**SOFTWARE REQUIRMENT SPECIFICATION**

**Software Requirements**:

 PHP 5.0

 APACHE HTTP Server

 Dreamweaver,FrontPage for Front End Programming

 Microsoft Windows or Linux

**Hardware Requirements**:

 Intel Pentium IV processor or equivalent or higher

 512 MB Ram or Higher

 20 GB HDD or Higher

 Network Connectivity

**DESIGN PHASE**

**1. Introduction**

**1.1) Scope and purpose**

The purpose of the design phase is to develop a clear understanding of

what the developer want people to gain from his/her project. As you the

developer work on the project, the test for every design decision should be

"Does this feature fulfill the ultimate purpose of the project?"

A purpose statement affects the design process by explaining what the

developer wants the project to do, rather than describing the project itself.

The Design Document will verify that the current design meets all of the

explicit requirements contained in the system model as well as the implicit

requirements desired by the customer.

**1.2) Overall System Design Objectives**

The overall system design objective is to provide an efficient, modular

design that will reduce the system’s complexity, facilitate change and

result in an easy implementation. This will be accomplished by designing

strongly cohesion system with minimal coupling. In addition, this document

will provide interface design models that are consistent user friendly and

will provide straight forward transition through the various system

functions.

**1.3) Structure of Design Document**

 *System Architecture Design* – The System architecture section has

detailed diagram of the system, server and client architecture.

 *Data Design* – The data Design include an ERD as well as Database

design.

 *Functional Design Description –* This section has the functional

partitioning from the SRS, and goes into great detail to describe each

function.

2. **System Architecture Design**

**2.1) System Architecture**

The SIMS is a system which contain major part which include: student

Detail, Student image and resume.

The user selects one of the available options as an input to the system.

According to the input by the user the system acts and the rest of the

functions are performed accordingly.The administartor can operate on any

student details.But the normal student or users can only access their

details of all the functionalities.

3. **Data Design**

**3.1) Entity Relationship Diagram :**

**Entity Relationship Diagram**

**4. Functional Design Description**

4.1 **Data Flow Diagram :**

**4.2 Decision Tree :**

5. **Conclusion**

Hence we can conclude that the design phase of the SIMS give us the

information of all the processes used in the project and their relation.

**TECHNOLOGY OVERVIEW**

The technology selected for implementing Student Information Management System

is PHP/MYSQL.Apache is used as the HTTP server.The development was done in a

‘windows’ environment using adobe dreamweaver CS5.

**PHP**

PHP is a general-purpose scripting language that is especially suited to serverside

web development where PHP generally runs on a web server.PHP code is

embedded into the HTML source document.Any PHP code in a requested file

is executed by the PHP runtime, usually to create dynamic web page content.It can

also be used for command-line scripting and client-side GUI applications. PHP can

be deployed on many web servers and operating systems, and can be used with

many relational database management systems (RDBMS). It is available free of

charge, and the PHP Group provides the complete source code for users to build,

customize and extend for their own use.

**MySQL**

MySQL is a relational database management system (RDBMS)[1] that runs as a

server providing multi-user access to a number of databases. MySQL is a popular

choice of database for use in web applications and is an open source product. The

process of setting up a MySQL database varies from host to host, however we will

end up with a database name, a user name and a password. Before using our

database, we must create a table. A table is a section of the database for storing

related information. In a table we will set up the different fields which will be used in

that table. Creating a table in phpMyAdmin is simple, we just type the name, select

the number of fields and click the ‘go’ button. we will then be taken to a setup screen

where you must create the fields for the database.Another way of creating databases

and tables in phpMyAdmin is by executing simple SQL statements.We have used

this method in order to create our database and tables.

**Apache**

The Apache HTTP Server is a web server software notable for playing a key role in

the initial growth of the World Wide Web. In 2009 it became the first web server

software to surpass the 100 million web site milestone. Apache is developed and

maintained by an open community of developers under the auspices of the Apache

Software Foundation. Since April 1996 Apache has been the most popular HTTP

server software in use. As of November 2010 Apache served over 59.36% of all

websites and over 66.56% of the first one million busiest websites.

**XAMPP**

XAMPP is a small and light Apache distribution containing the most common web

development technologies in a single package. Its contents, small size, and

portability make it the ideal tool for students developing and testing applications in

PHP and MySQL. XAMPP is available as a free download in two specific packages:

full and lite. While the full package download provides a wide array of development

tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills

Competition standards.The light version is a small package containing Apache HTTP

Server, PHP, MySQL, phpMyAdmin, Openssl, and SQLite.

**Obtaining and Installing XAMPP**

**PROJECT DESCRIPTION**

**Introduction**

Student Information Management System can be used by education institutes to

maintain the records of students easily. Achieving this objective is difficult using a

manual system as the information is scattered, can be redundant and collecting

relevant information may be very time consuming. All these problems are solved

using this project

The directory structure of the project is as follows:

.

**Description of root directory contents**

 **Images Directory :** This directory contains the images uploaded by the

students during registration process.Supported formats are the .jpg and .gif

files.

 **Resume Directory** : This Directory Contains resumes of students uploaded

during registration process of students.Files in this folder can be of .doc,.txt or

.pdf format.

 **Admin\_Edit\_Student\_Info.php** : Admin page for editing information of a

student.The administrator can change details of a student in this page.Though

facility of changing the image and resume are not yet provided but will be

provided in future versions of the project.

 **Admin\_Edit\_Student\_Info\_Handler.php** : Page handler for handling the

Admin\_Edit\_Student\_Info.php file.It writes the edited values in the database

on the server.

 **Admin\_Header.php** : Header file for pages accessible to administrator only.

 **Admin\_Home.php** : Home page for administrator after logging in process.

 **Admin\_Login.php** : Login page for administrator access.Shows appropriate

message for wrong username and/or password.

 **Admin\_Login\_handler.php** : Page handler for Admin\_Login.php page.It

checks the values provided with the values in the database.

 **Admin\_Student\_View\_Info.php** : Page to display student information to the

administrator.

 **Connect.php** : Page for database connectivity.It is used whenever database

values are required on the page.

 **DisplayAll.php** : Page to display all registered students to the

administrator.This facility is only available to the administrator.

 **Enable\_Disable\_handler.php** : Handler page for enabling/disabling of

students account facility.This functionality is only available to the

administrator.

 **Footer.php** : Footer file for all pages.

 **Header.php** : Header file for login page and homepage of the site.

 **Index.php** : Homepage of the website.

 **Logout.php** : Logout handler page.It Destroys all session variables thus

ending user session.

 **Search.php** : Search page to search students.It can only be used by

administrator.Students can be searched using different fields such as user

id,account status etc.

 **Search\_Result.php** : Page to display search results to the administrator.

 **Search\_View\_Result.php** : Page to display student information for any

selected search result.

 **Student\_Edit.php** : Page to edit student information.It can be accessed by

students.

 **Student\_Edit\_Handler.php** : Page handler for Student\_Edit.php.

 **Student\_Header.php** : Header file for student pages.

 **Student\_Home.php** : Home page for students after they log into their

respective accounts.

 **Student\_login.php** : Login Page for student login. Appropriate message is

displayed if the login is unsuccessful.

 **Student\_login\_handler.php** : Page handler for Student\_Login.php.It checks

the values provided with that in th the database.

 **Student\_Registration.php** : Student Registration page.The students enters

various details here for registration.

 **Student\_Registration\_handler.php** : Page Handler for handling the file

Student\_registration.php.It adds value to the student\_information table thus

creating a new user.

 **Student\_Reset\_Password.php** : Page for resetting password.It can be used

only by students.Administrator password can be changed only by changing

the values in the table directly.

 **Student\_Reset\_Password\_Handler.php** : Page handler for handling page

Student\_Reset\_Password.php.

 **Student\_View.php** : Page to display student profile with all the details of the

student.

 **Style.css** : Stylesheet for the whole site design.

 **Validation.js** : Javascript validations used for validation of form values.

Various form entries are validated at the client side using this file only.

**Description of database tables**

 **admin\_login :**

o user\_id : Stores user id of administrator(s).

o password : Stores password of the administrator(s).

o last\_login\_date : Stores the last login date of the administrator(s).

 **Student\_information :**

`

o student\_id : Stores user id of the student(s)

o student\_password : Stores password of the student(s)

o first\_name : Stores first name of the student(s)

o last\_name : Stores last name of the student(s)

o registration\_date : Stores the registration date of the student(s).

o gender : Stores the gender of the student(s).

o date\_of\_birth : Stores the date of birth of the student(s).

o student\_status : Stores the current status of the student account(s).

o contact\_no : Stores the contact number of the student(s).

o qualification : Stores student(s) qualification.

o city : Stores the city in which the student(s) lives.

o email1 : Stores primary email of the student(s).

o email2 : Stores secondary email of the student(s).

o address : Stores the address of the student(s).

o description : Stores description of the student(s).

o resume : Stores resume of students(s).

o image : Stores image of the student(s).

o last\_login\_date : Strores last login date of the student(s).

**Features**

The Website provides following functionalities to the users :

 **Administrator** :

o Login/Logout

o View student information

o Edit Student Information

o Enable/disable student accounts

o Search students

 **Student** :

o Login/Logout

o View profile

o Edit profile

o Change password

o Register new profile

**Source Code:-**

**Form.php**

<?php

session\_start();

$\_SESSION['message']=' ';

$mysqli = new mysqli('localhost','root','system','accounts') ;

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

{

if($\_POST['password']==$\_POST['confirmpassword'])

{

$username = $mysqli->real\_escape\_string($\_POST['username']);

$email=$mysqli->real\_escape\_string($\_POST['email']);

$password=md5($\_POST['password']);

$avatar\_path=$mysqli->real\_escape\_string('images/'.$\_FILES['avatar']['name']);

//checking required field is correct or not

//file type is image

if(preg\_match("!image!",$\_FILES['avatar']['type']))

{

//copy image to image folder

if(copy($\_FILES['avatar']['tmp\_name'],$avatar\_path))

{

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

$\_SESSION['avatar']=$avatar\_path;

$sql="INSERT INTO accounts(username,email,password,avatar)"

."VALUES('$username','$email','$password','$avatar\_path')";

}

//if query is successful,redirect to php page,done!

if($mysqli->query($sql)===true)

{

$\_SESSION['message']="Registeration successful! Added $username to database!";

header('location:welcome.php');

}

else{

$\_SESSION['message']='User could not be added to database!';

}

}

else{

$\_SESSION['message']='File upload failed!';

}

}

else{

$\_SESSION['message']='Please upload GIF,JPG OR PNG Images!';

}

}

else{

$\_SESSION['message']='Two password do not match!';

}

?>

<link href="//db.onlinewebfonts.com/c/a4e256ed67403c6ad5d43937ed48a77b?family=Core+Sans+N+W01+35+Light" rel="stylesheet" type="text/css"/>

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

<div class="body-content">

<div class="module">

<h1>Create an account</h1>

<form class="form" action="form.php" method="post" enctype="multipart/form-data" autocomplete="off">

<div class="alert alert-error"></div>

<input type="text" placeholder="User Name" name="username" required />

<input type="email" placeholder="Email" name="email" required />

<input type="password" placeholder="Password" name="password" autocomplete="new-password" required />

<input type="password" placeholder="Confirm Password" name="confirmpassword" autocomplete="new-password" required />

<div class="avatar"><label>Select your avatar: </label><input type="file" name="avatar" accept="image/\*" required /></div>

<input type="submit" value="Register" name="register" class="btn btn-block btn-primary" />

</form>

<br/><br/>

<h1>DELETING PORTION</h1>

<form class="form" action="formdel.php" method="post" enctype="multipart/form-data" autocomplete="off">

<div class="alert alert-error"></div>

<input type="text" placeholder="User Name" name="d\_username" required />

<input type="email" placeholder="Email" name="d\_email" required />

<input type="password" placeholder="Password" name="d\_password" autocomplete="new-password" required />

<input type="submit" value="DELETE" name="delete" class="btn btn-block btn-primary" />

<br/><br/>

</form>

<h1>VIEW ALL RECORDS</h1>

<form class="form" action="view.php" method="post" enctype="multipart/form-data" autocomplete="off">

<input type="submit" value="VIEW" class="btn btn-block btn-primary" name="view" />

</form>

</div>

</div>

**Formdel.php**

<?php

session\_start();

$\_SESSION['message']=' ';

$\_SESSION['d\_password']=$\_POST['d\_password'];

$\_SESSION['d\_email']=$\_POST['d\_email'];

$mysqli = new mysqli('localhost','root','system','accounts') ;

if(1)

{

header('location:delete.php');

}

?>

<link href="//db.onlinewebfonts.com/c/a4e256ed67403c6ad5d43937ed48a77b?family=Core+Sans+N+W01+35+Light" rel="stylesheet" type="text/css"/>

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

<div class="body-content">

<div class="module">

<h1>Create an account</h1>

<form class="form" action="form.php" method="post" enctype="multipart/form-data" autocomplete="off">

<div class="alert alert-error"></div>

<input type="text" placeholder="User Name" name="username" required />

<input type="email" placeholder="Email" name="email" required />

<input type="password" placeholder="Password" name="password" autocomplete="new-password" required />

<input type="password" placeholder="Confirm Password" name="confirmpassword" autocomplete="new-password" required />

<div class="avatar"><label>Select your avatar: </label><input type="file" name="avatar" accept="image/\*" required /></div>

<input type="submit" value="Register" name="register" class="btn btn-block btn-primary" />

</form>

<br/><br/>

<h1>DELETING PORTION</h1>

<form class="formd" action="delete.php" method="post" enctype="multipart/form-data" autocomplete="off">

<div class="alert alert-error"></div>

<input type="text" placeholder="User Name" name="d\_username" required />

<input type="email" placeholder="Email" name="d\_email" required />

<input type="password" placeholder="Password" name="d\_password" autocomplete="new-password" required />

<input type="submit" value="DELETE" name="delete" class="btn btn-block btn-primary" />

<br/><br/>

</form>

<h1>VIEW ALL RECORDS</h1>

<form class="form" action="delete.php" method="post" enctype="multipart/form-data" autocomplete="off">

<input type="submit" value="VIEW" class="btn btn-block btn-primary" name="view" />

</form>

</div>

</div>

**Exp.php**

<link rel="stylesheet" href="form.css">

<div class="body content">

<div class="welcome">

<?php session\_start();

$mysqli= new mysqli('localhost','root','system','accounts');

$email=$mysqli->real\_escape\_string($\_SESSION['d\_email']);

$password=md5($\_SESSION['d\_password']);

//$sql=mysql\_query("SELECT \* FROM accounts",$mysqli);

//$initial=mysql\_num\_rows($sql);

//$sql=mysql\_query("DELETE FROM accounts WHERE email='$email' AND password='$password'",$mysqli);

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

//$after=mysql\_num\_rows($sql);

if($mysqli->query("DELETE FROM accounts WHERE email='$email' AND password='$password'")===TRUE)

{

echo'<div class="alert alert-success"><h1>DELETION SUCCESSFUL</h1></div><br/><IMG src="project\thumbu.png">';

}

else

{

echo'<div class="alert alert-success"><h1>DELETION NOT SUCCESSFULL</h1></div><br/><IMG src="project\thumbd.png">';

}

?>

</div>

</div>

<?php

// after deletion tesult

$sql="SELECT username,avatar,email FROM accounts";

$result = $mysqli->query($sql);// $result = sql result objecct

?>

<div id="registered">

<span>ALL registered users</span><div id="registered">

<span>ALL registered users</span>

<?php

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

echo "<div class='userlist'><span>$row[username]</span><br/><span>$row[email]</span><br/>";

echo"<img src='$row[avatar]'></div>";

}

?>

</div>

**View.php**

<link rel="stylesheet" href="form.css">

<?php session\_start(); ?>

<div class="body content">

<div class="welcome">

<div class="alert alert-success"><?= $\_SESSION['message']?></div>

<span class="user"><img src='project\N.png alt="image cat load"'</span>

Welcome<span class="user"> GUEST</span>

<?php

$mysqli= new mysqli('localhost','root','system','accounts');

$sql="SELECT username,avatar,email FROM accounts";

$result = $mysqli->query($sql);// $result = sql result objecct

?>

<div id="registered">

<span>ALL registered users</span>

<?php

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

echo "<div class='userlist'><span>$row[username]</span><br/><span>$row[email]</span><br/>";

echo"<img src='$row[avatar]'></div>";

}

?>

</div>

**Welcome.php**

<link rel="stylesheet" href="form.css">

<?php session\_start(); ?>

<div class="body content">

<div class="welcome">

<div class="alert alert-success"><?= $\_SESSION['message']?></div>

<span class="user"><img src='<?=$\_SESSION['avatar']?>'</span>

Welcome <span class="user"><?=$\_SESSION['username']?></span>

<?php

$mysqli= new mysqli('localhost','root','system','accounts');

$sql="SELECT username,avatar,email FROM accounts";

$result = $mysqli->query($sql);// $result = sql result objecct

?>

<div id="registered">

<span>ALL registered users</span>

<?php

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

echo "<div class='userlist'><span>$row[username]</span><br/><span>$row[email]</span><br/>";

echo"<img src='$row[avatar]'></div>";

}

?>

</div>

**Delete.php**

<link rel="stylesheet" href="form.css">

<div class="body content">

<div class="welcome">

<?php session\_start();

$mysqli= new mysqli('localhost','root','system','accounts');

$email=$mysqli->real\_escape\_string($\_SESSION['d\_email']);

$password=md5($\_SESSION['d\_password']);

//$sql=mysql\_query("SELECT \* FROM accounts",$mysqli);

//$initial=mysql\_num\_rows($sql);

//$sql=mysql\_query("SELECT\* FROM accounts WHERE email='$email' AND password='$password'",$mysqli);

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

//$after=mysql\_num\_rows($sql);

if($mysqli->query("SELECT\* FROM accounts WHERE email='$email' AND password='$password'")===true)

{ echo"<h1>RECORD PRESENT</h1>";

if($mysqli->query("DELETE FROM accounts WHERE email='$email' AND password='$password'")===TRUE)

{

echo'<div class="alert alert-success"><h1>DELETION SUCCESSFUL</h1></div><br/><IMG src="project\thumbu.png">';

}

else

{

echo'<div class="alert alert-success"><h1>DELETION NOT SUCCESSFULL</h1></div><br/><IMG src="project\thumbd.png">';

}

}

else

{

echo"<h1>RECORD NOT PRESENT</h1></div><br/>";

}

?>

</div>

</div>

<?php

// after deletion tesult

$sql="SELECT username,avatar,email FROM accounts";

$result = $mysqli->query($sql);// $result = sql result objecct

?>

<div id="registered">

<span>ALL registered users</span><div id="registered">

<span>ALL registered users</span>

<?php

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

echo "<div class='userlist'><span>$row[username]</span><br/><span>$row[email]</span><br/>";

echo"<img src='$row[avatar]'></div>";

}

?>

</div>

**SNAPSHOTS**

This is the homepage/indexpage of the site.

This is the student login page.

This is the registration page where students can register online.

After successful registration the user account needs to be first enabled by the

administrator.The students can then login into their accounts with the id and

password they choose during registration.The above page shows student login page.

After logging in the student home page is opened as shown above.

The student can view their profile.The above page shows the user profile.The link to

the resume is shown if a resume was uploaded during registration otherwise its not

shown.

Students can edit their profiles by using the edit profile option on their homepage.the

above page is used for editing student information.

The students have the option to change their password.They need their old

password in order to change the password.

After successful operation a password change notification is displayed to the user.

The above page is the administrator login page. It requires the administrator

username and password.

This is the homepage of administrator.

This is the displayall page wherein the administrator can see all the registered

students.He can enable/disable their accounts as well as view/edit students

information.

This page shows the change in account status.after successful change a message is

displayed to the administrator.

This is the search page and it provides search function to the administrator.here we

search for all the students with theirs accounts enabled.

This is the search result display page.the administrator can view profile of any of the

searched students from here.If there are no results then an appropriate message is

displayed.

**SCOPE OF THE PROJECT**

 The Student Information Management System(SIMS) can be enhanced to

include some other functionality like marks,attendance management.

 Talent management of students based on their performance evaluation

can be added.

 Social networking can also be added wherein students can interact with

each other.

 Online class functionality can be added.

 Can evolve as an online institution.

 Functionality of chat and messages can be added.

 Online exam functionality can be added.

 Online resume builder functionality can also be added.

**CONTRIBUTION IN THE PROJECT**

Student information management system lead to a better organization structure

since the information management of the students is well structured and also lead to

better as well as efficient utilization of resources.

Student Information Management System can be used by education institutes to

maintain the records of students easily. Achieving this objective is difficult using a

manual system as the information is scattered, can be redundant and collecting

relevant information may be very time consuming. All these problems are solved

using this project

Our project Student Information Management System was developed by all three of

us.We, a team of three persons took a step by step approach in order to reach our

goal.We applied the knowledge we gained during our training period at **EN**

**Technologies Pvt. Ltd.** and developed this project **“STUDENT INFORMATION**

**MANAGEMENT SYSTEM”**.

**Bibliography**

PHP book by Vasvani (TMH publications).

Beginning PHP5 by WROX.

www.google.com.

www.wikipedia.com

www.w3schools.com

Informatics practices by Sumita Arora.

Head First PHP & MySQL by Lynn Beighley and Michael Morrison(O’Reilly)