

CONTENTS

College Certificate	03
Acknowledgement	04
Declaration	05
Abstract	06
1. Introduction	
1.1 Background	07
1.2 Description	07
1.3 What it can do?	08
1.4 Who can control it?	08
2. System Analysis	
2.1 Existing System	09
2.2 Proposed System	09
2.3 Feasibility Study	10
2.4 Software/ Hardware Requirements	11
2.5 System Architecture	12
2.6 Modules and its Description	13
2.7 UML Diagrams	15
3. Implementation	
3.1 Technologies Used	
3.1.1 BOOTSTRAP	17
3.1.2 HTML	18
3.1.3 PHP	23
3.1.4 JAVASCRIPT	25
3.1.5 MYSQL	27
4. Screens/ Forms	28
5. Future Enhancement	35
6. Conclusion	35
7. Bibliography	36

ABSTRACT

Student Information System is for maintaining and retrieving the students' information in an easy and robust manner. In this system admin can register a student and will have the access to delete, update or modify any student's data in the database.

Only the admin can have access to use this application. This is done by using the login form where the admin needs to login with his credentials to gain access for adding, updating, deleting, modifying and viewing students' details. While adding the students' details, the admin can either add a single student's information or upload the list of students belonging to a particular batch.

While deleting the students' details, the admin can either delete a single student's information or drop the table containing the details of an entire batch. While viewing a particular student's profile, admin can either search using his roll number or with his name along with the year of admission.

INTRODUCTION

STUDENT INFORMATION SYSTEM:

Student Information System focuses on presenting the students' information in an easy and robust manner. This is done by creating a stand-alone application which helps in retrieving this information providing us a simple user-friendly interface. Thus this application helps in reducing the manual work and automating the record generation process.

BACKGROUND:

There are many departments of administration in a college for the maintenance of students' data.

As there are plenty of records available regarding students, maintaining this information manually becomes a tough task as the information may be scattered or redundant making it time consuming job for collecting and retrieving the information regarding a particular student. This system allows the admin to maintain this information in a centralized manner.

DESCRIPTION:

The main objective of this application is to develop a system that enables the user to maintain the students' data in a centralized manner and automate the record generation process. This is stand-alone application that follows a client-server architecture where the admin is the client requesting to perform an operation and the server responds accordingly. The front-end is designed using **BOOTSTRAP**, a collective frame work for both **HTML** and **CSS**. The back-end database being used is **MYSQL** database for storing the information of students. The intermediate logic is written in **PHP** to interact with the server and retrieve the information from the database and **JAVASCRIPT** is used for doing the client-side validations.

WHAT CAN STUDENT INFORMATION SYSTEM DO?

STUDENT INFORMATION SYSTEM CAN:

1. Reduce the effort in maintaining the students' information manually.
2. Provide a centralized way for maintaining this information.
3. Automate the record generation process.
4. Provide a user-friendly interface for the admin to control it.
5. Adapt to any department or institution and can be controlled easily.

WHO CAN CONTROL IT?

This is a stand-alone application that provides a simple user-friendly interface to reduce the time taken and the amount of resources required for maintaining the students' information.

As this is a stand-alone application only the super user or admin can control it. The admin is given login credentials and he can only access the application and use it. This has a simple user-friendly interface that enables this application to be used anywhere, that is this application is **portable**. This application is developed to be used in any department and any institution or university, which makes application **adaptable**

SYSTEM ANALYSIS

EXISTING SYSTEM:

There are many departments of administration in a college for the maintenance of student data. This information can be the general details like name, address, performance, academics, achievements, etc. As there are plenty of records available for students, maintaining this information manually is a tough task because the information may be scattered or redundant.

PROPOSED SYSTEM:

Student Information System focuses on maintaining and retrieving this information in an easy and robust manner. This can be done by creating a stand-alone database application which helps in retrieving the student information providing us a simple interface for its maintenance. It also provides a centralized way to add, modify, update and delete students' information. Thus, this project helps in reducing the manual work and automating the record generation process in an educational institution.

Feasibility Study

The feasibility of the project is analyzed in this phase and is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the universities. For feasibility analysis, some understanding of the major requirements for the system is essential.

- Technical feasibility
- Social feasibility

Technical Feasibility:

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the student has enough experience using that technology. The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc in order to estimate if the new system will perform adequately or not.

Social Feasibility:

Determines whether the proposed system conflicts with legal requirements, (e.g. a data processing system must comply with the local data protection acts). When an organization/institution has either internal or external legal counsel, such reviews are typically standard. However, a project may face legal issues after completion if this factor is not considered at this stage. It is about the authorization.

Software/Hardware Requirements

Hardware Requirements:

- RAM: 512MB and above.
- Processor: Minimum 1GHZ
- Hard Disk: Minimum space of 40GB

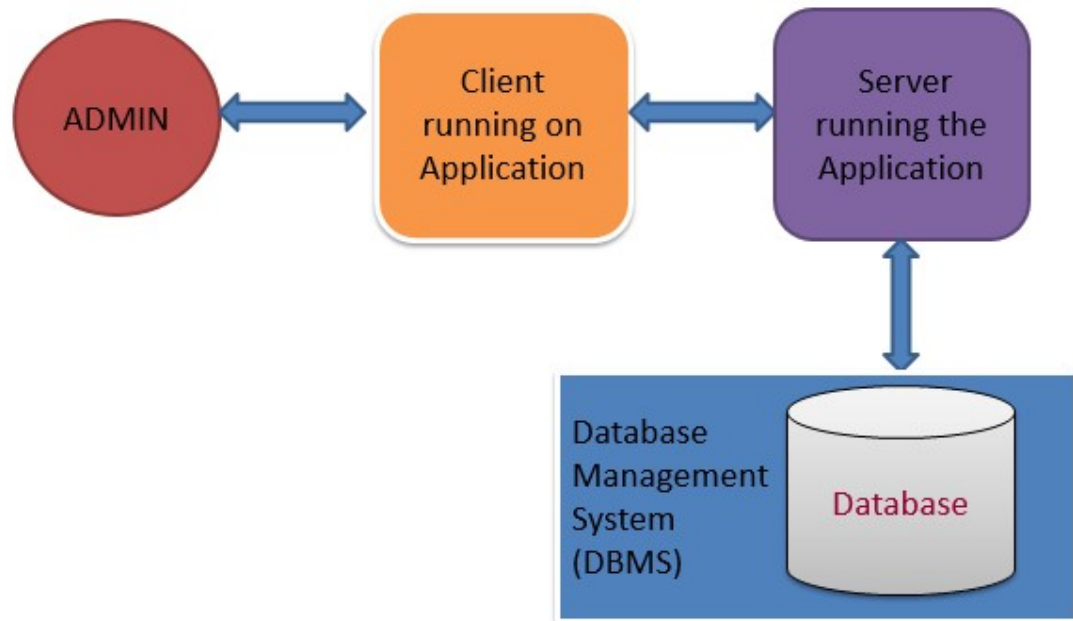
Software Requirements:

- Operating System: Windows or Linux
- Data Base: MYSQL
- Web Server: Apache Tomcat / (XAMPP Toolkit)

Languages required:

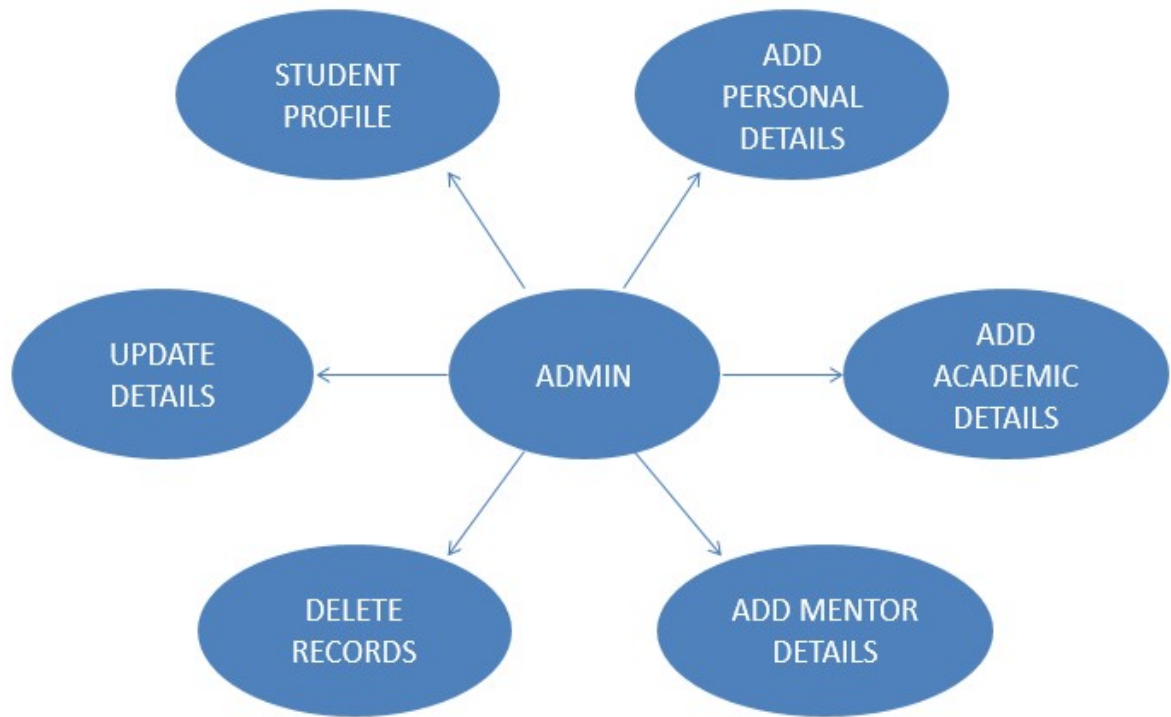
- BOOTSTRAP (HTML, CSS)
- JAVASCRIPT
- PHP
- MYSQL

System Architecture



This application follows a simple three-tiered structural model. Here, the three tiers are called presentation, application and storage respectively.

The user interface is the first tier (presentation), also called as the Front-End. A web server using some dynamic web content technology is the middle tier (application logic). A back-end database or data store that manages and provides access to the data is the third tier (storage).

Modules and its description

This application consists a single module which is the Admin module which in turn consists of several sub-modules like student profile, add personal details, add academic details, add mentor details, delete records, update details and so on.

Description:**1. Student Profile:**

This application provides an efficient way to retrieve the students' data. Admin can simply enter the roll number of the student whose profile is required. Otherwise, he can enter his/her name (part of name) and year of admission so that the list of students will be displayed and select the one we need. Even if the admin simply enters the year of admission, the entire list of students corresponding to that batch will be displayed.

2. Add Academic Details:

The admin can upload the academic details of students. The advantage in this application is the admin can either enter the details of a single student or upload a file containing the details of all students belonging to a particular batch.

3. Add Personal Details:

The admin can upload the personal details of students. The advantage in this application is the admin can either enter the details of a single student or upload a file containing the details of all students belonging to a particular batch.

4. Add Mentors Details:

The admin can upload the Mentors details of students. The admin can simply upload a file containing the details of all mentors belonging to a particular batch.

5. Update Details:

The admin can update both personal as well as the academic details of students one at a time.

6. Delete Records:

The admin can delete the records belonging to a particular student or a particular batch.

UML DIAGRAMS

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

GOALS OF UML:

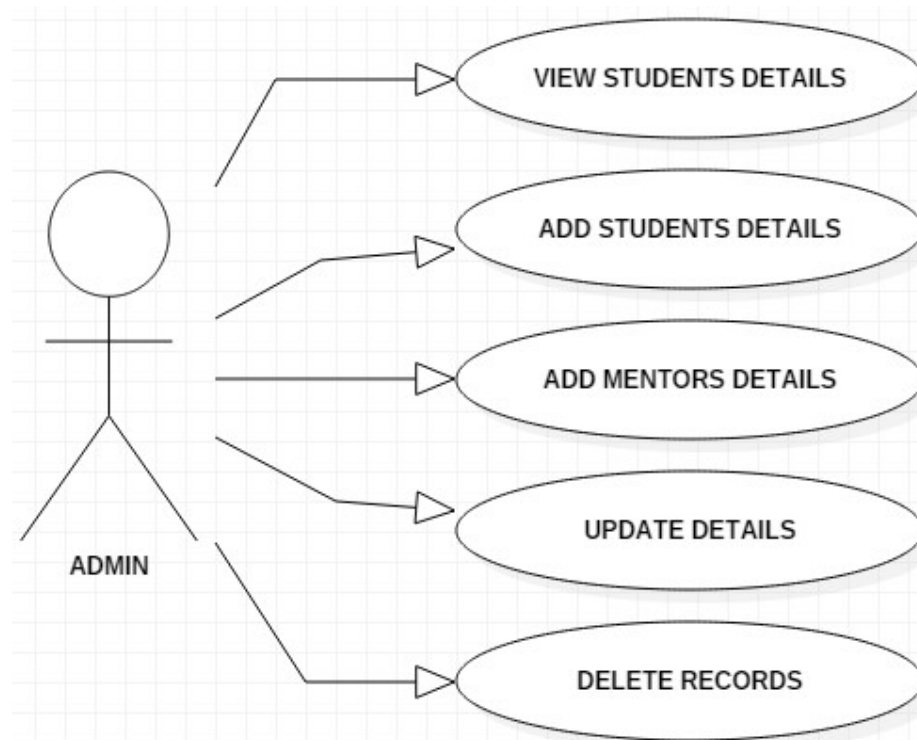
The primary goals in the design of the UML were:

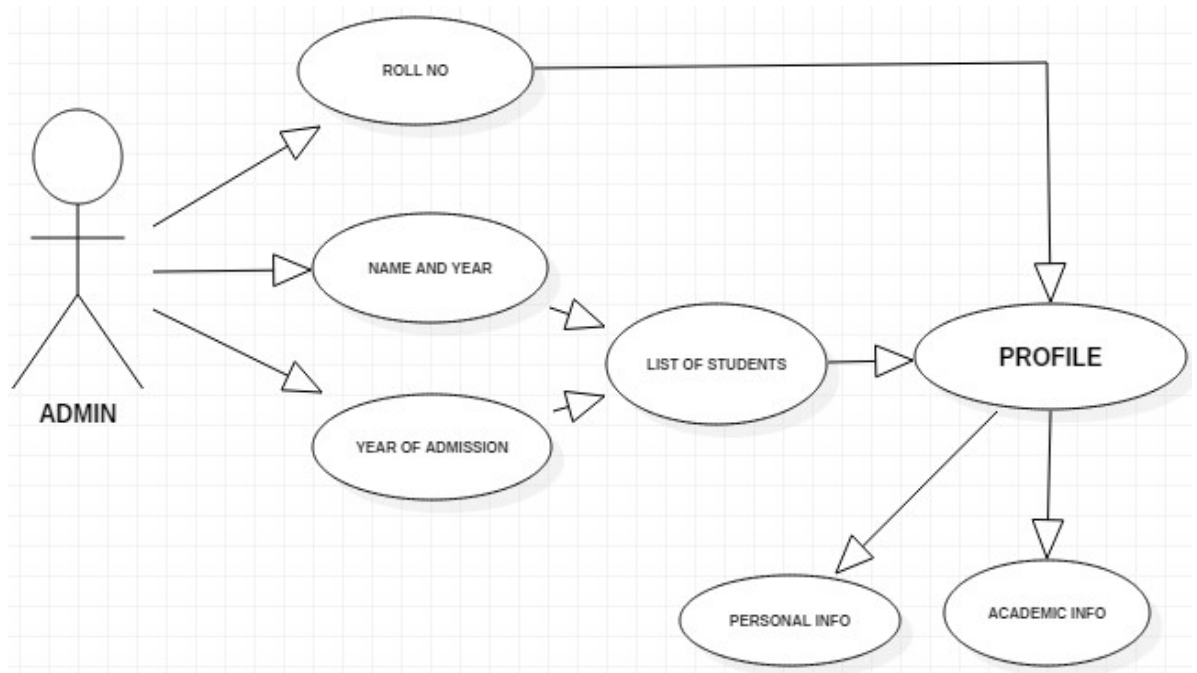
1. Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

Use Case Diagrams

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

USE CASE DIAGRAM OF STUDENT INFORMATION SYSTEM:



USE CASE DIAGRAM TO VIEW A STUDENT'S PROFILE:

IMPLEMENTATION

TECHNOLOGIES USED:

SOFTWARE:

Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is modular and consists of a series of less stylesheets that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

Stylesheets

Bootstrap provides a set of stylesheets that provide basic style definitions for all key HTML components. These provide a uniform, modern appearance for formatting text, tables and form elements.

Re-usable components

In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements. The components are implemented as CSS classes, which must be applied to certain HTML elements in a page.

JavaScript components

Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements,

including for example an auto- complete function for input fields. In version 2.0, the following JavaScript plugins are supported: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Typeahead.

HTML

Hyper Text Markup Language, commonly abbreviated as **HTML**, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically and, before the advent of Cascading Style Sheets (CSS), included cues for the presentation or appearance of the document (web page), making it a markup language, rather than a programming language.

HTML can also be used to add meta information to a webpage. Meta information is information about the web page. For example, the name of the person who made it. Meta information is not usually shown by web browsers. Cascading Style Sheets (CSS) and JavaScript can be included in HTML code. CSS is used to change how a webpage looks. JavaScript is used to add features to web pages and make them more interactive.

HTML elements form the building blocks of HTML pages. HTML allows images and other objects to be embedded and it can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Document type declaration:

HTML documents are required to start with a Document Type Declaration (informally, a "doctype"). In browsers, the doctype helps to define the rendering mode—particularly whether to use quirks mode.

The original purpose of the doctype was to enable parsing and validation of HTML documents by SGML tools based on the Document Type Definition (DTD). The DTD to which the DOCTYPE refers contains a machine-readable grammar specifying the permitted and prohibited content for a document conforming to such a DTD. Browsers, on the other hand, do not implement HTML as an application of SGML and by consequence do not read the DTD.

HTML5 does not define a DTD; therefore, in HTML5 the doctype declaration is simpler and shorter:

```
<!DOCTYPE html>
```

The Document Type Declaration `<!DOCTYPE html>` is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

Mark up:

HTML markup consists of several key components, including those called *tags* (and their *attributes*), character-based data types, character references and entity references. HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some represent empty elements and so are unpaired, for example ``. The first tag in such a pair is the *start tag*, and the second is the *end tag* (they are also called *opening tags* and *closing tags*).

Another important component is the HTML document *type* declaration, which triggers standards mode rendering.

The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages. The above example is made using sample code:


```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

(The text between `<html>` and `</html>` describes the web page, and the text between `<body>` and `</body>` is the visible page content. The markup text "`<title>This is a title</title>`" defines the browser page title.)

- The World Wide Web is composed primarily of HTML documents transmitted from web servers to web browsers using the Hypertext Transfer Protocol (HTTP). However, HTTP is used to serve images, sound, and other content, in addition to HTML. To allow the web browser to know how to handle each document it receives, other information is transmitted along with the document.
- An HTML Application (HTA; file extension ".hta") is a Microsoft Windows application that uses HTML and Dynamic HTML in a browser to provide the application's graphical interface.
- HTML defines several data types for element content, such as script data and style sheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

TAGS:

HTML uses special bits of programming language called "tags" to let the browser know how a webpage should look. The tags usually come in pairs: an *opening* tag to tell the browser when to start doing something, and an *ending* tag to tell the browser when to stop doing something. There are many different kinds of tags, and each one has a different purpose.

Opening tags have a keyword, such as "p," surrounded by angle brackets (< and >). For example, the tag `<p>` tells the browser to start a new paragraph. Ending tags look almost exactly the same, only they have a forward slash (/) added just before the keyword. For example, the tag `</p>` tells the browser to end a paragraph. A small number of tags, like `
` and `<hr>`, can be used without an ending tag.

Some tags only work in certain browsers. For example, the `<marquee>` tag, which is used to make a bit of writing slide across the page, only works in the Internet Explorer and Mozilla Firefox browsers. Other browsers simply ignore this tag and display the writing normally. Many web page creators avoid using these "non-standard" tags because they want their pages to look the same with all browsers.

ATTRIBUTES:

Most of the attributes of an element are name-value pairs, separated by "=" and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML. Leaving attribute values unquoted is considered unsafe.^[60] In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element, like the `ismap` attribute for the `img` element.

There are several common attributes that may appear in many elements:

- The `id` attribute provides a document-wide unique identifier for an element. This is used to identify the element so that stylesheets can alter its presentational properties, and scripts may alter, animate or delete its contents or presentation. Appended to the URL of the page, it provides a globally unique identifier for the element, typically a sub-section of the page.
- The `class` attribute provides a way of classifying similar elements. This can be used for semantic or presentation purposes. For example, an HTML document might semantically use the designation `class="notation"` to indicate that all elements with this class value are subordinate to the main text of the document. In presentation, such elements might be gathered together and presented as footnotes on a page instead of appearing in the place where they occur in the HTML source. Class

attributes are used semantically in micro formats. Multiple class values may be specified; for example `class="notation important"` puts the element into both the "notation" and the "important" classes.

- An author may use the `style` attribute to assign presentational properties to a particular element. It is considered better practice to use an element's `id` or `class` attributes to select the element from within a style sheet, though sometimes this can be too cumbersome for a simple, specific, or ad hoc styling.
- The `title` attribute is used to attach sub textual explanation to an element. In most browsers this attribute is displayed as a tooltip.
- The `lang` attribute identifies the natural language of the element's contents, which may be different from that of the rest of the document.

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for *Personal Home Page*, but it now stands for the recursive backronym *PHP: Hypertext Preprocessor*.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

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

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification.

Syntax:

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP, although non-PHP text is still subject to control structures described in PHP code. The most common delimiters are `<?php` to open and `?>` to close PHP sections. The shortened form `<?=` also exists. This short delimiter makes script files less portable, since support for them can be disabled in the local PHP configuration and it is therefore discouraged. However, there is no recommendation against the use of the echo short tag `<?=`. Prior to PHP 5.4.0, this short syntax for `echo()` only works with the `short_open_tag` configuration setting enabled, while for PHP 5.4.0 and later it is always available. The purpose of all these delimiters is to separate PHP code from non-PHP content, such as JavaScript code or HTML markup.

The first form of delimiters, `<?php` and `?>`, in XHTML and other XML documents, creates correctly formed XML processing instructions. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. PHP 5 introduced *type hinting* that allows functions to force their parameters

to be objects of a specific class, arrays, interfaces or callback functions. However, before PHP 7.0, type hints could not be used with scalar types such as integer or string.

Unlike function and class names, variable names are case sensitive. Both double-quoted (") and heredoc strings provide the ability to interpolate a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language, and statements are terminated by a semicolon. PHP has three types of comment syntax: `/* */` marks block and inline comments; `//` as well as `#` are used for one-line comments. The `echo` statement is one of several facilities PHP provides to output text, *e.g.*, to a web browser. In terms of keywords and language syntax, PHP is similar to the C style syntax. *if* conditions, *for* and *while* loops, and function returns are similar in syntax to languages such as C, C++, C#, Java and Perl.

JAVASCRIPT:

JavaScript, often abbreviated as "JS", is a high-level, dynamic, untyped, and interpreted run-time language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production; the majority of websites employ it, and all modern Web browsers support it without the need for plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two are distinct languages and differ greatly in their design. JavaScript was influenced by programming languages such as Self and Scheme.

The most common use of JavaScript is to add client-side behavior to HTML pages, also known as Dynamic HTML (DHTML). Scripts are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page. Some simple examples of this usage are:

- Loading new page content or submitting data to the server via Ajax without reloading the page (a social network might allow the user to post status updates without leaving the page).
- Animation of page elements, fading them in and out, resizing them, moving them, etc.
- Interactive content, for example games, and playing audio and video.
- Validating input values of a Web form to make sure that they are acceptable before being submitted to the server.
- Transmitting information about the user's reading habits and browsing activities to various websites. Web pages frequently do this for Web analytics, ad tracking, personalization or other purposes.

Because JavaScript code can run locally in a user's browser (rather than on a remote server), the browser can respond to user actions quickly, making an application more responsive. Furthermore, JavaScript code can detect user actions that HTML alone cannot, such as individual keystrokes. Applications such as Gmail take advantage of this: much of the user-interface logic is written in JavaScript, and JavaScript dispatches requests for information (such as the content of an e-mail message) to the server. The wider trend of Ajax programming similarly exploits this strength.

Example script

Below is a minimal example of a standards-conforming Web page containing JavaScript (using HTML 5 syntax) and the DOM:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <button id="hellobutton">Hello</button>
    <script>
      document.getElementById("hellobutton").onclick = function() {
        alert('Hello world!');           // Show a dialog
```

```
var myTextNode = document.createTextNode('Some new words.');
```

document.body.appendChild(myTextNode); // Append "Some new words" to the page

```
};
```

```
</script>
```

```
</body>
```

```
</html>
```

MYSQL:

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter,^[8] and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.^[9] For proprietary use, several paid editions are available, and offer additional functionality.

PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.

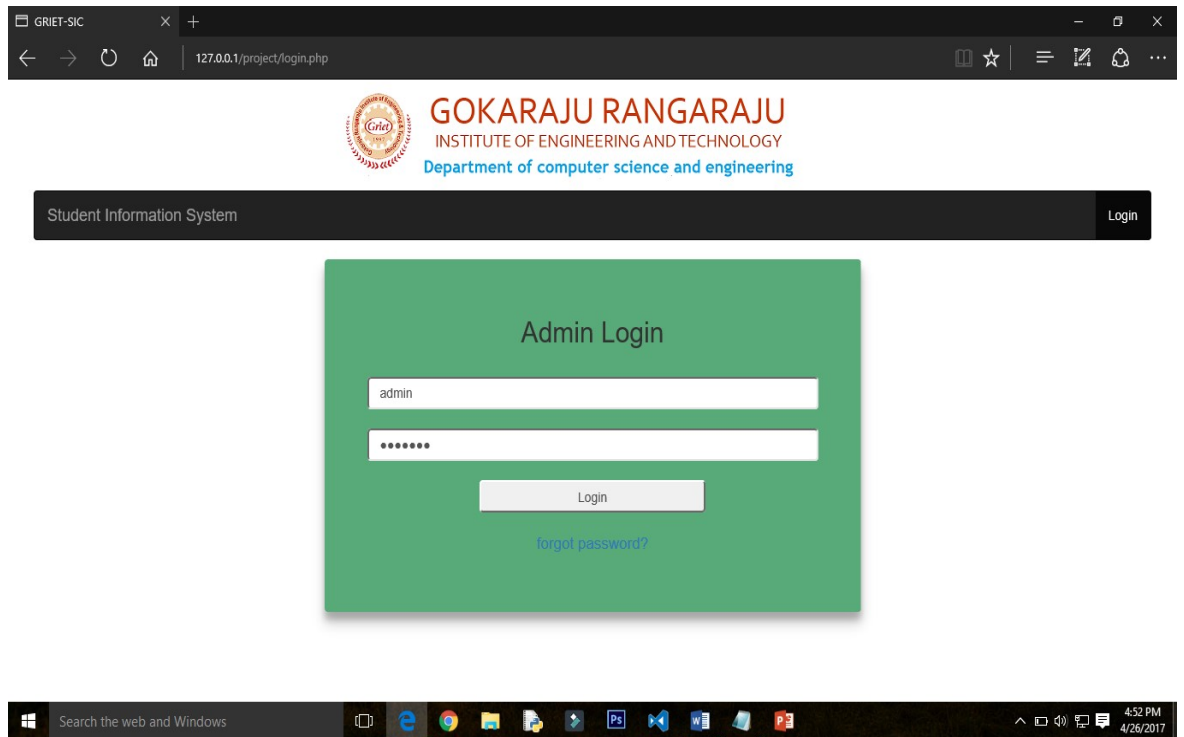
phpMyAdmin

phpMyAdmin is a free and open source tool written in PHP intended to handle the administration of MySQL with the use of a web browser. It can perform various tasks such as creating, modifying or deleting databases, tables, fields or rows; executing SQL statements; or managing users and permissions. The software, which is available in 78 languages, is maintained by *The phpMyAdmin Project*.

It can import data from CSV and SQL, and transform stored data into any format using a set of predefined functions, like displaying BLOB-data as images or download-links.

FRONT END OF WEB PAGE:

a) Login Page:



b) Home Page:

c) Searching for a student using Roll Number:

GRIET-SIC

127.0.0.1/project/index1.php

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
Department of computer science and engineering

Student Information System

Home Profile

Student Information Center

14241A05C7

Name

year of admission


submit

Search the web and Windows


4:49 PM
4/26/2017

d) Profile:

GRIET - SIC
127.0.0.1/project/student.php?i=14241A05C7&j=14


GOKARAJU RANGARAJU
 INSTITUTE OF ENGINEERING AND TECHNOLOGY
 Department of computer science and engineering

Student Profile
Home
profile
Back



Roll Number: 14241A05C7

Name: Dasari Bala Sundeep Krishna

Rank: 8814

Course: BTECH

Department: cse

Section: c

Semester: 06

Mentor: B. Padma Vijetha Dev
padmavijetha@gmail.com 8106999514

Personal details

DOB: 11/12/1996

Father: Dasari Latish Kumar

Email:

Phone1: [9502038394](tel:9502038394)

Phone2: [9912159986](tel:9912159986)

Address: jhansilatish@yahoo.co.in, Fl.No: 303, Metro Residency, KRCCR Colony, Bachupally, Qutubullapur, Ranga R

Academic details

Category	I-I	I-II	II-I	II-II	III-I	III-II	IV-I	IV-II
Total	694	667	736	637	650	--	--	--
Backlogs	0	0	0	0	0	--	--	--
Aggregate	84.12	80.85	89.21	87.86	89.66	--	--	--
Credits	25	25	25	25	25	--	--	--

Total Aggregate: 86.34

Total Backlogs: 0

Search the web and Windows
4:49 PM 4/26/2017

e) Searching for a student using name and year of admission:

The screenshot displays a web browser window with the address bar showing '127.0.0.1/project/index1.php'. The page header includes the GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY logo and the text 'Department of computer science and engineering'. Below the header is a navigation bar with 'Student Information System', 'Home', and 'Profile' links. The main content area features a green box titled 'Student Information Center' containing a search form. The form has three input fields: 'RollNo' (empty), 'sai' (text), and '2014' (dropdown menu). A 'submit' button is located below the fields. The Windows taskbar at the bottom shows the search bar and various application icons, with the system clock indicating 4:50 PM on 4/26/2017.

Student Information System

Home Profile

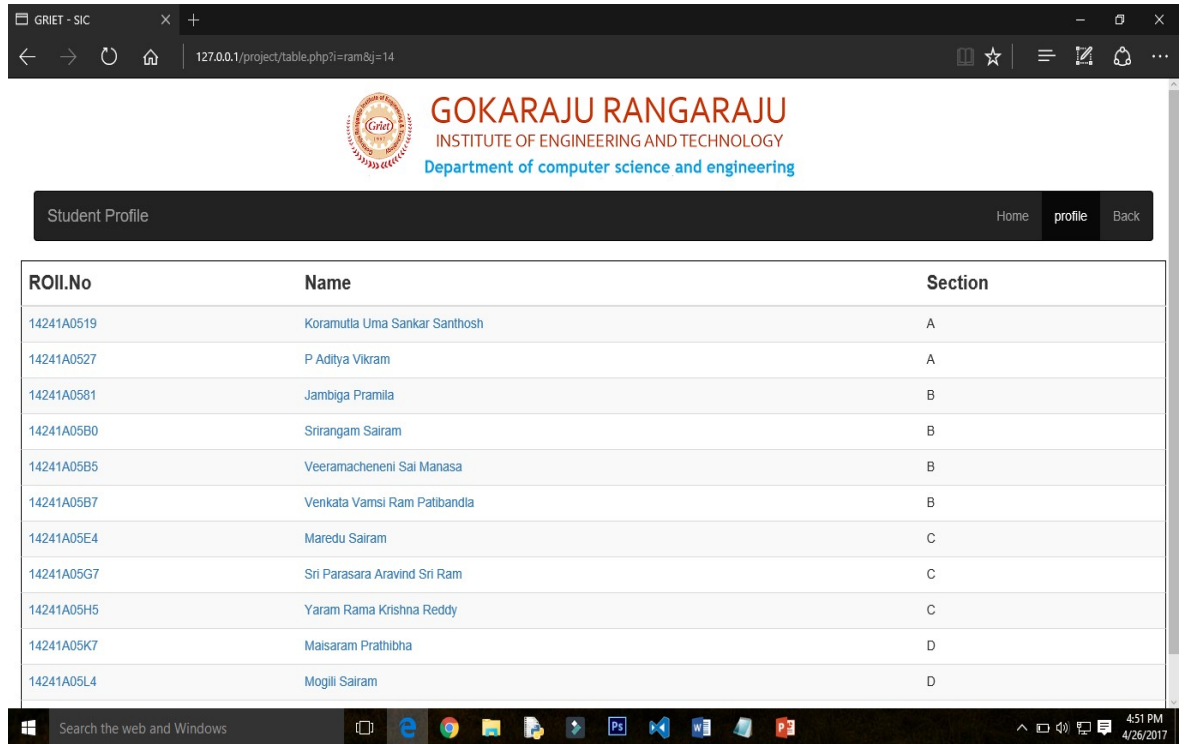
Student Information Center

RollNo

sai

2014

submit

f) List of students with the given name:

Student Profile

Home profile Back

ROII.No	Name	Section
14241A0519	Koramutla Uma Sankar Santhosh	A
14241A0527	P Aditya Vikram	A
14241A0581	Jambiga Pramila	B
14241A05B0	Srirangam Sairam	B
14241A05B5	Veeramacheneni Sai Manasa	B
14241A05B7	Venkata Vamsi Ram Patibandla	B
14241A05E4	Maredu Sairam	C
14241A05G7	Sri Parasara Aravind Sri Ram	C
14241A05H5	Yaram Rama Krishna Reddy	C
14241A05K7	Maisaram Prathibha	D
14241A05L4	Mogili Sairam	D

g) Adding students details:

The screenshot displays a web browser window with the address bar showing '127.0.0.1/project/scpersonal.php'. The page header includes the GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY logo and the text 'Department of computer science and engineering'. Below the header is a navigation bar with 'Student Information System' and 'New Students' links. The main content area features a large orange box titled 'Create Student Dataset'. Inside this box is a form with two radio buttons: 'Single Entry' (selected) and 'Bulk Upload'. A 'Submit' button is located below the form. The Windows taskbar at the bottom shows the search bar and various application icons, with the system clock indicating 4:51 PM on 4/26/2017.

FUTURE ENHANCEMENT

The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Improvements can be appended by changing the existing module or adding new modules. One important development that can be added to the project in future is we can host this and make it a web application by modifying the authentication and increasing the modules.

CONCLUSION

The entire project has been developed and deployed as per the requirements stated by the user, it is found to be bug free as per the testing standards that are implemented. Any specification untraced errors will be concentrated in the coming versions, which are planned to be developed. This system reduces the manual work by automating the process of maintaining students' information. This application can be used by any institution as it is very user friendly, portable and secured.

BIBLIOGRAPHY

Websites:

1. <https://www.coursera.org/learn/html-css-javascript-for-web-developers>.
2. <https://www.w3schools.com/php/default.asp>.
3. <https://www.w3schools.com/bootstrap/default.asp>.
4. https://www.w3schools.com/php/php_mysql_intro.asp.
5. <https://www.w3schools.com/css/default.asp>.
6. <https://www.w3schools.com/js/default.asp>.