STUDENT MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted By

AMIT KUMAR
University Roll No 1900290140006

AKHIL VERMA
University Roll No 1900290140004

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Mr. SD MISHRA

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

Session 2020-2021

PREFACE

This project **“Student Management System”** provides us a simple interface for maintainance of student information.It can be used by educational institutes or colleges 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.

Throughout the project the focus has been on presenting information in an easy and intelligible manner. The project is very useful for those who want to know about Student Information Management Systems and want to develop softwares/websites based on the same concept.

The project provides facilities like online registration and profile creation of students thus reducing paperwork and automating the record genreration process in an educational institution.

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report.

We are privileged to express our sense of gratitude to our respected teacher **Mr. SD MISHRA** whose unparalleled knowledge, moral fiber and judgment along with his know-how, was an immense support in completing the project.

We are also grateful to **Dr. AJAY KUMAR SHRIVASTAVA**,the Head of Department, Information Technology, for the brainwave and encouragement given.

We take this opportunity also to thank our friends and contemporaries for their cooperation and compliance.

Amit Kumar
Akhil Verma

TABLE OF CONTENTS

CERTIFICATE

Certified that **Amit Kumar (1900290140006) and Akhil Verma (19002900140004)**

have carried out the project work having **“Student Management System”** for

Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Amit Kumar (1900290140006)

Akhil Verma (1900290140004)

Signature of External Examiner **Signature of Project Supervisor**

Dr. Ajay kumar Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

SYNOPSIS

Abstract

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.

Name of the Project: Student

Management System

Objectives:

- Online registration of students
- Maintenance of student records
- Searching student records

Users Views:

- Administrator
- Student

Platform

Operating Systems: Microsoft Windows

Technologies Used:

- Front End: HTML and Javascript

- Web designing language: PHP
- RDBMS(Back end): MySQL

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

SOFTWARE REQUIRMENT SPECIFICATION

1. Introduction

1.1 Purpose:

The objective of **Student information System** is to allow the administrator of any organization to edit and find out the personal details of a student and allows the student to keep up to date his profile .It'll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So all the information about an student will be available in a few seconds.

Overall, it'll make Student Information Management an easier job for the administrator and the student of any organization.

The main purpose of this SRS document is to illustrate the requirements of the project **Student information System** and is intended to help any organization to maintain and manage its student's personal data.

1.2 Scope :

Without a **Student information System**, managing and maintaining the details of the student is a tedious job for any organization.

Student Information system will store all the details of the students including their background information, educational qualifications, personal details and all the information related to their resume .

Login module: Login module will help in authentication of user accounts .Users who have valid login id and password can only login into their respective accounts.

Search module: Suppose there are hundreds of students and from this we have to search a particular student and we know the name of the student .In manual system it is a tedious task though we know the name of the student, but using this module we can easily search the student by specifying the name of the student in the search criteria. Thus this module will help the administrator in searching the student with various criteria easily.

Registration Module and Account Management: This module will help the student get registered from anywhere if internet is present .This module will really simplify the task of on paper registration. Also after successful registration the user can update information and change their password as and when required.

User Management: This module will help the administrator in enabling/disabling a user account and updating user information as required.

Purpose of project is to maintain details of the students such as storing information about:

- Student id
- Student password
- Student name
- Student DOB
- Student mailing address
- Gender

- Registration date

- Student status

- Contact no

- Qualification

- City

- Resume

- Image

1.3 Definitions, Acronyms and Abbreviations :

- **Personal details:** Details of student such as user id, phone number, address, image, resume, e-mail address etc.

- **Contact details:** Details of contact associated with the student.

- **SRS:** System requirement Specification

- **WWW**: World Wide Web
- **Administrator:** A Login Id representing the user is an administrator & can access all the records details

1.4 Technologies :

- PHP.
- MYSQL
- JAVASCRIPT
- HTML
- CSS

1.5 Overview:

The rest of this SRS is organized as follows:
Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software.
Section 3 gives specific requirements which the software is expected to deliver. Some performance requirements and constraints are also given and deal with other Non-Functional Requirements.
Section 4 deals with External Interface Requirements like Hardware and Software Interface.

2. OVERALL DESCRIPTION

2.1 Product Perspective :

The website **Student Information System** is aimed towards recording a considerable number of student records and needs online assistance for managing records of students. Website should be user-friendly, 'quick to learn' and reliable website for the above purpose.

Student Information System is intended to be a stand-alone product and should not depend on the availability of other website. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the website.

2.2 Product Functions :

There are two different users who will be using this product:

- Administrator who can view and edit the details of any students.
- Students who can view their details as well as they can edit their details.

The features that are available to the Administrator are:

- An Administrator can login into the system and perform any of the available operations.
- Can enable/disable student.
- Can edit student information to the database.
- Can make search for a specific student.
- Can access all the details of the student.

The features that are available to the student are:

- Student can login into the system and can perform any of the available options.
- Can view his/her personal details.
- Can edit his/her personal details ⍰ Can upload his/her resume.
- Can upload his/her image.

2.3 User Classes and Characteristics :

There are mainly two kinds of users for the product.

The users include:

- Administrator

- Student

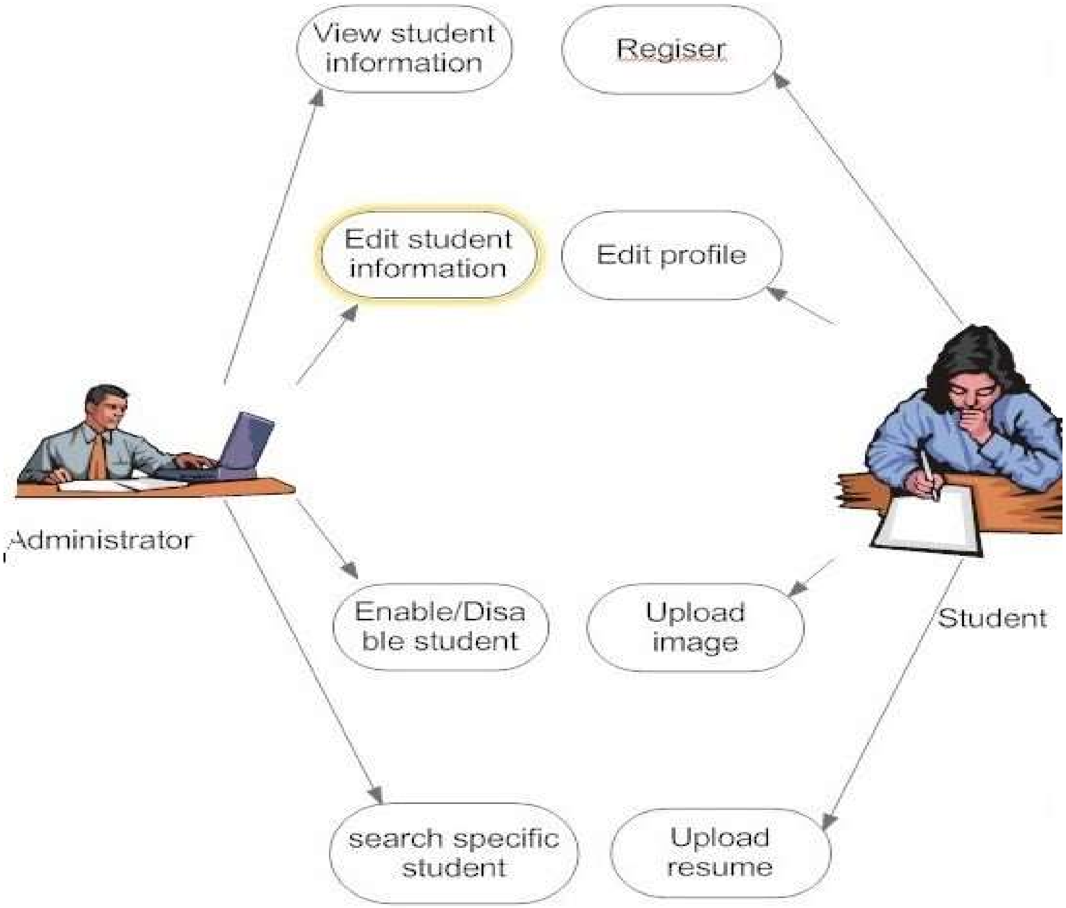
2.4 Operating Environment :

The product can run on any browser.

2.5 Constraints :

- Every user must be comfortable using computer.
- All operations are in English so user must have basic knowledge of English.

2.6 USE CASE MODEL :



Use Case Model

1. **Administrator:** Responsible for managing student records.

- Login into the website
- Update student details
- Search student details
- Display student details
- Enable/Disable student

2. **Student:** Has the access rights to view and edit their personal details.

- Login into the website
- Display student details
- Edit their details
- Upload their images
- Upload their resumes

2.7 Assumptions & dependencies

- Administrator is created in the system already.
- Roles and tasks are predefined.

3 Specific Requirements :

3.1 Use Case Reports

1. **Administrator:** Responsible for managing student details.

Use-case: Login into the website

Goal in context: Gain access to the website

Brief Description: This use case is used when the administrator wants to access the website to enable/disable/update the personal details of the student.

Preconditions: The Administrator must be logged onto the website in order for this use case to begin.

Basic Flow:

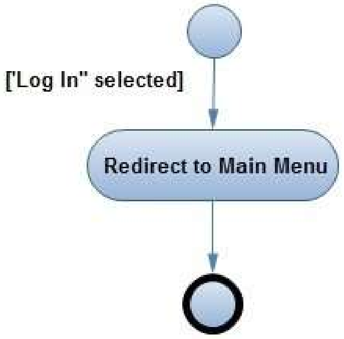
- The Website prompts the administrator for the user name and password.
- The Administrator enters the user name and password.
- The Website verifies the password and sets the user's authorization.
- The Administrator is given access to the Website to perform his tasks.

Alternative Flow:

- The administrator enters invalid username and password then he

  will not be allowed to enter the website**.**

Post conditions: The website state is unchanged by this use case.



Use Case Report- Login into the website

Use Case : Display student details

Goal in context: View the details of a student

Brief Description: This use case is used when the administrator wants to view the personal details of the students already existing in the database on the screen.

Preconditions:

- The Administrator must be logged into the system in order for this use case to begin
- The details of the student must pre-exist in the database
- The student id must be entered correctly.

Basic Flow:

- The Administrator logs onto the System.
- The Administrator search the student from following keys:-

  ○ Student id ○
  First/last name ○
  Registration date ○
  status

- The System prompts for the student detail from one of the above keys.
- The student details are displayed on the screen.

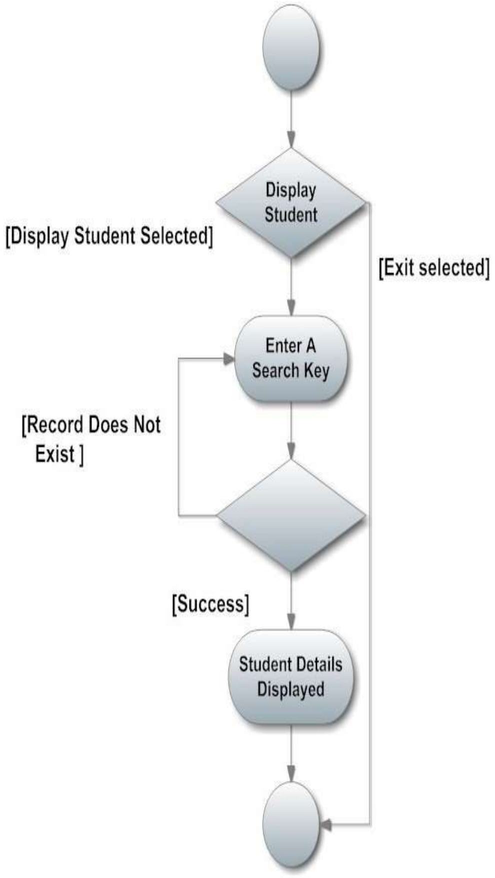
Alternative Flow:

Student Not Found

If in the **Display a student** sub-flows, a student with the specified id number does not exist, The system displays an error message. The Administrator can then enter a different id number or cancel the operation, at which point the use case ends.

Post conditions:

The student details are displayed on the screen already existing in the system. The state of the system remains unchanged.



Use Case Report-Display Student Details

Use Case : Edit student details

Goal in context: Edit the details of a student

Brief Description: This use case is used when the administrator wants to edit the personal details of the himself/herself already existing in the database.

Preconditions:

- The Administrator must be logged into the system in order for this use case to begin.
- The details of the student must pre-exist in the database

Basic Flow:

- The Administrator logs onto the System.
- The Administrator can edit following keys:-
    - First/last name
    - Gender
    - DOB
    - Contact no
    - Qualification
    - City
    - Email1
    - Email2
    - Address
    - Description
- The Website updates the database according to edited details.

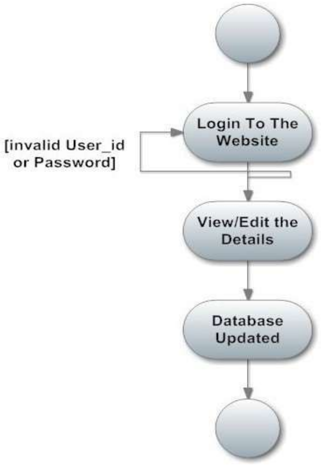
- The student details are edited in the database.

Alternative Flow:

There is no alternative flow of this use case diagram.

Post conditions:

The student details get updated in the database**.**



Use Case Report- Edit student detail into the website

2. Student

Use Case : student registration

Goal in context: Registration of a student

Brief Description: This use case is used when the student register himself/herself in the database online.

Preconditions:

- The Student must accessed the website in order for this use case to begin.
- The user id must be unique and entered correctly.

Basic Flow:

- The Student enters into the website.
- The student fill his/her details from the following keys:-

○ Student id

○ password

○ First/last

name ○ Status

○ Gender ○

DOB

○ Contact no

○ Qualification

○ City ○

Email1 ○

Email2 ○

Address ○

Description ○

Resume ○

Image

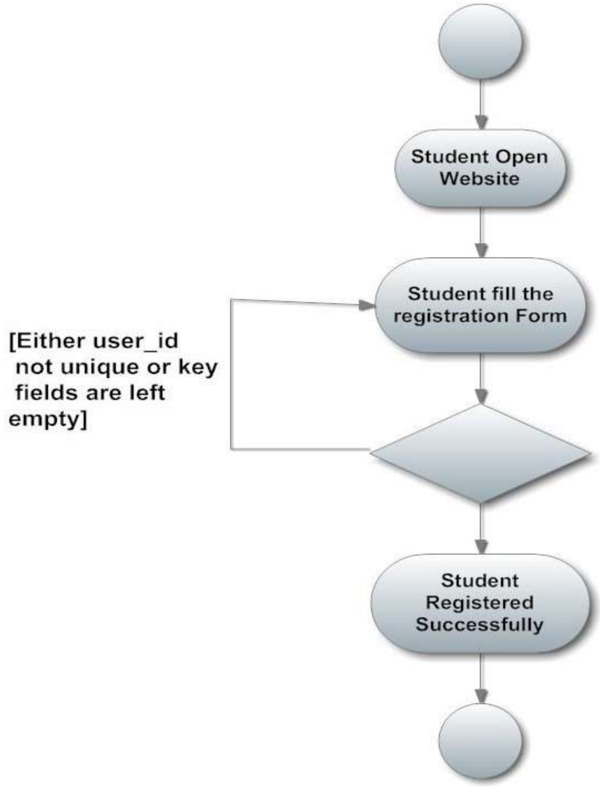
- The System details are added to the database.
- The student details are displayed on the screen.

Alternative Flow:

User ID not unique: if the user id entered is not unique then it will show an error message.

Post conditions:

The student get registered on the website and to login into that particular the administrator must enable it.


Student Open Website
Student fill the registration Form
[Either user_id not unique or key fields are left empty]
Student Registered Successfully

Use Case Report- Register student on website

Use-case: Login into the website

Goal in context: Gain access to the website

Brief Description: This use case is used when the student wants to access the website

Preconditions: The Administrator must enabled the particular student onto the website in order for this use case to begin.

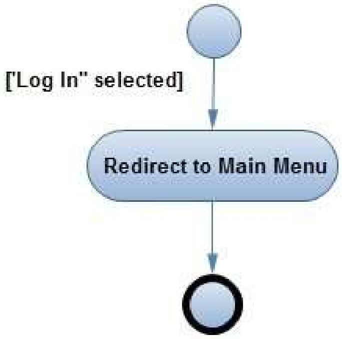
Basic Flow:

- The website prompts the student for the user name and password.
- The Student enters the user name and password.
- The website verifies the password and sets the user's authorization.
- The Student is given access to the website to perform his tasks.

Alternative Flow:

- The Student enters invalid username and password then he will not be allowed to enter the website**.**

Post conditions: The website state is unchanged by this use case.



Use Case Report- Login into the system

Use Case : Edit student details

Goal in context: Edit the details of a student

Brief Description: This use case is used when the student wants to edit the personal details of the himself/herself already existing in the database.

Preconditions:

- The Student must be logged into the system in order for this use case to begin.

- The details of the student must pre-exist in the database

- The student must be enabled by administrator.

Basic Flow:

- The Student logs onto the System.
- The Student can edit following keys:-
  - First/last name
  - Gender ○ DOB
  - Contact no ○ Qualification ○ City ○ Email1 ○ Email2 ○ Address ○ Description

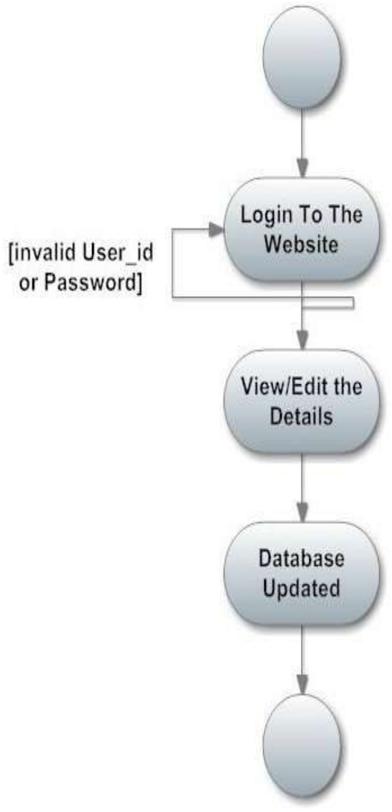
- The Website updates the database according to edited details.
- The student details are edited in the database.

Alternative Flow:

There is no alternative flow of this use case diagram.

Post conditions:

The student details get updated in the database**.**



Use Case Report- Edit Student Details Into Database

3.2 Functional Requirements :

- The Administrator will be given more powers (enable/disable/ update) than other users.
- It will be ensured that the information entered is of the correct format. For example name cannot contain numbers. In case if incorrect form of information is added, the user will be asked to fill the information again.
- The system can be accessed anytime.

3.3 Non- Functional Requirement :

3.2.1. Performance Requirements:

The proposed system that we are going to develop will be used as the Chief performance system for providing help to the organization in managing the whole database of the student studying in the organisation. Therefore, it is expected that the database would perform functionally all the requirements that are specified.

3.2.2. Safety Requirements:

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

3.2.3. Security Requirements:

We are going to develop a secured database. There are different categories of users namely Administartor ,Student who will be viewing either all or some specific information form the database.

Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, append etc. All other users only have the rights to retrieve the information about database.

3.4 Conclusion :

This SRS has given all the details of the application need to be built.

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

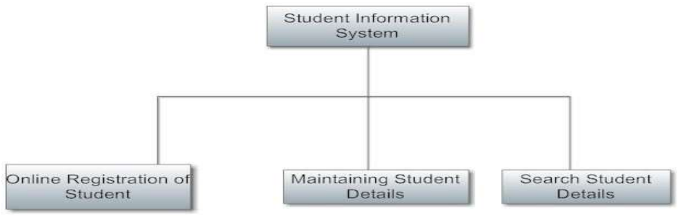
- *System Architecture Design* – The System architecture section has detailed diagram of the system, server and client architecture.
- *Data Design* – The data Design include an ERD as well as Database design.
- *Functional Design Description* – This section has the functional partitioning from the SRS, 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.

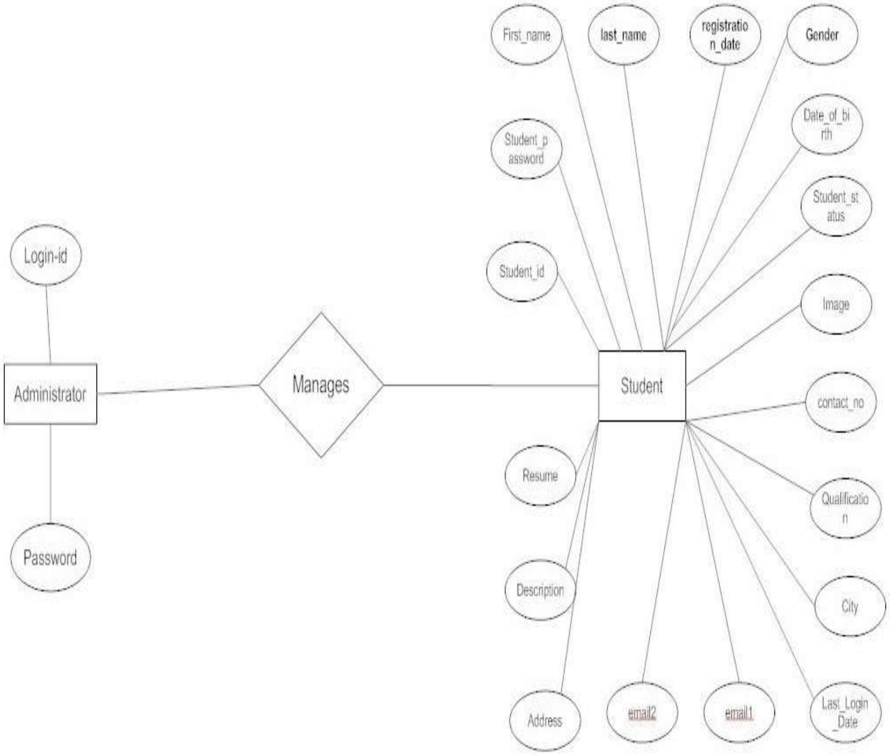
The user selects one of the available options as an input to the system. According to the input by the user the system acts and the rest of the functions are performed accordingly.The administartor can operate on any student details.But the normal student or users can only access their details of all the functionalities.


Student Information System
Online Registration of Student
Maintaining Student Details
Search Student Details

Architecture Diagram

3. Data Design

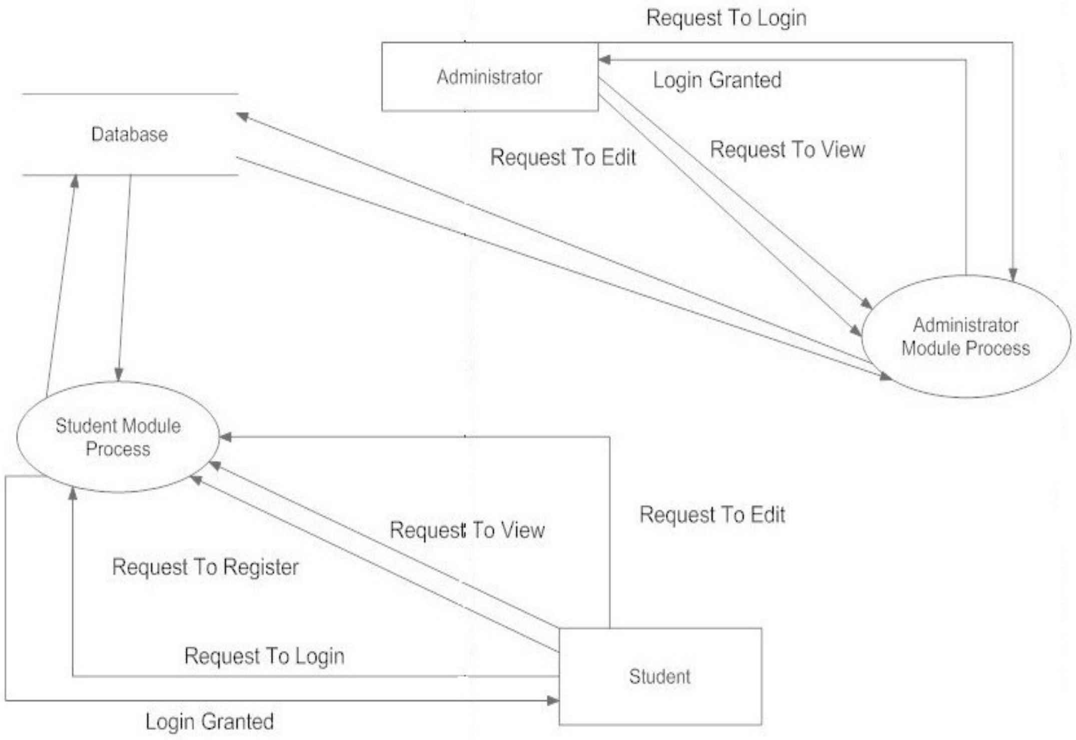
3.1) Entity Relationship Diagram :



Entity Relationship Diagram

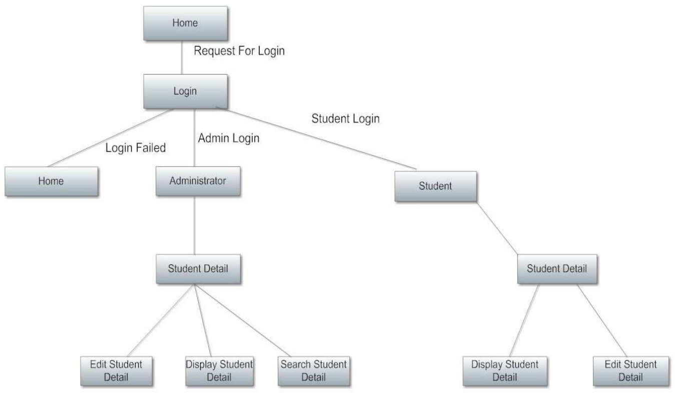
4. Functional Design Description

4.1 Data Flow Diagram :



Data Flow Diagram

4.2 Decision Tree :



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.

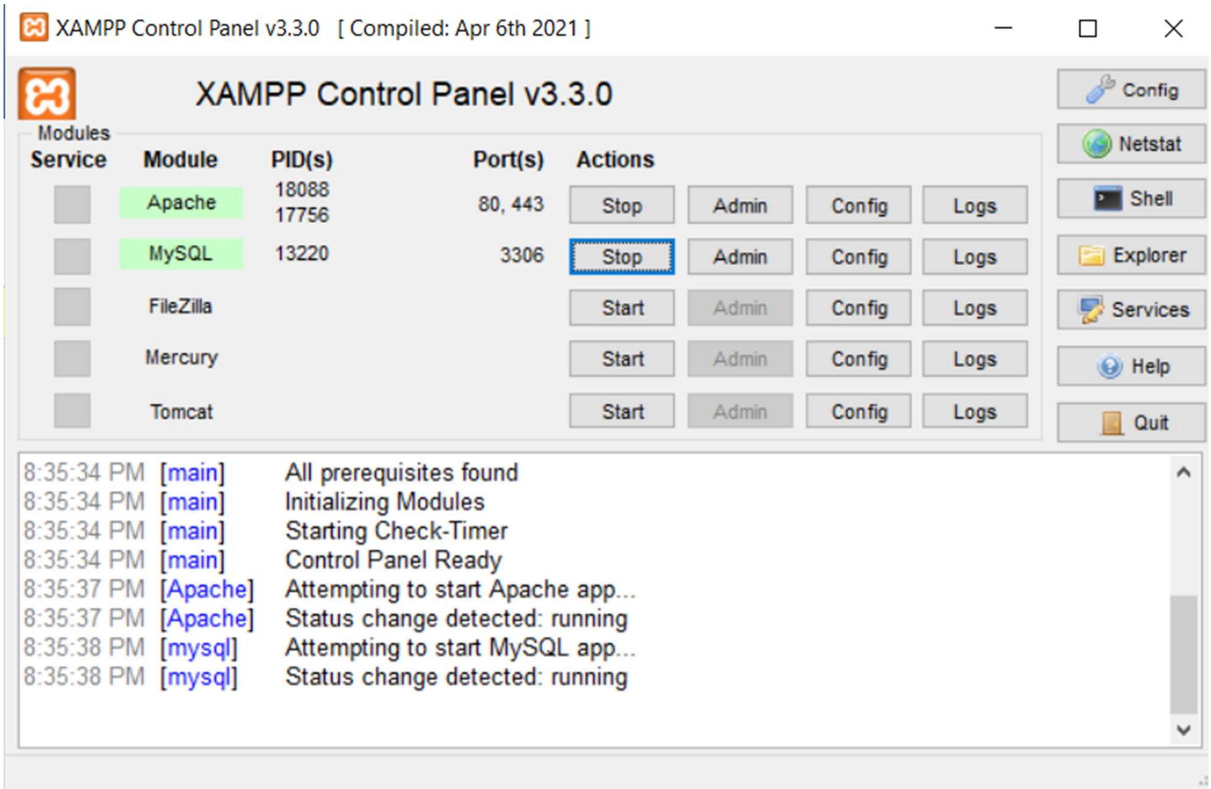
XAMPP

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

As previously mentioned, XAMPP is a free package available for download and use for various web development tasks. All XAMPP packages and add-ons are distributed through the Apache Friends website at the address: http://www.apachefriends.org/. Once on the website, navigate and find the Windows version of XAMPP and download the self-extracting ZIP archive. After downloading the archive, run and extract its contents into the root path of a hard disk or USB drive. For example, the extract path for a local Windows installation would simply be C:\. If extracted properly we will notice a new xampp directory in the root of your installation disk. In order to test that everything has been installed correctly, first start the Apache HTTP Server by navigating to the xampp directory and clicking on the apache_start.bat batch file.

Next we will test if the server is running correctly by opening an internet browser and typing http://localhost/ into the address bar. If configured correctly, we will be presented with a screen similar to that of the one below.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	18088 17756	80, 443	Stop	Admin	Config	Logs
	MySQL	13220	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
8:35:34 PM  [main]      All prerequisites found
8:35:34 PM  [main]      Initializing Modules
8:35:34 PM  [main]      Starting Check-Timer
8:35:34 PM  [main]      Control Panel Ready
8:35:37 PM  [Apache]    Attempting to start Apache app...
8:35:37 PM  [Apache]    Status change detected: running
8:35:38 PM  [mysql]     Attempting to start MySQL app...
8:35:38 PM  [mysql]     Status change detected: running
```

XAMPP splash screen.

In order to stop all Apache processes we do not close the running terminal application, but instead run another batch file in the xampplite directory called apache_stop.bat.

Creating a Database and Inserting Data

Now that we have run and tested Apache and PHP, the next step is running MySQL and creating a database and table which will hold information to be used by our website. In order to start MySQL, navigate to the xampp directory and run the mysql_start.bat batch file.The XAMPP package contains an application called phpMyAdmin which allows developers to administer and maintain MySQL databases.We will be using phpMyAdmin to create a database and table, and enter test data. Before testing phpMyAdmin, make sure that both Apache and MySQL are running by opening their respective batch files: apache_start.bat and mysql_start.bat. Along with Apache and MySQL running in the background, we type http://localhost/phpMyAdmin/ into our web browser. If successful we will be presented with a phpMyAdmin start page similar to the one shown below.

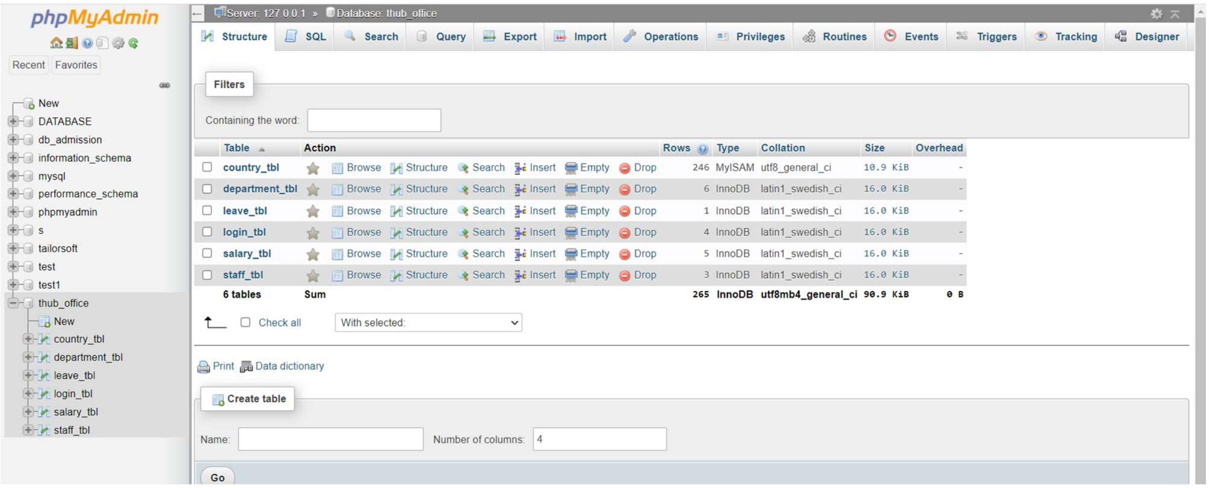
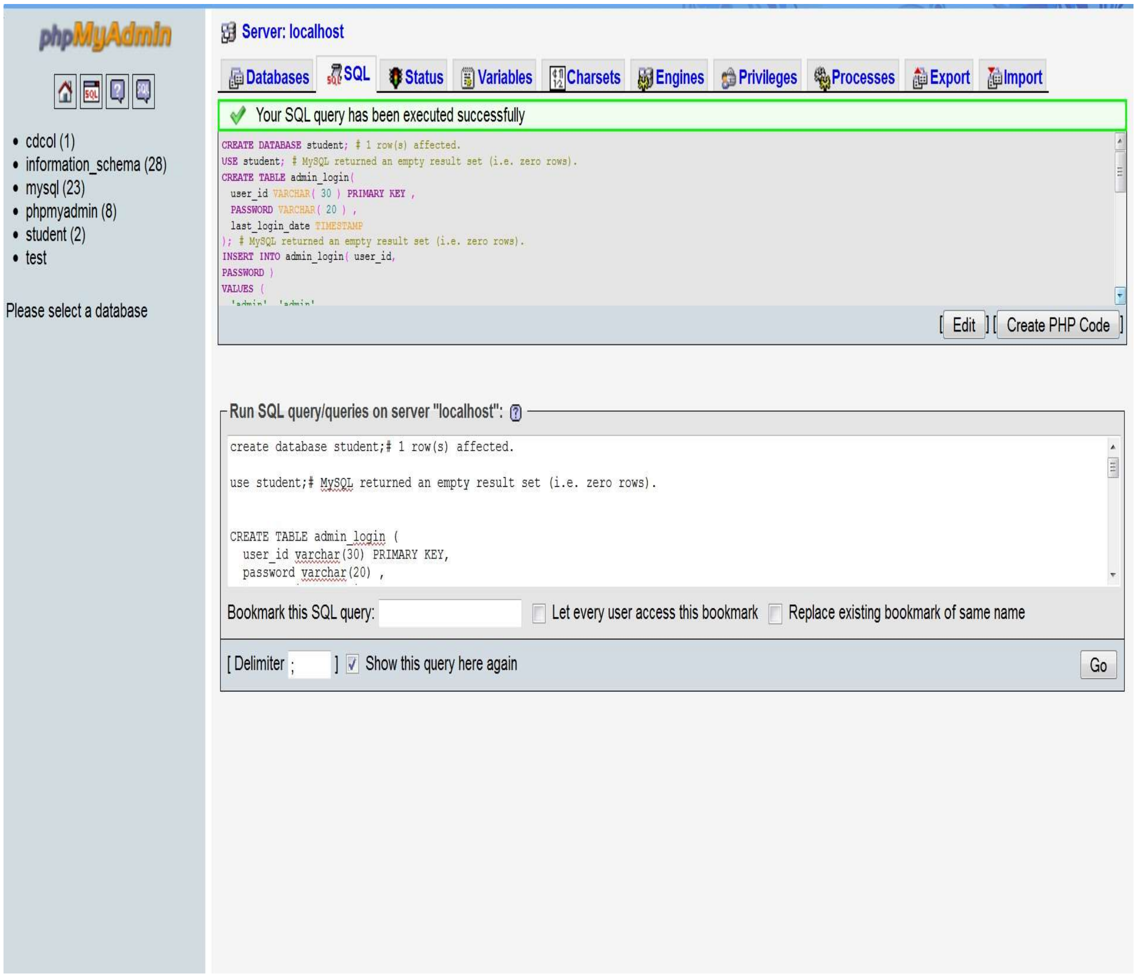


	Table	Action							Rows	Type	Collation	Size	Overhead
□	country_tbl	★	Browse	Structure	Search	Insert	Empty	Drop	246	MyISAM	utf8_general_ci	10.9 KiB	-
□	department_tbl	★	Browse	Structure	Search	Insert	Empty	Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	leave_tbl	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	login_tbl	★	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	salary_tbl	★	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	staff_tbl	★	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB	-
	6 tables	Sum							265	InnoDB	utf8mb4_general_ci	90.9 KiB	0 B

phpMyAdmin start page

The first step with phpMyAdmin running is creating a new database.We create a new database by directly executing SQL statements as shown below.The successful execution of the sql querry creates a database ‘student’ with two tables in it.The tabels are admin_login and student_information.We also inserted values in the admin table.The screenshot below shows the successful execution of the query thus creation of a database named student.



Your SQL query has been executed successfully

```
CREATE DATABASE student; # 1 row(s) affected.
USE student; # MySQL returned an empty result set (i.e. zero rows).
CREATE TABLE admin_login(
 user_id VARCHAR( 30 ) PRIMARY KEY ,
 PASSWORD VARCHAR( 20 ) ,
 last_login_date TIMESTAMP
); # MySQL returned an empty result set (i.e. zero rows).
INSERT INTO admin_login( user_id,
PASSWORD )
VALUES (
 'admin', 'admin'
```

Run SQL query/queries on server "localhost":

```
create database student;# 1 row(s) affected.

use student;# MySQL returned an empty result set (i.e. zero rows).


CREATE TABLE admin_login (
  user_id varchar(30) PRIMARY KEY,
  password varchar(20) ,
```

Bookmark this SQL query: ☐ Let every user access this bookmark ☐ Replace existing bookmark of same name

[Delimiter ;] ☑ Show this query here again

Creation of database in mysql using phpMyadmin

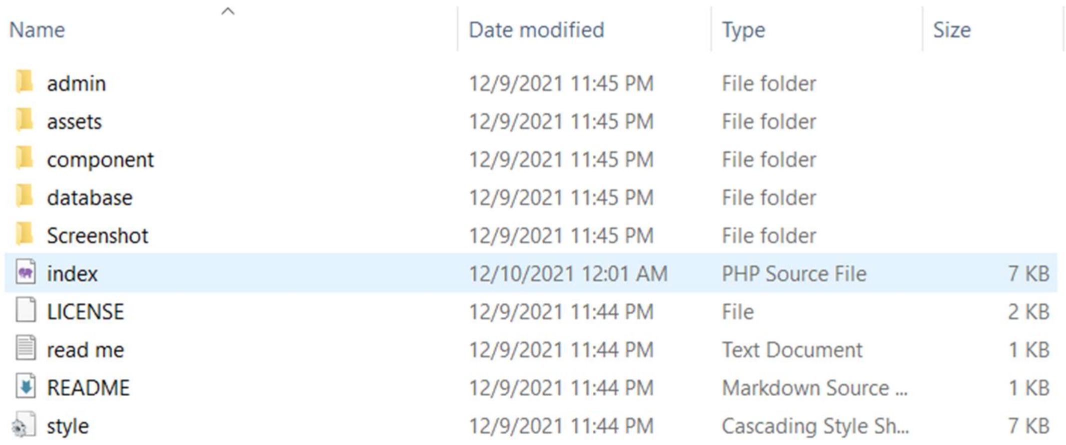
Thus we have learned to create a database in MYSQL by executing sql statements. After creating the database and tables we are now ready to use them in our website “Student Information Management System” .

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:



Name	Date modified	Type	Size
admin	12/9/2021 11:45 PM	File folder	
assets	12/9/2021 11:45 PM	File folder	
component	12/9/2021 11:45 PM	File folder	
database	12/9/2021 11:45 PM	File folder	
Screenshot	12/9/2021 11:45 PM	File folder	
index	12/10/2021 12:01 AM	PHP Source File	7 KB
LICENSE	12/9/2021 11:44 PM	File	2 KB
read me	12/9/2021 11:44 PM	Text Document	1 KB
README	12/9/2021 11:44 PM	Markdown Source ...	1 KB
style	12/9/2021 11:44 PM	Cascading Style Sh...	7 KB

.

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 :**

user_id : Stores user id of administrator(s). ○ password : Stores password of the administrator(s). ○ last_login_date : Stores the last login date of the administrator(s).

Student_information :`

○ student_id : Stores user id of the student(s) ○ student_password : Stores password of the student(s) ○ first_name : Stores first name of the student(s) ○ last_name : Stores last name of the student(s) ○ registration_date : Stores the registration date of the student(s). ○ gender : Stores the gender of the student(s). ○ date_of_birth : Stores the date of birth of the student(s).

○ student_status : Stores the current status of the student account(s). ○ contact_no : Stores the contact number of the student(s). ○ qualification : Stores student(s) qualification.

○ city : Stores the city in which the student(s) lives. ○ email1 : Stores primary email of the student(s). ○ email2 : Stores secondary email of the student(s). ○ address : Stores the address of the student(s). ○ description : Stores description of the student(s). ○ resume : Stores resume of students(s). ○ image : Stores image of the student(s). ○ last_login_date : Strores last login date of the student(s).

Features

The Website provides following functionalities to the users :

- **Administrator** :

  ○ Login/Logout ○ View student information ○ Edit Student Information ○ Enable/disable student accounts ○ Search students

- **Student** : ○ Login/Logout ○ View profile ○ Edit profile ○ Change password

  ○ Register new profile

Source Code

Index.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" href="Style.css" type="text/css"/>
<title>Home Page</title>
</head>
<body>
<table width="100%" height="100%" >
  <tr height="15%">
    <td><?php include 'Header.php';?></td>
  </tr>
  <tr height="80%">
    <td align="center" valign="baseline"><table width="70%">
      <tr height="10%">
        <td align="center"><a href="index.php" class="stylelink"
style="textdecoration:none">Home</a></td>
        <td align="right"><a href="Student_login.php" class="stylelink"
style="text-decoration:none">Student Login</a>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                <a                 href="Admin_login.php"                class="stylelink"
     style="textdecoration:none">Administrator Login</a></td>
      </tr>
      <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
      </tr>
      <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
      </tr>
    </table></td>
  </tr>
  <tr height="5%">
    <td><?php include 'Footer.php';?></td>
  </tr>
</table>
</body>
</html>
```

Student_login.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Login Page</title>
<link rel="stylesheet" href="Style.css" type="text/css"/>
<script type="text/javascript">
function validate()
{
        if(document.form1.st_id.value=="")
        {   alert("Please enter your login
        Id.");
        document.form1.st_id.focus();
        return false;
        }
        if(document.form1.st_pass.value=="")
        {
          alert("Please enter your password.");
        document.form1.st_pass.focus();
        return false;
        }
}
</script>
</head>
<body onLoad="javascript:document.form1.st_id.focus()">
```

```
<form name="form1" method="post" action="Student_login_handler.php"
onSubmit="return validate();">   <table width="100%" height="100%" >
  <tr>
   <td height="15%"><?php include 'Header.php';?></td>
  </tr>
  <tr>
   <td width="100%" height="80%" align="center" valign="baseline"><table
width="90%" >
    <tr>
     <td width="8%"><a href="index.php" class="stylelink"
style="textdecoration:none; font-family: &quot;Times New Roman&quot;,
Times, serif;">Home</a></td>
     <td width="35%" align="center">&nbsp;</td>
     <td width="27%">&nbsp;</td>
     <td width="30%" align="right"><a href="Student_Registration.php"
class="stylelink" style="text-decoration:none ; font-weight: bold;">New
Student Click Here</a></td>
    </tr>
    <tr>
     <td>&nbsp;</td>
     <td>&nbsp;</td>
     <td>&nbsp;</td>
     <td>&nbsp;</td>
    </tr>
    <tr>
     <td>&nbsp;</td>
     <td>&nbsp;</td>
     <td>&nbsp;</td>
     <td>&nbsp;</td>
    </tr>

     <?php if($_GET['flag'] == "success") { ?>
            <tr>
```

```
                <td class="stylegreen" colspan="4"
align="center">Congratulations! You Are successfully registered. You can use
your Login Id and Password to login to your account.</td>
            </tr>
            <?php
            }
            else if($_GET['flag'] == "exists") { ?>
            <tr>
                <td class="stylered" colspan="4" align="center">This Login
Id( <?=$_GET['student_id']?>) already exists.Please, try again with another
Login Id</td>
            </tr>
            <?php
                    }else if($_GET['flag'] == "error") {
            ?>
<tr>
                <td class="stylered" colspan="4" align="center">Error while
inserting data. Please, try again.</td>
            </tr>
            <?php
                    }
            ?>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
     <td colspan="4"><table width="30%"  border="1" align="center"
cellpadding="3" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#CCCCCC">
      <tr align="center" bgcolor="#999999">
       <td colspan="2" class="stylebig">Student Login Here</td>
        </tr>
      <tr bgcolor="#E1E1E1" class="stylesmall">
       <td width="35%" align="left" class="style7">Login Id : </td>
<td width="65%" align="left"><input name="st_id" type="text"
id="st_id"></td>
       </tr>
      <tr bgcolor="#E1E1E1" class="stylesmall">
       <td align="left" class="style7">Password:</td>
        <td align="left"><input name="st_pass" type="password"
id="st_pass"></td>
       </tr>
      <tr bgcolor="#E1E1E1">
       <td colspan="2" align="center">&nbsp;
                      <?php if($_GET['flag'] == "invalid") { ?>
        <span class="stylered">Invalid Login Id or Password</span>
                      <?php }?>
                      </td>
      </tr>
      <tr bgcolor="#E1E1E1">
       <td colspan="2" align="center"><input name="login" class="style10"
type="submit" id="login" value="Login">
        <input name="close" type="button" id="close" class="style10"
value="Close" onClick="self.location='index.php'">          </td>
        </tr>
     </table></td>
     </tr>
```

```
   </table></td>
  </tr>
  <tr>
   <td height="5%" align="center"><?php include 'Footer.php';?></td>
  </tr>
 </table>
</form>
</body>
</html>
```

Student_login_handler

```
<?php
      session_start();
      include 'Connect.php';
      $flag = "";
      $student_id = $_POST['st_id'];
      $st_pass = $_POST['st_pass'];
      $query = "select last_login_date from student_information where
student_id='$student_id' and student_password='$st_pass' and
student_status ='Enable'";
      $result = mysql_query($query,$link_id);
      if(mysql_error() != null){
      die(mysql_error());
      }
      if($date = mysql_fetch_array($result))
      {
            $lastdate = $date['last_login_date'];
            $date2 = date("d-m-Y h:i A",strtotime($lastdate));
            $_SESSION['userid'] = $_POST['st_id'];
            $_SESSION['lastlogin'] =$date2;
            $_SESSION['type'] = "Student";
            mysql_query("update student_information set
last_login_date=now() where student_id='$student_id'",$link_id);
        if(mysql_error() != null){ die(mysql_error());
        }
         header("location:Student_Home.php");
        die();
    }
    else
    {
        $flag = "invalid";
        header("location:Student_login.php?flag=$flag");
        die();
    }

?>
```

Student_Registration.php

```
<html>
<head>
<link rel="stylesheet" href="Style.css" type="text/css"/>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Student Registration Page</title>
<script src="Validation.js"></script>
<script type="text/javascript">
function validation()
{
    if(document.form1.first_name.value=="")
    {   alert("Please enter your first
    name.");
    document.form1.first_name.focus();
    return false;
    }
    if(document.form1.last_name.value=="")
```

```
{    alert("Please enter your last
name.");
document.form1.last_name.focus();
return false;
}
if(document.form1.dob.value=="")
{
  alert("Please enter your date of
birth.");    document.form1.dob.focus();
return false;
}
else
{
  var date = document.form1.dob.value;
var yes = checkDate(date);
  if(!yes)
        { alert("Please Enter a valid date of birth.");
                document.form1.dob.focus();  return
                false;
        }
 }
if(document.form1.email1.value=="")
{    alert("Please enter your primary
email.");
document.form1.email1.focus();    return
false;
}
else
{
  var isEmail = emailValidator(document.form1.email1.value);
if(!isEmail)
```

```
            { alert("Please enter a valid primary email.");
                        document.form1.email1.focus(); return
                        false;
            }
          }
          if(document.form1.address.value != "" &&
document.form1.address.value.length > 100){
                 alert("You can enter address upto 100 characters only.");
                           document.form1.address.focus(); return false;
          }
          if(document.form1.description.value != "" &&
document.form1.description.value.length > 200){ alert("You can enter
                 description upto 200 characters only.");
                 document.form1.description.focus(); return false;
          }
          if(document.form1.st_id.value=="")
          {    alert("Please enter your desired student
          id."); document.form1.st_id.focus();
            return false;
          }
          if(document.form1.st_pass.value=="")
          {
            alert("Please enter your desired
          password.");
          document.form1.st_pass.focus();    return
          false;
          }
          if(document.form1.retype.value=="")
          {
            alert("Please enter retype
          password.");
          document.form1.retype.focus();    return
          false;
```

```
           }
           if(document.form1.st_pass.value != document.form1.retype.value)
            {
              alert("Password and retype password are not same.");
                      document.form1.st_pass.value = "";
              document.form1.retype.value = "";
           document.form1.st_pass.focus();
              return false;
            }
}
</script>
</head>

<body onLoad="javascript:document.form1.first_name.focus()">
<form name="form1" method="post"
action="Student_Registration_handler.php" onSubmit="return validation()"
enctype="multipart/form-data">
  <table width="100%">
    <tr>
      <td width="100%" height="15%" align="center"><?php include
'Header.php';?></td>
    </tr>
    <tr>
      <td width="100%" height="80%" align="center"><table width="80%"
border="1" cellpadding="2" cellspacing="0" bordercolor="#CCCCCC">
        <tr bgcolor="#EEEEEE">
          <td colspan="4" align="center" class="stylemedium">Student
Information</td>
          </tr>
        <tr class="stylesmall">
          <td>First Name <span class="stylered">*</span> </td>
<td><input name="first_name" type="text" id="first_name"
maxlength="50"></td>
```

```
          <td>Last Name <span class="stylered">*</span> </td>
<td><input name="last_name" type="text" id="last_name"
maxlength="30"></td>
        </tr>
        <tr class="stylesmall">
          <td>Gender</td>
          <td><input name="gender" type="radio" value="Male" checked>
            Male<input name="gender" type="radio" value="Female">
            Female</td>
          <td>Date Of Birth <span class="stylered">*</span> </td>
<td><input name="dob" type="text" id="dob" size="10"
maxlength="10">                        DD-MM-YYYY</td>
        </tr>
        <tr class="stylesmall">
          <td>Qualification <span class="stylered">*</span> </td>
          <td><select name="qualification" id="qualification">
<option value="">-----select-----</option>
                               <option value="High School">High School</option>
                               <option value="Graduate">Graduate</option>
                               <option value="MCA">MCA</option>
                               <option value="BCA">BCA</option>
                               <option value="Master Degree">Master Degree</option>

          </select></td>
          <td>Contact No</td>
          <td><input name="contact_no" type="text" id="contact_no"
maxlength="20"></td>
        </tr>
                    <tr class="stylesmall">
          <td> City</td>
          <td><input name="city" type="text" id="city" maxlength="30"></td>
          <td>&nbsp;</td>
          <td>&nbsp;</td>
        </tr>
        <tr class="stylesmall">
          <td>Primary Email <span class="stylered">*</span> </td>
<td><input name="email1" type="text" id="email1" maxlength="100"></td>
          <td>Secondary Email</td>
          <td><input name="email2" type="text" id="email2"
maxlength="100"></td>
        </tr>
        <tr class="stylesmall">
          <td>Address</td>
          <td colspan="3"><textarea name="address" cols="45" rows="2"
id="address"></textarea></td>
        </tr>
                    <tr class="stylesmall">
          <td> Description</td>
          <td colspan="3"><textarea name="description" cols="45" rows="3"
id="description"></textarea></td>
        </tr>
        <tr bgcolor="#EEEEEE">
          <td colspan="4" align="center" class="stylemedium">Login
Information</td>
        </tr>
        <tr>
          <td colspan="4">&nbsp;</td>
          </tr>
        <tr class="stylesmall">
          <td>Desired ID <span class="stylered">*</span> </td>
          <td><input name="st_id" type="text" id="st_id" maxlength="20"></td>
          <td>&nbsp;</td>
          <td>&nbsp;</td>
        </tr>
        <tr class="stylesmall">
        <td>Password <span class="stylered">*</span> </td>
<td><input name="st_pass" type="password" id="st_pass"
maxlength="20"></td>
        <td align="right">Retype Password<span class="stylered"> *</span>
</td>
        <td><input name="retype" type="password" id="retype"
maxlength="20"></td>
      </tr>
                 <tr>
                   <td>&nbsp;</td>
                   <td>&nbsp;</td>
                   <td>&nbsp;</td>
                   <td>&nbsp;</td>
                   </tr>
                 <tr bgcolor="#EEEEEE">
                  <td colspan="4" align="center" class="stylemedium"> Resume
Information</td>
            </tr>
                 <tr class="stylesmall">
                   <td>Upload Resume </td>
                   <td colspan="3"><input name="resume" type="file" >
                   &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span
class="stylered">.doc , .txt, .pdf file only</span></td>
         </tr>
                 <tr>
        <td colspan="4">&nbsp;</td>
        </tr>
                 <tr align="center" class="stylemedium" bgcolor="#EEEEEE">
                   <td colspan="4">Image Information</td>
                   </tr>
                 <tr class="stylesmall">
                   <td>Upload Image </td>
                   <td colspan="3"><input type="file" name="image">
```

```
                    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span
class="stylered"><span class="stylered">.jpg file And .gif file
only</span></td>
                   </tr>
                   <tr>
         <td colspan="4"><span class="stylered">* <em>means fields are
compulsory</em> </span></td>
         </tr>
                   <tr>
         <td colspan="4" align="center"><input name="register" type="submit"
id="register" value="Register">
           <input name="reset" type="reset" id="reset" value="Reset">
<input name="close" type="button" id="close" value="Close"
onClick="self.location='Student_login.php'"></td>
         </tr>
      </table></td>
    </tr>
    <tr>
      <td width="100%" height="5%" align="center"><?php include
'Footer.php';?></td>
    </tr>
  </table>
</form>
</body>
</html>
```

Student_Registration_handler.php

```
<?php include 'Connect.php'; $flag =
          "success"; function rollbackData(){
          mysql_query(" ROLLBACK "); global
          $flag; $flag = "error"; if(mysql_error()
          != null){ die(mysql_error());
```

```
                }
        }
        $student_id = $_POST['st_id'];
        $student_pass = $_POST['st_pass'];
        $first_name = $_POST['first_name'];
        $last_name = $_POST['last_name'];
        $gender = $_POST['gender'];
        $contact_no = $_POST['contact_no'];
        $qualification= $_POST['qualification'];
        $city = $_POST['city'];
        $email1 = $_POST['email1'];
        $email2 = $_POST['email2'];
        $address = $_POST['address'];
        $description = $_POST['description'];
        $resumename = "";
        $imagename = "";
        $dobdate = date("Y-m-d",strtotime($_POST['dob']));
        /*
                This block is used to check whether the student_id already exits
in database.
        */
        $select_query="select student_id from student_information where
student_id = '$student_id'";
        $result_set = mysql_query($select_query,$link_id);

        if($row = mysql_fetch_array($result_set)){
                $flag="exists";
                header("location:Student_login.php?
flag=$flag&student_id=$student_id");
                die();
        }
        else{
                /*
```

```
            This block is used to insert the student record in database
            if the student_id is already not present in the database.
        */
        mysql_query("SET AUTOCOMMIT = 0 ");
        if(mysql_error() != null){
            die(mysql_error());

        }
        $query = "insert into
student_information(student_id,student_password,first_name,last_name,regis
tration_date,gender,date_of_birth,"; $query .=
"student_status,contact_no,qualification,city,email1,email2,address,descriptio
n)";
        $query .= "
values('$student_id','$student_pass','$first_name','$last_name',now(),'$gender
','$dobdate','Disable','$contact_no',";
        $query .=
"'$qualification','$city','$email1','$email2','$address','$description')";

        $result = mysql_query($query,$link_id);
        if(mysql_error() != null){
        die(mysql_error());
        }

        if($result){ if($_FILES['resume']['name']
            != ""){
                $filename = $_FILES['resume']['name'];
                $ext = strrchr($filename,".");
                $resumename = $student_id;
                $resumename .= "_".$filename; if($ext ==".txt" ||
                $ext ==".doc" || $ext ==".TXT" ||
$ext ==".DOC" || $ext ==".pdf" || $ext ==".PDF"){
                    $size = $_FILES['resume']['size'];
                    if($size > 0 && $size < 1000000){
                        $archive_dir = "resumes";
                        $userfile_tmp_name =
$_FILES['resume']['tmp_name'];

    if(move_uploaded_file($userfile_tmp_name, "$archive_dir/
$resumename")){
                            /* if image is successfully
uploaded then resumename is stored in database.
                            */
                            mysql_query("update
student_information set resume='$resumename' where
student_id='$student_id'", $link_id);
                            if(mysql_error() != null){
                                die(mysql_error());
                            }
                            $flag = "success";
                        }else{ rollbackData();
                        }
                    }
                    else{ rollbackData(); die("You can upload
                        resume of 1 MB
size only. Please, try again.");
                    }
                }
                else{ rollbackData(); die("You can upload
resume of .txt, .pdf, .doc extensions only. Please, try again.");
                }
            } if($_FILES['image']['name'] !=
            ""){
                $filename = $_FILES['image']['name'];
                $ext = strrchr($filename,".");
                $imagename = $student_id;
```

```
                                $imagename .="_". $filename; if($ext ==".jpg" ||
                                $ext ==".jpeg" || $ext ==".JPG" ||
$ext ==".JPEG" || $ext ==".gif" || $ext ==".GIF"){
                                        $size = $_FILES['image']['size'];
                                        if($size > 0 && $size < 1000000){
                                                $archive_dir = "images";
                                                $userfile_tmp_name =
$_FILES['image']['tmp_name'];

        if(move_uploaded_file($userfile_tmp_name, "$archive_dir/
$imagename")){
                                                        /* if image is successfully
uploaded then imagename is stored in database.
                                                        */
                                                        mysql_query("update
student_information set image='$imagename' where
student_id='$student_id'", $link_id);
                                                        $flag = "success";
                                                        if(mysql_error()!=null){
                                                        die(mysql_error());
                                                        }
                                                }
                                                else{ if(file_exists('resumes/' .
                                           $resumename)) { unlink('resumes/' .
$resumename);
                                                        }
                                                        rollbackData();
                                                }
                                        }
                                        else{ if(file_exists('resumes/' .
$resumename)) { unlink('resumes/' .
$resumename);
                                                }
```

```
                        rollbackData(); die("You can
                        upload image of 1 MB
size only. Please, try again.");
                    }
                }
                else{ if(file_exists('resumes/' . $resumename)) {
                    unlink('resumes/' . $resumename);
                    }
                    rollbackData(); die("You can upload
images of .jpg, .jpeg, .gif extensions only. Please, try again. ");
                }
            }
        }
        else{
            $flag="error";
        }
        if($flag == "success"){
            mysql_query(" COMMIT ");
            $flag="success";
            if(mysql_error() != null){
            die(mysql_error());
            }
            /*
                This block is used to send email to the
successfully registered users.
            */
            /*
            $to  = $email1;
            $subject = 'Congratulations';
            $message = 'Congratulations you are registered in our
site.\r\n\r\n';
            $message .= "Your Login Id : $student_id \r\n Password :
$student_pass";
```

```
            $headers = "From: info@sims.com\r\n";
            $headers .= 'X-Mailer: PHP' . phpversion();
            mail($to, $subject, $message, $headers);
            */
        }
        header("location:Student_login.php?flag=$flag");
        die();
    }
?>
```

Admin_Edit_Student_Info.php

```
<?php
    session_start();
    $session_id = $_SESSION['userid'];
    if($session_id == null){
    header("location:index.php");
    die();
    }

  include 'Connect.php';
  $student_id = $_REQUEST['student_id'];
$query = "select * from student_information where
student_id='$student_id'";
  $result = mysql_query($query, $link_id);
  $data = mysql_fetch_array($result);
?>
<html>
<head>
<link rel="stylesheet" href="Style.css" type="text/css"/>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Edit Student Information By Admin</title>
```

```
<script src="Validation.js"></script>
<script type="text/javascript">
function validation()
{
        if(document.form1.first_name.value=="")
        {    alert("Please enter your first
        name.");
        document.form1.first_name.focus();
        return false;
        }
        if(document.form1.last_name.value=="")
        {    alert("Please enter your last
        name.");
        document.form1.last_name.focus();
        return false;
        }
        if(document.form1.dob.value=="")
        {    alert("Please enter your date of
        birth.");    document.form1.dob.focus();
        return false;
        }
        else
        {
          var date = document.form1.dob.value;
        var yes = checkDate(date);
          if(!yes)
                { alert("Please Enter a valid date of birth.");
                        document.form1.dob.focus();  return
                        false;
                }
         }
        if(document.form1.email1.value=="")
```

```
          {    alert("Please enter your primary
          email.");
          document.form1.email1.focus();    return
          false;
          }
          else
          {
            var isEmail = emailValidator(document.form1.email1.value);
          if(!isEmail)
            { alert("Please enter a valid primary email.");
                          document.form1.email1.focus(); return
                          false;
            }
          }
          if(document.form1.email2.value != "")
          {
            var isEmail = emailValidator(document.form1.email2.value);
          if(!isEmail)
            {
                          alert("Please enter a valid secondary
                          email."); document.form1.email2.focus();
                          return false;
            }
          }
          if(document.form1.address.value != "" &&
document.form1.address.value.length > 100){ alert("You can enter
                address upto 100 characters only.")
                document.form1.address.focus(); return false;
          }
          if(document.form1.description.value != "" &&
document.form1.description.value.length > 200){
                alert("You can enter description upto 200 characters only.")
                            document.form1.description.focus(); return false;
          }

} function
SetAll()
 {
   document.form1.qualification.value="<?php echo $data['qualification'];?>";
var gen = "<?php echo $data['gender'];?>";    var gend =
document.form1.gender.length;    for(var i =0; i<gend; i++)
    {
             if(document.form1.gender[i].value == gen)
          document.form1.gender[i].checked=true;
          }
 }
</script> </head>

<body onLoad="javascript:SetAll()"> <form name="form1"
method="post" action="Admin_Edit_Student_Info_Handler.php"
onSubmit="return validation()">
 <input type="hidden" name="student_id" value="<?php echo $student_id; ?
>;">
  <table width="100%" height="100%" >
    <tr bgcolor="#E1E1E1">
      <td width="100%" height="15%" align="center"><?php include
'Admin_Header.php';?></td>
    </tr>
    <tr>
      <td width="100%" height="80%" align="center"><table width="80%"
border="1" cellpadding="2" cellspacing="0" bordercolor="#EEEEEE">
        <tr>
          <td colspan="4" align="center" bgcolor="#EEEEEE"
class="stylebig">Edit Student Information</td>
          </tr>
<tr>
```

```
                     <td colspan="4" align="center">&nbsp;
                    <?php  if($_GET['flag'] == "success") { ?>
                    <span class="stylered">Student Information updated
successfully.</span>
       <?php } else if($_GET['flag'] == "error") {  ?>
                    <span class="stylered">Error while updating student
information.Please, try again</span>
<?php }  ?> </td>
                    </tr>
       <tr class="stylesmall">
         <td width="19%" align="left">First Name : </td>
         <td width="30%" align="left"><input name="first_name" type="text"
id="first_name" value="<?php echo $data['first_name']; ?>"
size="25"maxlength="50"></td>
         <td width="17%" align="left">Last name</td>
         <td width="34%" align="left"><input name="last_name" type="text"
id="last_name" value="<?php echo $data['last_name'];?>" size="25"
maxlength="30"></td>
       </tr>
       <tr class="stylesmall">
         <td height="29" align="left">Gender : </td>
         <td align="left"><input name="gender" type="radio" value="Male">
           Male<input name="gender" type="radio" value="Female">
           Female</td>
         <td align="left">Date Of Birth</td>
         <td align="left"><input name="dob" type="text" id="dob" size="10"
maxlength="10" value="<?php echo date("d-m-
Y",strtotime($data['date_of_birth']));?>">
           DD-MM-YYYY</td>
       </tr>
       <tr class="stylesmall">
         <td align="left">Qualification : </td>
         <td align="left"><select name="qualification" id="qualification">
```

```
          <option value="High School">High School</option>
                         <option value="Graduate">Graduate</option>
                         <option value="MCA">MCA</option>
                         <option value="BCA">BCA</option>
                         <option value="Master Degree">Master Degree</option>
                   </select></td>
          <td align="left">Contact No</td>
          <td align="left"><input name="contact_no" type="text" id="contact_no"
value="<?php echo $data[$contact_no];?>" size="25" maxlength="20"></td>
        </tr>
        <tr class="stylesmall">
          <td align="left">Primary Email : </td>
          <td align="left"><input name="email1" type="text" id="email1"
value="<?php echo $data['email1'];?>" size="25" maxlength="100"></td>
          <td align="left">Secondary Email</td>
          <td align="left"><input name="email2" type="text" id="email2"
value="<?php echo $data['email2'];?>" size="25" maxlength="100"></td>
        </tr>
        <tr class="stylesmall">
          <td align="left">City : </td>
          <td colspan="3" align="left"><input name="city" type="text" id="city"
value="<?php echo $data['city'];?>" size="25" maxlength="30"></td>
          </tr>
        <tr class="stylesmall">
          <td align="left">Address : </td>
          <td colspan="3" align="left"><textarea name="address" rows="2"
cols="40"><?php echo $data['address'];?></textarea></td>
        </tr>
                   <tr class="stylesmall">
          <td align="left">Description : </td>
          <td colspan="3" align="left"><textarea name="description" rows="3"
cols="40"><?php echo $data['description'];?></textarea></td>
        </tr>
```

```
    <tr>
     <td colspan="4">&nbsp;</td>
    </tr>
    <tr>
     <td colspan="4" align="center"><input name="update" type="submit"
id="update" value="Update">
      <input name="close" type="button" id="close" value="Close"
onClick="self.location='DisplayAll.php'"></td>
     </tr>
   </table></td>
  </tr>
  <tr bgcolor="#E1E1E1">
   <td width="100%" height="5%" align="center"><?php include
'Footer.php';?></td>
  </tr>
 </table>
</form>
</body>
</html>
```

Admin Edit Student Info Handler

```
<?php
     session_start();
     $session_id = $_SESSION['userid'];
     if($session_id == null){
     header("location:index.php");
     die();
     }
     include 'Connect.php';
     $student_id = $_POST['student_id'];
     $first_name = $_POST['first_name'];
```

```
        $last_name = $_POST['last_name'];
        $gender = $_POST['gender'];
        $contact_no = $_POST['contact_no'];
        $qualification= $_POST['qualification'];
        $city = $_POST['city'];
        $email1 = $_POST['email1'];
        $email2 = $_POST['email2'];
        $address = $_POST['address'];
        $description = $_POST['description'];
        $dateofbirth = date("Y-m-d",strtotime($_POST['dob']));
        $flag = "";
    $query = "update student_information set
first_name='$first_name',last_name='$last_name',gender='$gender',date_of_
birth='$dateofbirth',";
        $query .=
"qualification='$qualification',contact_no='$contact_no',email1='$email1',email
2='$email2',city='$city',address='$address',description='$description' ";
    $query .= " where student_id='$student_id'"; $result =
mysql_query($query, $link_id); if(mysql_error() != null){
die(mysql_error());
        }
        if($result)
        {
                $flag = "success";
        }
else
        {
                 $flag = "error";
        }
        header("location:Admin_Edit_Student_Info.php?
flag=$flag&student_id=$student_id");   ?>
```

Validation.js

```
function checkDate(dobStr)
{ var arr = dobStr.split(/-/);
        if(arr.length != 3){ return
        false;
        } var ddStr =
        arr[0]; var mmStr
        = arr[1]; var yyStr
        = arr[2];
        if(isNaN(ddStr)){
        return false;
        }
        if(isNaN(mmStr)){
                return false;
        }
        if(isNaN(yyStr)){
                return false;
        }

        dd=parseInt(ddStr,10);
        mm=parseInt(mmStr,10);
        yy=parseInt(yyStr,10);

        if(dd < 1 || dd > 31){
                return false;
        }
        if(mm < 1 || mm > 12){
                return false;
        } if(yy < 1900){
        return false;
        } if(yy > 3000){
        return false;
```

```
}
switch (mm){
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: return
        true; case 4:
        case 6: case 9:
        case 11:
                if(dd<=30) return
                        true;
                else return false;
        case 2:
                if(yy%100==0)
                {       if(yy%400==0)
                        {
                                if(dd<=29) return
                                        true;
                                else return false;
                        }
                }
                else
                {       if(yy%4==0){
                                if(dd<=29) return
                                        true;
                                else return false;
                        }
                }
                if(dd<=28)
                {       return true;
```

```
            }
            else {      return
            false;
            }
        default :
            return false;
    }
}
function emailValidator(elem)
{
    var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/;
    if(elem.match(emailExp))
    { return true;
    }
    else
    { return false;
    }
}
```

Style.css

```
body { margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
.stylebig { font-size: 18px; font-family: "Times New
    Roman", Times, serif; font-weight: bold;
}
.stylemedium { font-size: 14px; font-family: "Times
    New Roman", Times, serif; font-weight: bold;
}
```

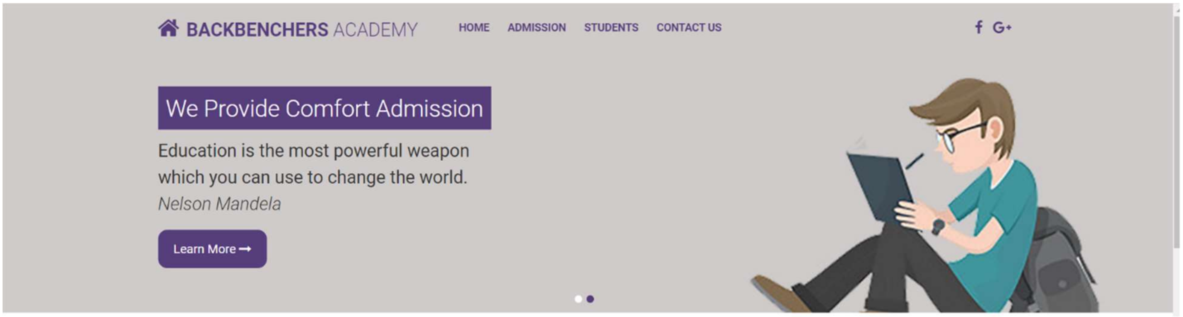
```
.stylesmall { font-family: "Times New Roman",
        Times, serif; font-size: 12px; font-weight: bold;
}
.stylegreen {font-family: "Times New Roman", Times, serif; font-size: 12px;
font-weight: bold; color: #007F00; }
.stylered { font-size: 12px; font-family: "Times New
        Roman", Times, serif; color: #FF0000; font-
        weight: bold;
}
.stylegrey {font-family: "Times New Roman", Times, serif; font-size: 12px; font-
weight: bold; color: #550055; }
.stylegreyre {font-family: "Times New Roman", Times, serif; font-size: 12px;
font-weight: bold; color: #CCC; }
.stylelink {font-family: "Times New Roman", Times, serif; font-size: 12px;
fontweight: bold; color: #AA0000; }
.stylelink2 {font-family: "Times New Roman", Times, serif; font-size: 12px;
font-weight: bold; color: #999; }
```

Connect.php

```
<?php
        $host = "localhost";
        $dbusername = "root";
        $dbpassword = "";
        $dbname = "student";

        $link_id = mysql_connect($host, $dbusername,
        $dbpassword); if(!$link_id){ die(mysql_error("Can`t Connect
        To database"));
        }
        else{
                $db = mysql_select_db($dbname, $link_id);
        }
        if(!$db){ die(mysql_error("Can`t select
                database"));
        }
        return;
?>
```

SNAPSHOTS

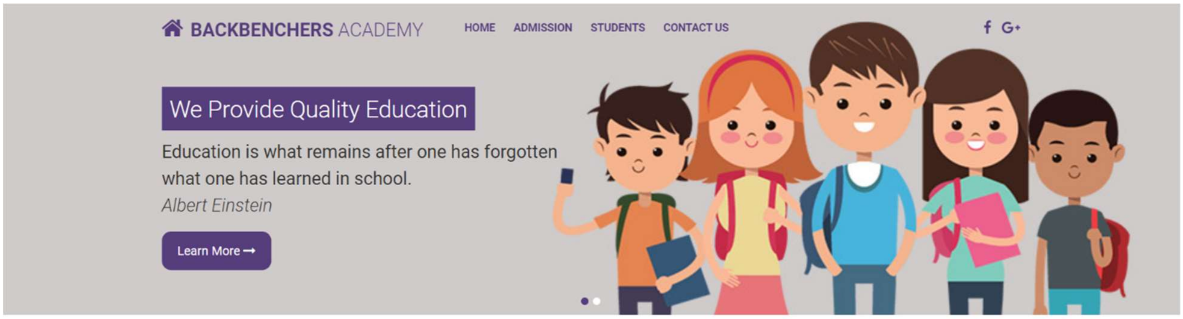


We Provide Comfort Admission

Education is the most powerful weapon which you can use to change the world.

Nelson Mandela

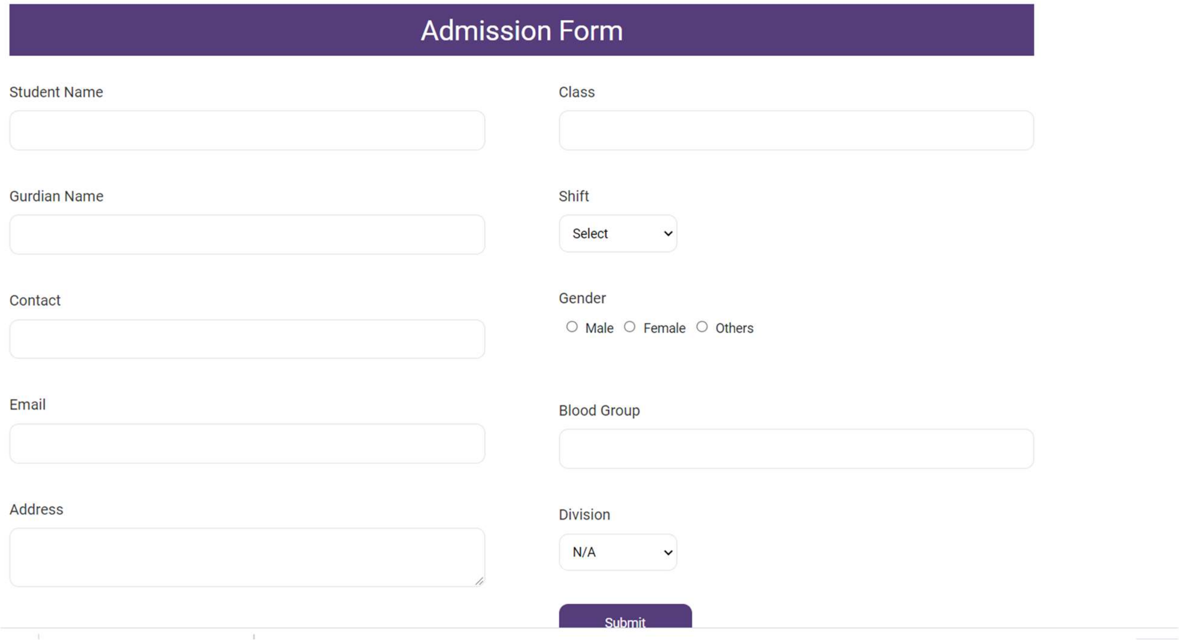
This is the homepage/indexpage of the site.



We Provide Quality Education

Education is what remains after one has forgotten what one has learned in school.

Albert Einstein



Admission Form

Student Name

Class

Gurdian Name

Shift

Select

Contact

Gender

○ Male ○ Female ○ Others

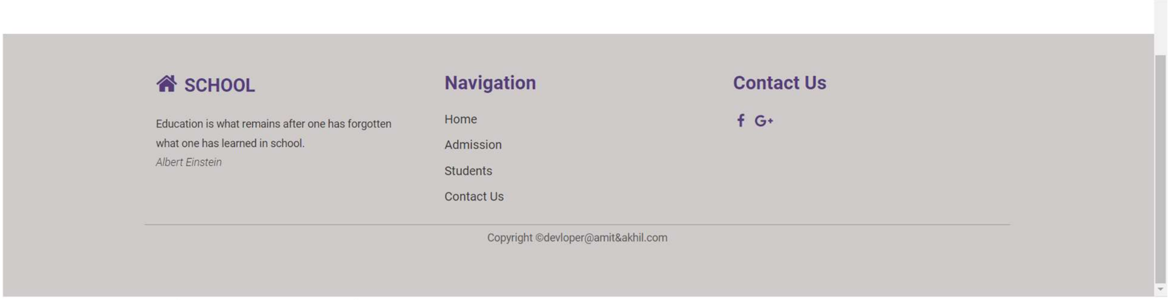
Email

Blood Group

Address

Division

N/A



SCHOOL

Education is what remains after one has forgotten what one has learned in school.

Albert Einstein

Navigation

Home

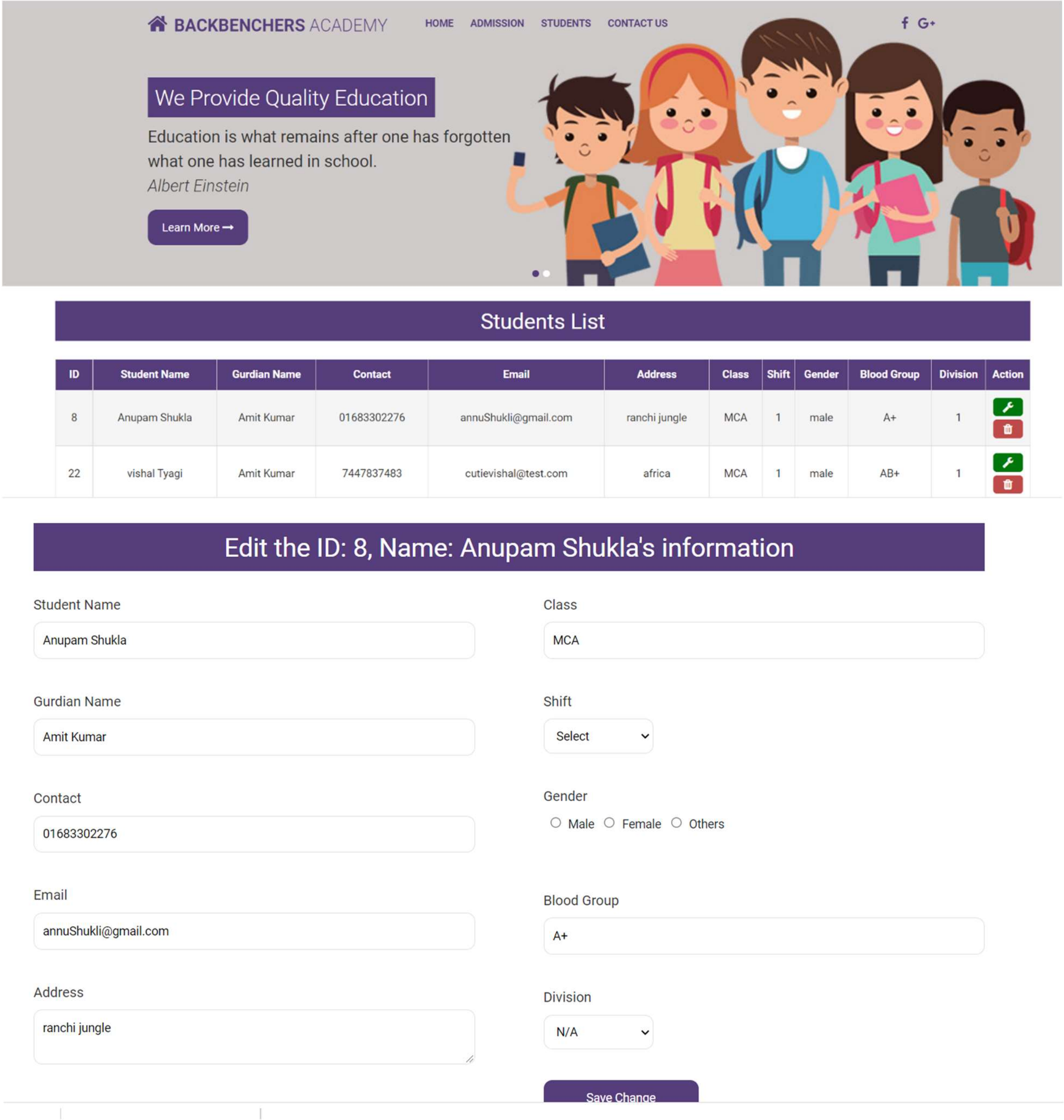
Admission

Students

Contact Us

Contact Us

Copyright ©devloper@amit&akhil.com



We Provide Quality Education

Education is what remains after one has forgotten what one has learned in school.

Albert Einstein

Students List

ID	Student Name	Gurdian Name	Contact	Email	Address	Class	Shift	Gender	Blood Group	Division	Action
8	Anupam Shukla	Amit Kumar	01683302276	annuShukli@gmail.com	ranchi jungle	MCA	1	male	A+	1	🔧 🗑
22	vishal Tyagi	Amit Kumar	7447837483	cutievishal@test.com	africa	MCA	1	male	AB+	1	🔧 🗑

Edit the ID: 8, Name: Anupam Shukla's information

Student Name: Anupam Shukla

Gurdian Name: Amit Kumar

Contact: 01683302276

Email: annuShukli@gmail.com

Address: ranchi jungle

Class: MCA

Shift: Select

Gender: ○ Male ○ Female ○ Others

Blood Group: A+

Division: N/A

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.

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)