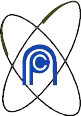
TAPAPUR ATOMIC POWER STATION 1&2



**Introduction to web page development**

**Using PHP**

Prepared by:

UMAIR AKHTAR

DEBYANI MITRA

SHWETA BARANWAL

3rd Year, COMPUTER ENGINEERING

MUMBAI UNIVERSITY

Under the guidance of

Mr. P. N. KAPOOR, SO/E

**CERTIFICATE**

This is to certify that Ms. DEBYANI MITRA, SHWETA BARANWAL & Mr. UMAIR AKHTAR 3rd year BE (Computer Engineering) SJCET College of Engineering, Palghar & Terna College of Engineering, Nerul has successfully completed the vocational implant training at TARAPUR ATOMIC POWER STATION from 10.12.2014 to 30.12.2014. During this period we studied and made a report on “Web Site Development”.

(P. N. Kapoor) (KVSN Murthy)

ME (I) SME (I)

SHRI S. K. Gupta

(Training Superintendent)

**ACKNOWLEDGEMENTS**

In an endeavor to study the intricacies of this project and at various stages when I was in need of guidance, I am grateful to all who helped me and wish to thank them.

I express my sincere gratitude towards my Project Guide Mr. Pankaj N. Kapoor, who guided me, instilled self-confidence within me and provided motivation throughout. I would also like to thank my Training Superintendent Mr. S.k.Gupta, my Course Co-ordinator Mr.K.V.S.N.Murthy, Ms. Smrutigandha Lade.

I extend my gratitude towards Nuclear Section staff for their timely and whole hearted support in the completion of this project.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR NO.** | **TABLE OF CONTENT** | **PG.NO.** |
| **1.0**  **1.1**  **1.2**  **1.3**  **1.4**  **1.5** | **Introduction of PHP**  **PHP Example**  **History of PHP**  **Data Types**  **Uses**  **Functions** | **5**  **6**  **6**  **8**  **9**  **10** |
| **2.0**  **2.1** | **What is AJAX?**  **What is Xampp Server?** | **11**  **11** |
| **3.0**  **3.1**  **3.2** | **What is SQL?**  **History**  **Queries** | **13**  **13**  **15** |
| **4.0** | **Jquery** | **16** |
| **5.0** | **Javascript** | **17** |
| **6.0**  **6.1** | **Source code**  **Screen shot** | **19**  **56** |
| **7.0** | **Conclusion** | **59** |
| **8.0** | **References** | **60** |

**1.0 Introduction to Php**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers Originally created by RasmusLerdorf in 1994, the reference implementation of PHP (powered by the Zend Engine) is now produced by The PHP Group.[ While PHP originally stood for Personal Home Page,it now stands for PHP: Hypertext Preprocessor, which is a recursive backronym.

PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page – for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

The canonical PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

Despite its popularity, no written specification or standard existed for the PHP language until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014, there is ongoing work on creating a formal PHP specification.

**1.1 An Introductory Example**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
    "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
    <head>  
        <title>Example</title>  
    </head>  
    <body>  
        <?php  
            echo "Hi, I'm a PHP script!";  
        ?>  
    </body>  
</html>

**1.2 History**

PHP development began in 1994 when RasmusLerdorf wrote a series of Common Gateway Interface (CGI) binaries in C, which he used to maintain his personal homepage. He extended them to add the ability to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could be used to build simple, dynamic web applications. Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" publicly to accelerate bug location and improve the code, on the Usenet discussion group comp.infosystems.www.authoring.cgi on June 8, 1995. This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.

Early PHP was not intended to be a new programming language, and grew organically, with Lerdorf. A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

One criticism of PHP is that it was not originally designed, but instead it was developed organically;among other things, this has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping",while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

ZeevSuraski and AndiGutmans rewrote the parser in 1997 and formed the base of PHP 3, changing the language's name to the recursive acronym PHP: Hypertext Preprocessor.Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP's core, producing the Zend Engine in 1999. They also founded Zend Technologies in Ramat Gan, Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released. As of August 2008 this branch reached version 4.4.9. PHP 4 is no longer under development nor will any security updates be released.

On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. In 2008 PHP 5 became the only stable version under development. Late static binding had been missing from PHP and was added in version 5.3.

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.

Over time, PHP interpreters became available on most existing 32-bit and 64-bit operating systems, either by building them from the PHP source code, or by using pre-built binaries. For the PHP versions 5.3 and 5.4, the only available Microsoft Windows binary distributions were 32-bit x86 builds,requiring Windows 32-bit compatibility mode while using Internet Information Services (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64-bit x86-64 builds available for Microsoft Windows.

**1.3 Data Types**

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit [signed](http://en.wikipedia.org/wiki/Signed_number_representations) [integer](http://en.wikipedia.org/wiki/Integer_(computer_science)) equivalent to the [C-language long type](http://en.wikipedia.org/wiki/C_variable_types_and_declarations). Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), [octal](http://en.wikipedia.org/wiki/Octal), [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal), and [binary](http://en.wikipedia.org/wiki/Binary_code) notations.

[Floating point](http://en.wikipedia.org/wiki/Floating_point) numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of [scientific notation](http://en.wikipedia.org/wiki/Scientific_notation). PHP has a native [Boolean](http://en.wikipedia.org/wiki/Boolean_datatype) type that is similar to the native Boolean types in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in [Perl](http://en.wikipedia.org/wiki/Perl) and C++.

The null data type represents a variable that has no value; *NULL* is the only allowed value for this data type.

Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources.

Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in [hashes](http://en.wikipedia.org/wiki/Hash_table)with both keys and values, and the two can be intermingled. PHP also supports [strings](http://en.wikipedia.org/wiki/String_(computing)), which can be used with single quotes, double quotes, nowdoc or [heredoc](http://en.wikipedia.org/wiki/Heredoc)syntax.

The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.

**1.4 Uses**

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

There are three main areas where PHP scripts are used.

1. Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming. See the installation instructions section for more information.
2. Command line scripting. You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on \*nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks. See the section about Command line usage of PHP for more information.
3. Writing desktop applications. PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution.

**1.5 Functions**

PHP has hundreds of functions provided by the core language functionality and thousands more available via various extensions; these functions are well documented in the online PHP documentation. Additional functions can be defined by the developer:

In PHP, normal functions are not first-class and can only be referenced by their name directly or dynamically by a variable containing the name of the function (referred to as "variable functions"). User-defined functions can be created at any time without being prototyped.Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined. Function calls must use parentheses, with the exception of zero-argument class constructor functions called with the PHP new operator, where parentheses are optional.

Until PHP 5.3, support for true anonymous functions or closures did not exist in PHP. While create\_function() exists since PHP 4.0.1, it is merely a thin wrapper around eval() that allows normal PHP functions to be created during program execution. Also, support for variable functions allows normal PHP functions to be used, for example, as callbacks or within function tables. PHP 5.3 added support for closures, which are true anonymous, first-class functions.

## **2.0 What is AJAX ?**

* AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS and Java Script.
* Conventional web application trasmit information to and from the sever using synchronous requests. This means you fill out a form, hit submit, and get directed to a new page with new information from the server.
* With AJAX when submit is pressed, JavaScript will make a request to the server, interpret the results and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

**2.1 XAMPP**

Xamppis a [free and open source](http://en.wikipedia.org/wiki/Free_software) [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [web server](http://en.wikipedia.org/wiki/Web_server) [solution stack](http://en.wikipedia.org/wiki/Solution_stack) package, consisting mainly of the [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server), [MySQL](http://en.wikipedia.org/wiki/MySQL) [database](http://en.wikipedia.org/wiki/Database), and [interpreters](http://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](http://en.wikipedia.org/wiki/PHP)and [Perl](http://en.wikipedia.org/wiki/Perl) [programming languages](http://en.wikipedia.org/wiki/Programming_language).

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.[[2]](http://en.wikipedia.org/wiki/XAMPP#cite_note-2) In practice, however, XAMPP is sometimes used to actually serve web pages on the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web). A special tool is provided to [password-protect](http://en.wikipedia.org/wiki/Password) the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in [MySQL](http://en.wikipedia.org/wiki/MySQL) and [SQLite](http://en.wikipedia.org/wiki/SQLite) among others.

Once XAMPP is installed, it is possible to treat a [localhost](http://en.wikipedia.org/wiki/Localhost) like a remote host by connecting using an [FTP](http://en.wikipedia.org/wiki/File_Transfer_Protocol) client. Using a program like [FileZilla](http://en.wikipedia.org/wiki/FileZilla) has many advantages when installing a [content management system](http://en.wikipedia.org/wiki/Content_management_system) (CMS) like [Joomla](http://en.wikipedia.org/wiki/Joomla) or [WordPress](http://en.wikipedia.org/wiki/WordPress). It is also possible to connect to localhost via FTP with an [HTML editor](http://en.wikipedia.org/wiki/HTML_editor).

The default FTP user is "newuser", the default FTP password is "wampp". The default MySQL user is "root" while there is no default MySQL password.

Components:-

* Apache 2.4.10
* MySQL 5.6.21
* PHP 5.6.3
* phpMyAdmin 4.2.11
* FileZilla FTP Server 0.9.41
* [Tomcat](http://en.wikipedia.org/wiki/Apache_Tomcat) 7.0.56 (with mod\_proxy\_ajp as connector)
* Strawberry Perl 5.16.3.1 Portable
* XAMPP Control Panel 3.2.1 (from hackattack142)
* OpenSSL 1.0.1j

**3.0 What is Sql?**

SQL is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language and a data manipulation language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks."Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, though, most SQL code is not completely portable among different database systems without adjustments.

**3.1 History**

SQL was initially developed at [IBM](http://en.wikipedia.org/wiki/IBM) by [Donald D.Chamberlin](http://en.wikipedia.org/wiki/Donald_D._Chamberlin) and [Raymond F. Boyce](http://en.wikipedia.org/wiki/Raymond_F._Boyce) in the early 1970s. This version, initially called *SEQUEL* (*Structured English Query Language*), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, [System R](http://en.wikipedia.org/wiki/IBM_System_R), which a group at[IBM San Jose Research Laboratory](http://en.wikipedia.org/wiki/IBM_Almaden_Research_Center) had developed during the 1970s. The acronym SEQUEL was later changed to SQL because "SEQUEL" was a [trademark](http://en.wikipedia.org/wiki/Trademark) of the[UK-based](http://en.wikipedia.org/wiki/United_Kingdom) [Hawker Siddeley](http://en.wikipedia.org/wiki/Hawker_Siddeley) aircraft company.

In the late 1970s, Relational Software, Inc. (now [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation)) saw the potential of the concepts described by Codd, Chamberlin, and Boyce, and developed their own SQL-based [RDBMS](http://en.wikipedia.org/wiki/RDBMS) with aspirations of selling it to the [U.S. Navy](http://en.wikipedia.org/wiki/United_States_Navy), [Central Intelligence Agency](http://en.wikipedia.org/wiki/Central_Intelligence_Agency), and other [U.S. government](http://en.wikipedia.org/wiki/Federal_government_of_the_United_States) agencies. In June 1979, Relational Software, Inc. introduced the first commercially available implementation of SQL, [Oracle](http://en.wikipedia.org/wiki/Oracle_Database) V2 (Version2) for [VAX](http://en.wikipedia.org/wiki/VAX) computers.

After testing SQL at customer test sites to determine the usefulness and practicality of the system, IBM began developing commercial products based on their System R prototype including [System/38](http://en.wikipedia.org/wiki/IBM_System/38), [SQL/DS](http://en.wikipedia.org/wiki/IBM_SQL/DS), and [DB2](http://en.wikipedia.org/wiki/IBM_DB2), which were commercially available in 1979, 1981, and 1983, respectively.

### 3.2 Queries

The most common operation in SQL is the query, which is performed with the declarative [SELECT](http://en.wikipedia.org/wiki/Select_(SQL)) statement. SELECT retrieves data from one or more [tables](http://en.wikipedia.org/wiki/Table_(database)), or expressions. Standard SELECT statements have no persistent effects on the database. Some non-standard implementations of SELECT can have persistent effects, such as the SELECT INTO syntax that exists in some databases.

Queries allow the user to describe desired data, leaving the [database management system (DBMS)](http://en.wikipedia.org/wiki/Database_management_system) responsible for [planning](http://en.wikipedia.org/wiki/Query_plan), [optimizing](http://en.wikipedia.org/wiki/Query_optimizer), and performing the physical operations necessary to produce that result as it chooses.

A query includes a list of columns to include in the final result, immediately following the SELECT keyword. An asterisk ("\*") can also be used to specify that the query should return all columns of the queried tables. SELECT is the most complex statement in SQL, with optional keywords and clauses that include:

* The [FROM](http://en.wikipedia.org/wiki/From_(SQL)) clause, which indicates the table(s) to retrieve data from. The FROM clause can include optional [JOIN](http://en.wikipedia.org/wiki/Join_(SQL)) subclauses to specify the rules for joining tables.
* The [WHERE](http://en.wikipedia.org/wiki/Where_(SQL)) clause includes a comparison predicate, which restricts the rows returned by the query. The WHERE clause eliminates all rows from the result set where the comparison predicate does not evaluate to True.
* The GROUP BY clause is used to project rows having common values into a smaller set of rows. GROUP BY is often used in conjunction with SQL aggregation functions or to eliminate duplicate rows from a result set. The WHERE clause is applied before the GROUP BY clause.
* The [HAVING](http://en.wikipedia.org/wiki/Having_(SQL)) clause includes a predicate used to filter rows resulting from the GROUP BY clause. Because it acts on the results of the GROUP BY clause, aggregation functions can be used in the HAVING clause predicate.
* The [ORDER BY](http://en.wikipedia.org/wiki/Order_by_(SQL)) clause identifies which columns to use to sort the resulting data, and in which direction to sort them (ascending or descending). Without an ORDER BY clause, the order of rows returned by an SQL query is undefined.

**4.0 Jquery**

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. Used by over 60% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today. jQuery is free, open source software, licensed under the MIT License.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX framework and ASP.NET MVC Framework while Nokia has integrated it into the Web Run-Time widget development platform. jQuery has also been used in MediaWiki since version 1.16.

jQuery includes the following features:

* DOM element selections using the multi-browser open source selector engine Sizzle, a spin-off of the jQuery project
* DOM manipulation based on CSS selectors that uses elements' names and attributes, such as id and class, as criteria to select nodes in the DOM
* Events
* Effects and animations

**5.0 Javascript**

JavaScript is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also used in server-side network programming with frameworks such as Node.js, game development and the creation of desktop and mobile applications.

JavaScript is classified as a prototype-based scripting language with dynamic typing and first-class functions. This mix of features makes it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics. The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by Self and Scheme programming languages.

JavaScript is also used in environments that aren't web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation.

JavaScript has been standardized in the ECMAScript language specification.

**6.0 Source Code**

* **Login.php**

<html>

<head>

<meta charset="utf8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

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

<script>

$(function() {

$("#loginForm").submit(function(event) {

//alert('1');

$.post("adminsubmit.php", $("#loginForm").serialize(), function(data) {

//alert(data);

$("#successmessage").html(data);

});

event.preventDefault();

});

});

</script>

</head>

<body>

<div>

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

<div>

<form id="loginForm" method="post" action="login-action.php" >

<table id="body" class="boxu">

<tr><td><a id="loginButton"><span>Admin Login</span></a></td></tr>

<tr>

<td><label for="username">Username</label></td>

<td><input type="text" name="username" id="username" value="" required/></td>

</tr>

<tr>

<td><label for="password">Password</label></td>

<td><input type="password" name="password" id="password" value="" required/></td>

</tr>

<tr><td><input type="submit" id="login" value="Login" /></td>

<td>

<div id="successmessage" style="color:#00000; font-weight:bold;"></td>

</tr>

</div>

</table>

</form>

</div><!-- Login box Ends Here -->

</div><!-- Login container Ends Here -->

</body>

</html>

* **Login-action.php**

<?php

include\_once 'admin-class.php';

$admin = new itg\_admin();

$admin->\_login\_action();

* **Admin-class.php**

<?php

/\*\* Include the database file \*/

include\_once 'db/db.php';

/\*\*

\* The main class of login

\* All the necesary system functions are prefixed with \_

\* examples, \_login\_action - to be used in the login-action.php file

\* \_authenticate - to be used in every file where admin restriction is to be inherited etc...

\* @author Swashata <swashata@intechgrity.com>

\*/

class itg\_admin {

/\*\*

\* Holds the script directory absolute path

\* @staticvar

\*/

static $abs\_path;

/\*\*

\* Store the sanitized and slash escaped value of post variables

\* @var array

\*/

var $post = array();

/\*\*

\* Stores the sanitized and decoded value of get variables

\* @var array

\*/

var $get = array();

/\*\*

\* The constructor function of admin class

\* We do just the session start

\* It is necessary to start the session before actually storing any value

\* to the super global $\_SESSION variable

\*/

public function \_\_construct() {

session\_start();

//store the absolute script directory

//note that this is not the admin directory

self::$abs\_path = dirname(dirname(\_\_FILE\_\_));

//initialize the post variable

if($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$this->post = $\_POST;

if(get\_magic\_quotes\_gpc ()) {

//get rid of magic quotes and slashes if present

array\_walk\_recursive($this->post, array($this, 'stripslash\_gpc'));

}

}

//initialize the get variable

$this->get = $\_GET;

//decode the url

array\_walk\_recursive($this->get, array($this, 'urldecode'));

}

/\*\*

\* Sample function to return the nicename of currently logged in admin

\* @global ezSQL\_mysql $db

\* @return string The nice name of the user

\*/

public function get\_username() {

$username = $\_SESSION['admin\_login'];

global $db;

$info = $db->get\_row("SELECT `username` FROM `login` WHERE `username` = '" . $db->escape($username) . "'");

if(is\_object($info))

return $info->username;

else

return '';

}

/\*\*

\* Checks whether the user is authenticated

\* to access the admin page or not.

\*

\* Redirects to the login.php page, if not authenticates

\* otherwise continues to the page

\*

\* @access public

\* @return void

\*/

public function \_authenticate() {

//first check whether session is set or not

if(!isset($\_SESSION['admin\_login'])) {

//check the cookie

if(isset($\_COOKIE['username']) && isset($\_COOKIE['password'])) {

//cookie found, is it really someone from the

if($this->\_check\_db($\_COOKIE['username'], $\_COOKIE['password'])) {

$\_SESSION['admin\_login'] = $\_COOKIE['username'];

header("location: main.php");

die();

}

else {

header("location: login.php");

die();

}

}

else {

header("location: login.php");

die();

}

}

}

/\*\*

\* Check for login in the action file

\*/

public function \_login\_action() {

//insufficient data provided

if(!isset($this->post['username']) || $this->post['username'] == '' || !isset($this->post['password']) || $this->post['password'] == '') {

header ("location: login.php");

}

//get the username and password

$username = $this->post['username'];

$password = md5(sha1($this->post['password']));

//check the database for username

if($this->\_check\_db($username, $password)) {

//ready to login

$\_SESSION['admin\_login'] = $username;

//check to see if remember, ie if cookie

if(isset($this->post['remember'])) {

//set the cookies for 1 day, ie, 1\*24\*60\*60 secs

//change it to something like 30\*24\*60\*60 to remember user for 30 days

setcookie('username', $username, time() + 1\*24\*60\*60);

setcookie('password', $password, time() + 1\*24\*60\*60);

} else {

//destroy any previously set cookie

setcookie('username', '', time() - 1\*24\*60\*60);

setcookie('password', '', time() - 1\*24\*60\*60);

}

header("location: main.php");

}

else {

header ("location: login.php");

}

die();

}

/\*\*

\* Check the database for login user

\* Get the password for the user

\* compare md5 hash over sha1

\* @param string $username Raw username

\* @param string $password expected to be md5 over sha1

\* @return bool TRUE on success FALSE otherwise

\*/

private function \_check\_db($username, $password) {

global $db;

$user\_row = $db->get\_row("SELECT \* FROM `login` WHERE `username`='" . $db->escape($username) . "'");

//general return

if(is\_object($user\_row) && md5($user\_row->password) == $password)

return true;

else

return false;

}

/\*\*

\* stripslash gpc

\* Strip the slashes from a string added by the magic quote gpc thingy

\* @access protected

\* @param string $value

\*/

private function stripslash\_gpc(&$value) {

$value = stripslashes($value);

}

/\*\*

\* htmlspecialcarfy

\* Encodes string's special html characters

\* @access protected

\* @param string $value

\*/

private function htmlspecialcarfy(&$value) {

$value = htmlspecialchars($value);

}

/\*\*

\* URL Decode

\* Decodes a URL Encoded string

\* @access protected

\* @param string $value

\*/

protected function urldecode(&$value) {

$value = urldecode($value);

}

}

* **Adminsubmit.php**

<?php

if ($\_POST) {

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

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

} else {

$username = "";

$password = "";}

//Database connection

$filePath = 'db/config.php';

require\_once ($filePath);

$conn = mysql\_connect($server, $username, $password);

//new mysqli('localhost',$user,$pass,$db) or die("unable to connect");

if (!$conn) {

echo "connection not established";

exit ;

}

if (!mysql\_select\_db('computerdetails', $conn)) {

echo 'Could not select database';

exit ;}

// Secure the credentials

if ($\_POST) {

$username = mysql\_real\_escape\_string($\_POST['username']);

$password = mysql\_real\_escape\_string($\_POST['password']);

} else {

$username = "";

$password = "";}

$query = "SELECT COUNT(\*) AS `total` FROM `login` WHERE `username` = '$username' AND `password` = '$password'";

$result = mysql\_query($query);

$row = mysql\_fetch\_assoc($result);

if ($row['total'] == 1) {

header('Location: main.php');

//echo '<div style="color:#008000; font-weight:bold;">Login Successful<br><a href="main.php">View Registered Users</a></div>';

} else {

echo '<div style="color:#008000; font-weight:bold;">Incorrect username or password</div>';

}

?>

* **Logout.php**

<?php

/\*\*

\* The logout file

\* destroys the session

\* expires the cookie

\* redirects to login.php

\*/

session\_start();

session\_destroy();

setcookie('username', '', time() - 1\*24\*60\*60);

setcookie('password', '', time() - 1\*24\*60\*60);

header("location: login.php");

?>

* **Main.php**

<?php

include\_once 'admin-class.php';

$admin = new itg\_admin();

$admin->\_authenticate();

?>

<html>

<head>

<meta charset="utf8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

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

</head>

<body>

<div id="nav">

<div id="nav\_wrapper">

<ul><li><a href="#">Server Details</a><ul>

<li><a href="serverdetailsadd.php">Add</a></li>

<li><a href="serverdetailsview.php">View</a></li>

</ul></li>

<li> <a href="#">PC details</a><ul>

<li><a href="#">Add</a></li>

<li><a href="#">View</a></li>

</ul></li>

<li> <a href="#">ITequip Details</a><ul>

<li><a href="#">Add</a></li>

<li><a href="#">View</a></li>

</ul>

</li>

</ul>

</div>

</div>

<!-- <fieldset>

<legend>Welcome <?php echo $admin->get\_username(); ?></legend>

<p>

Here are some of the basic informations

</p>

<p>

Username: <?php echo $\_SESSION['admin\_login']; ?>

</p>

<p>

Email:

</p>

</fieldset> -->

<p>

<input type="button" id="button" onclick="javascript:window.location.href='logout.php'" value="logout" />

</p>

</body>

</html>

* **Serverdetailsadd.php**

<?php

include\_once 'admin-class.php';

$admin = new itg\_admin();

$admin->\_authenticate();

?>

<?php include ("db/db.php"); ?>

<?php

require\_once("db/config.php");

$conn = mysql\_connect($server, $username, $password);

mysql\_select\_db("computerdetails") or die(mysql\_error());

if ( !isset ($\_GET['id']) || $\_GET['id'] == "" ) {

include ("404.php");

die();

}

$requested\_id = mysql\_real\_escape\_string($\_GET['id']);

$sql = "SELECT \* FROM serverdetailsform WHERE id= '{$requested\_id}' LIMIT 1";

$query = mysql\_query($sql) or die(mysql\_error());

$edit\_rows = mysql\_fetch\_assoc($query);

if (isset($\_POST['edit'])) {

$id = mysql\_real\_escape\_string($\_POST['id']);

$servername = mysql\_real\_escape\_string($\_POST['servername']);

$ipaddress = mysql\_real\_escape\_string($\_POST['ipaddress']);

$macaddress = mysql\_real\_escape\_string($\_POST['macaddress']);

$os = mysql\_real\_escape\_string($\_POST['os']);

$licensekey = mysql\_real\_escape\_string($\_POST['licensekey']);

$sqldata = mysql\_real\_escape\_string($\_POST['sqldata']);

$sqlversion = mysql\_real\_escape\_string($\_POST['sqlversion']);

$sqllicensekey = mysql\_real\_escape\_string($\_POST['sqllicensekey']);

$loai = mysql\_real\_escape\_string($\_POST['loai']);

$loch = mysql\_real\_escape\_string($\_POST['loch']);

$antivirus = mysql\_real\_escape\_string($\_POST['antivirus']);

$antivirusvalidity = mysql\_real\_escape\_string($\_POST['antivirusvalidity']);

$remarks = mysql\_real\_escape\_string($\_POST['remarks']);

$sql = "UPDATE serverdetailsform SET servername = '{$servername}', ipaddress = '{$ipaddress}', macaddress = '{$macaddress}', os = '{$os}', licensekey = '{$licensekey}', sqldata = '{$sqldata}', sqlversion = '{$sqlversion}', sqllicensekey = '{$sqllicensekey}', loai = '{$loai}', loch = '{$loch}', remarks = '{$remarks}', antivirus = '{$antivirus}', antivirusvalidity = '{$antivirusvalidity}' WHERE id = '{$requested\_id}'";

mysql\_query($sql) or die(mysql\_error());

header("location: serverdetailsupdate.php?id='{$requested\_id}'&success");

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<style type="text/css">

table, th{

border:1px solid #699;

}

td {

padding: 10px;

}

thead {

background: #D2E6EA;

}

</style>

<meta charset=utf-8" />

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

<title>the project</title>

<meta charset="utf8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

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

<script type="text/javascript" src="js/jquery-1.10.2.js"></script>

<script type="text/javascript" src="js/jquery-ui-1.10.4.js"></script>

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

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

<script type="text/javascript">

jQuery(function($){

$("#macaddress").mask("AA-AA-AA-AA-AA-AA"); $("#licensekey").mask("AAAAA/AAAAA/AAAAA/AAAAA/AAAAA");

$("#sqllicensekey").mask("AAAAA-AAAAA-AAAAA-AAAAA-AAAAA");

/\* $("#ipaddress").mask("0ZZ.0ZZ.0ZZ.0ZZ", {

translation: {

'Z': {

pattern: /[0-9]/, optional: true

}

}

}); \*/

});

$(function(){

$("#computerdetails").submit(function(event){

//alert('1');

$.post( "serverdetailssubmit.php", $("#computerdetails").serialize(), function(data){

//alert(data);

$("#message").html(data);

});

event.preventDefault();

});

});

</script>

<link rel="stylesheet" href="css/jquery-ui-1.10.4.custom.css">

<script type="text/javascript" src="js/jquery-1.10.2.js"></script>

<script type="text/javascript" src="js/jquery-ui-1.10.4.js"></script>

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

<script type="text/javascript">

// When the browser is ready...

$(function() {

// Setup form validation on the #register-form element

$("#computerdetails").validate({

// Specify the validation rules

rules : {

ipaddress : {

minlength : 11,

maxlength : 15

},

macaddress : {

minlength : 17,

maxlength : 17

},

licensekey : {

length : 25

},

sqlversion : {

digits : true

},

sqllicensekey : {

length : 25

},

agree : "required"

},

// Specify the validation error messages

messages : {

ipaddress : {

minlength : "Enter a valid ip address",

maxlength : "Enter a valid ip address"

},

macaddress : {

minlength : "Enter a valid mac address",

maxlength : "Enter a valid mac address"

},

licensekey : {

length : "Enter 25 digits"

},

sqlversion : {

digits : "Enter only numbers"

},

sqllicensekey : {

length : "Enter 25 digits"

},

agree : "Please accept our policy"

},

submitHandler : function(form) {

form.submit();

}

});

});

</script>

<script src="js/jquery.input-ip-address-control.js"></script>

<script>

$(function(){

$('#ipaddress').ipAddress();

});

</script>

<SCRIPT LANGUAGE="JavaScript">

function SecListBox(ListBox,text,value)

{

try

{

var option=document.createElement("OPTION");

option.value=value;

option.text=text;

ListBox.options.add(option)

}

catch(er)

{

alert(er)

}

}

function FirstListBox()

{

try

{

var count=document.getElementById("lstBox").options.length;

for(i=0;i<count;i++)

{

if(document.getElementById("lstBox").options[i].selected)

{

SecListBox(document.getElementById("loai"),document.getElementById("lstBox").options[i].value,document.getElementById("lstBox").options[i].value);document.getElementById("lstBox").remove(i);

break

}

}

}

catch(er)

{

alert(er)

}

}

function SortAllItems()

{

var arr=new Array();

for(i=0;i<document.getElementById("lstBox").options.length;i++)

{

arr[i]=document.getElementById("lstBox").options[i].value}arr.sort();

RemoveAll();

for(i=0;i<arr.length;i++)

{

SecListBox(document.getElementById("lstBox"),arr[i],arr[i])}}function RemoveAll(){try{document.getElementById("lstBox").options.length=0

}

catch(er)

{

alert(er)

}

}

function SecondListBox()

{

try

{

var count=document.getElementById("loai").options.length;

for(i=0;i<count;i++)

{

if(document.getElementById("loai").options[i].selected){SecListBox(document.getElementById("lstBox"),document.getElementById("loai").options[i].value,document.getElementById("loai").options[i].value);document.getElementById("loai").remove(i);

break

}

}

SortAllItems()

}

catch(er)

{

alert(er)

}

}

</SCRIPT>

</head>

<body>

<div id="nav">

<div id="nav\_wrapper">

<ul>

<li><a href="#">Server Details</a>

<ul>

<li><a href="serverdetailsadd.php">Add</a>

</li>

<li><a href="serverdetailsview.php">View</a>

</li>

<li><a href="main.php">Index</a>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</div>

</div>

<p>

<input type="button" id="button" onclick="javascript:window.location.href='logout.php'" value="logout" />

</p>

<br><br>

<div class="container">

<div class="content">

<?php

/\*print\_r($edit\_rows);\*/

?>

<?php if ( isset ($\_GET['success'])) { ?>

<h3 style="color: #71B23B;">The record was updated successfully!</h3>

<?php } ?>

<div class="separator"></div>

<h2 align="center">Update Serverdetails Form</h2>

<div class="separator"></div>

<form method="POST" action="<?php echo basename($\_SERVER['PHP\_SELF']); ?>?id=<?php echo $requested\_id; ?>">

<table align="center">

<tr>

<td>Server Name:</td>

<td><input type="text" name="servername" id="servername" placeholder="Input server name here...." value="<?php echo $edit\_rows['servername'];?>"></td>

</tr>

<tr>

<td>IP address</td>

<td><input type="text" name="ipaddress" id="ipaddress" class="form-control" value="<?php echo $edit\_rows['ipaddress'];?>"></td>

</tr>

<tr>

<td>MAC address</td>

<td><input type="text" name="macaddress" id="macaddress" value="<?php echo $edit\_rows['macaddress'];?>"></td>

</tr>

<tr>

<td>OS</td>

<td><input type="text" name="os" id="os" value="<?php echo $edit\_rows['os'];?>"></td>

</tr>

<tr>

<td>License Key</td>

<td><input type="text" name="licensekey" id="licensekey" value="<?php echo $edit\_rows['licensekey'];?>"></td>

</tr>

<tr>

<td>SQL Data (Yes/No) </td>

<td><input type="text" name="sqldata" id="sqldata" value="<?php echo $edit\_rows['sqldata'];?>"></td>

</tr><tr>

<td>SQL Version</td>

<td><input type="text" name="sqlversion" id="sqlversion" value="<?php echo $edit\_rows['sqlversion'];?>"></td>

</tr><tr>

<td>SQL License Key</td>

<td><input type="text" name="sqllicensekey" id="sqllicensekey" value="<?php echo $edit\_rows['sqllicensekey'];?>"></td>

</tr><tr>

<td>List of Application Installed</td>

<td><select size="5" name="lstBox" id="lstBox" multiple>

<option value="A">A</option>

<option value="D">D</option>

<option value="C">C</option>

<option value="E">E</option>

<option value="B">B</option>

</select>

<input name="add" type="button" value="Add" onclick="FirstListBox();" />

<input name="remove" type="button" value="Remove" onclick="SecondListBox();"/>

<select size="5" name="loai" id="loai" value="<?php echo $edit\_rows['loai'];?>" multiple>

</select></td>

</tr><tr>

<td>List of Connected Hardware</td>

<td><input type="text" name="loch" id="loch" value="<?php echo $edit\_rows['loch'];?>"></td>

</tr><tr>

<td>Antivirus (Yes/No)</td>

<td>

<input type="text" name="antivirus" id="antivirus" value="<?php echo $edit\_rows['antivirus'];?>">

</td>

</tr><tr>

<td>Antivirus Validity</td>

<td><input type="text" name="antivirusvalidity" id="antivirusvalidity" value="<?php echo $edit\_rows['antivirusvalidity'];?>"></td>

</tr><tr>

<td>Remarks</td>

<td><textarea rows="4" cols="50" name="remarks" value="<?php echo $edit\_rows['remarks'];?>"></textarea></td>

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

<td>

<input type="submit" name="edit" value="Update">

<input type="hidden" name="id" value="" value="<?php echo $edit\_rows['id'];?>">

</td>

</tr>

</table>

</form>

<div class="separator"></div>

</div>

</div>

</body>

</html>

* **Serverdetailssubmit.php**

<?php

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

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

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

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

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

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

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

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

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

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

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

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

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

//Database connection

require\_once("db/config.php");

$conn=mysql\_connect($server, $username, $password); //new mysqli('localhost',$user,$pass,$db) or die("unable to connect");

if(!$conn)

{

echo "connection not established";

exit;

}

if (!mysql\_select\_db('computerdetails', $conn)) {

echo 'Could not select database';

exit;

}

//mysql query to insert value to database

$sql = "INSERT INTO serverdetailsform (id, servername, ipaddress, macaddress, os, licensekey, sqldata, sqlversion, sqllicensekey, loai, loch, remarks, antivirus, antivirusvalidity) VALUES ('$id','$servername', '$ipaddress', '$macaddress', '$os', '$licensekey', '$sqldata', '$sqlversion', '$sqllicensekey', '$loai', '$loch', '$remarks', '$antivirus', '$antivirusvalidity')";

//echo $sql;

$result = mysql\_query($sql, $conn);

//if value inserted successfully display success message

if($result)

{

echo '<div style="color:#008000; font-weight:bold;">Registered successfully..!!</div>';

//echo 'COMMENTS :'.$comments;

}else

{

//error message

echo '<div style="color:#c24f00; font-weight:bold;">unable to register !!</div>';

}

?>

* **Serverdetailsview.php**

<?php

include\_once 'admin-class.php';

$admin = new itg\_admin();

$admin->\_authenticate();

?>

<?php

require\_once("db/config.php");

$conn = mysql\_connect($server, $username, $password);

mysql\_select\_db("computerdetails") or die(mysql\_error());

$queryData = mysql\_query("SELECT \* FROM serverdetailsform") or die(mysql\_error());

if (isset($\_GET['delete'])) {

$multiple = $\_GET['multiple'];

$i = 0;

$sql = "DELETE FROM serverdetailsform ";

foreach ($multiple as $item\_id) { $i ++;

if ($i == 1){

$sql .= "WHERE id = ". mysql\_real\_escape\_string($item\_id) . "";

} else {

$sql .= "OR id =". mysql\_real\_escape\_string($item\_id) . "";

}

}

mysql\_query($sql) or die(mysql\_error());

header("location: " . $\_SERVER['PHP\_SELF']);

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

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

<style type="text/css">

body {

font-family: Arial, sans-serif;

font-size: 14px;

line-height: 1.6;

text-align: center;

}

#wrapper {

margin: 0;

width: 650px;

}

table, th, td {

border:1px solid #699;

}

td {

padding: 10px;

}

thead {

background: #D2E6EA;

}

</style>

</head>

<body>

<div id="nav">

<div id="nav\_wrapper">

<ul>

<li><a href="#">Server Details</a>

<ul>

<li><a href="serverdetailsadd.php">Add</a>

</li>

<li><a href="serverdetailsview.php">View</a>

</li>

<li><a href="main.php">Index</a>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</div>

</div>

<p>

<input type="button" id="button" onclick="javascript:window.location.href='logout.php'" value="logout" />

</p>

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

<div id="wrapper">

<?php if (mysql\_num\_rows($queryData) > 0): ?>

<form action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="GET">

<table width="100%">

<thead>

<tr>

<td>

<div><input type="submit" name="delete" value="Delete"></div>

</td>

<td>Server Name</td>

<td>Ip Address</td>

<td>Mac Address</td>

<td>Operating System</td>

<td>License Key</td>

<td>SQL Data</td>

<td>SQL Version</td>

<td>SQL License Key</td>

<td>List of applications installed</td>

<td>List of connected hardware</td>

<td>Antivirus</td>

<td>Antivirus Validity</td>

<td>Remarks</td>

<td>Edit</td>

</tr>

</thead>

<tbody>

<tr>

<?php while ($row = mysql\_fetch\_assoc($queryData)) {?>

<td>

<input type="radio" name="multiple[]" value="<?php echo $row['id'];

?>" checked>

</td>

<td><?php echo $row['servername'] ?></td>

<td><?php echo $row['ipaddress'] ?></td>

<td><?php echo $row['macaddress'] ?></td>

<td><?php echo $row['os'] ?></td>

<td><?php echo $row['licensekey'] ?></td>

<td><?php echo $row['sqldata'] ?></td>

<td><?php echo $row['sqlversion'] ?></td>

<td><?php echo $row['sqllicensekey'] ?></td>

<td><?php echo $row['loai'] ?></td>

<td><?php echo $row['loch'] ?></td>

<td><?php echo $row['antivirus'] ?></td>

<td><?php echo $row['antivirusvalidity'] ?></td>

<td><?php echo $row['remarks'] ?></td>

<td><a href="serverdetailsupdate.php?id=<?php echo urlencode($row['id']); ?>">Edit</a>

</td>

</tr>

<?php } ?>

</tbody>

</table>

</form>

<?php else: ?>

<h2>No data to display</h2>

<?php endif; ?>

</div>

</script>

</body>

</html>

* **Serverdetailsupdate.php**

<?php

include\_once 'admin-class.php';

$admin = new itg\_admin();

$admin->\_authenticate();

?>

<?php include ("db/db.php"); ?>

<?php

require\_once("db/config.php");

$conn = mysql\_connect($server, $username, $password);

mysql\_select\_db("computerdetails") or die(mysql\_error());

if ( !isset ($\_GET['id']) || $\_GET['id'] == "" ) {

include ("404.php");

die();

}

$requested\_id = mysql\_real\_escape\_string($\_GET['id']);

$sql = "SELECT \* FROM serverdetailsform WHERE id= '{$requested\_id}' LIMIT 1";

$query = mysql\_query($sql) or die(mysql\_error());

$edit\_rows = mysql\_fetch\_assoc($query);

if (isset($\_POST['edit'])) {

$id = mysql\_real\_escape\_string($\_POST['id']);

$servername = mysql\_real\_escape\_string($\_POST['servername']);

$ipaddress = mysql\_real\_escape\_string($\_POST['ipaddress']);

$macaddress = mysql\_real\_escape\_string($\_POST['macaddress']);

$os = mysql\_real\_escape\_string($\_POST['os']);

$licensekey = mysql\_real\_escape\_string($\_POST['licensekey']);

$sqldata = mysql\_real\_escape\_string($\_POST['sqldata']);

$sqlversion = mysql\_real\_escape\_string($\_POST['sqlversion']);

$sqllicensekey = mysql\_real\_escape\_string($\_POST['sqllicensekey']);

$loai = mysql\_real\_escape\_string($\_POST['loai']);

$loch = mysql\_real\_escape\_string($\_POST['loch']);

$antivirus = mysql\_real\_escape\_string($\_POST['antivirus']);

$antivirusvalidity = mysql\_real\_escape\_string($\_POST['antivirusvalidity']);

$remarks = mysql\_real\_escape\_string($\_POST['remarks']);

$sql = "UPDATE serverdetailsform SET servername = '{$servername}', ipaddress = '{$ipaddress}', macaddress = '{$macaddress}', os = '{$os}', licensekey = '{$licensekey}', sqldata = '{$sqldata}', sqlversion = '{$sqlversion}', sqllicensekey = '{$sqllicensekey}', loai = '{$loai}', loch = '{$loch}', remarks = '{$remarks}', antivirus = '{$antivirus}', antivirusvalidity = '{$antivirusvalidity}' WHERE id = '{$requested\_id}'";

mysql\_query($sql) or die(mysql\_error());

header("location: serverdetailsupdate.php?id='{$requested\_id}'&success");

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<style type="text/css">

table, th{

border:1px solid #699;

}

td {

padding: 10px;

}

thead {

background: #D2E6EA;

}

</style>

<meta charset=utf-8" />

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

<title>the project</title>

<meta charset="utf8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

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

<script type="text/javascript" src="js/jquery-1.10.2.js"></script>

<script type="text/javascript" src="js/jquery-ui-1.10.4.js"></script>

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

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

<script type="text/javascript">

jQuery(function($){

$("#macaddress").mask("AA-AA-AA-AA-AA-AA");

$("#licensekey").mask("AAAAA/AAAAA/AAAAA/AAAAA/AAAAA");

$("#sqllicensekey").mask("AAAAA-AAAAA-AAAAA-AAAAA-AAAAA");

/\* $("#ipaddress").mask("0ZZ.0ZZ.0ZZ.0ZZ", {

translation: {

'Z': {

pattern: /[0-9]/, optional: true

}

}

}); \*/ });

$(function(){

$("#computerdetails").submit(function(event){

//alert('1');

$.post( "serverdetailssubmit.php", $("#computerdetails").serialize(), function(data){

//alert(data);

$("#message").html(data);

});

event.preventDefault();

});

}); </script>

<link rel="stylesheet" href="css/jquery-ui-1.10.4.custom.css">

<script type="text/javascript" src="js/jquery-1.10.2.js"></script>

<script type="text/javascript" src="js/jquery-ui-1.10.4.js"></script>

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

<script type="text/javascript">

// When the browser is ready...

$(function() {

// Setup form validation on the #register-form element

$("#computerdetails").validate({

// Specify the validation rules

rules : {

ipaddress : {

minlength : 11,

maxlength : 15

},

macaddress : {

minlength : 17,

maxlength : 17

},

licensekey : {

length : 25

},

sqlversion : {

digits : true

},

sqllicensekey : {

length : 25

},

agree : "required"

},

// Specify the validation error messages

messages : {

ipaddress : {

minlength : "Enter a valid ip address",

maxlength : "Enter a valid ip address"

},

macaddress : {

minlength : "Enter a valid mac address",

maxlength : "Enter a valid mac address"

},

licensekey : {

length : "Enter 25 digits"

},

sqlversion : {

digits : "Enter only numbers"

},

sqllicensekey : {

length : "Enter 25 digits"

},

agree : "Please accept our policy"

},

submitHandler : function(form) {

form.submit();

}

});

});

</script>

<script src="js/jquery.input-ip-address-control.js"></script>

<script>

$(function(){

$('#ipaddress').ipAddress();

});

</script>

<SCRIPT LANGUAGE="JavaScript">

function SecListBox(ListBox,text,value)

{

try

{

var option=document.createElement("OPTION");

option.value=value;

option.text=text;

ListBox.options.add(option)

}

catch(er)

{

alert(er)

}

}

function FirstListBox()

{

try

{

var count=document.getElementById("lstBox").options.length;

for(i=0;i<count;i++)

{

if(document.getElementById("lstBox").options[i].selected)

{

SecListBox(document.getElementById("loai"),document.getElementById("lstBox").options[i].value,document.getElementById("lstBox").options[i].value);document.getElementById("lstBox").remove(i);

break

}

}

}

catch(er)

{

alert(er)

}

}

function SortAllItems()

{

var arr=new Array();

for(i=0;i<document.getElementById("lstBox").options.length;i++)

{

arr[i]=document.getElementById("lstBox").options[i].value}arr.sort();

RemoveAll();

for(i=0;i<arr.length;i++)

{

SecListBox(document.getElementById("lstBox"),arr[i],arr[i])}}function RemoveAll(){try{document.getElementById("lstBox").options.length=0

}

catch(er)

{

alert(er)

}

}

function SecondListBox()

{

try

{

var count=document.getElementById("loai").options.length;

for(i=0;i<count;i++)

{

if(document.getElementById("loai").options[i].selected){SecListBox(document.getElementById("lstBox"),document.getElementById("loai").options[i].value,document.getElementById("loai").options[i].value);document.getElementById("loai").remove(i);

break

}

}

SortAllItems()

}

catch(er)

{

alert(er)

}

}

</SCRIPT>

</head>

<body>

<div id="nav">

<div id="nav\_wrapper">

<ul>

<li><a href="#">Server Details</a>

<ul>

<li><a href="serverdetailsadd.php">Add</a>

</li>

<li><a href="serverdetailsview.php">View</a>

</li>

<li><a href="main.php">Index</a>

</li>

</ul>

</li>

</ul>

</li>

</ul>

</div>

</div>

<p>

<input type="button" id="button" onclick="javascript:window.location.href='logout.php'" value="logout" />

</p>

<br><br>

<div class="container">

<div class="content">

<?php

/\*print\_r($edit\_rows);\*/

?>

<?php if ( isset ($\_GET['success'])) { ?>

<h3 style="color: #71B23B;">The record was updated successfully!</h3>

<?php } ?>

<div class="separator"></div>

<h2 align="center">Update Serverdetails Form</h2>

<div class="separator"></div>

<form method="POST" action="<?php echo basename($\_SERVER['PHP\_SELF']); ?>?id=<?php echo $requested\_id; ?>">

<table align="center">

<tr>

<td>Server Name:</td>

<td><input type="text" name="servername" id="servername" placeholder="Input server name here...." value="<?php echo $edit\_rows['servername'];?>"></td>

</tr>

<tr>

<td>IP address</td>

<td><input type="text" name="ipaddress" id="ipaddress" class="form-control" value="<?php echo $edit\_rows['ipaddress'];?>"></td>

</tr>

<tr>

<td>MAC address</td>

<td><input type="text" name="macaddress" id="macaddress" value="<?php echo $edit\_rows['macaddress'];?>"></td>

</tr>

<tr>

<td>OS</td>

<td><input type="text" name="os" id="os" value="<?php echo $edit\_rows['os'];?>"></td>

</tr>

<tr>

<td>License Key</td>

<td><input type="text" name="licensekey" id="licensekey" value="<?php echo $edit\_rows['licensekey'];?>"></td>

</tr>

<tr>

<td>SQL Data (Yes/No) </td>

<td><input type="text" name="sqldata" id="sqldata" value="<?php echo $edit\_rows['sqldata'];?>"></td>

</tr><tr>

<td>SQL Version</td>

<td><input type="text" name="sqlversion" id="sqlversion" value="<?php echo $edit\_rows['sqlversion'];?>"></td>

</tr><tr>

<td>SQL License Key</td>

<td><input type="text" name="sqllicensekey" id="sqllicensekey" value="<?php echo $edit\_rows['sqllicensekey'];?>"></td>

</tr><tr>

<td>List of Application Installed</td>

<td><select size="5" name="lstBox" id="lstBox" multiple>

<option value="A">A</option>

<option value="D">D</option>

<option value="C">C</option>

<option value="E">E</option>

<option value="B">B</option>

</select>

<input name="add" type="button" value="Add" onclick="FirstListBox();" />

<input name="remove" type="button" value="Remove" onclick="SecondListBox();"/>

<select size="5" name="loai" id="loai" value="<?php echo $edit\_rows['loai'];?>" multiple>

</select></td>

</tr><tr>

<td>List of Connected Hardware</td>

<td><input type="text" name="loch" id="loch" value="<?php echo $edit\_rows['loch'];?>"></td>

</tr><tr>

<td>Antivirus (Yes/No)</td><td>

<input type="text" name="antivirus" id="antivirus" value="<?php echo $edit\_rows['antivirus'];?>">

</td></tr><tr>

<td>Antivirus Validity</td>

<td><input type="text" name="antivirusvalidity" id="antivirusvalidity" value="<?php echo $edit\_rows['antivirusvalidity'];?>"></td>

</tr><tr>

<td>Remarks</td>

<td><textarea rows="4" cols="50" name="remarks" value="<?php echo $edit\_rows['remarks'];?>"></textarea></td>

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

<td>

<input type="submit" name="edit" value="Update">

<input type="hidden" name="id" value="" value="<?php echo edit\_rows['id'];?>">

</td></tr>

</table>

</form>

<div class="separator"></div></div>

</div>

</body>

</html>

* **Style.css**

body{

padding: 0;

margin: 0;

font-family: Arial;

font-size: 100%;

}

#nav{

background-color: #222;

}

/\* #nav\_wrapper{

width: 960px;

margin: 0 auto;

text-align: left;

} \*/

#nav ul{

background-color: #222;

list-style-type: none;

padding: 0;

margin: 0;

position: relative;

float: left;

}

#nav ul li{

display: inline-block;

float: left;

}

#nav ul li a:hover{

background-color: #333;

color: #699;

}

#nav ul li a,visited{

color: #fff;

display: block;

padding: 15px;

text-decoration: none;

}

#nav ul li:hover ul{

display: block;

}

#nav ul ul{

display: none;

position: absolute;

background-color: #222;

border: 5px solid #222;

border-top: 0;

}

#nav ul ul li{

display: block;}

#nav ul ul li a:hover{

color: #699;}

#boxd {

width:300px;

height:10px;

background-color:#C2AD99;

position:fixed;

margin-left:-150px; /\* half of width \*/

margin-top:-150px; /\* half of height \*/

top:50%;

left:50%;}

.boxu {

width:400px;

height:10px;

background-color:#C2AD99;

position:fixed;

margin-left:-150px; /\* half of width \*/

margin-top:-150px; /\* half of height \*/

top:50%;

left:50%;}

input#button{

float: right;

cursor:pointer; /\*forces the cursor to change to a hand when the button is hovered\*/

padding:5px 25px; /\*add some padding to the inside of the button\*/

background:#222; /\*the colour of the button\*/

border:1px solid #33842a; /\*required or the default border for the browser will appear\*/

/\*give the button curved corners, alter the size as required\*/

-moz-border-radius: 10px;

-webkit-border-radius: 10px;

border-radius: 10px;

/\*give the button a drop shadow\*/

-webkit-box-shadow: 0 0 4px rgba(0,0,0, .75);

-moz-box-shadow: 0 0 4px rgba(0,0,0, .75);

box-shadow: 0 0 4px rgba(0,0,0, .75);

/\*style the text\*/

color:#fff;

font-size:1.1em;}

/\*\*\*NOW STYLE THE BUTTON'S HOVER AND FOCUS STATES\*\*\*/

input#button:hover, input#gobutton:focus{

background-color :#333; /\*make the background a little darker\*/

/\*reduce the drop shadow size to give a pushed button effect\*/

color:#699;

-webkit-box-shadow: 0 0 1px rgba(0,0,0, .75);

-moz-box-shadow: 0 0 1px rgba(0,0,0, .75);

box-shadow: 0 0 1px rgba(0,0,0, .75);}

* **db/db.php**

<?php

/\*\*

\* The db.php file which initiates a connection to the database

\* and gives a global $db variable for access

\* @author Swashata <swashata@intechgrity.com>

\* @uses ezSQL MySQL

\*/

/\*\* edit your configuration \*/

$dbuser = 'root';

$dbname = 'computerdetails';

$dbpassword = '';

$dbhost = 'localhost';/\*\* Stop editing from here, else you know what you are doing ;) \*//\*\* defined the root for the db \*/

if(!defined('ADMIN\_DB\_DIR'))

define('ADMIN\_DB\_DIR', dirname(\_\_FILE\_\_));

require\_once ADMIN\_DB\_DIR . '/ez\_sql\_core.php';

require\_once ADMIN\_DB\_DIR . '/ez\_sql\_mysql.php';

global $db;

$db = new ezSQL\_mysql($dbuser, $dbpassword, $dbname, $dbhost);

* **Config.php**

<?php

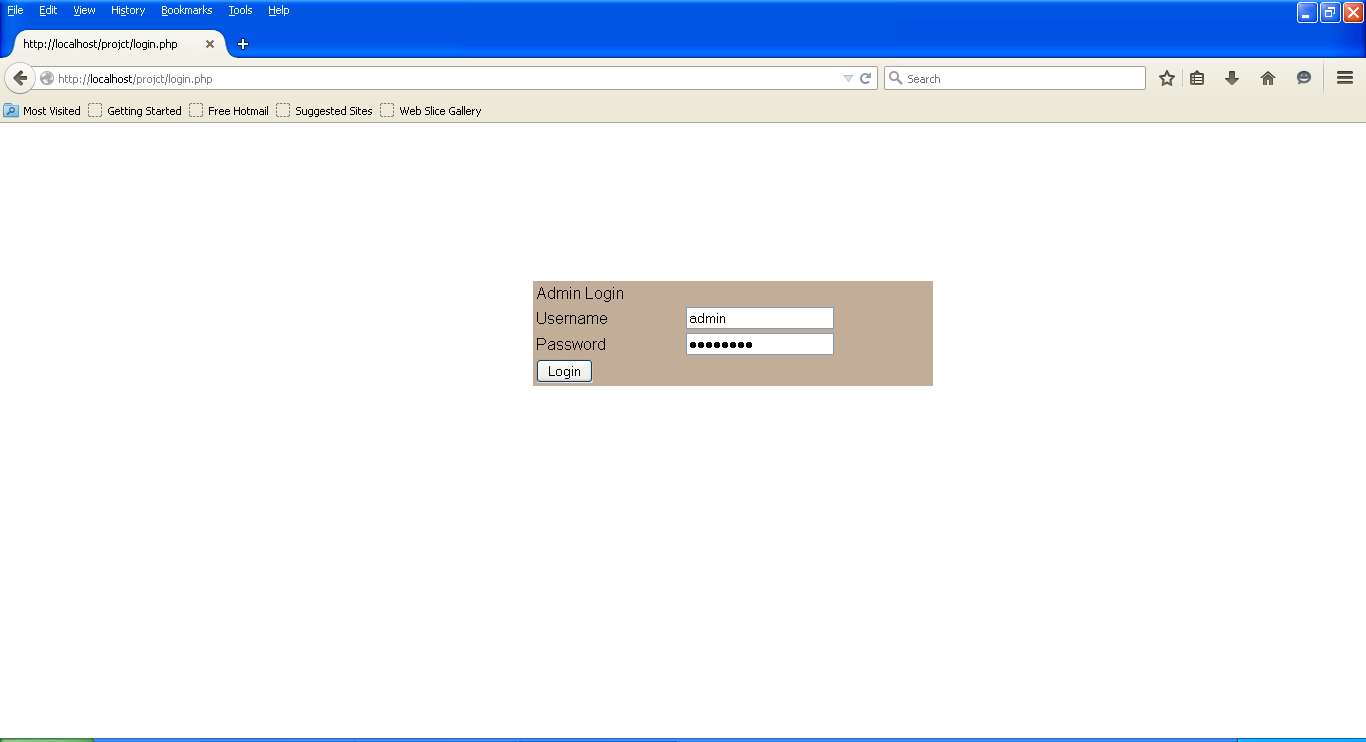
$username='root';

$password='';

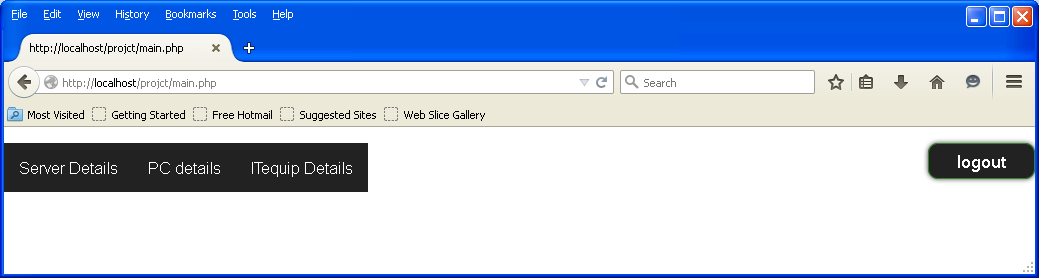
$server='localhost';?>

**6.1 Snapshots**

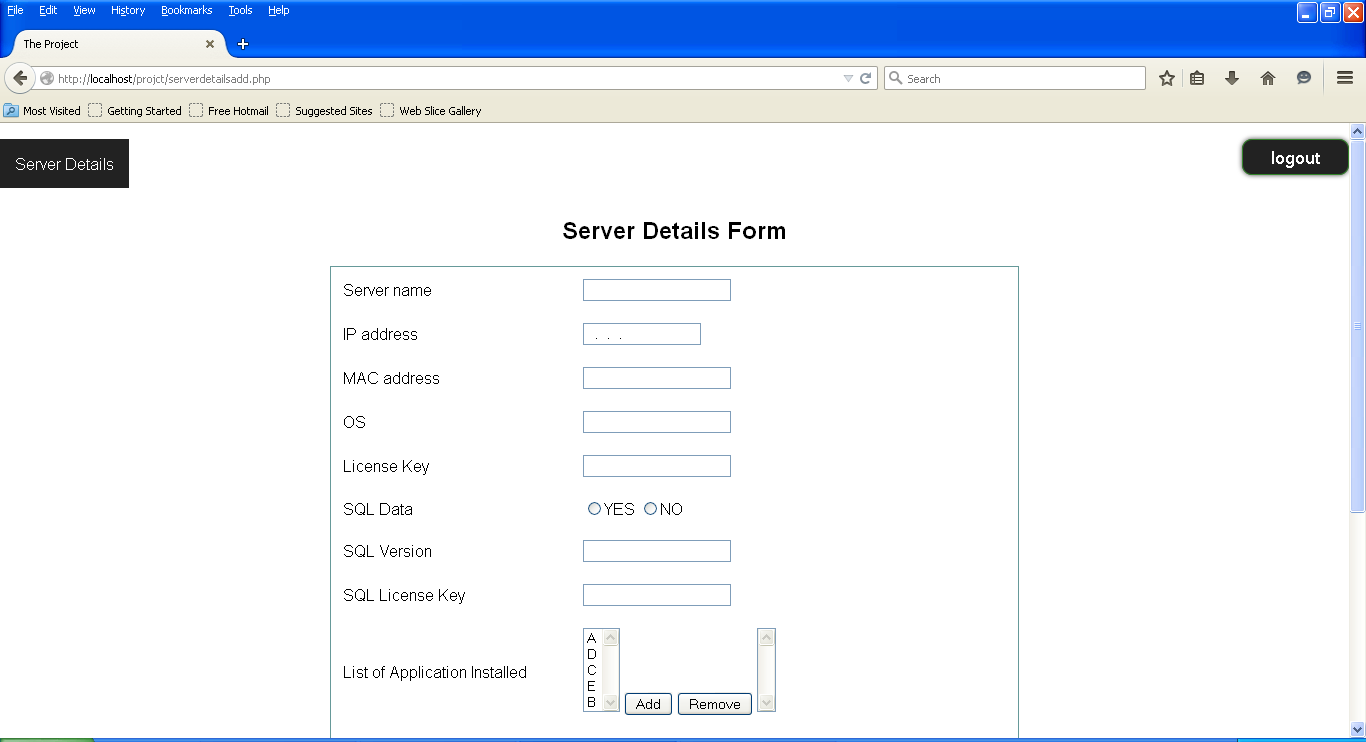
**login.php**

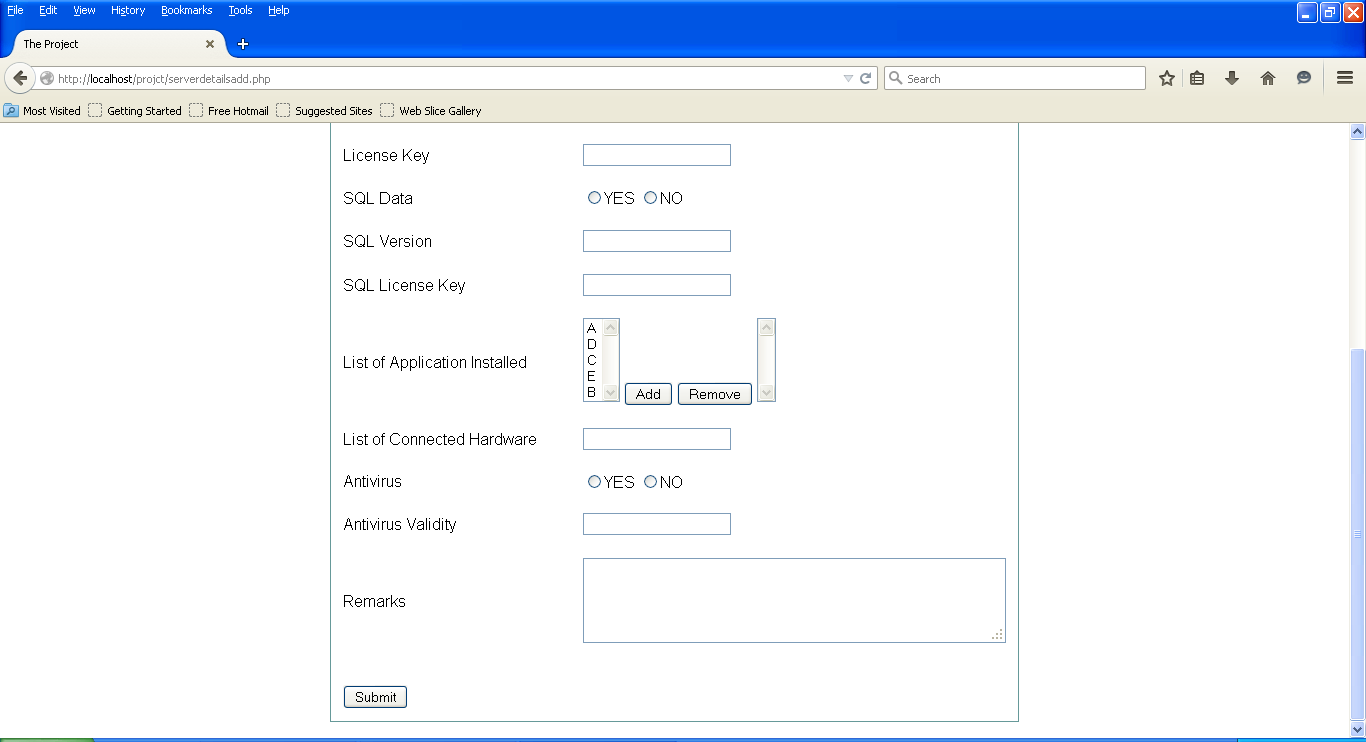
****

**Main.php**

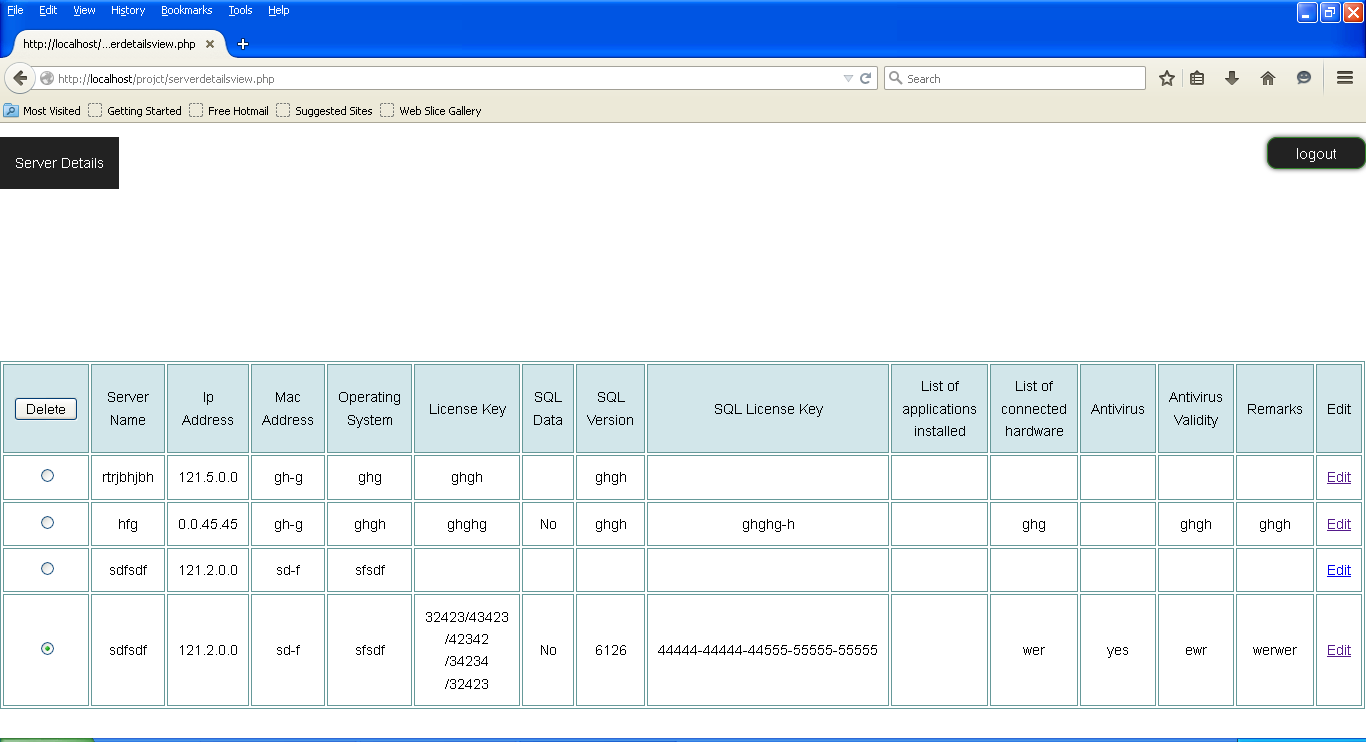
****

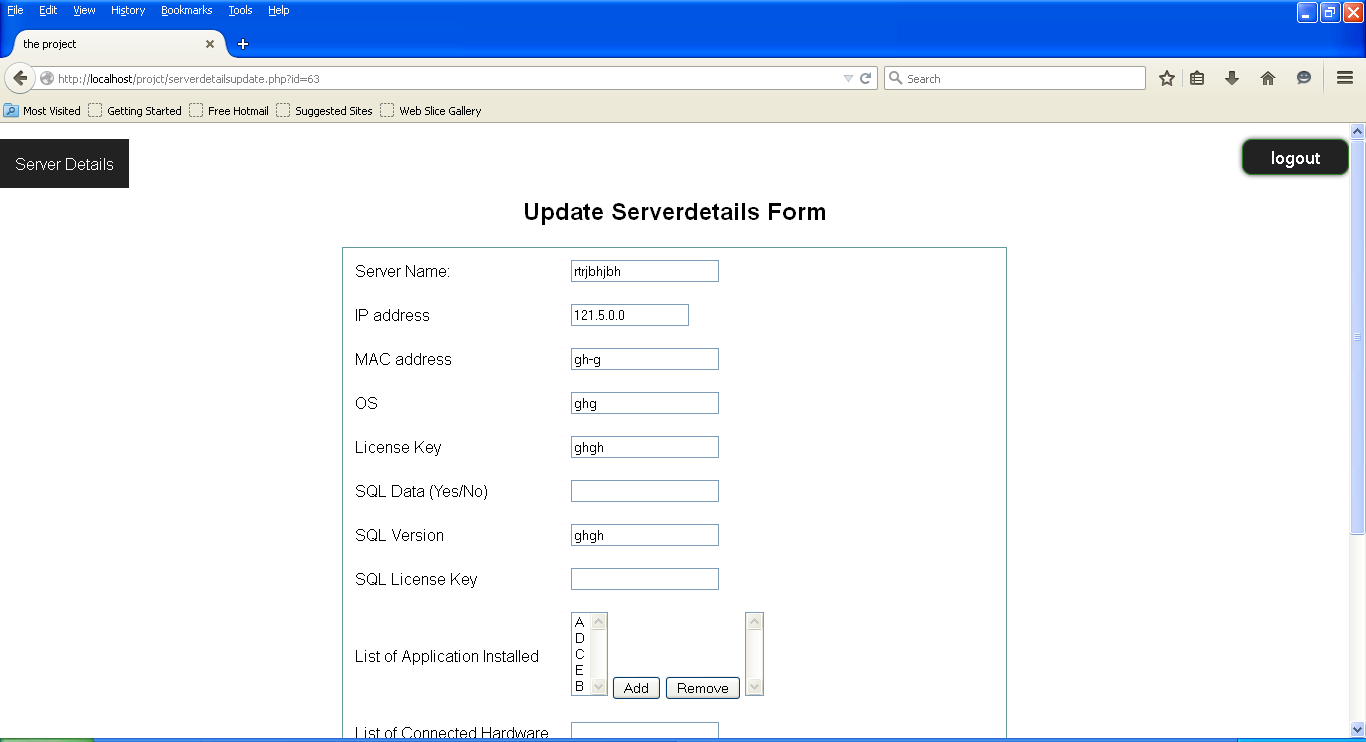
**Serverdetailsadd.php**

****

****

**Serverdetailsview.php**

**Serverdetailsupdate.php**

****

**7.0 Conclusion**

During the course period we have studied and observed how to make web sites using php. We made a project for storing the server and PC information into the SQL database. The project consist of different modules to add, update, delete and viewing the stored server and PC information. This project is useful for plant to keep the records of important systems in organized manner. Same program can be extended further to incorporate other important system information.

In that we have used different languages like

* javascript
* css
* php
* html
* jquery
* sql.

**10.0 References**

* [www.google.com](http://www.google.com)
* [www.w3schools.com](http://www.w3schools.com)
* www.developphp.com