**A PROJECT REPORT ON**

**GRIET STUDENT PORTAL**

(A Main Project report submitted in partial fulfillment of the requirements for the award of the degree of)

BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY

(2010-2014)

By

PRUDHVI RAJ MUDUNURI (11241A12A3)

VARUN KODAVEETI (11241A12A3)

CH.SAMBI REDDY (11241A12A3)

MOHD.IRFAN (11241A12A3)

Under the Esteemed Guidance of

SUKANYA ,Assistant Professor

(Department of IT)



GOKARAJU RANGARAJU INSTITUTE OF

ENGINEERING AND TECHNOLOGY (Bachupally, Kukatpally, Hyderabad – 500 090 )



CERTIFICATE

This is to certify that it is a bonafide record of dissertation work entitled “**GRIET STUDENT PORTAL**” done by **PRUDHVI RAJ MUDUNURI(11241A12A3),VARUN KONDAVEETI(11241A12A3),CH.SAMBIREDDY(11241A12A3),MOHD.IRFAN(11241A12A3),** students of **B.Tech (IT)** in the Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology during the period 2010-2014 in the partial fulfillment of the requirements for the award of degree of B.Tech in Information Technology. The results presented in this dissertation have been verified and are found to be satisfactory. This work is not submitted to any other university for the award of any Degree/Diploma.

**ACKNOWLEDGEMENT**

I wish to express our deep gratitude to our guide **SUKANYA**, Assistant Professor in the Department of Information Technology, for all the advice, encouragement and constant support she has given us throughout our project work. This work would not have been possible without her support and valuable suggestions.

We are grateful to **Dr.Y.Vijayalata**, Head of the Department of InformationTechnology and the Members of the Project Review Committee **Dr.Y.Vijayalata, Mrs. K. Prasanna Lakshmi, Mr.Nagendra, Mr.Gopala Krishna, Mr.Vijendar Reddy,** and **Mr. S Palaniappan**, for their valuable suggestions**.**

We are also grateful to **Dr. Jandhyala N Murthy**, Principal and **Prof. P.S.Raju**, Director of GRIET for giving us the necessary facilities to carry out our project work successfully We would like to thank all our friends for their help and constructive criticism during our project work.

**PRUDHVI RAJ MUDUNURI (11241A12A3)**

**VARUN KODAVEETI (11241A12A3)**

**CH.SAMBI REDDY (11241A12A3)**

**MOHD.IRFAN (11241A12A3)**

**ABSTRACT**

The project is web portal of Information Technology Department and

works as an interface between the students and the Faculty of the Department.

This project provides students following

• profile of faculty associated with the department

• View the updates about the Department

• View the information about the Faculties

• View the Photo Gallery of the Department

• To provide access to student the details about his/her attendance, marks and can also have access to online classes.•It also provides access to the professor’s remarks on the student.

•Student portal also contains a student professor interaction where the student can post his/her doubts and the professor can answer them.

**“GRIET STUDENT PORTAL” maintains the details of various asked**

**questions posted by the students to the faculty of consult subject and also**

**download the study materials of the subject uploaded by the faculty.**

**TABLE OF CONTENTS**

**1. INTRODUCTION** ..................................................................................................

1.1 PURPOSE OF THE PROJECT .............................................................................................

1.2 PROJECT APPROACH .....................................................................................................

1.3 PROJECT GOALS AND OBJECTIVES................................................................................

1.4 ROLES AND RESPONSIBILITIES .................................................................................

**2. SYSTEM ANALYSIS** ..............................................................................................

2.1 EXISTING SYSTEM ........................................................................................................

2.2 PROPOSED SYSTEM ......................................................................................................

2.3 REQUIREMENT ANALYSIS ...............................................................................................

2.4 REQUIREMENT SPECIFICATIONS .................................................................................

2.5 FEASIBILITY STUDY .......................................................................................................

**3. SYSTEM DESIGN** .................................................................................................

3.1 MODULE DESCRIPTION .................................................................................................

3.2 UML DIAGRAMS ................................................................................................

3.3 ENTITY RELATIONSHIP DIAGRAMS ...............................................................................

3.4 UML DIAGRAMS ........................................................................................................

3.5 HARDWARE AND SOFTWARE REQUIREMENTS……………………………………..

**4. TECHNOLOGIES USED** .......................................................................................

4.1 HTML .....................................................................................................................

4.2 CSS ........................................................................................................................

4.3 PHP ........................................................................................................................

4.4 MYSQL ....................................................................................................................

4.5 XAMPP ....................................................................................................................

5. CONCLUSION ...............................................................................................

6.BIBLIOGRAPHY........................................................................................

**LIST OF FIGURES**

**UML DIAGRAMS .....................................................................................**

**3.1 ENTITY RELATIONSHIP DIAGRAM FOR IT DEPARTMENT PORTAL ..............**

**3.2 USE CASE DIAGRAM FOR ADMINISTRATOR ...........................................**

**3.3 USE CASE DIAGRAM FOR FACULTY ......................................................**

**3.4 USE CASE DIAGRAM FOR STUDENT ............................................**

**3.5 STATE CHART DIAGRAM FOR ADMINISTRATOR ...........................**

**3.6 STATE CHART DIAGRAM FOR FACULTY ......................................**

**3.7 STATE CHART DIAGRAM FOR STUDENT .....................................**

**3.8 SEQUENCE DIAGRAM FOR ADMINISTRATOR .................................**

**3.9 SEQUENCE DIAGRAM FOR FACULTY ..............................................**

**3.10 SEQUENCE DIAGRAM FOR STUDENT ................................................................**

**INTRODUCTION**

**1.1Purpose of the Project**

GRIET STUDENT PORTAL is an online web portal for the students and faculties. It

is an interactive service for students and information seekers of GRIET Information

Technology Department. It is easy to access and use for communication between

faculties and students. It can also be used for keeping track of information and

documents of students and faculties as well.

**1.2 Modules**

This section gives an outline of the way we developed the application,

including the highest level milestones:

**Module I:** Gathering of information regarding the various technologies to be used to

develop a IT department portal.

**Module II:** We learned the technologies to be used for the developed during this phase

by listening to tutorials which helped us gain knowledge about the implementation of

the technologies.

**Module III:** Developing was next aspect and did it by using various technologies

Joomla and Xampp server.

**Modules IV:** We installed the Joomla using the Xampp server and developed our

application using various settings and coding.

**Phase V:** As this application was meant to be a cross-platform one,we had to ensure

that it works on different platforms.For this,we made it run on various operating

systems and on different browsers.

**Project Goals and objectives**

The application goals and objectives for this project will focus on providing the

students:

•view marks ,attendance.

• View the updates about the Department

• View the information about the Faculties

• View the Photo Gallery of the Department

• Cross check attendance with Attendence tracker

**Roles and Responsibilities**

The student and faculty are provided with the unique user id and password by

the administrator. With this both the students and faculty are interacted among themselves and ask queries, read remarks.

**SYSTEM ANALYSIS**

**2.1Existing System**

There is no such system available for department. The students cannot communicatewith teachers properly. There is a great problem in keeping recordof daily attendance, performance of each student, profile of each person associated with the department, and there was a lot of difficulties for the student regarding interaction with the faculties and viewing the current updates.

**2.2Proposed System**

The proposed system is to make an online web portal for IT Department, so that everyone can get the information about the department, students can easily communicate with teachers from anywhere. And can keep records of every student easily and efficiently. The proposed system could be accessed from any corner of the world on net.

**2.3 Requirement analysis**

The requirement phase basically consists of three activities:

2.3.1 Requirement Analysis

2.3.2 Requirement Specification

2.3.3 Requirement Validation

2.3.1 Requirement Analysis

Requirement Analysis is a software engineering task that bridges the gap between system level software allocation and software design. It provides the system engineer to specify software function and performance indicate software’s interface with the other system elements and establish constraints that software must meet.

The basic aim of this stage is to obtain a clear picture of the needs and requirements of the end-user and also the organization. Analysis involves interaction between the clients and the analysis. Usually analysts research a problem any asking questions and reading existing documents. The analysts have to uncover the real needs of the user even if they don’t know them clearly. During analysis it is essential that a complete and consistent set of specifications emerge for the system. Here it is essential to resolve the contradictions that could emerge from information got from various parties. This is essential to ensure that the final specifications are consistent. It may be divided into 3 areas of effort.

2.3.1.1)Problem recognition

2.3.1.2)Evaluation and synthesis

2.3.1.3) Specification

Each Requirement analysis method has a unique point of view. However all analysis methods are related by a set of operational principles. They are

• The information domain of the problem must be represented and understood.

• The functions that the software is to perform must be defined.

• The behavior of the software as a consequence of external events must be defined.

• The models that depict information, function and behavior must be partitioned in a

hierarchical or layered fashion.

• The analysis process must move from essential information to implementation detail.

**2.3.1.1 Problem Recognition**

The main problem here is the more time is taken to answer the queries of students. This has to be eliminated. A comprehensive solution has to be developed which will facilities to answer the queries in a faster and more efficient way.

**2.3.1.2 Evaluation and Synthesis:**

The system has to be designed only after complete evaluation of the existing one, upon which we can see that a lot depends on the medium of communication. In the proposed system the information about FAQ’s, processing of the request and the time taken to answer the query are very effective and convenient. So this has to be used such that there is no waste of time.

**2.3.1.3 Specification**

This division provided the required format for the FAQ-info ware. The appearance of forms, and their field names, the different screens he desired, the stages of this database etc., were all given. The system has been built following all the specifications.

**2.4 Requirements specifications**

**2.4.1 Specification Principles:**

Software Requirements Specification plays an important role in creating quality software solutions. Specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation. Requirements may be specified in a variety of ways. However there are some guidelines worth following:

• Representation format and content should be relevant to the problem

• Information contained within the specification should be nested

• Diagrams and other notational forms should be restricted in number and consistent in

use.

• Representations should be revisable.

**2.4.2 Software Requirements Specifications:**

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, and indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements.

**2.4.3 An outline of the Software Requirements Specification:**

A simplified outline can be given for the framework of the specifications. This is according to the IEEE Standards. 14

**2.5 Feasibility study**

All projects are feasible given unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time.Three key considerations are involved in the feasibility analysis.

**2.5.1 Economic Feasibility:**

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

**2.5.2 Technical Feasibility:**

Technical feasibility centers around the existing computer system (hardware, software, etc.,) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible.

**2.5.3 Operational Feasibility:**

People are inherently resistant to change, and computers have been known to facilitate change. It is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

**SYSTEM DESIGN**

**3.1 Module Description**

The various modules use in the Computer Science Department Web Portal is explained below.

**3.1.1 Students**:

This is a separate webpage for students. They can download study materials, tutorials and ask queries to the faculty.

**3.1.2 Faculty**:

This module gives profile information of various faculties associated to the department. It gives profile information of all the faculties. Anyone can contact to these faculties through available email addresses.

**3.1.3 Queries**:

With this module, students can ask the questions and in reply faculty can answer them.

**3.1.4Tutorials**:

In this module, students can download the tutorials of the topics which are not in the syllabus. This module could be edited by the administrator of the web portal.

**3.1.5 Academics**:

The syllabus, time tables and academic calendar of various branches associated with Information technology Department is available here.

**3.5 Hardware and Software Requirements**

**3.5.1)Hardware Specifications:**

• Processor: Intel i3.

• CPU speed: 833 MHz to 2.53GHz or more.

• RAM: min 120 MB.

• Hard Disk: min60GB.

**3.5.2)Software specifications:**

• Operating system: windows 2000, 2003, XP, vista, 7 or later.

• Technologies: Joomla, PHP.

• Web Server: Xampp server.

• DataBase: My SQL.

**CHAPTER 5**

**TECHNOLOGIES IMPLEMENTED**

**5.1 HTML**

Hypertext Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, the second tag is the end tag(they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML web pages.

**5.1.1 Elements**

HTML documents are composed entirely of HTML elements that, in their most general form have three components: a pair of tags, a "start tag" and "end tag"; some attributes within the start tag; and finally, any textual and graphical content between the start and end tags, perhaps including other nested elements. The HTML element is everything between and including the start and end tags. Each tag is enclosed in angle brackets. 38

The general form of an HTML element is therefore: <tag attribute1="value1" attribute2="value2">content</tag>. Some HTML elements are defined as empty elements and take the form <tag attribute1="value1" attribute2="value2" >. Empty elements may enclose no content, for instance, the BR tag or the inline IMG tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character, "/", and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case.

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. This data usually includes the MIME type (e.g. text/html or application/xhtml+xml) and the character encoding (see Character encoding in HTML).

In modern browsers, the MIME type that is sent with the HTML document may affect how the document is initially interpreted. A document sent with the XHTML MIME type is expected to be well-formed XML; syntax errors may cause the browser to fail to render it. The same document sent with the HTML MIME type might be displayed successfully, since some browsers are more lenient with HTML.

**5.1.2 Structural markup describes the purpose of text**

For example, <h2>Golf</h2> establishes "Golf" as a second-level heading. Structural markup does not denote any specific rendering, but most web browsers have default styles for element formatting. Content may be further styled using Cascading Style Sheets (CSS).

**5.1.3.Presentational markup describes the appearance of the text, regardless of**

**its purpose**

For example <b>boldface</b> indicates that visual output devices should render "boldface" in bold text, but gives little indication what devices that are unable to do this should do.In case of both <b>bold</b> and <i>italic</i> there are other elements that may have equivalent visual renderings but which are more semantic in nature, such as <strong>strong text</strong> and <em>emphasised text</em> respectively. It is easier to see how an aural user agent should interpret the latter two elements. However, they are not equivalent to their presentational counterparts: it would be undesirable for a screenreader to emphasize the name of a book, for instance, but on a screen such a name would be italicized. Most Presentational markup elements have become deprecated under the HTML 4.0 specification, in favor of using CSS for styling.

**5.1.4 Data types**

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.

**5.2 CASCADING STYLE SHEETS** (**CSS**)

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing

the presentation semantics (the look and formatting) of a document written in

a markup language. Its most common application is to style web pages written

in HTML and XHTML, but the language can also be applied to any kindof

XML document, including plain XML,SVG and xul

CSS is designed primarily to enable the separation of document content

(written in HTML or a similar markup language) from document presentation,

including elements such as the layout, colors, and fonts. This separation can improve

content accessibility, provide more flexibility and control in the specification of

presentation characteristics, enable multiple pages to share formatting, and reduce

complexity and repetition in the structural content (such as by allowing for table less

web design).

40

CSS can also allow the same markup page to be presented in different styles

for different rendering methods, such as on-screen, in print, by voice (when read out

by a speech-based browser or screen reader) and on Braille-based, tactile devices.

It can also be used to allow the web page to display differently depending on

the screen size or device on which it is being viewed. While the author of a document

typically links that document to a CSS style sheet, readers can use a different style

sheet, perhaps one on their own computer, to override the one the author has

specified.

CSS specifies a priority scheme to determine which style rules apply if more

than one rule matches against a particular element. In this so-called *cascade*, priorities

or *weights* are calculated and assigned to rules, so that the results are predictable.CSS

has a simple syntax and uses a number of English keywords to specify the names of

various style properties.

A style sheet consists of a list of *rules*. Each rule or rule-set consists of one or

more *selectors*, and a *declaration block*. A declaration-block consists of a list

of *declarations* in braces. Each declaration itself consists of a *property*, a colon (:),

and a *value*. If there are multiple declarations in a block, a semi-colon (;) must be

inserted to separate each declaration CSS information can be provided from various

sources.

CSS style information can be in a separate document or it can be embeddedinto an HTML document. Multiple style sheets can be imported. Different styles canbe applied depending

on the output device being used; for example, the screen versioncan be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.

Priority scheme for CSS sources (from highest to lowest priority):

• Author styles (provided by the web page author), in the form of:

• **Inline styles**, inside the HTML document, style information on a single element,

specified using the style attribute

• **Embedded style**, blocks of CSS information inside the HTML itself

• **External style sheets**, i.e., a separate CSS file referenced from the document

**5.2.1 Separation of content from presentation**

CSS facilitates publication of content in multiple presentation formats based

on nominal parameters. Nominal parameters include explicit user preferences,

different web browsers, the type of device being used to view the content (a desktop

computer or mobile Internet device), the geographic location of the user and many

other variables.

**5.2.2 Site-wide consistency**

When CSS is used effectively, in terms of inheritance and "cascading," a

global style sheet can be used to affect and style elements site-wide. If the situation

arises that the styling of the elements should need to be changed or adjusted, these

changes can be made by editing rules in the global style sheet. Before CSS, this sort

of maintenance was more difficult, expensive and time-consuming.

**5.2.3 Bandwidth**

A style sheet, internal or external, will specify the style once for a range of

HTML elements selected by class, type or relationship to others. This is much more

efficient than repeating style information inline for each occurrence of the element.

An external style sheet is usually stored in the browser cache, and can therefore be

used on multiple pages without being reloaded, further reducing data transfer over a

network.

**5.2.4 Page reformatting**

With a simple change of one line, a different style sheet can be used for the

same page. This has advantages for accessibility, as well as providing the ability to

tailor a page or site to different target devices. Furthermore, devices not able to

understand the styling still display the content.

**5.2.5 Accessibility**

Without CSS, web designers must typically lay out their pages with techniques

that hinder accessibility for vision-impaired users, like HTML tables (see Table less

web design Accessibility).

**5.3 PHP**

**PHP** is a widely used, general-purpose scripting language that was originally

designed for web development, to produce dynamic web pages. It can be embedded

into HTML and generally runs on a web server, which needs to be configured to

process PHP code and create web page content from it. It can be deployed on most

web servers and on almost every operating system and platform free of charge. PHP is

installed on over 20 million websites and 1 million web servers.

PHP was originally created by RasmusLerdorf in 1995 and has been in continuous

development ever since. The main implementation of PHP is now produced by The

PHP Group and serves as the de facto standard for PHP as there is no formal

specification. PHP is free software released under the PHP License, which is

incompatible with the GNU General Public License (GPL) because of restrictions on

the use of the term PHP.

PHP has evolved to include a command line interface capability and can also be used

in standalone graphical applications.

**5.3.1What is PHP?**

PHP is a widely-used open source general-purpose scripting language that is

especially suited for web development and can be embedded into HTML. Nice, but

what does that mean? An example:

**Example A1n introductory example**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>Example</title>

</head>

<body>

<?php

echo "Hi, I'm a PHP script!";

?>

</body>

</html>

Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages

contain HTML with embedded code that does "something" (in this case, output "Hi,

I'm a PHP script!"). The PHP code is enclosed in special start and end processing

instructions <?php and ?> that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is

executed on the server, generating HTML which is then sent to the client. The client

would receive the results of running that script, but would not know what the

underlying code was. You can even configure your web server to process all your

HTML files with PHP, and then there's really no way that users can tell what you

have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but

offers many advanced features for a professional programmer. Don't be afraid reading

the long list of PHP's features. You can jump in, in a short time, and start writing

simple scripts in a few hours.

**5.3.2 What can PHP do?**

Anything. PHP is mainly focused on server-side scripting, so you can do anything any

other CGI program can do, such as collect form data, generate dynamic page content,

or send and receive cookies. But PHP can do much more.

There are three main areas where PHP scripts are used.

Server-side scripting. This is the most traditional and main target field for PHP. You

need three things to make this work. The PHP parser (CGI or server module), a web

server and a web browser. You need to run the web server, with a connected PHP

installation. You can access the PHP program output with a web browser, viewing the

PHP page through the server. All these can run on your home machine if you are just

experimenting with PHP programming. See the installation instructions section for

more information.

Command line scripting. You can make a PHP script to run it without any server or

browser. You only need the PHP parser to use it this way. This type of usage is ideal

for scripts regularly executed using cron (on \*nix or Linux) or Task Scheduler (on

Windows). These scripts can also be used for simple text processing tasks. See the

section about Command line usage of PHP for more information.

Writing desktop applications. PHP is probably not the very best language to create a

desktop application with a graphical user interface, but if you know PHP very well,

and would like to use some advanced PHP features in your client-side applications

you can also use PHP-GTK to write such programs. You also have the ability to write

cross-platform applications this way. PHP-GTK is an extension to PHP, not available

in the main distribution. If you are interested in PHP-GTK, visit » its own website.

PHP can be used on all major operating systems, including Linux, many Unix variants

(including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC

OS, and probably others. PHP has also support for most of the web servers today.

This includes Apache, Microsoft Internet Information Server, Personal Web Server,

Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami,

OmniHTTPd, and many others. For the majority of the servers, PHP has a module, for

the others supporting the CGI standard, PHP can work as a CGI processor.

One of the strongest and most significant features in PHP is its support for a wide

range of databases. Writing a database-enabled web page is incredibly simple. The

following databases are currently supported:

Adabas D

dBase

Empress

FilePro (read-only)

Hyperwave

IBM DB2

Informix

**5.5 Mysql**

**5.5.1 What is MySQL?**

MySQL is a database. MySQL is the most popular open-source database

system. The free MySQL database is very often used with PHP. A database holds one

or multiple tables.

The data in MySQL is stored in database objects called tables.

A table is a collection of related data entries and it consists of columns and rows.

Databases are useful when storing information categorically. A company may have a

database with the following tables: "Employees", "Products", "Customers" and

"Orders".

**5.5.2 Database Tables**

A database most often contains one or more tables. Each table is identified by

a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

Below is an example of a table called "Persons":

|  |  |  |  |
| --- | --- | --- | --- |
| LastName | FirstName | Address | City |
| Hansen | Ola | Timoteivn 10 | Sandnes |
| Svendson | Tove | Borgvn 23 | Sandnes |
| Pettersen | Kari | Storgt 20 | Stavanger |

Ingres

InterBase

FrontBase

mSQL

Direct MS-SQL

MySQL

ODBC

Oracle (OCI7 and OCI8)

Ovrimos

PostgreSQL

SQLite

Solid

Sybase

Velocis

Unix dbm

**5.3.3 Why use PHP-Nuke and not static HTML pages?**

Because managing large sites with only static HTML pages is dangerous for your

health.

Because through the dynamic pages, users can interact (Forum, chat)

Because through the dynamic pages we can offer value added services (restricted

areas, various services based on user classification...)

Because the information is more easily catalogued.

Because with a few PHP pages we recall a lot of information.

Because keeping the contents up-to-date does not demand particular technical

expertise and can be managed by anyone (by Davis Batistes).

It is very intuitive and easy to learn (by Anonymous)

It is easy to modify by those who intend to personalize the program (By Arus)

It is easy to use by the lesser experts among us.

**5.5.3Queries**

A query is a question or a request.

With MySQL, we can query a database for specific information and have a recordset

returned.

Look at the following query:

SELECT LastName FROM Persons

The query above selects all the data in the "LastName" column from the "Persons"

table, and will return a recordset like this:

|  |
| --- |
| LastName |
| Hansen |
| Svendson |
| Pettersen |

**5.5.4 Facts about MySQL Database**

One great thing about MySQL is that it can be scaled down to support

embedded database applications. Perhaps it is because of this reputation that many

people believe that MySQL can only handle small to medium-sized systems.

The truth is that MySQL is the de-facto standard database for web sites that support

huge volumes of both data and end users (like Friendster, Yahoo and Google).

**5.6 XAMPP**

Developer(s) Apache Friends

Stable release 1.7.2/2013-04-10

Operating system windows

Type WAMP,MAMP,SAMP,LAMP

License GPL

Website <http://www.apachefriends.org/en/xamp.html>

XAMPP (pronounced /ˈzæmp/ or /ˈɛks.æmp) is a free and open sourcecrossplatformweb

server package, consisting mainly of the Apache HTTP Server,

MySQLdatabase, and interpreters for scripts written in the PHP and Perlprogramming

languages.

X (meaning cross-platform)

Apache HTTP Server

MySQL

PHP

Perl

The program is released under the terms of the GNU General Public License and acts

as a free web server capable of serving dynamic pages. XAMPP is available for

Microsoft Windows, Linux, Solaris, and Mac OS X, and is mainly used for web

development projects.

**5.6.1 Requirements and features**

XAMPP requires only one zip, tar or exe file to be downloaded and run, and

little or no configuration of the various components that make up the web server is

required. XAMPP is regularly updated to incorporate the latest releases of

Apache/MySQL/PHP and Perl. It also comes with a number of other modules

including OpenSSL and phpMyAdmin.Because its user interface is considered simple

to use, it is sometimes called the "lazy man's WAMP/LAMP installation".Installing

XAMPP takes less time than installing each of its components separately. Selfcontained,

multiple instances of XAMPP can exist on a single computer, and any

given instance can be copied from one computer to another.

It is offered in both a full, standard version and a smaller version (known as

XAMPP Lite). Add-on extension modules also are available for free.

Us Officially, XAMPP's designers intended it for use only as a development tool,

to allow website designers and programmers to test their work on their own

computers without any access to the Internet. To make this as easy as possible, many

important security features are disabled by default. In practice, however, XAMPP is

sometimes used to actually serve web pages on the World Wide Web. A special tool

is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in

MySQL and SQLite among others

**CHAPTER-7**

**CONCLUSION**

As there was no such system available for department by which the students can

communicate with teachers properly. This project will provide the students proper medium to

communicate with the faculties over there problems regarding studies and other topics.

There was a great problem in keeping record of daily attendance, performance of each

student, profile of each person associated with the department.

There were a lot of difficulties for the student regarding the current updates of the

department. All the latest updates of the departments are provided on the website, such as all

the information is uploaded on the website as quick as possible

**CHAPTER-8**

**BIBILIOGRAPHY**

**References:**

References for the project development were taken from the following books and websites:

SQL Handbook by Judith S. Bowman 4th Edition

SQL Fundamentals by John J. Pattrick 3rd Edition

MySQL by Baron Schwartz 3rdEditionsss

Software Engineering by Roger Pressman.

HTML Black Book by Holzner

PHP: The Complete Reference by Steve Holzner

**Websites:**

http://www.w3schools.com/html/default.asp

http://www.w3schools.com/css/default.asp

http://www.w3schools.com/php/

http://www.joomla.org/

http://www.buildajoomlawebsite.com/

http://dev.mysql.com/doc/refman/5.0/en/tutorial.html