

Abstract

Any institute has a complex and voluminous data for each student. To access these data of the students at any time is a huge task and this is a great disadvantage to any academic institute. Student portal is an abstract way to show the student's details at one platform. The portal allows students to explore through their hectic and voluminous data like semester results, Back paper results, Account details, Library details, Attendance of their respective subjects, Notices from different departments of the institutions. This project can be also be used to access from anywhere through internet i.e. registered users can be able to login to their accounts by directly accessing the website and then signing in with their username and password anytime and anywhere.

INTRODUCTION

The purpose of the project is to develop an online website which covers all details of college like student attendance details ,event details, News/circular ,Account details, library information, Research , Innovation &incubation, Marks details, Photo gallery, Academics, Admissions, Student life etc.

This is a systematic collection of extremely complex and voluminous data. It is an abstract way to show the student's details at one platform.

This system can be used by the students, admin and faculties.

It is very simple to use for the novice users also. Admin and faculties can frequently add and manage the portal whenever they can.

The student portal will have a large number of users, a good user interface, systematically designed, carefully implemented thoroughly tested. It will help the students as well as the faculties to communicate in a better way. Students can view their marks, library related information.

This project can be also be used to access from anywhere through internet i.e. registered users can be able to login to their accounts by directly accessing the website and then signing in with their username and password anytime and anywhere.

Advantages over existing system:

- It eliminates human effort in searching.
- It increases efficiency and speed.
- It significantly increases users' satisfaction.
- It provides easy way to access the extremely complex & voluminous student's data at one platform.

Features:

The portal allows students to explore through their hectic and voluminous data like semester results, Backpaper results, Account details, Library details, Attendance of their respective subjects , Notices from different departments of the institutions.

Admin and faculties can add and manage the student's information like their attendance, notice, account details, attendance details, library details, and their semester results including backpaper results.

This portal is able to provide a basic and easy interchange of information i.e. it is an efficient way to remove the communication gaps between the student and their large academic data.

Student portal works even at when the network traffic is high. Server has a power backup as well as a database backup. It is compatible with most of the operating systems i.e. previous and latest ones

Their work has been made more efficient and simple since it has a good interactive interface, reliable database and is thoroughly tested with almost every possible test cases.

FEASIBILITY STUDY

Technical Feasibility:

- Proposed project is beneficial only if it can turn into an online system that will meet the requirement.
- MySQL Server provides high efficiency in field of data updating and retrieval and allows access to huge amount of database.
- HTML is platform independent and operates at the client side. So user can access this service from any system.
- PHP allows us to perform operations on MySQL server database which would otherwise not be possible with only HTML.
- Validation control enables input validation. It prevents giving unnecessary load to the server.
- The proposed information system can be developed using MySQL server, PHP, Adobe Dreamweaver, Mozilla,chrome,edge, HTML, AJAX, JavaScript, etc.

Operational Feasibility

Student:

- An account with valid Registration number and password is created by the student.

- Student can access his or her respective account by supplying a Registration number and password.
- After login the student can access the option like providing feedback ,look through pages like research, about the institution, can change their password, view their profile which shows their personal details, semester results, backpaper results, attendance of their respective subjects, library details showing respective book no, and author name issued, account details showing their dues and notices from different departments of the institution.

Admin:

- With the valid username and password admin can enter the system.
- Admin enters &manages all the related student's information in the database system.
- Admin can delete data from database.
- Admin has a privilege to update database table.
- Admin also retrieves the complain and suggestion put by the user.

Faculty:

- An account with valid Faculty id and password is created by the Faculty.
- Faculty can access his or her respective account by supplying a Faculty id and password.
- Faculty has a privilege to update database table.
- Faculty enters all the related student's information in the database system.

HARDWARE AND SOFTWARE REQUIREMENT

SOFTWARE SELECTION:

When the analysis process got over the next step was to decide which should be the correct platform to develop the project and for that what are the best-suited software's and hardware.

Let us have a look on the software requirements.

To develop this new application the user requirement is simple. There must be screens through which the user can enter and display data. There should be one database in which the data can be store. There should be software, which will create report as per the requirement.

Software Requirements:-

Operating system	: WINDOWS 7 and above
Database	: MySQL
Server	: Apache
Web Browser	: All browsers
Web Page Development	: Adobe Dream Weaver CS6
Web Page Style sheet	: Html, CSS3, Bootstrap
Scripting Language	: Java Script, PHP, JQuery, Ajax

HARDWARE SELECTION:

The decision of the software's to be use doesn't solve the problem of the hardware selection, which is compactable to the software's.

Before Going for a correct hardware selection the following things must be taken into consideration.

1. The availability of the hardware.

2. The Compatibility with the software.
 3. Services provided for the hardware.
 4. Cost of the hardware.
 5. The period of installation of the hardware's with the software.
-
- Intel Pentium IV Processor or higher.
 - Minimum 1 GB Main Memory.
 - CPU speed : 2.6GHz or higher
 - Monitor: EGA / SVGA (display), 800X600 24 bits True Colour.
 - Standard Keyboard: 106 Keys with Separate Function Keys & Numeric Pad.
 - Mouse: PS /2 Optical mouse.
 - CD-ROM: Required

TECHNICAL DESCRIPTION

What Is PHP?

PHP is a server-side scripting language designed specifically for the Web. Within an HTML page, you can embed PHP code that will be executed each time the page is visited. Your PHP code is interpreted at the Web server and generates HTML or other output that the visitor will see.

PHP was conceived in 1994 and was originally the work of one man, Rasmus Lerdorf. It was adopted by other talented people and has gone through three major rewrites to bring us the broad, mature product we see today. As of January 2001, it was in use on nearly five million domains worldwide, and this number is growing rapidly. You can see the current number at <http://www.php.net/usage.php>

PHP is an Open Source product. You have access to the source code. You can use it, alter it, and redistribute it all without charge.

PHP originally stood for *Personal Home Page*, but was changed in line with the GNU recursive naming convention (GNU = Gnu's Not Unix) and now stands for *PHP Hypertext Preprocessor*. The current major version of PHP is 4. This version has seen some major improvements to the language, The home page for PHP is available at <http://www.php.net>

What Is MySQL?

MySQL (pronounced *My-Ess-Que-Ell*) is a very fast, robust, *relational database management system(RDBMS)*. A database enables you to efficiently store, search, sort, and retrieve data. The MySQL server controls access to your data to ensure that multiple users can work with it concurrently, to provide fast access to it, and ensure that only authorized users can obtain access. Hence, MySQL is a multi-user, multi-threaded server. It uses *SQL (Structured Query Language)*, the standard database query language worldwide. MySQL has been publicly available since

1996, but has a development history going back to 1979. It has now won the Linux Journal Readers' Choice Award three years running.

MySQL is now available under an Open Source license, but commercial licenses are also available if required.

Why Use PHP and MySQL?

When setting out to build an e-commerce site, there are many different products that you could use. You will need to choose hardware for the Web server, an operating system, Web server software, a database management system, and a programming or scripting language.

Some of these choices will be dependent on the others. For example, not all operating systems will run on all hardware, not all scripting languages can connect to all databases, and so on.

In this book, we do not pay much attention to your hardware, operating system, or Web server software. We don't need to. One of the nice features of PHP is that it is available for Microsoft Windows, for many versions of UNIX, and with any fully-functional Web server. MySQL is similarly versatile.

To demonstrate this, the examples in this book have been written and tested on two popular setups:

- Linux using the Apache Web server
- Microsoft Windows 2000 using Microsoft Internet Information Server (IIS) Whatever hardware, operating system, and Web server you choose, we believe you should seriously consider using PHP and MySQL.

Some of PHP's Strengths

Some of PHP's main competitors are Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP), and Allier Cold Fusion.

In comparison to these products, PHP has much strength including the following:

- High performance
- Interfaces to many different database systems
- Built-in libraries for many common Web tasks
- Low cost
- Ease of learning and use
- Portability
- Availability of source code

A more detailed discussion of these strengths follows.

Performance:

PHP is very efficient. Using a single inexpensive server, you can serve millions of hits per day. Benchmarks published by Zend Technologies (<http://www zend.com>) show PHP outperforming its competition.

Database Integration:

PHP has native connections available to many database systems. In addition to MySQL, you can directly connect to Postgre SQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, InterBase, and Sybase databases, among others.

Using the *Open Database Connectivity Standard (ODBC)*, you can connect to any database that provides an ODBC driver. This includes Microsoft products, and many others.

Built-in Libraries:

Because PHP was designed for use on the Web, it has many built-in functions for performing many useful Web-related tasks. You can generate GIF images on-the-fly, connect to other network services, send email, work with cookies, and generate PDF documents, all with just a few lines of code.

Learning PHP:

The syntax of PHP is based on other programming languages, primarily C and Perl. If you already know C or Perl, or a C-like language such as C++ or Java, you will be productive using PHP almost immediately.

Portability:

PHP is available for many different operating systems. You can write PHP code on the free Unix like operating systems such as Linux and FreeBSD, commercial Unix versions such as Solaris and IRIX, or on different versions of Microsoft Windows.

Your code will usually work without modification on a different system running PHP.

Source Code:

You have access to the source code of PHP. Unlike commercial, closed-source products, if there is something you want modified or added to the language, you are free to do this. You do not need to wait for the manufacturer to release patches. You don't need to worry about the

manufacturer going out of business or deciding to stop supporting a product.

Some of MySQL's Strengths

Some of MySQL's main competitors are PostgreSQL, Microsoft SQL Server, and Oracle.

MySQL has many strengths, including high performance, low cost, easy to configure and learn, portable, and the source code is available.

A more detailed discussion of these strengths follows.

Performance:

MySQL is undeniably fast. You can see the developers' benchmark page at <http://web.mysql.com/benchmark.html>. Many of these benchmarks show MySQL to be orders of magnitude faster than the competition.

Ease of Use:

Most modern databases use SQL. If you have used another RDBMS, you should have no trouble adapting to this one. MySQL is also easier to set up than many similar products.

Portability:

MySQL can be used on many different UNIX systems as well as under Microsoft Windows.

Source Code:

As with PHP, you can obtain and modify the source code for MySQL.

Applying Software Engineering to Web Development

As you probably already know, software engineering is the application of a systematic, quantifiable approach to software development. That is, it is the application of engineering principles to software development.

It is also an approach that is noticeably lacking in many Web projects. This is for two main

Reasons:

The first reason is that Web development is often similar to the development of written reports.

It is an exercise in document structure, graphic design, and production. This is a document oriented paradigm. This is all well and good for static sites of small to medium size, but as we increase the amount of dynamic content in Web sites to the level in which Web sites offer services rather than documents, this paradigm no longer fits. Many people do not think to use software engineering practices for a project at all. The second reason software engineering practices are not used is that Web application development is different from normal application development in many ways. We deal with much shorter lead times, a constant pressure to have the site built **now**. Software engineering is all about doing things in an orderly, planned manner, and spending time on planning. With Web projects, often the perception is that we don't have the time to plan.

When we fail to plan Web projects, we end up with the same problems as if we fail to plan any software project: buggy applications, missed deadlines, and unreadable code.

The trick, then, is in finding the parts of software engineering that work in this new discipline of Web application development, and discarding the parts that don't.

PHP and databases

PHP has become widely popular due to its capacity to use varied and powerful database systems. Website content is made dynamic, interactive and flexible with the help of a database.

Databases are collection of data, stored separately in such a manner at you can easily recover it. Database information is stored in table format. The tables are divided into rows and columns, separating the data uniformly. Each row in a table represents a single record like name and address. In each column of a table, a uniform record is maintained like the first name of an individual and his/her contact number. A database in such format arranges a record according to the values available in the column exactly like a spreadsheet program.

In this way, you can easily retrieve a record from a database without keeping in mind how the data has been arranged into a table.

Basically, most of the database systems use the SQL. There are several Database Management Systems available in the Market.

Mysql, Microsoft SQL Server, Oracle are among others. MySQL is the most popular Open Source Database Management System and for this 'Open Source' nature it is widely used among the web applications that use PHP.

PHP performs various inbuilt functions with certain databases such as MySQL database, SQL Server, Oracle and others. PHP supports databases in multiple ways. It supports various databases as:

- Informix

- DBM (Berkeley)
- MSSQL (Microsoft)
- Sybase
- Oracle 8
- PostgreSQL (Berkeley, open source)
- MySQL (Open source)

There are certain databases which are supported by PHP through protocol-based functions. These include:

- ODBC
- LDAP
- DBM style

Database concept

The term ‘data’ suggests a record with certain necessary information.

A database in computer is a structured record collection.

They are stored in computer systems. A database structure is prepared by arranging a data in a database model patterns. There are three types of databases that are frequently used:

- Relational model
- Hierarchical model
- Network model

A database is organized in a computer with a database management system. Using this software, a computer performs different Related functions. These include recovering data, storing, Adding, deleting and modifying data. An extensive part of a database depends on different managing factors like integrity, presentation, concurrency and recovery from hardware failures. We can divide a database management system into two categories - desktop databases and server databases. The desktop databases are generally targeted towards individual user application. The server databases target towards the authenticity and uniformity of a data.

A database is basically of two types - flat file and relational database.

Flat file

Flat file databases usually store small amounts of data. They are easily read and edited. Basically, they are arranged in a series and are accordingly analyzed. Flat files are best to store simple data types. They may become complicated if you store complex data structures.

Relational database

The relational model is the most commonly used database in the present scenario. MySQL, Microsoft SQLServer and Oracle are best examples of relational databases. In a relational database, tables are used to represent some interlinked objects. In the relational model, databases are arranged for maintaining integrity.

DATA FLOW DIAGRAM

Data flow diagram

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The DFD is also known as a data flow graph or a bubble chart.

Context diagram

The top-level diagram is often called a “context diagram”. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study.

Types of Data Flow Diagram

Data Flow Diagrams are of two types as follows:

1. Physical DFD
2. Logical DFD

1. *Physical DFD:*

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs shows actual devices, departments, and people etc., involved in the current system

2. Logical DFD

Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

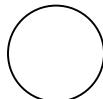
Basic Notation

The Basic Notation used to create a DFD's are as follows:

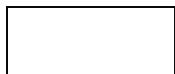
- ***Dataflow:*** Data move in a specific direction from an origin to a destination.



- ***Process:*** People, procedures, or devices that use or produce (transform) data.

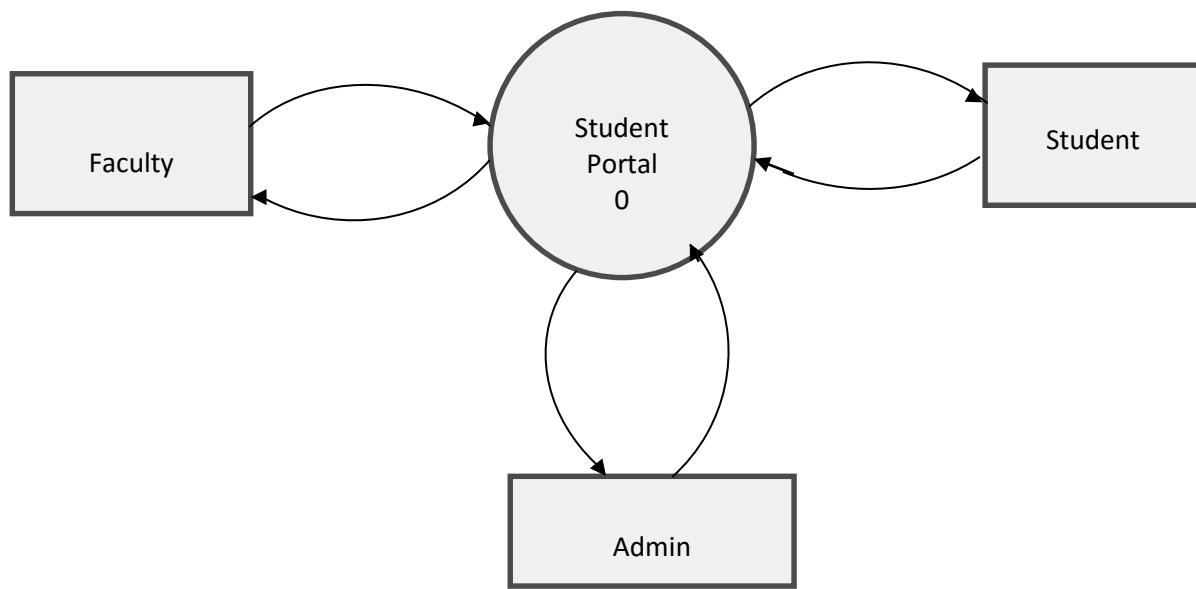


- ***Source:*** External sources or destination of data, which may be People, programs, organizations or other entities.

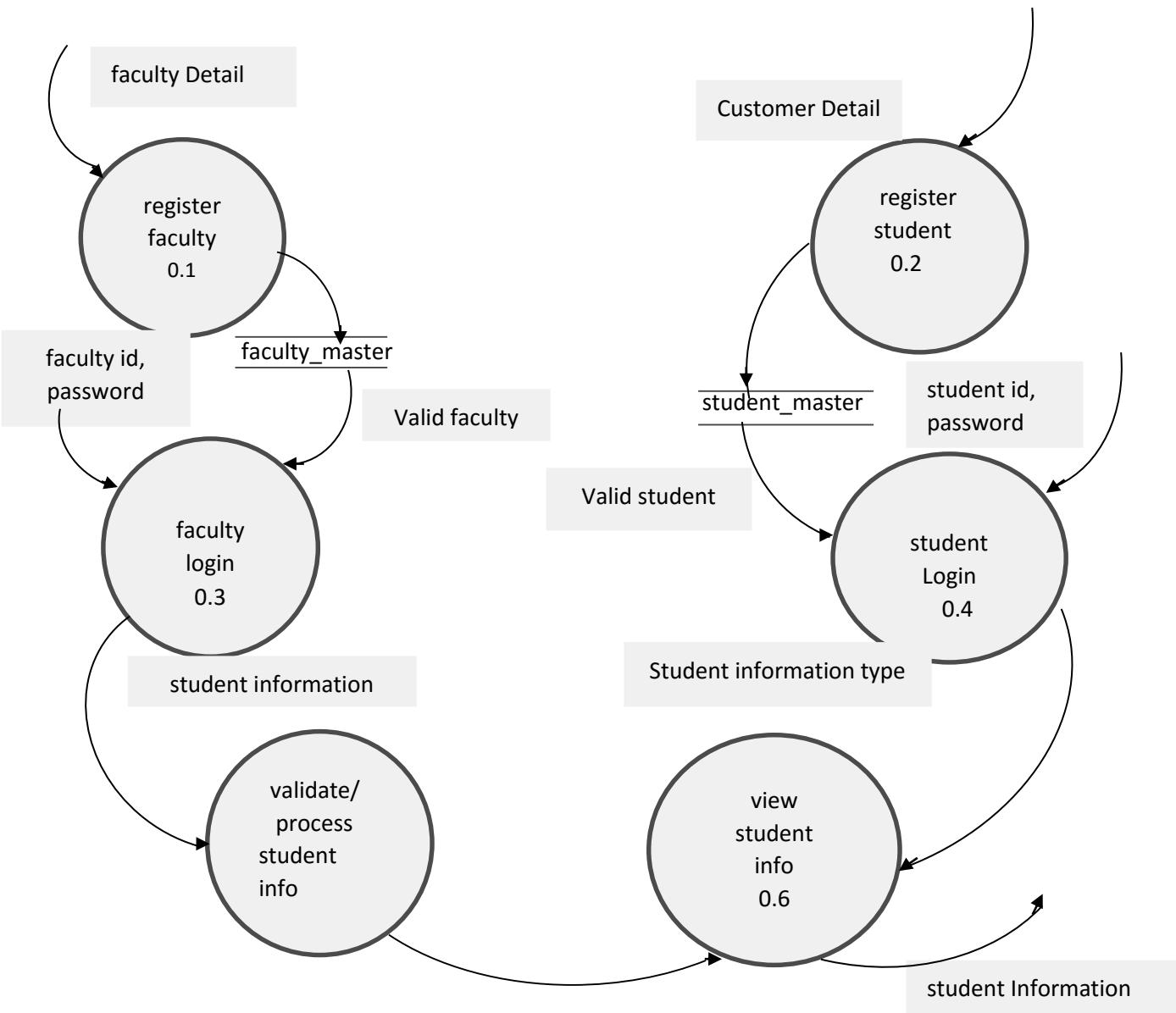


- ***Data Store:*** Here data are stored or referenced by a process in the System.

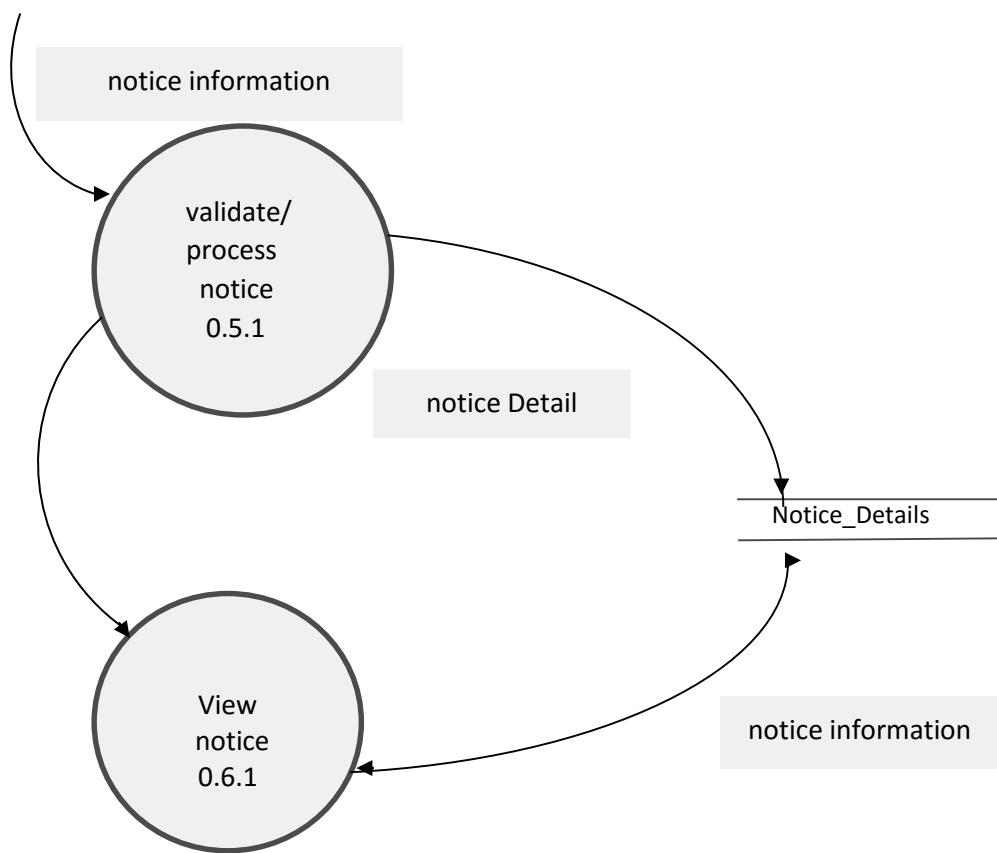
Context Diagram

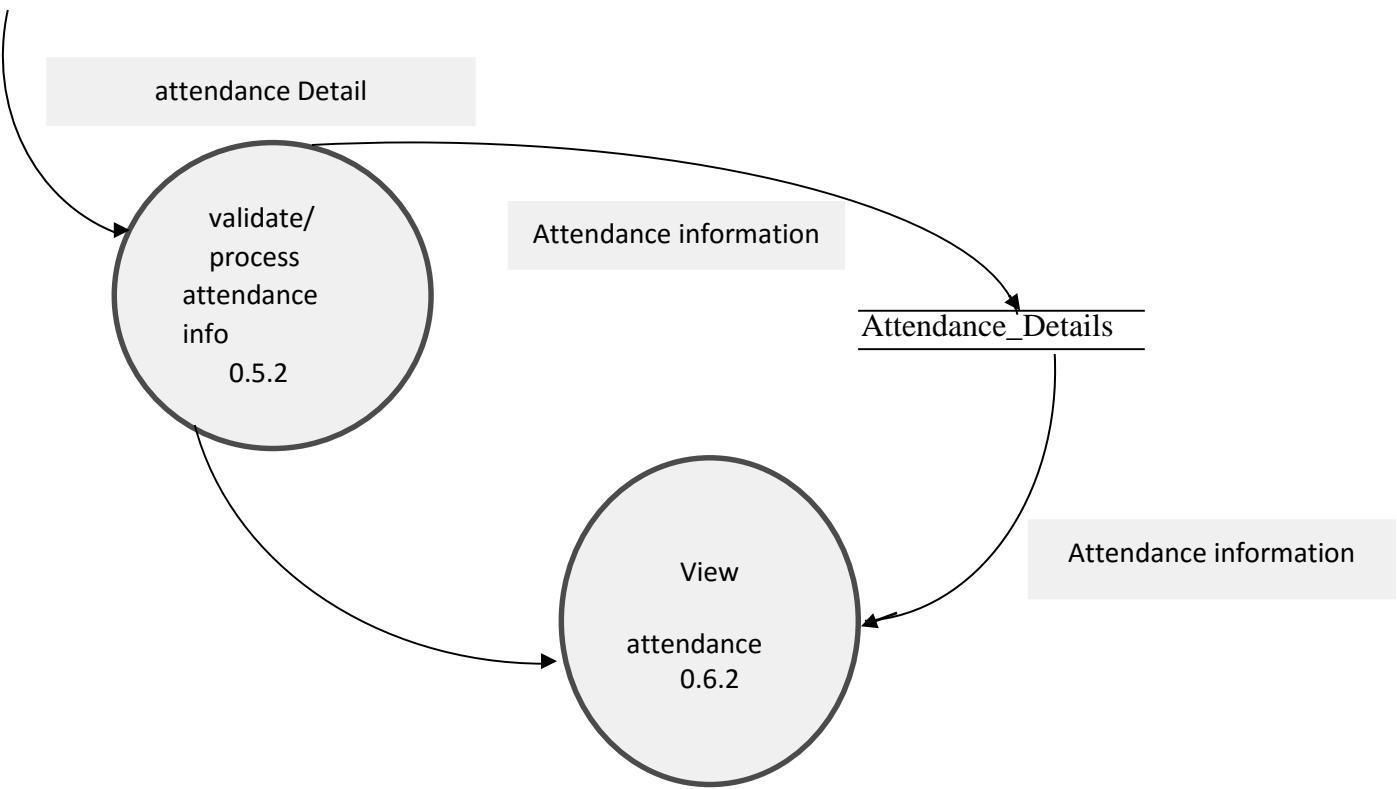


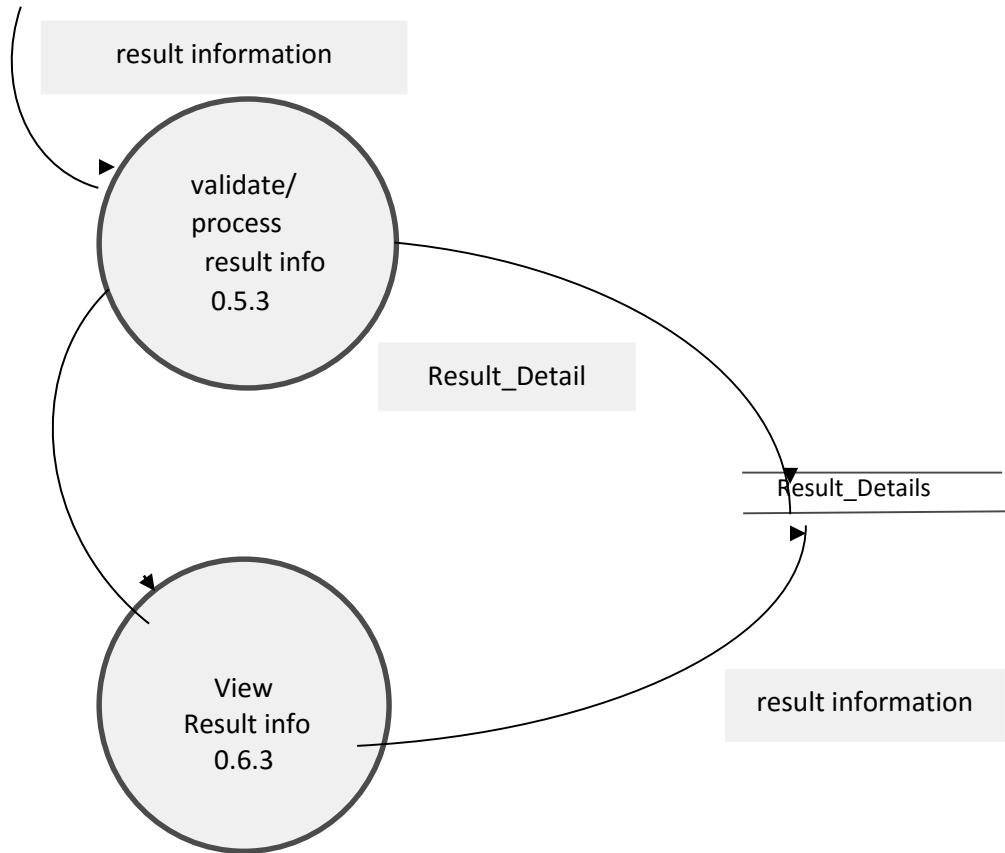
Level 1 DFD



Level 2 DFD







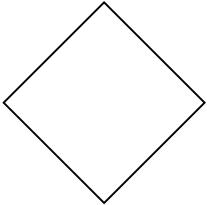
ENTITY RELATIONSHIP DIAGRAM

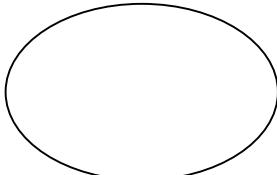
Entity Relationship Diagram (ER-Diagram):

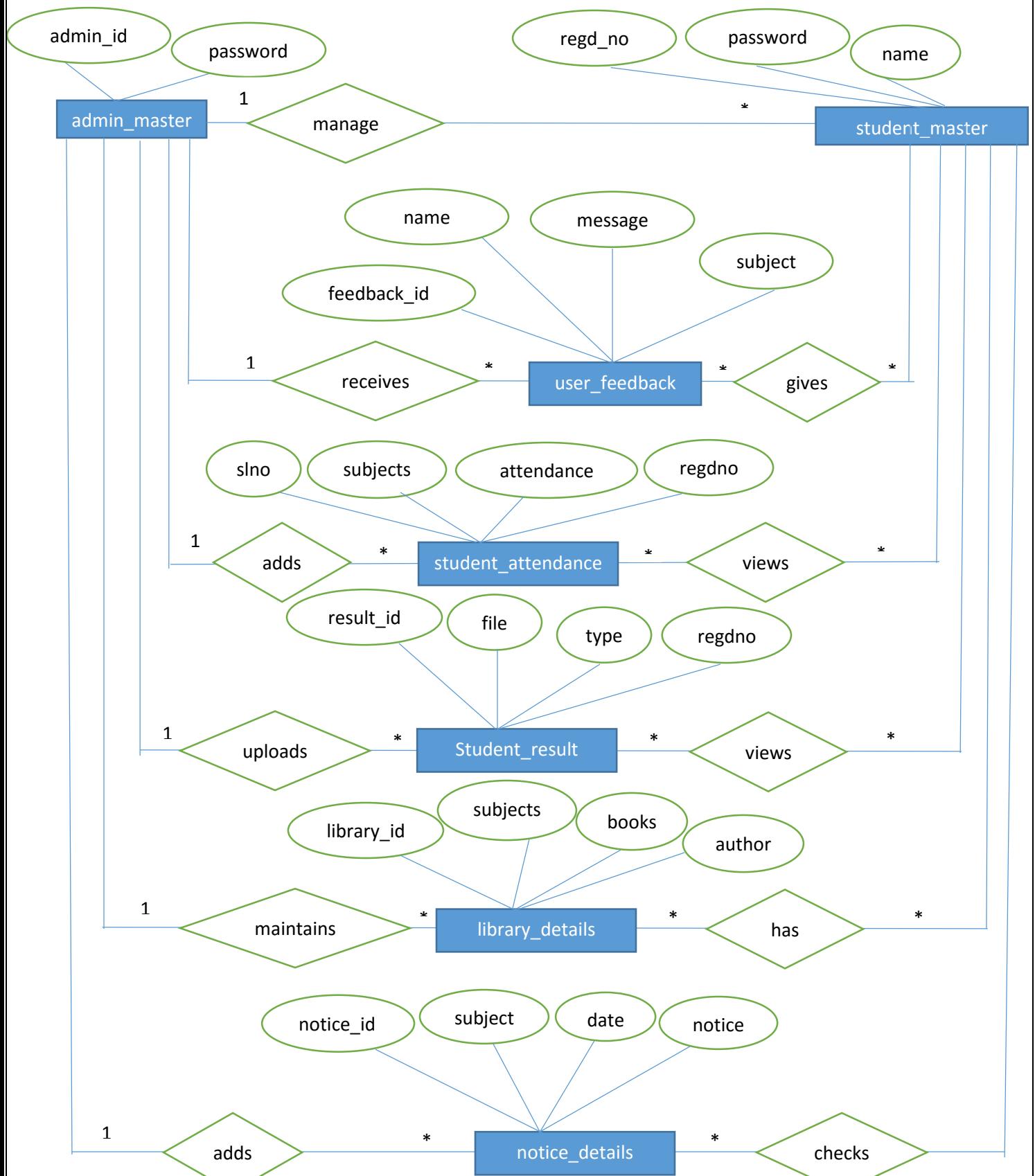
Entity relationship diagram is used in modern database software. Software engineering is to illustrate logical structure of database. It is a relational schema database, modelling method, used to model a system and approach .this approach is commonly used in database design. The diagrams created using this method are called Entity-Relationship Diagram.

Entity-Relationship Diagram depicts the various relationships among entities, considering each objective as an entity. Entity is represented as rectangle and relationship between data objects .the ER diagram is a notation that is used to conduct the data modelling activity.

Basic Notation:

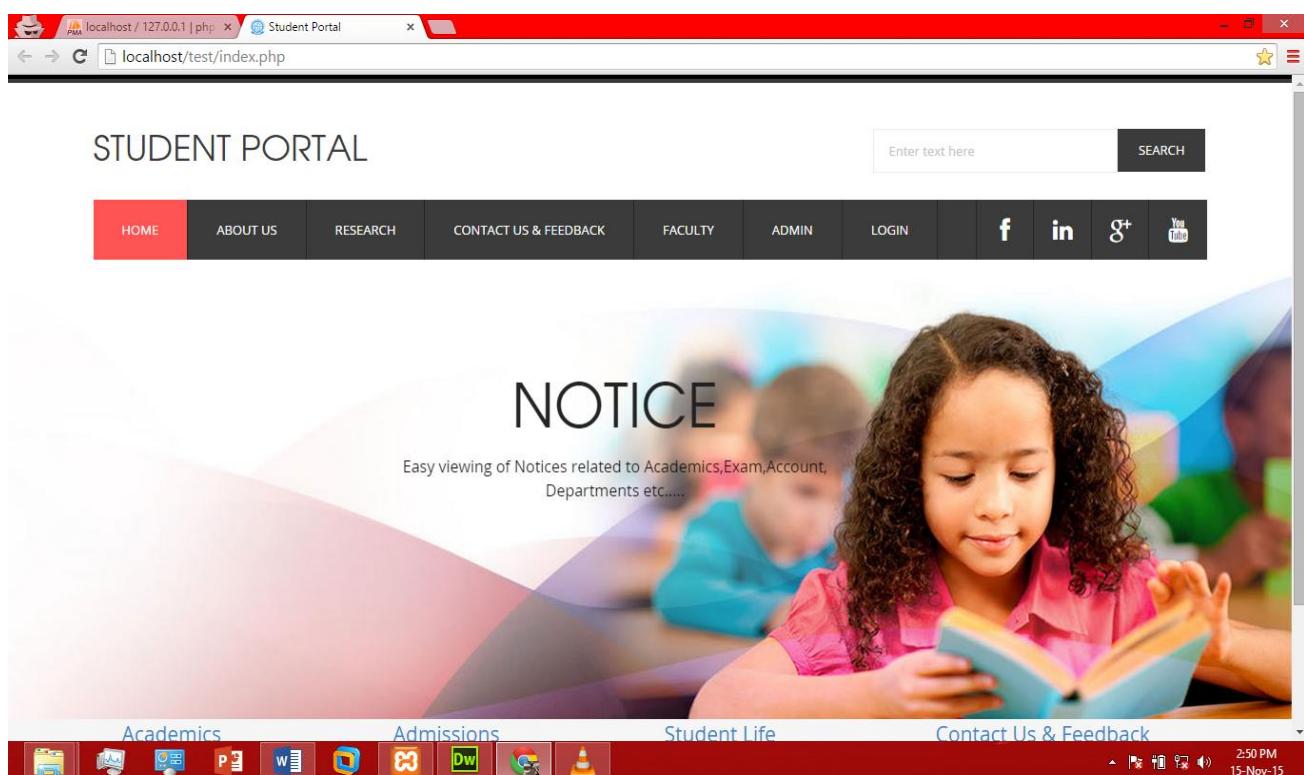
Name	Notation	Description
Entity		An entity is a single object about which data can be stored it is subjected of a table. Entity and their relationships are modelled through the use of the entity relationships diagram.
Relationship		A relationship is named connection or association between entities or used to relate two or more entities with common attributes or meaningful interaction between objects.

Attributes		Attributes are the properties of the entities and the relationships are the descriptions of the entity. Attributes are elementary pieces of information attached to an entity.
-------------------	---	--



Screen Shots

Index page:



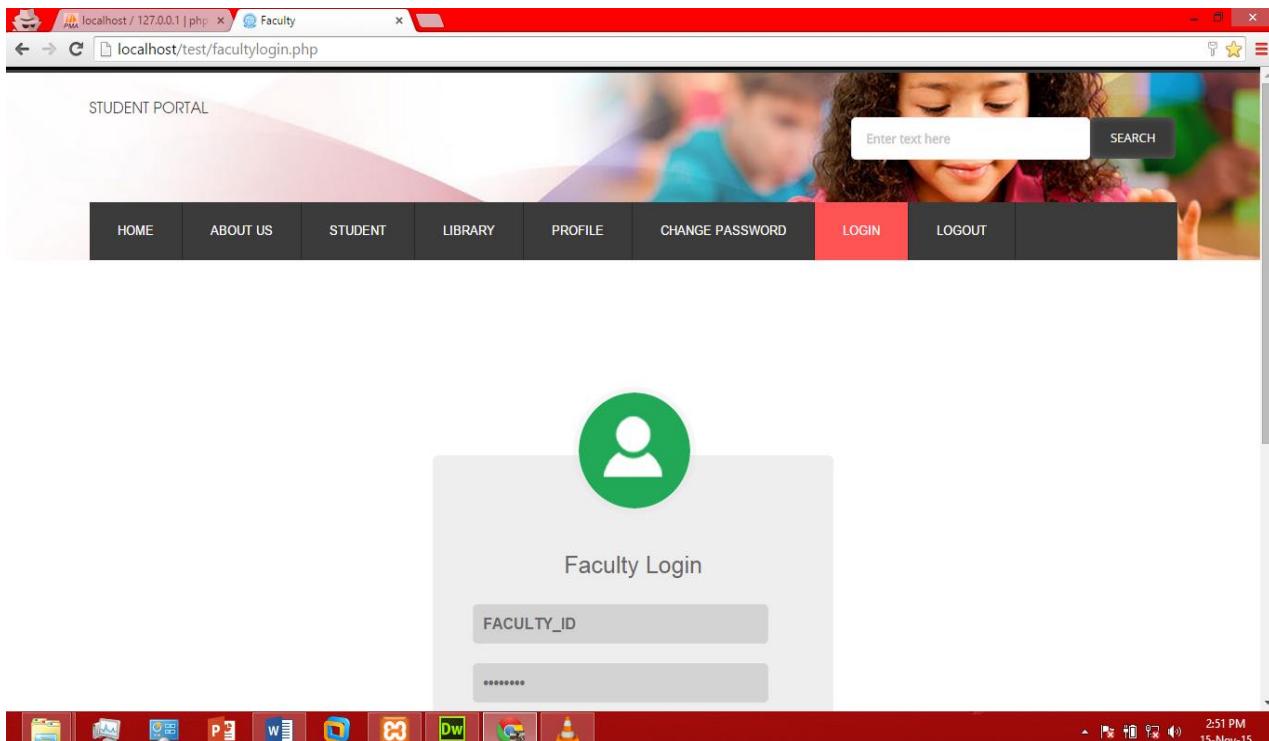
Student login page:

The screenshot shows a web browser window titled "Student Portal" with the URL "localhost/test/studentlogin.php". The page has a header with "STUDENT PORTAL" and a background image of a young girl. A navigation bar includes links for HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, ADMIN, and LOGIN. Social media icons for Facebook, YouTube, Google+, and LinkedIn are also present. The main content area contains fields for "REGD. NO." and "Password", with "Login" and "SignUp" buttons. Below these are links for "Forgot Password?" and "Contact Us & Feedback". The footer features icons for various software applications like Microsoft Office and Adobe Photoshop, along with copyright and system information.

Admin login page:

The screenshot shows a web browser window with the URL `localhost/test/adminlogin.php`. The page has a header with the text "STUDENT PORTAL" and a search bar. Below the header is a navigation menu with links: HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, FACULTY, ADMIN (which is highlighted in red), and LOGOUT. To the right of the menu are social media icons for Facebook, LinkedIn, Google+, and YouTube. The main content area contains a form for "Admin Login" with fields for "Admin ID." and a password, and a "Login" button. At the bottom of the page is a footer with links to Academics, Admissions, Student Life, and Contact Us & Feedback. The footer also includes a copyright notice, a row of small icons representing various software or services, and a system status bar showing the time as 2:50 PM and the date as 15-Nov-15.

Faculty Login page:



Student Profile Page:

The screenshot shows a student profile page titled "STUDENT PORTAL". The top navigation bar includes links for STUDENT, CONTACTUS & FEEDBACK, RESEARCH, PROFILE, CHANGE PASSWORD, and LOGOUT, along with social media icons for Facebook, LinkedIn, Google+, and YouTube. A search bar is also present. The main content area displays a profile for "Saurabh Bharti" with fields for Name (Saurabh Bharti), Registration No. (1201214131), and a "SHOW" button. Below this are sections for "Results" (First Semester) and "Type" (Regular). The "Notices" section includes an "Events" link. The bottom of the page features a toolbar with various application icons and a status bar showing the date and time.

This screenshot shows a more detailed view of the student profile page. On the left, there is a sidebar with fields for Name (Saurabh Bharti), Registration No. (1201214131), Email (saurabhbharti9@gmail.com), Mobile No. (9040899268), Security Question (1), Answer (ppse), Branch (CSE), Semester (8th), Year (2012), and D.O.B. To the right, there is a table showing subjects and their details:

Subjects	Total Class	Classes(P)	Percentage
SC	4	1	25
II	4	2	50
ES	4	2	50
ECS	4	2	50
PDS	4	2	50
XXXX	4	0	0

Below this table is a "Library Details" section with fields for ED, CG, PPSE, PSC, and IDSP.

Faculty Profile Page:

The screenshot shows a faculty profile page on a local host server. The top navigation bar includes links for STUDENT, CONTACTUS & FEEDBACK, RESEARCH, PROFILE, CHANGE PASSWORD, and LOGOUT. It also features social media icons for Facebook, LinkedIn, Google+, and YouTube. The main content area starts with a "Profile" section featuring a photo of a female student and a file upload form. Below this is a "Details..." section with fields for Name (purnima khuntia) and Faculty_Id. A horizontal toolbar at the bottom contains various application icons.

Student localhost / 127.0.0.1 / stu... Saurabh

localhost/test/facultyprofile.php

STUDENT CONTACTUS & FEEDBACK RESEARCH PROFILE CHANGE PASSWORD LOGOUT f in g+ YouTube

Profile

Add Student's information...

Choose File No file chosen

UPLOAD

Aug 18, 2015 Comments Admin 124 Views

Details...

Name purnima khuntia

Faculty_Id.

1:58 PM 22-Nov-15

Admin Home Page:

The screenshot shows a web browser window titled "Admin" with the URL "localhost/test/adminhome.php". The page has a header with the text "STUDENT PORTAL" and a background image of a young girl. A navigation menu at the top includes links for HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, FACULTY, ADMIN (which is highlighted in red), and LOGOUT. Social media icons for Facebook, LinkedIn, Google+, and YouTube are also present. Below the menu, there are two main sections: "Manage..." and "Add...". The "Manage..." section has a dropdown menu set to "Student" with a "Manage" button. The "Add..." section has a dropdown menu set to "Notice" with an "Add" button. At the bottom of the page is a dark red footer bar containing links for Academics, Admissions, Student Life, Contact Us & Feedback, and a series of small icons representing various services. The footer also displays the time "1:57 PM" and the date "22-Nov-15".

User Feedback Page:

CONTACT US AT..

Post Box No. 21, Techno Park, Jatni-752050
Bhubaneswar, Odisha
Phone no: +91-(0674)-2490962, 2490965, 2490964
Fax no: +91-(0674)-2490962
E-mail us at - info@kist.ac.in

CORPORATE OFFICE..

Plot No -76, Satya Nagar
Near Kali mandir, Bhubaneswar
Tel. No.: +91-(0674)- 2575872,
Contact Mobile No : +91-9090959524, 9090959525,
9437940440

Give Your Feedback

Name

E-Mail Id

Contact No

Subject

Message

SEND US

Academics Admissions Student Life Contact Us & Feedback

8:16 PM 22-Nov-15

Add Attendance Page:

STUDENT PORTAL

Attendance Entry...

Batch: 2015

Semester: 1st

Branch: Computer Science & Enggineering

Date: Sat Apr 30, 2016

Show

Sl.No	Regd No	Name	SC	II	ES	ECS	PDS	XXXXX
1	1201214061	Prakash	<input type="checkbox"/>					
2	1201214062	Sidhartha Kumar Singh	<input type="checkbox"/>					
3	1201214131	Saurabh Bharti	<input type="checkbox"/>					

update Select All

Add Notice Page:

The screenshot shows a web browser window titled "localhost / 127.0.0.1 / stu". The page header features a "STUDENT PORTAL" logo with a background image of a young girl. A navigation bar at the top includes links for HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, FACULTY, ADMIN (which is highlighted in red), LOGOUT, and social media icons for Facebook, LinkedIn, Google+, and YouTube. Below the navigation bar, the main content area is titled "Add Notice". It contains three input fields: "Subject" (set to "Events"), "Title" (empty), and "Notice" (empty). At the bottom of the form is a red-bordered "ADD" button. The browser's status bar at the bottom right shows the time as 1:58 PM and the date as 22-Nov-15.

This screenshot is identical to the one above, showing the "Add Notice" page. The only difference is that the "ADD" button has been highlighted with a red border, indicating it is the active or selected element.

Add Library Details:

The screenshot shows a web browser window titled "localhost / 127.0.0.1 / stuk" with the URL "localhost/test/addlibrarydetails.php". The page has a header "STUDENT PORTAL" and a navigation bar with links: HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, FACULTY, ADMIN (highlighted in red), LOGOUT, and social media icons for Facebook, LinkedIn, Google+, and YouTube. Below the header is a banner image of a young girl. The main content area is titled "Add Library Details". It contains three dropdown menus: "Year" (2015), "Branch" (Computer Science & Enggineering), and "Semester" (1st). A "Show" button is located below the semester dropdown. The bottom of the screen shows a taskbar with various application icons and a system tray indicating the date and time as 22-Nov-15 at 2:02 PM.

The screenshot shows the same web browser window as the previous one, but the "Show" button is now highlighted with a red box. Below it, there is a search field labeled "Regd No:" with a placeholder "Subjects...". Below the search field is a table with columns "Subjects", "Book", and "Author". The table rows contain subjects ED, CG, PPSE, PSC, IDSP, and RTS, each with empty input fields for Book and Author. At the bottom of the table is a "UPDATE" button.

Add Account Details:

The screenshot shows a web browser window titled "localhost / 127.0.0.1 / stuk" with the URL "localhost/test/addaccount.php". The page has a header "STUDENT PORTAL" with a background image of a young girl. Below the header is a navigation bar with links: HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, FACULTY, ADMIN (highlighted in red), LOGOUT, and social media icons for Facebook, LinkedIn, Google+, and YouTube. The main content area is titled "Add Account Details" and contains four dropdown menus: "Year" (2015), "Branch" (Computer Science & Enggineering), "Semester" (1st), and "Regd No:". At the bottom of the page is a toolbar with various application icons and a status bar showing "1:59 PM 22-Nov-15".

This screenshot shows the same "localhost/test/addaccount.php" page as above, but with different input values. The "Semester" dropdown now shows "2nd" instead of "1st". The "Regd No:" field is empty. Below these fields is a section titled "Account Details" with an empty text area. At the bottom left is a red "UPDATE" button. The page footer includes navigation links for Academics, Admissions, Student Life, and Contact Us & Feedback. The status bar at the bottom right shows "8:27 PM 22-Nov-15".

Add Results:

The screenshot shows a web browser window titled "localhost / 127.0.0.1 / stu...". The page header includes "STUDENT PORTAL" and a navigation menu with links: HOME, ABOUT US, CONTACT US & FEEDBACK, RESEARCH, FACULTY, ADMIN (highlighted in red), LOGOUT, and social media icons for Facebook, LinkedIn, Google+, and YouTube. Below the header is a large image of a young girl looking down at a device. The main content area is titled "Add Result" and contains fields for "Regd No:" (with a placeholder "1234567890"), "Branch" (set to "Computer Science & Enggineering"), "Semester" (set to "1st"), and "Year" (set to "6th"). A toolbar at the bottom features icons for various applications like Microsoft Office and Adobe Photoshop. The status bar at the bottom right shows the time as 1:58 PM and the date as 22-Nov-15.

This screenshot shows the same "Add Result" page as above, but with additional fields for "Result File" (a file input field containing "1201214131.html") and "Type" (a dropdown menu set to "Regular"). Below these fields is a red-bordered "UPLOAD" button. The rest of the page structure, including the header, footer, and system tray, is identical to the first screenshot.

Data Table:

Student_master

localhost / 127.0.0.1 / stu... > localhost/phpmyadmin/index.php?db=studentportal&token=046c9df88735e53f9a323b1e7bbfd4be

phpMyAdmin

127.0.0.1 > studentportal > student_master

Browse Structure SQL Search Insert Export Import Operations Tracking Triggers

LIMIT 0, 30

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

+ Options

	fname	email	regdno	password	sque	ans	contactno	status	dob	permanent	temporary	photo	branch	semest
<input type="checkbox"/>	prakash ranjan behra	asdf@gmail.com	1201214061	prakas	1	ppse	12345		1				image	
<input type="checkbox"/>	Sidhartha Kumar Singh	sidhartha.sing591@gmail.com	1201214062	did.123	4	824210	8984075709	1	03/12/1995	gaya	jaydevvihar		image	
<input type="checkbox"/>	madhu kumari	madhusomuhk1084@gmail.c	1201214106	madhu	2	saurabh	7205552558						image	
<input type="checkbox"/>	hrishaw kumar	hrk7aa@gmail.com	1201214112	hrk112	4	800023	8763452134	1	15/12/93	patna	bbsr		image	
<input type="checkbox"/>	Nishant	nishant5543211@gmail.com	1201214121	9031737278	4	831009	8093253585	1	20/08/1994	lakheesarai	jamsedpur		image	
<input type="checkbox"/>	Om Prakash Jha	omprakashjh32@gmail.com	1201214127	opj	1	math	8984605575						image	
<input type="checkbox"/>	Saurabh Bharti	saurabhbharti9@gmail.com	1201214131	saurabh	4	824206	+919040899268	1	26/04/1995	Gaya.bihar	jaydevvihar odisha		image	

Check All / Uncheck All With selected: Change Delete Export

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

2:40 PM
15-Nov-15

Student Portal > localhost / 127.0.0.1 / stu... > localhost/phpmyadmin/index.php?db=studentportal&token=3401b15d2847bd312db8c114cc736461

phpMyAdmin

127.0.0.1 > studentportal > student_master

Browse Structure SQL Search Insert Export Import Operations Tracking Triggers

	name	email	regdno	password	sque	ans	contactno	status	dob	permanent	temporary	photo	branch	semest
<input type="checkbox"/>	Prakash Ranjan Behera	asdf@gmail.com	1201214061	prakash	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	Sidhartha Kumar Singh	asdf@gmail.com	1201214062	sidhartha	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	pritam priya	asdf@gmail.com	1201214063	pritam	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	jyoti kumari singh	asdf@gmail.com	1201214064	jyoti	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	gulshan kumar	asdf@gmail.com	1201214065	gulshan	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	kumar piyush	asdf@gmail.com	1201214066	kumar	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	madhusmita biswal	asdf@gmail.com	1201214067	madhusmita	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	riya maity	asdf@gmail.com	1201214068	riya	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	raj kumar murmu	asdf@gmail.com	1201214069	raj	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	vivek kumar	asdf@gmail.com	1201214070	vivek	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	vishal kumar	asdf@gmail.com	1201214071	vishal	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	bishwajeet kumar singh	asdf@gmail.com	1201214072	bishwajeet	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	suman bharti	asdf@gmail.com	1201214073	suman	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	mani sankar sangrahi	asdf@gmail.com	1201214074	mani	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	ruhi kumar	asdf@gmail.com	1201214075	ruhi	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	sukriti mondal	asdf@gmail.com	1201214076	sukriti	1	ppse	0123456789	1				images/blog_pic1.jpg		
<input type="checkbox"/>	manisha	asdf@gmail.com	1201214078	manisha	1	ppse	0123456789	1				images/blog_pic1.jpg		

8:34 PM
22-Nov-15

faculty_master

The screenshot shows the phpMyAdmin interface for a database named 'studentportal'. The current table is 'faculty_master'. A successful update query has been run:

```
UPDATE `studentportal`.`faculty_master` SET `dob` = 'xx/xx/yyyy' WHERE `faculty_master`.`faculty_id` = 'pk1';
```

The table data is displayed in a grid:

	name	faculty_id	password	sque	answer	email_id	contactno	status	photo	department	dob	pad
<input type="checkbox"/>	purnima khuntia	pk1	pk	3	me	purnimakhuntia@gmail.com	0123456789	1	pk1.png	cse	xx/xx/yyyy	bbs

Below the table, there are buttons for 'Edit', 'Copy', 'Delete', and 'Export'. The status bar at the bottom right shows the time as 8:37 PM and the date as 22-Nov-15.

admin_master

The screenshot shows the phpMyAdmin interface for the 'admin_master' table. The table structure is as follows:

adminid	password
hrk	hrk
nishu	nishu
om	om
sb	sb

Below the table, there are buttons for 'Edit', 'Copy', 'Delete', 'Change', and 'Export'. The 'password' column contains the same value ('hrk', 'nishu', 'om', 'sb') for all rows.

attendance_details

The screenshot shows the phpMyAdmin interface for the 'studentportal' database. The 'attendance_details' table is selected. The table has columns: attendance_id, year, branch, semester, subjects, attendance, and regdno. The data shows 19 rows of attendance records for CSE students in the 7th semester. The 'Attendance' column lists various subject codes separated by commas. The 'regdno' column contains unique student registration numbers.

	attendance_id	year	branch	semester	subjects	attendance	regdno
<input type="checkbox"/>	1	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	63,58,50,45,55,65	1201214061
<input type="checkbox"/>	2	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	71,54,56,62,64,61	1201214062
<input type="checkbox"/>	3	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	83,79,78,83,70,73	1201214063
<input type="checkbox"/>	4	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	69,54,68,64,70,76	1201214064
<input type="checkbox"/>	5	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	31,42,48,48,57,42	1201214065
<input type="checkbox"/>	6	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	31,33,38,48,48,55	1201214066
<input type="checkbox"/>	7	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	94,92,94,88,82,94	1201214067
<input type="checkbox"/>	8	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	69,58,68,69,64,64	1201214068
<input type="checkbox"/>	9	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	6,4,10,14,9,10	1201214069
<input type="checkbox"/>	10	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	57,50,54,69,50,55	1201214070
<input type="checkbox"/>	11	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	23,25,14,12,9,16	1201214071
<input type="checkbox"/>	12	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	20,21,14,12,25,18	1201214072
<input type="checkbox"/>	13	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	54,50,50,64,68,58	1201214073
<input type="checkbox"/>	14	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	17,17,22,24,43,18	1201214074
<input type="checkbox"/>	15	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	69,75,66,71,64,73	1201214075
<input type="checkbox"/>	16	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	69,54,60,69,57,67	1201214076
<input type="checkbox"/>	17	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	71,62,52,60,50,76	1201214078
<input type="checkbox"/>	18	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	77,62,78,79,84,76	1201214079
<input type="checkbox"/>	19	2012	CSE	7th	ED,CG,PPSE,PSC,IDS,RTS	46,58,44,67,66,55	1201214080

library_details

The screenshot shows the phpMyAdmin interface for the 'studentportal' database. The left sidebar lists tables, and the main area displays the 'library_details' table with 22 rows of data. The columns are: ID, Action, Year, Branch, Class, Books, and Notes.

Action	ID	Year	Branch	Class	Books	Notes
Edit Copy Delete	4	2012	CSE	6th	PM,MPMC,OS,CD,IWT,CE	***** , ***** , ***** , ***** , s.shar
Edit Copy Delete	6	2012	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	32000,31563,****,33090,27300
Edit Copy Delete	7	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	32033,33103,****,20083,27300
Edit Copy Delete	8	2012	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	****,****,29599,22574,25334,2
Edit Copy Delete	9	2012	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	****,Maurya,R.Mall,S.N Deepa,Prokis,R.Mall
Edit Copy Delete	10	2012	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	***** ,***** ,***** ,***** ,*
Edit Copy Delete	11	2012	CSE	2nd	Data Structure,Physics/Chemistry,Mechanics/Thermod...	****,13590,****,1844,**** ,
Edit Copy Delete	12	2012	CSE	5th	SS,OE,CO,JAVA,DCCN,TC	***** ,***** ,5605,16948,***** ,
Edit Copy Delete	13	2012	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	***** ,31586,19455,22575,1237
Edit Copy Delete	14	2012	CSE	6th	PM,MPMC,OS,CD,IWT,CE	28108,11750,10849,***** ,**** ,
Edit Copy Delete	15	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	***** ,33099,15897,***** ,**** ,33
Edit Copy Delete	16	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	32097,31569,22198,13134,252
Edit Copy Delete	17	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	***** ,***** ,***** ,***** ,*****
Edit Copy Delete	18	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	***** ,***** ,***** ,***** ,*****
Edit Copy Delete	19	2013	CSE	1st	programming in c,mathematics- 1,physics/chemistry,t...	****,*,17302,*, , ,
Edit Copy Delete	20	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	**,31575,**** ,**** ,**** ,33242
Edit Copy Delete	21	2013	CSE	7th	ED,CG,PPSE,PSC,IDSP,RTS	***** ,***** ,***** ,***** ,*****
Edit Copy Delete	22	2013	CSF	7th	FD CG PPSF PSC IDSP RTS	**** 16850 22587 31602 2730

account_details

The screenshot shows the phpMyAdmin interface for a database named 'studentportal'. The current table being viewed is 'account_details'. The table structure includes columns: account_id, regdno, semester, year, branch, and amount. There are 21 rows of data listed, each with edit, copy, and delete options. The data shows various student IDs, registration numbers, and amounts ranging from 34000.00 to 99000.00.

	account_id	regdno	semester	year	branch	amount
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1	1201214131	7th	2012	CSE	72750.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	1201214061	7th	2012	CSE	42000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	5	1201214066	7th	2012	CSE	57000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	6	1201214069	7th	2012	CSE	74000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	7	1201214071	7th	2012	CSE	74000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	8	1201214072	7th	2012	CSE	74000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	9	1201214073	7th	2012	CSE	62100.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	10	1201214074	7th	2012	CSE	34000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	11	1201214076	7th	2012	CSE	37000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	12	1201214078	7th	2012	CSE	24500.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	13	1201214079	7th	2012	CSE	89000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	14	1201214080	7th	2012	CSE	29000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	15	1201214081	7th	2012	CSE	47500.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	16	1201214082	7th	2012	CSE	74000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	17	1201214083	7th	2012	CSE	44000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	18	1201214084	7th	2012	CSE	65000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	19	1201214085	7th	2012	CSE	34000.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	20	1201214087	7th	2012	CSE	132500.00
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	21	1201214089	7th	2012	CSE	99000.00

notice_details

The screenshot shows the phpMyAdmin interface for a database named 'studentportal'. The current table is 'notice_details'. The SQL query executed is:

```
SELECT *  
FROM `notice_details`  
LIMIT 0 , 30
```

The results show one row with the following data:

notice_id	subject	title	notice	date
1	Department	Attendance	The students those who do't have minimum % of atte...	2015-11-14

Below the table, there are 'Query results operations' buttons: Print view, Print view (with full texts), Export, Display chart, Create view.

student_result

Student Portal localhost / 127.0.0.1 / stu... Surabh... x

localhost/phpmyadmin/index.php?db=studentportal&token=3401b15d2847bd312db8c114cc736461

phpMyAdmin 127.0.0.1 » studentportal » student_result

Browse Structure SQL Search Insert Export Import Operations Tracking Triggers

(Recent tables) ...

studentportal

account_details admin_master attendance_details branch_sem_year faculty_master library_details notice_details student_master student_result subject_details user_feedback

Create table

	resultid	regdno	branch	semester	year	file	type	
<input type="checkbox"/>	Edit Copy Delete	16	1201214061	CSE	1st	2012	1201214061_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	17	1201214062	CSE	1st	2012	1201214062_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	18	1201214063	CSE	1st	2012	1201214063_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	19	1201214064	CSE	1st	2012	1201214064_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	20	1201214065	CSE	1st	2012	1201214065_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	21	1201214066	CSE	1st	2012	1201214066_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	22	1201214067	CSE	1st	2012	1201214067_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	23	1201214068	CSE	1st	2012	1201214068_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	24	1201214069	CSE	1st	2012	1201214069_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	25	1201214070	CSE	1st	2012	1201214070_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	26	1201214071	CSE	1st	2012	1201214071_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	27	1201214072	CSE	1st	2012	1201214072_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	28	1201214073	CSE	1st	2012	1201214073_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	29	1201214074	CSE	1st	2012	1201214074_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	30	1201214075	CSE	1st	2012	1201214075_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	31	1201214075	CSE	1st	2012	1201214075_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	32	1201214076	CSE	1st	2012	1201214076_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	33	1201214077	CSE	1st	2012	1201214077_1st_regular.html	regular
<input type="checkbox"/>	Edit Copy Delete	34	1201214078	CSE	1st	2012	1201214078_1st_regular.html	regular

8:46 PM
22-Nov-15

subject_details

The screenshot shows the phpMyAdmin interface for the 'studentportal' database. The 'subject_details' table is selected. The table has columns: bsyid, subject1, subject2, subject3, subject4, subject5, and subject6. There are 8 rows of data.

bsyid	subject1	subject2	subject3	subject4	subject5	subject6
1	programming in c	mathematics-1	Physics/Chemistry	Thermodynamics/Mechanics	English communication skill	BE
2	programming in c	mathematics-1	physics/chemistry	thermodynamics/mechanics	English communication skill	BE
3	programming in c	mathematics-1	physics/chemistry	thermodynamics/mechanics	English communication skill	BE
4	programming in c	Mathematics-1	Physics/Chemistry	Thermodynamics/Mechanics	English communication skill	BE
5	programming in c	Mathematics-1	Physics/Chemistry	Thermodynamics/Mechanics	English Communication Skill	BE
6	programming in c	Mathematics-1	Physics/Chemistry	Thermodynamics/Mechanics	English communication skill	BE
7	programming in c	Mathematics-1	Physics/Chemistry	Thermodynamics/Mechanics	English Communication Skill	BE
8	programming in c	Mathematics-1	Physics/Chemistry	Thermodynamics/Mechanics	English communication	BE

branch_sem_year

The screenshot shows the phpMyAdmin interface for a MySQL database named 'studentportal'. The current table being viewed is 'branch_sem_year'. The table structure includes columns: bsysid, branch, semester, and year. The data contains 20 entries, each with a unique bsysid, a branch name (cse, etc, aei, mechanical, electrical, civil, mba), a semester (1st), and a year (2015, 2014, 2013, 2015, 2014, 2013, 2015, 2014, 2013, 2015, 2014, 2013, 2015, 2014, 2013, 2015, 2014, 2013, 2015, 2014). The interface also displays other tables in the database and various management tools.

bsysid	branch	semester	year
1	cse	1st	2015
2	cse	1st	2014
3	cse	1st	2013
4	etc	1st	2015
5	etc	1st	2014
6	etc	1st	2013
7	aei	1st	2015
8	aei	1st	2014
9	aei	1st	2013
10	mechanical	1st	2015
11	mechanical	1st	2014
12	mechanical	1st	2013
13	electrical	1st	2015
14	electrical	1st	2014
15	electrical	1st	2013
16	civil	1st	2015
17	civil	1st	2014
18	civil	1st	2013
19	mba	1st	2015
20	mba	1st	2014

user_feedback

The screenshot shows the phpMyAdmin interface for the 'studentportal' database. The left sidebar lists tables: account_details, admin_master, attendance_details, branch_sem_year, faculty_master, library_details, notice_details, student_master, student_result, subject_details, and user_feedback. The 'user_feedback' table is selected. The main area displays the data from the 'user_feedback' table:

	feedback_id	name	email	contactno	subject	message	date
<input type="checkbox"/>	8	saurabh bharti	saurabhbharti9@gmail.com	9040899268	Review about the site	A great way to access complex and voluminous stude...	15/11/15
<input type="checkbox"/>	9	abhimanyu	abhimanyupal.pal@gmail.com	8984020457	review	This project is awesome in whole branch of kist.....	18/11/15

Below the table, there are buttons for 'Check All / Uncheck All With selected', 'Change', 'Delete', and 'Export'. There are also 'Show' and 'Query results operations' sections.

Coding

Student Login

```
<?php  
if(isset($_GET['loginbtn']))  
{  
$regdno=$_GET['regdno'];  
$password=$_GET['password'];  
$sql="select * from student_master where regdno='".$regdno."' and  
password='".$password."';  
include 'db.php';  
$q=mysql_query($sql);  
if($arr=mysql_fetch_array($q))  
{  
$status=$arr['status'];  
if($status=='1')  
{  
$_SESSION['regdno']=$regdno;  
header('location:studenthome.php');  
}  
else  
{  
echo "<script>  
alert('Account is blocked');  
window.location.href='addattendance.php';  
</script>";  
}  
}
```

```
else
{
echo "<script>
alert('Invalid username or password');
window.location.href='addattendance.php';
</script>";
}
}
?>
<?php
if(isset($_GET['signup']))
echo "<script>
window.location.href='studentregister.php';
</script>";
?
?>
```

Student Registration

```
<?php
if(isset($_GET['submitbtn']))
{
$name=$_GET['name'];
$email=$_GET['email'];
$username=$_GET['username'];
$password=$_GET['password'];
$sque=$_GET['sque'];
$answer=$_GET['answer'];
```

```

$phone=$_GET['phone'];

include 'db.php';

$sql="insertintostudent_master
values('".$name."','".$email."','".$username."','".$password."','".$sque."','".$answer.
"',".$phone."','".$1."','".$images/blog_pic1.jpg','".$."");

$i=mysql_query($sql);

if($i)

{

echo "<script>

alert('Account created');

window.location.href='studentlogin.php';

</script>";

}

else

{

echo "<script>

alert('Oops!!! Error');

window.location.href='studentregister.php';

</script>";

}

}

?>

```

Add Library Details:-

```

<?php

if(isset($_GET['updatelibrarybtn']))

{

```

```
$year=$_GET['year'];
$branch=$_GET['branch'];
$semester=$_GET['semester'];
$regdno=$_GET['regdno'];
$subj1=$_GET['subj1'];
$subj2=$_GET['subj2'];
$subj3=$_GET['subj3'];
$subj4=$_GET['subj4'];
$subj5=$_GET['subj5'];
$subj6=$_GET['subj6'];
$subjs=$subj1.'. '.$subj2.'. '.$subj3.'. '.$subj4.'. '.$subj5.'. '.$subj6;
```

```
$s1=$_GET['s1'];
$s2=$_GET['s2'];
$s3=$_GET['s3'];
$s4=$_GET['s4'];
$s5=$_GET['s5'];
$s6=$_GET['s6'];
$book=$s1.'. '.$s2.'. '.$s3.'. '.$s4.'. '.$s5.'. '.$s6;
```

```
$a1=$_GET['a1'];
$a2=$_GET['a2'];
$a3=$_GET['a3'];
$a4=$_GET['a4'];
$a5=$_GET['a5'];
```

```

$a6=$_GET['a6'];
$author=$a1.'.$.a2.'.$.a3.'.$.a4.'.$.a5.'.$.a6;

include 'db.php';

$sql="insert           into           library_details
values('".$year."','".$branch."','".$semester."','".$subjs."','".$book."','".$regdno."','".$
.$author."');

$i=mysql_query($sql);

if($i)
{
$regdno=$_GET['regdno'];

echo "<script>
alert('Library details updated');
window.location.href='addlibrarydetails.php';
</script>";

}
else
{
echo "<script>
alert('Not Updated Try Again!!!');
window.location.href='addlibrarydetails.php';
</script>";

}
}

?>

```

```
<script type="text/javascript">

$(document).ready(function(e) {
    $("#showsubj").click(function()
    {
        var year=$("#year").val();
        var branch=$("#branch").val();
        var semester=$("#semester").val();
        $.ajax({
            type:'post',
            url:'fetchsubj1.php',
            data:{yr:year,branch:branch,semester:semester},
            success: function(res)
            {
                $("#subjectlist").html(res);
            }
        });
    });
});

</script>
```

Add Account Details:-

```
<?php  
if(isset($_GET["addaccountbtn"]))  
{  
  
$regdno=$_GET['regdno'];  
$branch=$_GET['branch'];  
$semester=$_GET['semester'];  
$year=$_GET['year'];  
$account=$_GET['account'];  
  
  
  
include 'db.php';  
$sql="insert into account_details  
values","",.$regdno."",".$semester."",".$year."",".$branch."",".$account."");  
$i=mysql_query($sql);  
if($i)  
{  
echo "<script>  
alert('Account Details updated');  
window.location.href='addaccount.php';  
</script>";  
}  
}  
?  
?>
```

Add Attendance:-

```
<html>
<head>
<script type="text/javascript">
$(document).ready(function(e) {
$("#showsubj").click(function()
{
var year=$("#year").val();
var branch=$("#branch").val();
var semester=$("#semester").val();
var date=$("#date").val();

$.ajax({
type:'post',
url:'fetchsubj_attendance.php',
data:{year:year,branch:branch,semester:semester,date:date},
success: function(res)
{
$("#attendancelist").html(res);
}
});
});
});
});</script>
</head>
```

Add Attendance details:-

```
<?php  
include 'error.php';  
$year=$_POST['year'];  
$branch=$_POST['branch'];  
$semester=$_POST['semester'];  
$date=$_POST['date'];  
  
include 'db.php';  
{  
$sql=mysql_query("select * from subject_details as t1 left join branch_sem_year as  
t2      on      t1.bsyid=t2.bsyid      where      t2.branch=\"".$branch."\"      and  
t2.semester=\"".$semester."\" and t2.year=\"".$year."\"");  
  
$sql1=mysql_query("select * from student_master where semester=\"".$semester."  
and year=\"".$year."\" and branch=\"".$branch."\" ");  
  
echo'<div id="page-wrap" class="fa-2x">';  
echo '<table class="table table-bordered">';  
echo '<tr>';  
echo '<th>Sl.No</th>';  
echo '<th>Regd No</th>';  
echo '<th>Name</th>';  
  
while($arr=mysql_fetch_array($sql))
```

```

{
echo '<th>' . $arr["subject1"] . '</th>';
echo '<th>' . $arr["subject2"] . '</th>';
echo '<th>' . $arr["subject3"] . '</th>';
echo '<th>' . $arr["subject4"] . '</th>';
echo '<th>' . $arr["subject5"] . '</th>';
echo '<th>' . $arr["subject6"] . '</th>';
echo '</tr>';
$j=0;
$array_regdno="";
$array_name="";
mysql_set_charset("utf8");
while($arr1=mysql_fetch_array($sql1))
{
$array_regdno=$array_regdno.','.$arr1["regdno"];
$array_name=$array_name.','.$arr1["fname"];
$s1 = $arr['subject1'];
$s2 = $arr['subject2'];
$s3 = $arr['subject3'];
$s4 = $arr['subject4'];
$s5 = $arr['subject5'];
$s6 = $arr['subject6'];
}

```

```

$j++;

echo'<tr><input type="hidden" name="subj1" value= '.$arr["subject1"].'>';
echo '<td> '.$j.' </td>';
echo '<td> '.$arr1["regdno"].' </td>';
echo '<td> '.$arr1["fname"].' </td>';
echo '<td>
<input type="hidden" name="subj2" value= '.$arr["subject2"].'>
<input      class="cb"      type="checkbox"      value="1"      name="cb1"
id="ch'.$arr["subject1"].$j.'" /></td>';

echo '<td>
<input type="hidden" name="subj3" value= '.$arr["subject3"].'>
<input      class="cb"      type="checkbox"      value="1"      name="cb2"
id="ch'.$arr["subject2"].$j.'" /></td>';

echo '<td>
<input type="hidden" name="subj4" value= '.$arr["subject4"].'>
<input      class="cb"      type="checkbox"      value="1"      name="cb3"
id="ch'.$arr["subject3"].$j.'" /></td>';

echo '<td>
<input type="hidden" name="subj6" value= '.$arr["subject5"].'>
<input      class="cb"      type="checkbox"      value="1"      name="cb4"
id="ch'.$arr["subject4"].$j.'" /></td>';

```

```

echo '<td>

<input type="hidden" name="subj6" value= '.$arr["subject6"].'>
<input      class="cb"      type="checkbox"      value="1"      name="cb5"
id="ch'.$arr["subject5"].$j.'" /></td>';

echo '<td>

<input type="hidden" name="regdno" value= '.$arr1["regdno"].'>
<input      class="cb"      type="checkbox"      value="1"      name="cb6"
id="ch'.$arr["subject6"].$j.'" /></td>';

echo'</tr><input type="hidden" name="fname" value= '.$arr1["fname"].'>';
}

}

echo '</table>';

echo'<label class="fa-btn btn-1 btn-1e"><input type="button" value="update"
name"updatebtn" id="up_btn"></label>';

echo'<label class="fa-btn btn-1 btn-1e"><input type="button" value="Select All"
name"checkbtn" id="check_btn"></label>';

echo'<input      type="hidden"      value='.$array_regdno.'      name="regdno"
id="regdno_id"/><br>';

echo'<input      type="hidden"      value="'.$array_name.'"      name="fname"
id="name_id"/><br>';

```

```

echo'<input type="hidden" value="'.$branch.'" name="branch" id="branch"/><br>';
echo'<input      type="hidden"      value="'.$semester.'"      name="semester"
id="semester"/><br>';
echo'<input type="hidden" value="'.$year.'" name="year" id="year"/><br>';
echo'<input type="hidden" value="'.$date.'" name="date" id="date"/><br>

echo'<input type="hidden" value='.$j.' id="totalstudent"/><br>';
echo'<input type="hidden" value='.$s1.' name="s_s1" id="sub1"/><br>';
echo'<input type="hidden" value='.$s2.' name="s_s2" id="sub2"/><br>';
echo'<input type="hidden" value='.$s3.' name="s_s3" id="sub3"/><br>';
echo'<input type="hidden" value='.$s4.' name="s_s4" id="sub4"/><br>';
echo'<input type="hidden" value='.$s5.' name="s_s5" id="sub5"/><br>';
echo'<input type="hidden" value='.$s6.' name="s_s6" id="sub6"/><br>

}

?>

<script type="text/javascript">
$(document).ready(function(e) {
    $("#up_btn").on('click',function()
    {
        var a_regdno=$("#regdno_id").val();
        var a_fname=$("#name_id").val();

        var branch=$("#branch").val();

```

```

var semester=$("#semester").val();
var year=$("#year").val();
var date=$("#date").val();

var sub1=$("#sub1").val();
var sub2=$("#sub2").val();
var sub3=$("#sub3").val();
var sub4=$("#sub4").val();
var sub5=$("#sub5").val();
var sub6=$("#sub6").val();
var totalstudent=$("#totalstudent").val();
//for each subjects's attendance.

var sa1="";
for(var i=1;i<=totalstudent;i++)
{
if($("#ch"+sub1+i).is(':checked'))
{
sa1=sa1+',1';
}
else
{
sa1=sa1+',0';
}
}

}


```

```

var sa2="";
for(var i=1;i<=totalstudent;i++)
{
    if($("#ch"+sub2+i).is(':checked'))
    {
        sa2=sa2+',1';
    }
    else
    {
        sa2=sa2+',0';
    }
}

var sa3="";
for(var i=1;i<=totalstudent;i++)
{
    if($("#ch"+sub3+i).is(':checked'))
    {
        sa3=sa3+',1';
    }
    else
    {
        sa3=sa3+',0';
    }
}

```

```
}
```

```
}
```

```
var sa4="";
for(var i=1;i<=totalstudent;i++)
{
if($("#ch"+sub4+i).is(':checked'))
{
sa4=sa4+',1';
}
else
{
sa4=sa4+',0';
}
}
```

```
var sa5="";
for(var i=1;i<=totalstudent;i++)
{
if($("#ch"+sub5+i).is(':checked'))
{
sa5=sa5+',1';
}
}
```

```

else
{
sa5=sa5+',0';
}

}

var sa6="";
for(var i=1;i<=totalstudent;i++)
{
if($("#ch"+sub6+i).is(':checked'))
{
sa6=sa6+',1';
}
else
{
sa6=sa6+',0';
}
}

$.ajax({
type:'post',
url:'ajax_attendance_update.php',

```

```

data:{ a_regdno:a_regdno,sub1:sub1,sub2:sub2,sub3:sub3,sub4:sub4,sub5:sub5,sub
6:sub6,sa1:sa1,sa2:sa2,sa3:sa3,sa4:sa4,sa5:sa5,sa6:sa6,a_fname:a_fname,branch:br
anch,semester:semester,year:year,date:date},

success: function(res)

{

alert(res);

window.location="add_attendance.php";

}

});

});

});

});

</script>

```

Manage Attendance:-

```

<head>

<script type="text/javascript">

$(document).ready(function(e) {

$("#showattendance").click(function()

{

var year=$("#year").val();

var branch=$("#branch").val();

var semester=$("#semester").val();

```

```

/*
var
limit=$("#limit").val();*/
$.ajax({
    type:'post',
    url:'fetchattendance.php',
    data:{yr:year,branch:branch,semester:semester},
    success: function(res)
    {
        $("#studentlist").html(res);
    }
});

});
});
});
</script>
</head>

<body>
<div class="main_bg">
<div class="container">
<div class="main row">

<div class="col-md-6">

```

```
<div class="contact-form">
<h2>Manage Attendance</h2>
<form method="get">
<div>
<legend>Batch</legend>
<span><select name="year" id="year" size="1" class="form-control">
<option value="2015">2015</option>
<option value="2014">2014</option>
<option value="2013">2013</option>
<option value="2012">2012</option>
</select>
</span>
</div>
<div>
<legend>Branch</legend>
<span><select name="branch" id="branch" size="1" class="form-control">
<option value="cse">Computer Science & Enggenieering</option>
<option value="etc">ETC</option>
<option value="aei">AEI</option>
<option value="mechanical">Mechanical</option>
<option value="electrical">Electrical</option>
<option value="civil">Civil</option>
<option value="mba">MBA</option>
</select>
</span>
```

```
<legend>Percentage Limit</legend>
<span><input type="text" class="form-control" id="limit" name="limit"/></span>
</div>*!-->
<br>
<br>
<div>
<label class="fa-btn btn-1 btn-1e">
<input type="button" value="show" name="showattendance"
id="showattendance"></label>
</div>
</div>
</div>
</div>
<div id="studentlist">
</div>
</div>
</div>
<div class="clearfix">
</div>
</div>
</div>
</div>
</div>
</form>
```

```
</body>
```

Manage Library Details:-

```
<head>
```

```
<script type="text/javascript">  
$(document).ready(function(e) {  
    $("#showlibrary").click(function()  
    {  
        var year=$("#year").val();  
        var branch=$("#branch").val();  
        var semester=$("#semester").val();  
        $.ajax({  
  
            type:'post',  
            url:'fetchlibrary.php',  
            data:{yr:year,branch:branch,semester:semester},  
            success: function(res)  
            {  
                $("#studentlist").html(res);  
            }  
  
        });  
    });  
});  
</script>  
</head>
```

TESTING

INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed.

Finally we arrive at system testing, where the software and other system elements are tested as a whole.

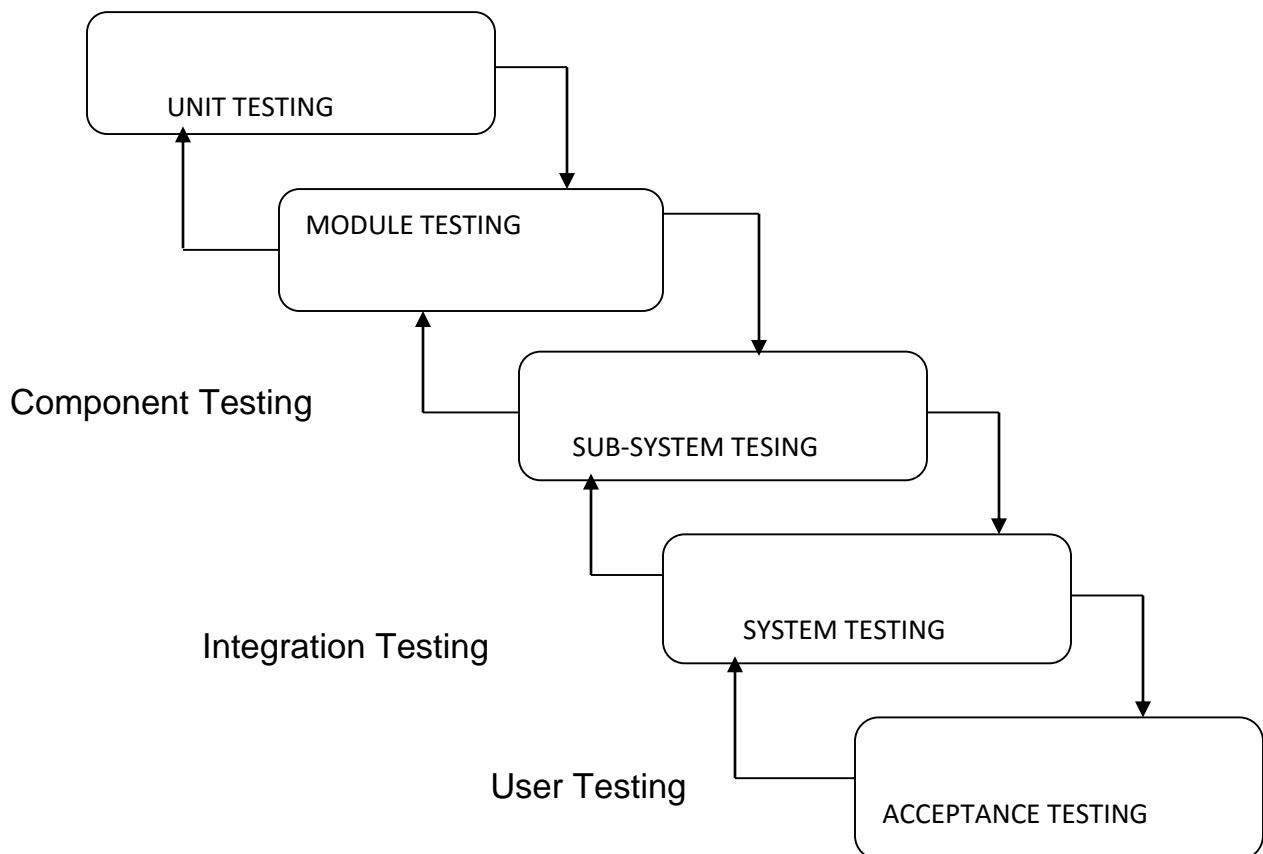


Figure:- Steps of testing

Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

2. BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$$V(G) = E - N + 2 \text{ or}$$

$$V(G) = P + 1 \text{ or}$$

$$V(G) = \text{Number Of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on particular condition is traced to uncover any possible errors.

4. DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible.

The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

Test Cases:

Test case 1

Test case 1: Registration	Priority (H, L): High	
Test Objective: For Verifying Registration		
Test Description: “User (student, faculty) enters the required fields, presses register button”, client program contacts with server, server contacts with the database, database updates and sends result to the user.		
Requirements Verified: Yes		
Test Environment: Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program.		
Test Setup/Pre-Conditions: Apache server should be in running state. All the mandatory fields must be entered.		
Actions	Expected Results	
The user will register to access application.	Displays the respective Pages	
Pass: Yes	Conditions pass: Yes	Fail: No
Problems / Issues:	NIL	
Notes: Successfully registered		

Test Case 2:-

Test case 2: Verifying Authentication.	Priority (H, L): High	
Test Description: “User(student, faculty) enters username and password and presses <u>submit button</u> ”, client program contacts with server, server contacts with the database, database checks for authentication and sends result as a valid user.		
Requirements Verified: Yes		
Test Environment: Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program.		
Test Setup/Pre-Conditions: Apache and Database server should be in running state. Username and Password fields should be entered.		
Actions	Expected Results	
The user presses submit button.	Displays Users profile page.	
Pass: yes	Conditions pass: yes	Fail: No
Problems / Issues: NIL		
Notes:User Verified successfully		

Test case 3:-

Test case 3: Add attendance	Priority (H, L): High	
Test Objective: For Adding attendance		
Test Description: “faculty enters the required fields, presses add button”, client program shows the respective subjects showing blank fields for attendance for respective subjects.		
Requirements Verified: Yes		
Test Environment: Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program.		
Test Setup/Pre-Conditions: Apache server should be in running state. All the mandatory fields must be entered.		
Actions	Expected Results	
The user will Enter the details and clicks add button.	The attendance will be uploaded	
Pass: Yes	Conditions pass: Yes	Fail: No
Problems / Issues: NIL		
Notes: attendance Successfully added.		

Test case 4:-

Test case 3: Add library details	Priority (H, L): High	
Test Objective: For Adding library details		
Test Description: “faculty enters the required fields, presses add button”, client program shows the respective subjects showing blank fields for attendance for respective subjects.		
Requirements Verified: Yes		
Test Environment: Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program.		
Test Setup/Pre-Conditions: Apache server should be in running state. All the mandatory fields must be entered.		
Actions	Expected Results	
The user will Enter the details and clicks add button.	The library details will be uploaded	
Pass: Yes	Conditions pass: Yes	Fail: No
Problems / Issues: NIL		
Notes: library details Successfully added.		

Test case 5:-

Test case 3: Add student result	Priority (H, L): High	
Test Objective: For adding student result		
Test Description: “faculty enters the required fields, presses add button”, client program shows the respective subjects showing blank fields for attendance for respective subjects.		
Requirements Verified: Yes		
Test Environment: Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program.		
Test Setup/Pre-Conditions: Apache server should be in running state. All the mandatory fields must be entered.		
Actions	Expected Results	
The user will Enter the details and clicks add button.	The student result will be uploaded	
Pass: Yes	Conditions pass: Yes	Fail: No
Problems / Issues: NIL		
Notes: student result Successfully added.		

Conclusion

The “student portal” has been successfully completed. The goal of the system is achieved and problems are solved. The package is developed in a manner that it is user friendly and required help is provided at different levels.

The project can be easily used in the process of decision making. Different types of reports can be generated which help the management to take correct decision and reduce the time delay which automatically increases the institute’s work standards as well as the economical state of the institute.

This system never decreases the manpower but helps the development of available manpower and optimizes the manpower by which institute’s standards and capabilities can be scaled to higher dimensions.

REFERENCE AND BIBLIOGRAPHY

REFERENCES

Database

Database Systems concept by Silverschatz, Korth, Sudarsan Pub. McGraw Hill.

Schaum's Outline of fundamental of relational databases by Ramon AMT, P Steve Suehring, MySql Bible.

KCushman, Pub: MC-graw Hill.

Web Page Design

Integrated HTML and CSS: A Smarter, Faster Way to Learn, VirginiaDeBolt, Pub: Sybex®

Dreamweaver MX 2004: The Complete Reference, Second Edition (Osborne Complete Reference Series)

High Performance JavascriptBy Kyle Lundon.

PHP and MYSQL

PHP AND MYSQL:Publisher: A press By: W. Jason Gilmore

PHP 5 Power Programming (AndiGutmans, StigBakken, et al)

Software /Project Management

IT project Management, On Track From Start to Finish, Second Edition, Joseph Phillips, McGraw-Hill/Osborne © 2004

Software Engineering, FIFTH EDITION, Roger S. Pressman, Ph.D.Pub: Mc-Gaw hill

Software Engineering, Rajib Mall

We have also made an earnest attempt to explore the “Internet Service” as much as we could. Here are some of the sites which we have taken help from.

www.google.com

www.sqlserver.com

BIBLIOGRAPHY

An Introduction to PHP AND MYSQL

- W. Jason Gilmore

Integrated HTML and CSS

- Virginia DeBolt

Dreamweaver MX 2004

- Osborne

Fundamental of relational databases

- Ramon AMT, P Steve Suehring

High Performance Javascript

- Kyle Landon

At the end, as the maker of this project I proudly credit success to all those quarters (as mentioned above) from which i have received tons of help and support towards my creation.