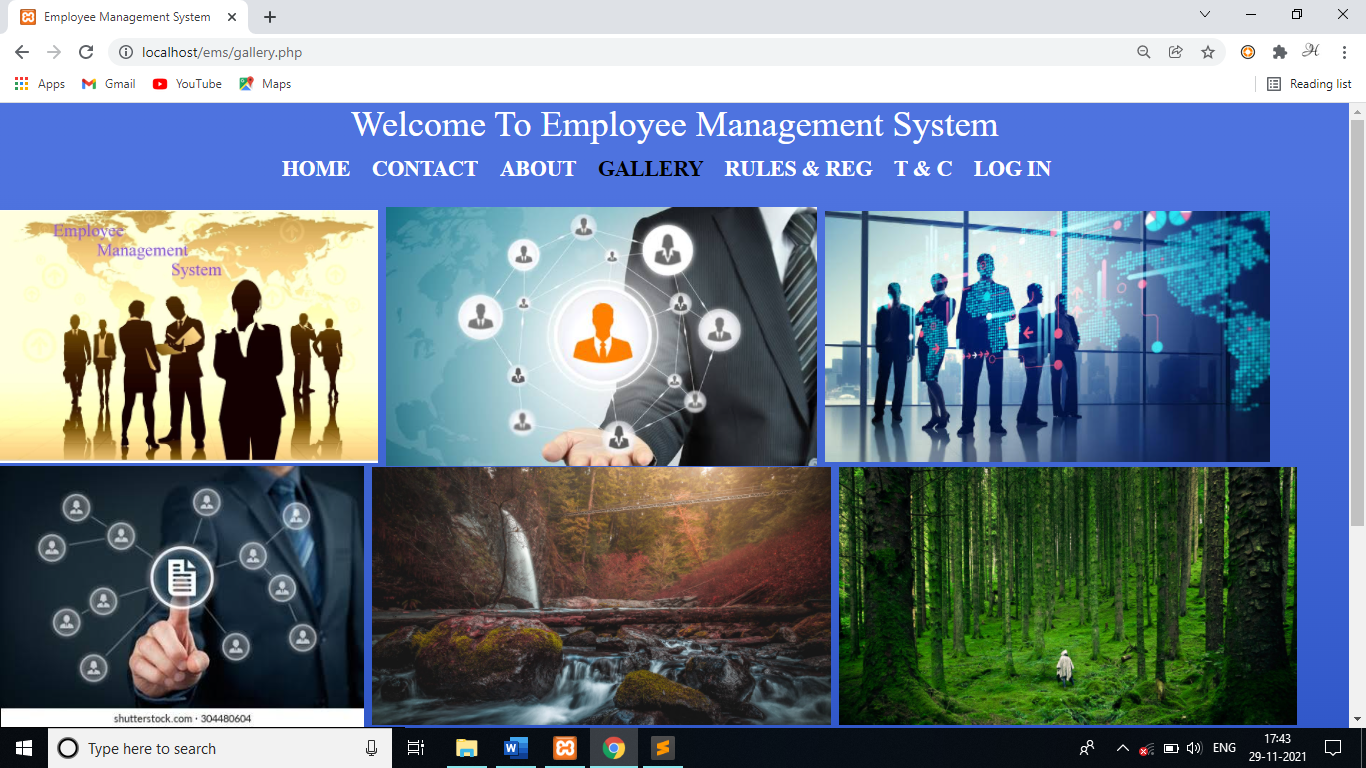
1. **Leave Apply Module**

This module is used to the employee to apply for a leave by providing the reason for getting leave . Admin can Accept or reject the leave ,then the employee can view the status of the leave.

**ORGANISATION OF WEBSITE**





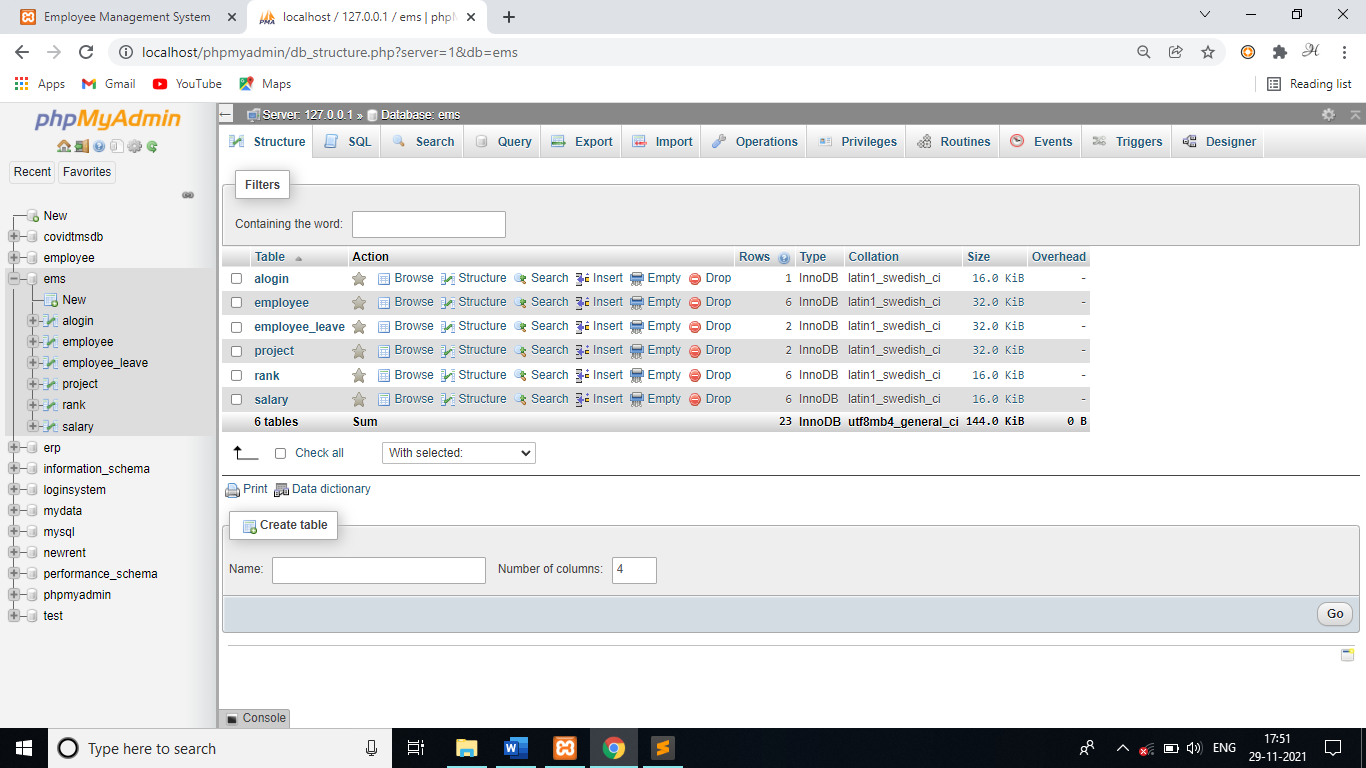




**DATABASE DESIGN**

**DATABASE NAME:** bustick.sql

It contains 6 tables



**TableName: employee**

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Datatype** | **Size** |
| Id | Varchar | 11 |
| firstName | Varchar | 30 |
| lastName | Varchar | 20 |
| Email | Varchar | 20 |
| Password | text |  |
| Birthday | date |  |
| Gender | Varchar | 10 |
| Contact | Varchar | 20 |
| Nid | int | 20 |
| Address | Varchar | 100 |
| Dept | Varchar | 20 |
| Degree | Varchar | 20 |
| Pic | text |  |

**TableName: employee\_leave**

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Datatype** | **Size** |
| Id | int | 10 |
| Token | int | 10 |
| Start | date |  |
| End | date |  |
| Reason | Varchar | 50 |
| Status | Varchar | 20 |

**TableName: project**

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Datatype** | **Size** |
| Pid | int | 10 |
| Eid | int | 10 |
| Pname | Varchar | 20 |
| Duedate | date |  |
| subdate | date |  |
| Mark | int | 10 |
| Status | Varchar | 20 |

**TableName: rank**

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Datatype** | **Size** |
| Eid | int | 10 |
| Points | int | 10 |

**TableName: salary**

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Datatype** | **Size** |
| Id | int | 10 |
| Base | int | 10 |
| Bonus | int | 10 |
| Total | int | 20 |

**CHAPTER IV**

**SYSTEM TESTING**

For any software that is newly developed, primary importance has been given to testing of the system. It is the responsibility of the developer to detect all possible errors in the software before handling it to the user or Customer. Since it is an Open source website it needs intensive testing to provide user friendly infrastructure.

**TESTING:**

Testing is the process by which developer will generate a set of test data, which gives maximum probability of finding all types of errors that can occur in the software. Testing is an important phase in the software development cycle of any system design. Testing indicates the futility of attempting to detect all errors through testing alone. Exhaustive testing of all paths is seldom feasible, and even if it is feasible it does not guarantee detection of missing path errors, computational errors or domain errors. These considerations re-emphasize the need for systematic analysis and design for continuous verification of work products, so that errors are removed prior to implementation.

Apart from Black box testing, which tests the behavioral aspects of the system and white box testing, which tests the functional aspects of the system, there are various testing strategies to carry out software testing at the successful level. They are explained below as follows:

**UNIT TESTING:**

Unit testing or First-Level Testing comprises of the set of tests performed by an individual programmer prior to integration of the Modules or units into a larger system. The system modules login, administrator, hall-manager, customer modules are being tested.

**INTEGRATION TESTING:**

Integration Testing is the systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. There are two approaches for Integration testing. The approach followed in this project is the bottom down approach as it follows object oriented language structure.

**TOP-DOWN INTEGRATION:**

It is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main module. Modules subordinate to the main module control module are incorporated into the structure in either a depth-first or breadth first manner.

**REGRESSION TESTING:**

The Regression testing is also made along with integration testing as whenever any module is added to previous status of the project, the side effects may intend to appear on the system.

**ALPHA TESTING:**

The Alpha testing is conducted at the developer’s site by the customer. As per their need the required modifications are also made successfully.

**CHAPTER V**

**CONCLUSION**

After various analysis, the errors are rectified and it is executed successfully without any bugs in XAMPP server under localhost. The project is an unofficial try from us and it has successfully paid off to be one of the standing model for better output.

**Advantages of the System**

The “Employee Management System” is found to have the following advantages:

* Good communication interface between company Admin and employee of company.
* Promotes good relationship with employee to reduce workload and make employee to feel good .
* Reduce Cost, Time, Space, Workload and reduce time consumption and easy to maintain records.

The Project thus is successfully designed, coded and implemented successfully.

**CHAPTER VI**

**BIBLIOGRAPHY**

**6.1 BOOKS**

1. PHP for absolute beginners – Jason Lengstorf – Apress Publications – Distributed by Springer Online
2. Build Your Own Database driven website using PHP and MYSQL – Kevin Yank – 3rd edition – Sitepoint publications
3. Deitel & Deitel - “Internet and WWW – How to program? Pearson Education, 2005 (Units I, II and III)
4. Core PHP Programming – Leo Atkinson – II Edition – PHI Edition
5. PHP and MySQL – Vikram Vaswami – Mc Graw Hill
6. MySQL/PHP Database Applications – Brad Bulger, Jay Greenspan, David Wall – Second Edition – Wiley Publication.
7. Teach yourself PHP within 24 Hours – SAMS Publication

**6.2 WEBSITES**

1. www.w3schools.com
2. **php**master.com/
3. **php**.net/manual/en/getting-started.**php**
4. www.**learnphp**tutorial.com/
5. www.**learnphp**.in/

**APPENDIX A**

**SOFTWARE DESCRIPTION**

1. **PHP AND MYSQL:**

PHP stands for Hypertext Preprocessor and is a server-side language. This means that the script is run on your web server, not on the user's browser, so you do not need to worry about compatibility issues. PHP is relatively new (compared to languages such as Perl (CGI) and Java) but is quickly becoming one of the most popular scripting languages on the internet. What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

You may be wondering why you should choose PHP over other languages such as Perl or even why you should learn a scripting language at all. I will deal with learning scripting languages first. Learning a scripting language, or even understanding one, can open up huge new possibilities for your website. Although you can download pre-made scripts from sites like Hotscripts, these will often contain advertising for the author or will not do exactly what you want. With an understanding of a scripting language you can easily edit these scripts to do what you want, or even create your own scripts.  
  
Using scripts on your website allows you to add many new 'interactive' features like feedback forms, guestbooks, message boards, counters and even more advanced features like portal systems, content management, advertising managers etc. With these sort of things on your website you will find that it gives a more professional image. As well as this, anyone wanting to work in the site development industry will find that it is much easier to get a job if they know a scripting language.

As mentioned earlier, PHP is a server-side scripting language. This means that, although your users will not need to install new software, you web host will need to have PHP set up on their server. It should be listed as part of your package but if you don't know if it is installed you can find out using the first script in this tutorial. If you server does not support PHP you can ask your web host to install it for you as it is free to download and install. If you need a low cost web host which supports PHP I would recommmend HostRocket.

Writing PHP on your computer is actually very simple. You don't need any specail software, except for a text editor (like Notepad in Windows). Run this and you are ready to write your first PHP script.

The first PHP script you will be writing is very basic. All it will do is print out all the information about PHP on your server. Type the following code into your text editor:

<?  
phpinfo();  
?>  
As you can see this actually just one line of code. It is a standard PHP function called phpinfo which will tell the server to print out a standard table of information giving you information on the setup of the server.

**1.1 PHP WITH HTML FORMS:**

If you know a little HTML, then you know that the FORM tags can be used to interact with your users. Things that can be added to a form are the likes of text boxes, radio buttons, check boxes, drop down lists, text areas, and submit buttons. A basic HTML form with a textbox and a Submit button looks like this:

<html>  
<head>  
<title>A BASIC HTML FORM</title>

</head>  
<body>

<FORM NAME ="form1" **METHOD** =" " **ACTION** = "">

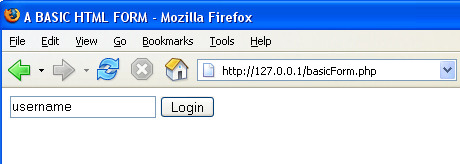
<INPUT TYPE = "TEXT" VALUE ="username">  
<INPUT TYPE = "Submit" Name = "Submit1" VALUE = "Login">

</FORM>  
</body>  
</html>

Some familiarity with the above is assumed. But we'll discuss the **METHOD**, **ACTION** and **SUBMIT** attributes in the form above, because they are important.

The above form can be found in the files you download. It's in the **scripts** folder, and is called **basicForm.php**. Use it as a template, if you like.

So, create the form above. Save your work as **basicForm.php**. (This name will be VERY important!) Start your server, and make sure the form loads ok in your browser. You should be able to see a text box and a Submit button. Here's what it should look like:



If a user comes to your site and has to login, for example, then you'll need to get the details from textboxes. Once you get the text that the user entered, you then test it against a list of your users (this list is usually stored on a database, which we'll see how to code for in a later section). First, you need to know about the HTML attributes METHOD, ACTION and SUBMIT. We'll explore these in the next few sections.

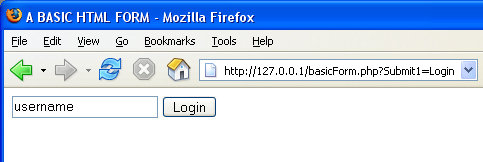
If you look at the first line of our form from the previous page, you'll notice a **METHOD** attribute:

<FORM NAME ="form1" **METHOD =**" " ACTION = "">

The **Method** attribute is used to tell the browser how the form information should be sent. The two most popular methods you can use are GET and POST. But our METHOD is blank. So change it to this:

<FORM NAME ="form1" **METHOD ="GET"** ACTION = "">

To see what effect using GET has, save your work again and then click the Submit button on your form. You should see this:



The thing to notice here is the address bar. After **basicForm.php**, we have the following:

**?Submit1=Login**

This is a consequence of using the GET method. The data from the form ends up in the address bar. You'll see a question mark, followed by form data. In the image above, **Submit1** was the NAME of the button, and **Login** was the VALUE of the button (the text on the button). This is what is being returned by the GET method. You use the GET method when the data you want returned is not crucial information that needs protecting.

**1.3 PHP WITH MYSQL:**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching, and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

So now a days we use relational database management systems (RDBMS) to store and manager huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational DataBase Management System (RDBMS)** is software that:

* Enables you to implement a database with tables, columns, and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

MySQL is a fast, easy-to-use RDBMS used being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

You can establish MySQL database using **mysql** binary at command prompt.

**Example:**

Here is a simple example to connect to MySQL server from command prompt:

|  |
| --- |
| [root@host]# mysql -u root -p  Enter password:\*\*\*\*\*\* |

This will give you mysql> command prompt where you will be able to execute any SQL command. Following is the result of above command:

|  |
| --- |
| Welcome to the MySQL monitor. Commands end with ; or \g.  Your MySQL connection id is 2854760 to server version: 5.0.9  Type 'help;' or '\h' for help. Type '\c' to clear the buffer. |

In above example we have used **root** as a user but you can use any other user. Any user will be able to perform all the SQL operation which are allowed to that user.

You can disconnect from MySQL database any time using **exit** command at mysql> prompt.

|  |
| --- |
| mysql> exit  Bye |
|  |

**MySQL Connection using PHP Script:**

PHP provides **mysql\_connect()** function to open a database connection. This function takes five parameters and returns a MySQL link identifier on success, or FALSE on failure.

**Syntax:**

|  |
| --- |
| connection mysql\_connect(server,user,passwd,new\_link,client\_flag); |

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| server | Optional - The host name running database server. If not specified then default value is **localhost:3036**. |
| user | Optional - The username accessing the database. If not specified then default is the name of the user that owns the server process. |
| passwd | Optional - The password of the user accessing the database. If not specified then default is an empty password. |

You can disconnect from MySQL database anytime using another PHP function **mysql\_close()**. This function takes a single parameter which is a connection returned by **mysql\_connect()** function.

**Syntax:**

|  |
| --- |
| bool mysql\_close ( resource $link\_identifier ); |

If a resource is not specified then last opened database is closed. This function returns true if it closes connection successfully otherwise it returns false.

**Example:**

Try out following example to connect to a MySQL server:

|  |
| --- |
| <html>  <head>  <title>Connecting MySQL Server</title>  </head>  <body>  <?php  $dbhost = 'localhost:3036';  $dbuser = 'guest';  $dbpass = 'guest123';  $conn = mysql\_connect($dbhost, $dbuser, $dbpass);  if(! $conn )  {  die('Could not connect: ' . mysql\_error());  }  echo 'Connected successfully';  mysql\_close($conn);  ?>  </body>  </html> |

**APPENDIX B**

**SAMPLE SOURCE CODE**

**Index.php**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Proper</title>

<meta charset="utf-8">

<link rel="shortcut icon" type="image/x-icon" href="css/images/favicon.ico">

<link rel="stylesheet" href="css/style.css" type="text/css" media="all">

<link rel="stylesheet" href="css/flexslider.css" type="text/css" media="all">

<script src="js/jquery-1.7.2.min.js"></script>

<!--[if lt IE 9]><script src="js/modernizr.custom.js"></script><![endif]-->

<script src="js/jquery.flexslider-min.js"></script>

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

<style type="text/css">

<!--

#wrapper .shell .footer .copy {

color: #000;

}

-->

</style>

</head>

<body>

<!-- wrapper -->

<div id="wrapper">

<!-- shell -->

<div class="shell">

<!-- container -->

<div class="container">

<!-- header -->

<header class="header">

<h1><a href="#">Material Management</a></h1>

<nav id="navigation">

<ul>

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

<li><a href="mentry.php">MaterialEntry</a></li>

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

<li><a href="contact.php">Contact</a></li>

</ul>

</nav>

<div class="cl">&nbsp;</div>

</header>

<!-- end of header -->

<div class="main">

<!-- slider -->

<div class="flexslider">

<ul class="slides">

<li> <img src="css/images/slide-img1.jpg" alt="">

<div class="slide-cnt">

<h2>We Manage Materials<br>

at rapid pace</h2>

<p>We Provide handful support to all products that comes in and out of the company.</p>

</div>

<li> <img src="css/images/slide-img1.jpg" alt="">

<div class="slide-cnt">

<h2>Quality Products Promotion<br>

to Vendors</h2>

<p>We deliver high quality prooducts to vendors at all levels of manufacturing, produciing and vendoor level delivery.</p>

</div>

<li> <img src="css/images/slide-img1.jpg" alt="">

<div class="slide-cnt">

<h2>Regular backups <br>

of transaction</h2>

<p>we offer regular backup oof products and maintain data in an effective manner.</p>

</div>

</li>

</ul>

</div>

<!-- end of slider -->

<!-- cols -->

<section class="cols">

<div class="col">

<h3><a href="#">Developer Tools</a></h3>

<ul>

<li>Dotnet 2010,</li>

<li> Open-Source systems, </li>

<li>Android Applications</li>

<li><img src="images/1436496977691.png" width="128" height="128" alt="gh"></li>

</ul>

</div>

<div class="col">

<h3><a href="#">Other Products</a></h3>

<ul>

<li>ERP Products, </li>

<li>Billing Software </li>

</ul>

<p><img src="images/Analysis-256.png" width="161" height="131" alt="hj"></p>

</div>

<div class="col">

<h3><a href="#">Placement</a></h3>

<p align="justify">We offer placement services to students and help them promote their career effectively.

</p>

<p align="justify"><img src="images/Blue-User-Authentication-Icon.png" width="121" height="102" alt="ty"></p>

</div>

<div class="col">

<h3><a href="#">Consulting</a></h3>

<p>We Promote consultancy services to people around the world</p>

<p><img src="images/company\_team\_icon\_teamwork\_work\_network\_management\_corporate\_organization\_hierarchy-512.png" width="135" height="124" alt="rt"></p>

</div>

<div class="cl">&nbsp;</div>

</section>

<!-- end of cols -->

<!-- box -->

<!-- end of box -->

<!-- services -->

<section class="services">

<div class="widget">

<h3>Crisp, Clean &amp; Creative Design</h3>

<p>Everything is possible when you make a click with your fingers. We have launched new software for commercial purpose and each are pay roll software, E-shopping software, school attendance software, real estate software, invoice software, finger print software etc…</p>

</div>

<div class="widget contact-widget">

<h3>Contact Us</h3>

<ul>

<li><strong>REGISTERED OFFICE</strong><br>

<br>

No.45 Kamaraj Salai, VVP NAGAR, Thantachavadi,<br>

Pondicherry - 605 009</li>

</ul>

</div>

<div class="widget socials-widget">

<h3>Get Social</h3>

<p>Join us at</p>

<a href="#" class="facebook-ico">facebook</a> <a href="#" class="twitter-ico">twitter</a> <a href="#" class="rss-ico">rss</a> <a href="#" class="in-ico">in</a> <a href="#" class="skype-ico">skype</a> <a href="#" class="google-ico">google</a> </div>

<div class="cl">&nbsp;</div>

</section>

<!-- end of services -->

</div>

<!-- end of main -->

</div>

<!-- end of container -->

<div class="footer">

<p class="copy">Copyright &copy; 2016 All Rights Reserved.</p>

</div>

</div>

<!-- end of shell -->

</div>

<!-- end of wrappert -->

</body>

</html>

**Reserve.php**

<!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>

<title>BusTick</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

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

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/droid\_sans\_400-droid\_sans\_700.font.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.main .header .header\_resize .logo h6 {

color: #004080;

}

.main .header .header\_resize .logo h2 {

color: #004080;

}

.main .header .header\_resize .logo h3 strong {

color: #004080;

}

.fbg .fbg\_resize .clr form table tr td {

color: #000040;

}

.fbg .fbg\_resize div form table tr td {

color: #000040;

}

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header\_resize">

<div class="logo">

<h1><a href="index.html">BUS TICKET RESERVATION</a></h1>

<h3><strong>CUDDALORE BUS STAND</strong></h3>

</div>

<div class="searchform">

<h2><a href="index.php">Signout </a></h2>

</div>

<div class="clr"></div>

<div class="menu\_nav">

<ul>

<li class="active"><a href="cushome.php"><span>Home</span></a></li>

<li><a href="vbus.php"><span>VIEW BUS</span></a></li>

<li><a href="vroute.php"><span>VIEW ROUTE</span></a></li>

<li><a href="reserve.php"><span>RESERVE</span></a></li>

<li><a href="savail.php"><span>SEAT AVAILABILITY</span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="clr"></div>

</div>

</div>

<div class="fbg">

<div class="fbg\_resize">

<div >

<p>&nbsp;</p>

<?php

if(isset($\_POST['t1']))

{

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("bustick");

$a=$\_POST["t1"];

$b=$\_POST["t2"];

$c=$\_POST["t3"];

$d=$\_POST["t4"];

$e=$\_POST["t5"];

$f=$\_POST["t6"];

$g=$\_POST["t7"];

$h=$\_POST["t8"];

$i=$\_POST["t9"];

$j=$\_POST["t10"];

$sql = "INSERT INTO tickbook VALUES('$a','$b','$c','$d','$e','$f','$g','$h','$i','$j')";

$result = mysql\_query($sql);

$msg="Thank you for Booking in our bus";

}

?>

<form action="reserve.php" method="post" name="f1">

<table width="635" height="337" border="3" align="center">

<tr>

<td width="231" height="34"><strong>Enter the Ticket ID</strong></td>

<td width="375"><label>

<input type="text" name="t1" id="t1" />

</label></td>

</tr>

<tr>

<td height="42"><strong>Enter the Bus Code</strong></td>

<td><label>

<select name="t2" id="t2">

<?php

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("bustick");

$sql = mysql\_query("SELECT distinct(busid) FROM busdet");

while($row = mysql\_fetch\_array($sql))

{

echo ' <option>';

echo $row['busid'];

echo '</option>';

}

?>

</select>

</label></td>

</tr>

<tr>

<td height="36"><strong>Enter the Customer Name</strong></td>

<td><label>

<input type="text" name="t3" id="t3" />

</label></td>

</tr>

<tr>

<td><strong>Enter the Route</strong></td>

<td><label>

<select name="t4" id="t4">

<?php

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("bustick");

$sql = mysql\_query("SELECT \* FROM routedet");

while($row = mysql\_fetch\_array($sql))

{

echo ' <option>';

echo $row['timep'];

echo '</option>';

}

?>

</select>

</label></td>

</tr>

<tr>

<td><strong>Enter the Starting Point</strong></td>

<td><label>

<select name="t5" id="t5">

<?php

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("bustick");

$sql = mysql\_query("SELECT distinct(startp) FROM routedet");

while($row = mysql\_fetch\_array($sql))

{

echo ' <option>';

echo $row['startp'];

echo '</option>';

}

?>

</select>

</label></td>

</tr>

<tr>

<td><strong>Enter the Booking Date</strong></td>

<td><label>

<input type="text" name="t6" id="t6" />

</label></td>

</tr>

<tr>

<td><strong>Enter destination</strong></td>

<td><label>

<select name="t7" id="t7">

<?php

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("bustick");

$sql = mysql\_query("SELECT distinct(endp) FROM routedet");

while($row = mysql\_fetch\_array($sql))

{

echo ' <option>';

echo $row['endp'];

echo '</option>';

}

?>

</select>

</label></td>

</tr>

<tr>

<td><strong>Enter Number of Seats</strong></td>

<td><label>

<input type="text" name="t8" id="t8" />

</label></td>

</tr>

<tr>

<td height="30"><strong>Enter Seat Number</strong></td>

<td><label>

<input type="text" name="t9" id="t9" />

</label></td>

</tr>

<tr>

<td><strong>Enter Status</strong></td>

<td><label>

<select name="t10" id="t10">

<option>reserved</option>

<option>unreserved</option>

</select>

</label></td>

</tr>

<tr>

<td>&nbsp;</td>

<td><label>

<input type="submit" name="b1" id="b1" value="Submit" />

<input type="reset" name="b2" id="b2" value="Reset" />

</label></td>

</tr>

<tr>

<td>&nbsp;</td>

<td><p>&nbsp;</p>

<p><?php if(isset($\_POST['t1'])) echo $msg; ?></p></td>

</tr>

</table></form>

<p>&nbsp;</p>

</div>

</div>

</div>

<div class="footer">

<div class="footer\_resize">

Guided by : Mr. Anand Christy

Project Work by : Bhavani-Ramya

<div style="clear:both;"></div>

</div>

</div>

</div>

</body>

</html>

savail.php

<form id="form1" name="form1" method="post" action="savail.php">

<p>&nbsp;</p>

<p>Enter the Bus ID

<label>

<select name="t1" id="t1">

<?php

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("bustick");

$sql = mysql\_query("SELECT distinct(busid) FROM busdet");

while($row = mysql\_fetch\_array($sql))

{

echo ' <option>';

echo $row['busid'];

echo '</option>';

}

?>

</select>

Enter the Date

<input type="text" name="t2" id="t2" />

</label>

<label>

<input type="submit" name="b1" id="b1" value="Submit" />

</label>

</p>

<table width="936" height="37" border="1">

<tr align="center">

<td width="159"><h5><strong>STARTING POINT</strong></h5></td>

<td width="162"><h5><strong>ENDING POINT</strong></h5></td>

<td width="122"><h5><strong>ROUTE</strong></h5></td>

<td width="132"><h5><strong>NO OF SEATS</strong></h5></td>

<td width="144"><h5><strong>SEAT NUMBERS</strong></h5></td>

<td width="177"><h5><strong>STATUS</strong></h5></td>

</tr>

<?php

if(isset($\_POST['t1']) && isset($\_POST['t2']))

{

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "bustick";

$a=$\_POST['t1'];

$b=$\_POST['t2'];

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

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

}

$sql = "SELECT \* FROM tickbook where bcode='".$a."' and bdate='".$b."'";

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

if ($result->num\_rows > 0) {

// output data of each row

while($row = $result->fetch\_assoc()) {

echo "<tR><TD>".$row["spoint"]."</TD>". "<TD>".$row["dest"]."</TD>"."<TD>".$row["route"]."</TD>"."<TD>".$row["noseats"]."</TD>"."<TD>".$row["seatno"]."</TD>"."<TD>".$row["status"]."</TD></tr>"."<br>";

}

} else {

echo "0 results";

}

$conn->close();

}

?>

</table>

</form>

Vbus.php

<form action="vbus.php" method="post" name="f1">

<table width="426" height="117" border="1" align="center">

<tr>

<td width="158">Enter Starting Point</td>

<td width="441"><label>

<select name="t1" id="t1">

<option>cuddalore</option>

<option>villupuram</option>

<option>chidambaram</option>

<option>pondicherry</option>

</select>

</label></td>

</tr>

<tr>

<td>Enter Destination</td>

<td><label>

<select name="t2" id="t2">

<option>cuddalore</option>

<option>chidambaram</option>

<option>villupuram</option>

<option>pondicherry</option>

</select>

</label></td>

</tr>

<tr>

<td>&nbsp;</td>

<td><label>

<input type="submit" name="b1" id="b1" value="Submit" />

</label></td>

</tr>

</table>

<div>

<h3>View Bus details </h3>

<table width="811" height="37" border="1">

<tr class="heade">

<td><h5><strong>BUS ID</strong></h5></td>

<td><h5><strong>CORPORATION NAME</strong></h5></td>

<td><h5><strong>TIMINGS</strong></h5></td>

<td><h5><strong>NO.OF SEATS</strong></h5></td>

</tr>

<?php

if(isset($\_POST['t1']))

{

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "bustick";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

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

}

$sql = "SELECT \* FROM busdet where startp='".$\_POST["t1"]."' and endp='".$\_POST["t2"]."'";

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

if ($result->num\_rows > 0) {

// output data of each row

while($row = $result->fetch\_assoc()) {

echo "<tR><TD>".$row["busid"]."</TD>". "<TD>".$row["corp"]."</TD>"."<TD>".$row["timeg"]."</TD>"."<TD>".$row["totseats"]."</TD></tr>"."<br>";

}

} else {

echo "0 results";

}

$conn->close();

}

?>

</table>

Login.php

<?php

if(isset($\_POST['t1']) && isset($\_POST['t2']))

{

$uname=$\_POST['t1'];

$pword=$\_POST['t2'];

$tous=$\_POST['t3'];

mysql\_connect("localhost","root","");

mysql\_select\_db("bustick");

if($tous=="admin")

{

$q1="select \* from login where uname='".$uname."' and pword='".$pword."'" ;

$b1=mysql\_query($q1) or

die ("Invalid Username");

if(mysql\_num\_rows($b1)>0)

{

header('location: http://localhost/BUSTICKET/admhome.php');

}

}

if($tous=="customer")

{

$q2="select \* from login where uname='".$uname."' and pword='".$pword."'";

$b2=mysql\_query($q2) or

die ("Invalid Username");

if(mysql\_num\_rows($b2)>0)

{

header('location: http://localhost/BUSTICKET/cushome.php');

}

}

}

?>

**APPENDIX 3**

**SCREENSHOTS**

