

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI - 590018.**



**A MINI - PROJECT REPORT ON
“ONLINE COLLEGE PORTAL”**

*submitted in partial fulfilment of requirements for the fifth semester
DATABASE MANAGEMENT SYSTEM LABORATORY*

*for the course of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE*

submitted by

SHASHWAT SHAHI
(1ST17CS148)

SHUBHAM CHAKRABORTY
(1ST17CS152)

*under the guidance of
Mrs. Bilwashree H
Asst. Prof, Dept. of CSE
&
Mrs. Divyashree N
Asst. Prof, Dept. of CSE*



**Department of Computer Science & Engineering
SAMBHRAM INSTITUTE OF TECHNOLOGY
BENGALURU 560097.**

SAMBHRAM INSTITUTE OF TECHNOLOGY

M.S. PALYA, JALAHALLI EAST, BENGALURU - 560 097.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the mini-project entitled “**ONLINE COLLEGE PORTAL**” has been carried out by **SHUBHAM CHAKRABORTY (1ST17CS148)** and **SHUBHAM CHAKRABORTY (1ST17CS152)**, bonafide students of **Sambhram Institute of Technology** in partial fulfilment of requirements for the fifth semester **Database Management System Laboratory**, prescribed by the **Visvesvaraya Technological University (VTU)**, Belagavi during the academic year **2019-2020**. It is also hereby ensured that all corrections/suggestions indicated for internal assessment have been incorporated while submitting this report. This report has been approved as it satisfies the academic requirements with respect to the project work prescribed for the said course.

Mrs. Bilwashree H

Asst.Professor, Dept. Of CSE

Dr. T John Peter

Head, Dept. of CSE, SaIT

External Viva

Names of the examiners

Signatures with date

1. _____

2. _____

ACKNOWLEDGEMENT

While presenting this mini-project entitled “**ONLINE COLLEGE PORTAL**” entrusted to us, we feel that it is our duty to acknowledge the guidance and help provided for its success.

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavour to a successful culmination.

We are grateful to our institution, **Sambhram Institute of Technology** for providing us, a congenial atmosphere to carry out this project with all possible amount of ease.

We thank **Dr. H.G Chandrakanth**, Principal, Sambhram Institute of Technology, and **Dr. T John Peter**, HOD, Department of Computer Science & Engineering, for their constant support.

We extend our thanks to our esteemed guides, **Mrs. Bilwashree H & Mrs. Divyashree N** Assistant Professors, Department of Computer Science & Engineering, for providing us with the right guidance and advice at the crucial junctures and for showing us the right way.

We are also indebted to our parents and friends for their continued moral and material support throughout the course of this project and helping us in showcasing the project’s presentation.

Our thanks to all teaching, non-teaching staffs and others who have contributed bits, bytes and words to accomplish this project.

SHASHWAT SHAHI

SHUBHAM CHAKRABORTY

ABSTRACT

The purpose of the project entitled as “**ONLINE COLLEGE PORTAL**” is to reduce the consumption of time during maintaining the records of college. Separate divisions are providing to maintain the records of Faculties, Students, Subjects and other important information. Our System also provides an easy way not only to automate all functionalities of a college, but also to provide full functional reports to top management of college with the finest of details about any aspect of college. In other words, our portal has, following objectives: Simple database is maintained. Easy operations for the operator of the system. User interfaces are user friendly and attractive. It takes very less time for the operator to use the system. Traditionally, it was done manually. The main function of system is to register and store Student details and Faculty details and retrieve these details as and when required, and also to manipulate these details meaningfully. The **ONLINE COLLEGE PORTAL** can be entered using a username and password. It is accessible either by an Administrator or Student and Faculty. Only Administrator can update data into the database and user only can add the data while signup otherwise can only read data related to him/her. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

TABLE OF CONTENTS

CHAPTER	CHAPTER NAME	PAGE NO
CHAPTER 1	INTRODUCTION TO DATABASE	1
1.1	Database system environment	3
1.2	Advantages of using DBMS approach	5
1.3	Architecture of database	6
CHAPTER 2	USER REQUIREMENTS DEFINITION	8
2.1	Overview	8
2.1.1	Objectives	8
2.2	Functionalities of the application	8
2.2.1	User classes and characteristics	9
2.3	Description of the relations	9
CHAPTER 3	SYSTEM REQUIREMENTS SPECIFICATION	
3.1	Introduction	13
3.1.1	Hardware requirements	13
3.1.2	Software requirement	13
CHAPTER 4	ANALYSIS	14
4.1	Existing system	14
4.2	Proposed system	14
4.3	Feasibility study	14
4.3.1	Economic Feasibility	14
4.3.2	Technical Feasibility	15
4.3.3	Operational Feasibility	15

CHAPTER 5	SYSTEM DESIGN	16
5.1	Bootstrap	16
5.2	HTML	17
5.3	Cascading system style sheets(CSS)	18
5.4	MYSQLI	19
5.5	PHP	22
5.6	ER DIAGRAM	23
5.7	SCHEMA DIAGRAM	24
CHAPTER 6	RESULT AND ANALYSIS	26
6.1	User sign in page	26
6.2	User signup page	27
6.3	User Dashboard	27
CONCLUSION		39
BIBLIOGRAPHY		40

LIST OF FIGURES

Fig. No	DESCRIPTION	Page No
1.1	The database system environment	3
1.2	Client - Server architecture	6
5.1	Sample screenshot of HTML code	17
5.2	Sample screenshot of CSS code	19
5.3	Sample screenshot of MYSQLII code	21
5.4	Sample screenshot of PHP code	22
5.5	Entity Relationship Diagram	23
5.6	Schema Diagram	24
6.1	User login page	26
6.2	User signup page	27
6.3	Dashboard for Faculty	27
6.4	Dashboard for Student	28
6.5	Dashboard for Administrator	28
6.6	Student Details page	29
6.7	Student Academics Details Page	29
6.8	Students IAMarks Page(View Permission)	30
6.9	Students IAMarks Page(Update Permission)	30
6.10	Database Structure	31
6.11	Added Stored Procedures	31
6.12	Table Structure (Academics)	32
6.13	Table Structure (Branch)	32
6.14	Table Structure (Class)	33
6.15	Table Structure (Faculty)	33
6.16	Table Structure (IAMarks)	34

6.17	Table Structure (Semsec)	34
6.18	Table Structure (Student)	35
6.19	Table Structure (Subject)	35
6.20	Table Structure (Users)	36

LIST OF TABLES

Table No	DESCRIPTION	Page No
2.1	User	9
2.2	Students	10
2.3	Faculty	10
2.4	Class	11
2.5	Academics	11
2.6	Branch	11
2.7	IAMarks	12
2.8	Subject	12
2.9	Semsec	12

CHAPTER 1

INTRODUCTION TO DATABASE

A **database** is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database.

The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the **miniworld** or the universe of discourse (**UOD**). Changes to the miniworld are reflected in the database.
- A database is a **logically coherent collection of data** with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database may be generated and maintained manually or it may be computerized. For example, a library card catalog is a database that may be created and maintained manually. A computerized database may be created and maintained either by a group of application programs written specifically for that task or by a database management system. We are only concerned with computerized databases in this report.

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data.

- **Constructing** the database is the process of storing the data on some storage medium that is controlled by the DBMS.
- **Manipulating** a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data.
- **Sharing** a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by sending queries or requests for data to the DBMS. A query typically causes some data to be retrieved; a transaction may cause some data to be read and some data to be written into the database.

Other important functions provided by the DBMS include protecting the database and maintaining it over a long period of time. Protection includes system protection against hardware or software malfunction (or crashes) and security protection against unauthorized or malicious access. A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.

It is not absolutely necessary to use general-purpose DBMS software to implement a computerized database.

We could write our own set of programs to create and maintain the database, in effect creating our own special-purpose DBMS software. In either case—whether we use a general-purpose DBMS or not—we usually have to deploy a considerable amount of complex software. In fact, most DBMSs are very complex software systems.

To complete our initial definitions, we will call the database and DBMS software together a database system.

1.1 DATABASE SYSTEM ENVIRONMENT

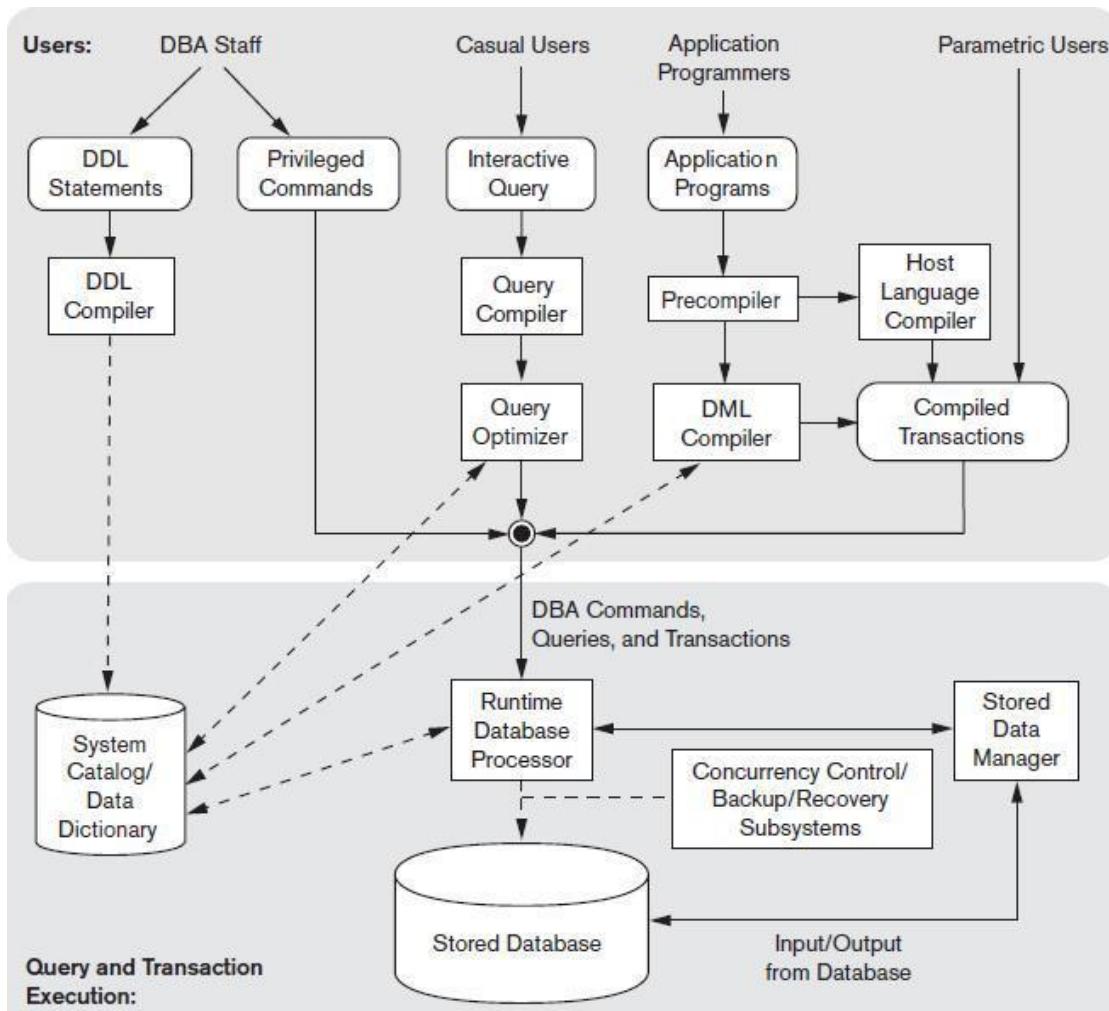


Fig 1.1: The database system environment

The database and the DBMS catalog are usually stored on disk. Access to the disk is controlled primarily by the operating system (OS), which schedules disk read/write.

Many DBMSs have their own buffer management module to schedule disk read/write, because this has a considerable effect on performance. Reducing disk read/write improves performance considerably. A higher-level stored data manager module of the DBMS controls access to DBMS information that is stored on disk, whether it is part of the database or the catalog.

- The **DBA staff**, casual users work with interactive interfaces to formulate queries.

- **Application programmers** create programs using some host programming languages, and parametric users do data entry work by supplying parameters to predefined transactions. The DBA staff works on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands.
- The **DDL compiler** processes schema definitions, specified in the DDL, and stores descriptions of the schemas (meta-data) in the DBMS catalog. The catalog includes information such as the names and sizes of files, names and data types of data items, storage details of each file, mapping information among schemas, and constraints. In addition, the catalog stores many other types of information that are needed by the DBMS modules, which can then look up the catalog information as needed.
- **Casual users** and persons with occasional need for information from the database interact using some form of interface, which we call the interactive query interface. These queries are parsed and validated for correctness of the query syntax, the names of files and data elements, and so on by a query compiler that compiles them into an internal form. This internal query is subjected to query optimization.
- The **query optimizer** is concerned with the rearrangement and possible reordering of operations, elimination of redundancies, and use of correct algorithms and indexes during execution. It consults the **system catalog** for statistical and other physical information about the stored data and generates executable code that performs the necessary operations for the query and makes calls on the runtime processor.
- The **runtime database processor** executes (1) the privileged commands, (2) the executable query plans, and (3) the canned transactions with runtime parameters. It works with the system catalog and may update it with statistics. It also works with the stored data manager, which in turn uses basic operating system services for carrying out low-level input/output (read/write) operations between the disk and main memory. It is now common to have the client program that accesses the DBMS running on a separate computer from the computer on which the database resides.

- The former is called the client computer running a DBMS client software and the latter is called the database server. In some cases, the client accesses a middle computer, called the application server, which in turn accesses the database server.

1.2 ADVANTAGES OF USING THE DBMS APPROACH

Due to its centralized nature, the database system can overcome the disadvantages of the file system-based system.

- **Data independency:** Application program should not be exposed to details of data representation and storage DBMS provides the abstract view that hides these details.
- **Efficient data access:** DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently.
- **Data integrity and security:** Data is accessed through DBMS, it can enforce integrity constraints. E.g.: Inserting salary information for an employee.
- **Data Administration:** When users share data, centralizing the data is an important task, Experience professionals can minimize data redundancy and perform fine tuning which reduces retrieval time.
- **Concurrent access and Crash recovery:** DBMS schedules concurrent access to the data. DBMS protects user from the effects of system failure.
- **Reduced application development time:** DBMS supports important functions that are common to many applications.

1.3 ARCHITECTURE OF DATABASE

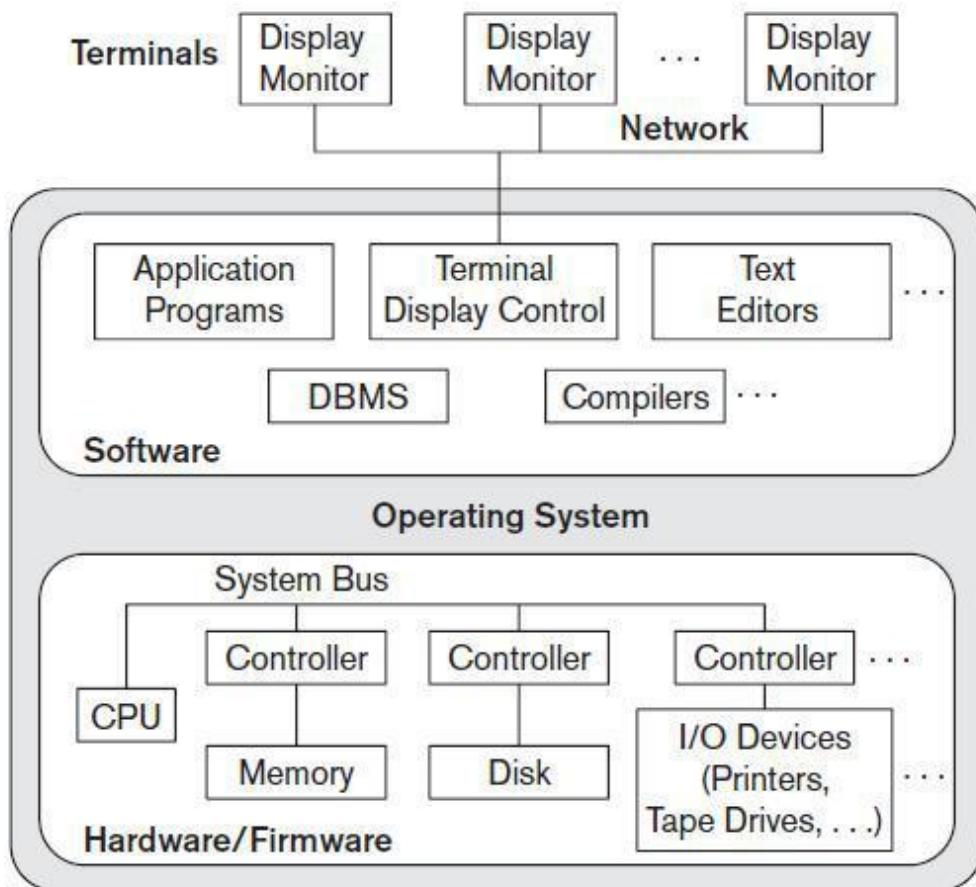


Fig 1.2: A physical, centralized and a basic Client - Server architecture

The idea is to define **specialized servers** with specific functionalities. For example, it is possible to connect a number of PCs or small workstations as clients to a **file server** that maintains the files of the client machines. Another machine can be designated as a **printer server** by being connected to various printers; all print requests by the clients are forwarded to this machine.

Thus, the resources provided by specialized servers can be accessed by many client machines. The **client machines** provide the user with the appropriate interfaces to utilize these servers, as well as with local processing power to run local applications. This concept can be carried over to other software packages, with specialized programs such as a CAD (computer-aided design) package being stored on specific server machines and being made accessible to multiple clients.

A **client** in this framework is typically a user machine that provides user interface capabilities and local processing. When a client requires access to additional functionality such as database access that does not exist at that machine, it connects to a server that provides the needed functionality.

A **server** is a system containing both hardware and software that can provide services to the client machines, such as file access, printing, archiving, or database access.

CHAPTER 2

USER REQUIREMENTS DEFINITION

2.1 OVERVIEW

- Online College Portal includes registration of students and faculties, storing their details into the system. The system has a facility to give a unique id for every user (student and faculties) and stores the details of every user by submitting it.
- It includes facilities like updating, entering and viewing details related to each user with different permissions for each category of users.
- The current manual system is slow, laborious and error prone to computerize the same for quicker efficient result and user satisfaction which resolve around transaction-oriented activities.

2.1.1 OBJECTIVES

- To automate the existing system by relating all of the records of the above mentioned operations in a logical manner such that the recorded transactions can be replaced and accepted without major changes and problems.
- To eliminate paper work by using a computerized system.
- To save time and money.
- To trap most of the data entry errors.

2.2 FUNCTIONALITIES OF THE APPLICATION

- Provision of quick access to the records maintained.
- Showing important details so that important decisions could be made easily.
- Showing computerization of student's and faculties details which make the maintenance of records easier.
- Quicker updation of records of users and departments of college.
- Proper maintenance of student's academics details.
- Timely updation of student's IAMarks by the faculties.

- Provision the computerization of the Online College Portal will help greatly in maintaining proper information about the users.
- Provide effective and efficient querying techniques.

2.2.1 USER CLASSES AND CHARACTERISTICS

Any user(Student and Faculty) with basic computer skills can make use of the application. The user should have only been briefed about the functionality of the system before he/she can start using the system.

2.3 DESCRIPTION OF THE RELATIONS (TABLES)

(I) Users: This table is used to store details of the users (For Login)

Si.no	Name	Type	Description
1	Id	Varchar(10)	ID
2	Email	Varchar (100)	Email ID
3	Usertype	Varchar(100)	Type of User
4	Password	Varchar (100)	Password

Table 2.1: User Table

(II) Student : This table is used to store the details of the Student

Si.no	Name	Type	Description
1	usn	Varchar(10)	Student ID
2	firstname	Varchar(50)	First name
3	lastname	Varchar(50)	Last name
4	semester	int(1)	Semester
5	section	Varchar(1)	Section
6	branchid	Varchar(3)	Branch ID
7	contactno	Int(10)	Contact Number
8	email	Varchar(30)	Email ID

Table 2.2: Students Table

(III) Faculty: This table contains the details of the Faculties

Si.no	Name	Type	Description
1	id	Varchar(10)	ID
2	Firstname	Varchar(30)	First Nname
3	Lastname	Varchar(30)	Last Name
4	designation	Varchar(30)	Designation
5	qualification	Varchar(30)	Qualification
6	branchid	Varchar(3)	Branch ID
7	Contactno	int(10)	Contact Number
8	Email	Varchar(30)	Email ID

Table 2.3: Faculty Table

(IV) Class : This table contains the details of classes in the college.

Si.no	Name	Type	Description
1	usn	Varchar(10)	ID
2	ssid	int(11)	Section ID

Table 2.4: Class Table

(V) Academics : This table contains the academic details of the Students.

Si.no	Name	Type	Description
1	Usn	Varchar(10)	Student ID
2	Ssc	Float	Senior Secondary
3	Hsc	Float	Higher Secondary
4	first	Float	First Semester Marks
5	second	Float	Second Semester Marks
6	third	Float	Third Semester Marks
7	fourth	Float	Fourth Semester Marks
8	fifth	Float	Fifth Semester Marks
9	sixth	Float	Sixth Semester Marks
10	seventh	Float	Seventh Semester Marks
11	eighth	Float	Eighth Semester Marks

Table 2.5: Academics Table

(VI) Branch : This table contains the details of branches offered by the college.

Si.no	Name	Type	Description
1	branch_id	Varchar(3)	Branch ID
2	branch_name	Varchar(50)	Branch Name

Table 2.6: Branch Table

(VII) IAMarks : This table contains the details of Internal Marks of all the Students.

Si.no	Name	Type	Description
1	usn	Varchar(10)	Student ID
2	subcode	Varchar(10)	Subject Code
3	ssid	int(10)	Section ID
4	test1	int(10)	Test 1 Marks
5	test2	int(10)	Test 2 Marks
6	test3	int(10)	Test 3 Marks
7	finalia	int(10)	Final IA Marks

Table 2.7: IAMarks Table

(VIII) Subject : This table contains the details of Subjects present in each Semester.

Si.no	Name	Type	Description
1	subcode	Varchar(10)	Subject ID
2	title	Varchar(100)	Title of Subject
3	sem	int(1)	Semester
4	credits	int(1)	Credits of Subject

Table 2.8: Subject Table

(IX) Semsec : This table contains the details of Sections of the Semesters of each Branch.

Si.no	Name	Type	Description
1	Ssid	int(10)	Section ID
2	sem	Int(1)	Semester
3	sec	Varchar(1)	Section
4	branch	Varchar(3)	Branch ID

Table 2.9: Semsec Table

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

3.1.1 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

PROCESSOR	:	Intel/AMD
RAM	:	1 GB
HARD DISK	:	80 GB

3.1.2 SOFTWARE REQUIREMENTS

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

OPERATING SYSTEM	:	Windows XP /Vista/7/8 /10
FRONT END	:	Html, CSS , Java script (Web Browser)
SERVER SIDE SCRIPT	:	PHP
DATABASE	:	MYSQLI

CHAPTER 4

ANALYSIS

4.1 EXISTING SYSTEM

Colleges currently use a manual system for the management and maintenance of critical information of students and faculties. The current system requires numerous paper forms, with data stores spread through out the College Management Infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the college and may lead to inconsistencies in data in various data stores.

4.2 PROPOSED SYSTEM

The Online College Portal is designed for any college to replace their existing manual paper based system. The new system is to control the information of students and faculties. Managing students records, Managing students Academic Records, Managing Students IA Marks and Managing Faculties Records. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

4.3 FEASIBILITY STUDY

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

4.3.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

4.3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

4.3.3 OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

SYSTEM DESIGN AND IMPLEMENTATION

5.1 BOOTSTRAP

Getting Started

Here, you will learn how easy it is to create a web page using Bootstrap. Before begin, be sure to have a code editor and some working knowledge of HTML and CSS.

If you're just starting out in web development, start learning from here » www.tutorialrepublic.com/html-tutorial/

Well, let's get straight into it.

Downloading the Bootstrap Files

There are two versions available for download, *compiled Bootstrap* and *Bootstrap source files*. You can download Bootstrap files from here» <https://getbootstrap.com/docs/3.3/getting-started/> Compiled download contains compiled and minified version of CSS and JavaScript files as well as icons in font format for faster and easier web development, while the source contain original source files for all CSS and JavaScript, along with a local copy of the docs.

For the purpose of better understanding we'll focus on the compiled Bootstrap files. It saves your time because you don't have to bother every time including separate files for individual functionality. It will also increase the performance of your website and saves the precious bandwidth when you decided to move your site on production because of lesser HTTP request and download size since files are compiled and minified.

Understanding the File Structure

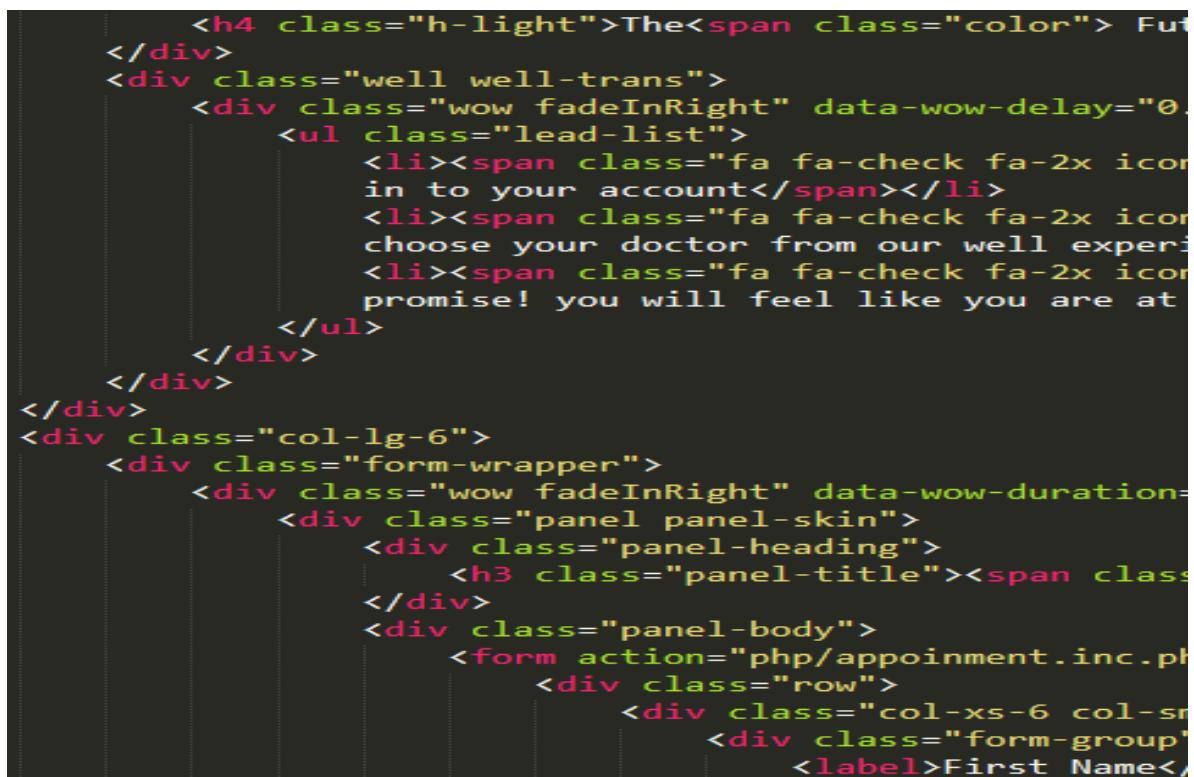
Once downloaded the compiled Bootstrap, unzip the compressed folder to see the structure. You'll find the following file structure and contents.

```
bootstrap/
|—— css/
| |—— bootstrap.css
| |—— bootstrap.min.css
| |—— bootstrap-theme.css
| |—— bootstrap-theme.min.css
```

```
|—— js/
| |—— bootstrap.js
| |—— bootstrap.min.js
|—— fonts/
| |—— glyphicons-halflings-regular.eot
| |—— glyphicons-halflings-regular.svg
| |—— glyphicons-halflings-regular.ttf
| |—— glyphicons-halflings-regular.woff
```

5.2 HTML

HTML or Hypertext Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets. HTML tags most commonly come in pairs, although some tags represent empty elements and so are unpaired, for example .



A screenshot of an HTML code editor displaying a portion of an HTML document. The code includes various HTML tags such as `<h4>`, `<div>`, ``, ``, ``, `<form>`, and `<label>`. The code is styled with CSS classes like `h-light`, `color`, `well well-trans`, `wow fadeInRight`, and `lead-list`. The content discusses medical services and promises to the user.

```
<h4 class="h-light">The<span class="color"> Future</h4>
</div>
<div class="well well-trans">
    <div class="wow fadeInRight" data-wow-delay="0.5s">
        <ul class="lead-list">
            <li><span class="fa fa-check fa-2x icon"> in to your account</span></li>
            <li><span class="fa fa-check fa-2x icon"> choose your doctor from our well experi
            <li><span class="fa fa-check fa-2x icon"> promise! you will feel like you are at
        </ul>
    </div>
</div>
<div class="col-lg-6">
    <div class="form-wrapper">
        <div class="wow fadeInRight" data-wow-duration="1s">
            <div class="panel panel-skin">
                <div class="panel-heading">
                    <h3 class="panel-title"><span class="color"> Book An Appoin
                </div>
                <div class="panel-body">
                    <form action="php/appointment.inc.php">
                        <div class="row">
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>First Name</label>
                                    <input type="text" name="first_name" value="John" />
                                </div>
                            </div>
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Last Name</label>
                                    <input type="text" name="last_name" value="Doe" />
                                </div>
                            </div>
                        </div>
                        <div class="row">
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Email Address</label>
                                    <input type="text" name="email" value="john.doe@example.com" />
                                </div>
                            </div>
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Phone Number</label>
                                    <input type="text" name="phone" value="123-456-7890" />
                                </div>
                            </div>
                        </div>
                        <div class="row">
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Appointment Type</label>
                                    <select name="appointment_type">
                                        <option value="Consultation">Consultation</option>
                                        <option value="Treatment">Treatment</option>
                                        <option value="Prescription">Prescription</option>
                                    </select>
                                </div>
                            </div>
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Date</label>
                                    <input type="text" name="date" value="2023-10-15" />
                                </div>
                            </div>
                        </div>
                        <div class="row">
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Time</label>
                                    <input type="text" name="time" value="10:00 AM" />
                                </div>
                            </div>
                            <div class="col-xs-6 col-sm-6">
                                <div class="form-group">
                                    <label>Notes</label>
                                    <input type="text" name="notes" value="None" />
                                </div>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>
```

Fig 5.1: Sample screenshot of HTML code

The first tag (like `<div>`) in a pair is the start tag, and the second tag (like `</div>`) is the end tag (they are also called opening tags and closing tags. Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag).

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 describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

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 documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

5.3 CASCADING STYLE SHEETS (CSS)

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content 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 . Below is the example of CSS.

```
th
{
    padding: 10px 10px;
    text-align: center;
    font-weight: 500;
    font-size: 12px;
    color: black;
    text-transform: uppercase;
}
td
{
    padding: 10px 10px;
    text-align: center;
    vertical-align: middle;
    font-weight: 300;
    font-size: 12px;
    color: white;
    border-bottom: solid 1px rgba(255,255,255,0.1);
}
```

Fig 5.2: Sample screenshot of CSS code

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 file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

5.4 MYSQLI

MYSQLI is developed, distributed, and supported by Oracle Corporation. MYSQLI is a database system used on the web it runs on a server. MYSQLI is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MYSQLI can be compiled on a number of platforms.

The data in MYSQLI is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically.

FEATURES OF MYSQLI:

Internals and portability:

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms.
- Tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool.
- Uses multi-layered server design with independent modules.

Security:

- A privilege and password system that is very flexible and secure, and that enables host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

Scalability and Limits:

- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- Support for up to 64 indexes per table (32 before MySQL 4.1.2). Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 767 bytes for InnoDB tables, or 1000 for MyISAM; before MySQL 4.1.2, the limit is 500 bytes. An index may use a prefix of a column for CHAR, VARCHAR, BLOB, or TEXT column types.

CONNECTIVITY:

Clients can connect to MySQL Server using several protocols:

- Clients can connect using TCP/IP sockets on any platform.
- On Windows systems in the NT family (NT, 2000, XP, 2003, or Vista), clients can connect using named pipes if the server is started with the --enable-named-pipe option. In MySQL 4.1 and higher, Windows servers also support shared-memory connections if started with the --shared-memory option. Clients can connect through shared memory by using the --protocol=memory option.
- On UNIX systems, clients can connect using Unix domain socket files.

LOCALIZATION:

- The server can provide error messages to clients in many languages.
- All data is saved in the chosen character set.

CLIENTS AND TOOLS:

- MYSQLI includes several client and utility programs. These include both command-line programs such as MySQLIdump and MySQLIadmin, and graphical programs such as MySQLI Workbench.
- MYSQLI Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the MySQLIcheck client. MySQLI also includes myisamchk , a very fast command-line utility for performing these operations on MyISAM tables.
- MySQLI programs can be invoked with the --help or -? option to obtain online assistance.

WHY TO USE MYSQLI:

- Leading open source RDBMS
- Ease of use – No frills
- Fast
- Robust
- Security
- Multiple OS support
- Free
- Technical support
- Support large database— up to 50 million rows, file size limit up to 8 Million TB

```
<?php

    $userprofile = $_SESSION['user_name'];
    if($userprofile==true)
    {
        }

    else
    {
        header("location:login.php");
    }

    $query = "SELECT * FROM users WHERE Username='$userprofile'";
    $data = mysqli_query($conn,$query);
    $result = mysqli_fetch_assoc($data);

?>
```

Fig 5.3: Sample screenshot of MYSQLI code

5.5 PHP

WHAT IS PHP?

PHP is an acronym for "PHP Hypertext Preprocessor"

PHP is a widely-used, open source scripting language

PHP scripts are executed on the server

PHP costs nothing, it is free to download and use

WHAT IS PHP FILE?

PHP files can contain text, HTML, CSS, JavaScript, and PHP code

PHP code are executed on the server, and the result is returned to the browser as plain HTML

PHP files have extension ".php"

WHAT CAN PHP DO?

PHP can generate dynamic page content

PHP can create, open, read, write, delete, and close files on the server

PHP can collect form data

PHP can send and receive cookies

PHP can add, delete, modify data in your database

PHP can restrict users to access some pages on your website

PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

```
<?php

$adminprofile = $_SESSION['admin_name'];
if($adminprofile==true)
{
}
else
{
    header("location:admin.php");
}

?>
```

Fig 5.4: Sample screenshot of PHP code

5.6 ER DIAGRAM

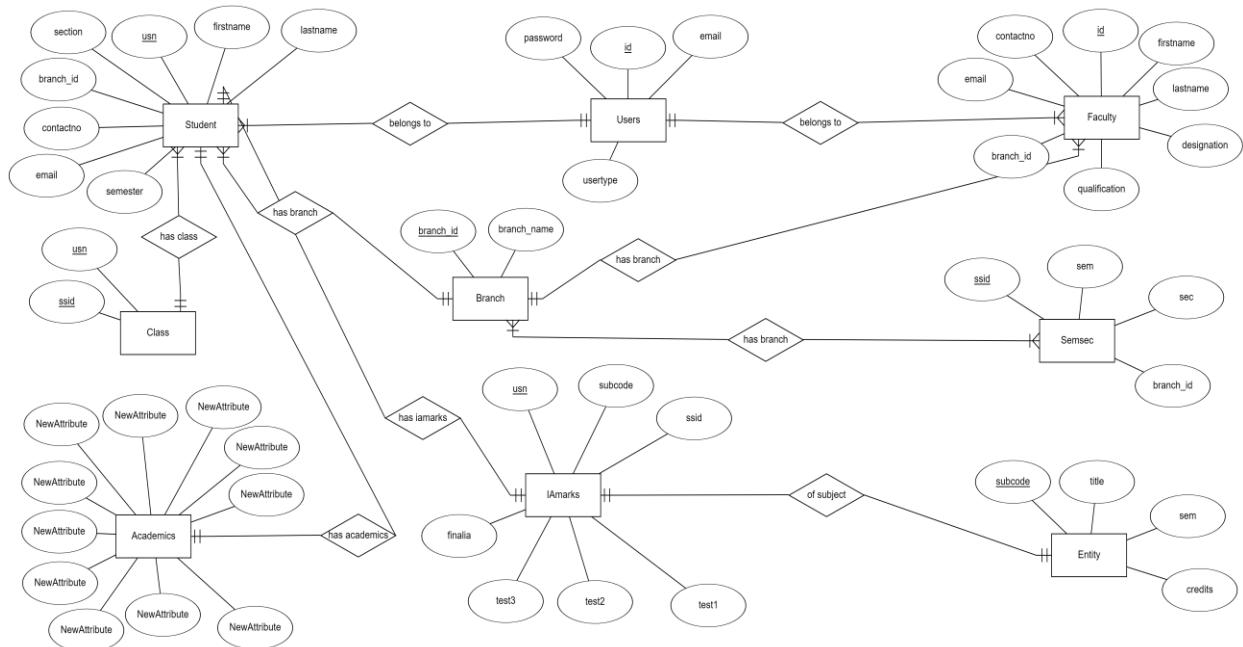


Fig 5.5: Entity Relationship Diagram

Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

Facts about ER Diagram Model:

- ER model allows you to draw Database Design
- It is an easy to use graphical tool for modeling data
- Widely used in Database Design
- It is a GUI representation of the logical structure of a Database
- It helps you to identifies the entities which exist in a system and the relationships between those entities

5.6 SCHEMA DIAGRAM

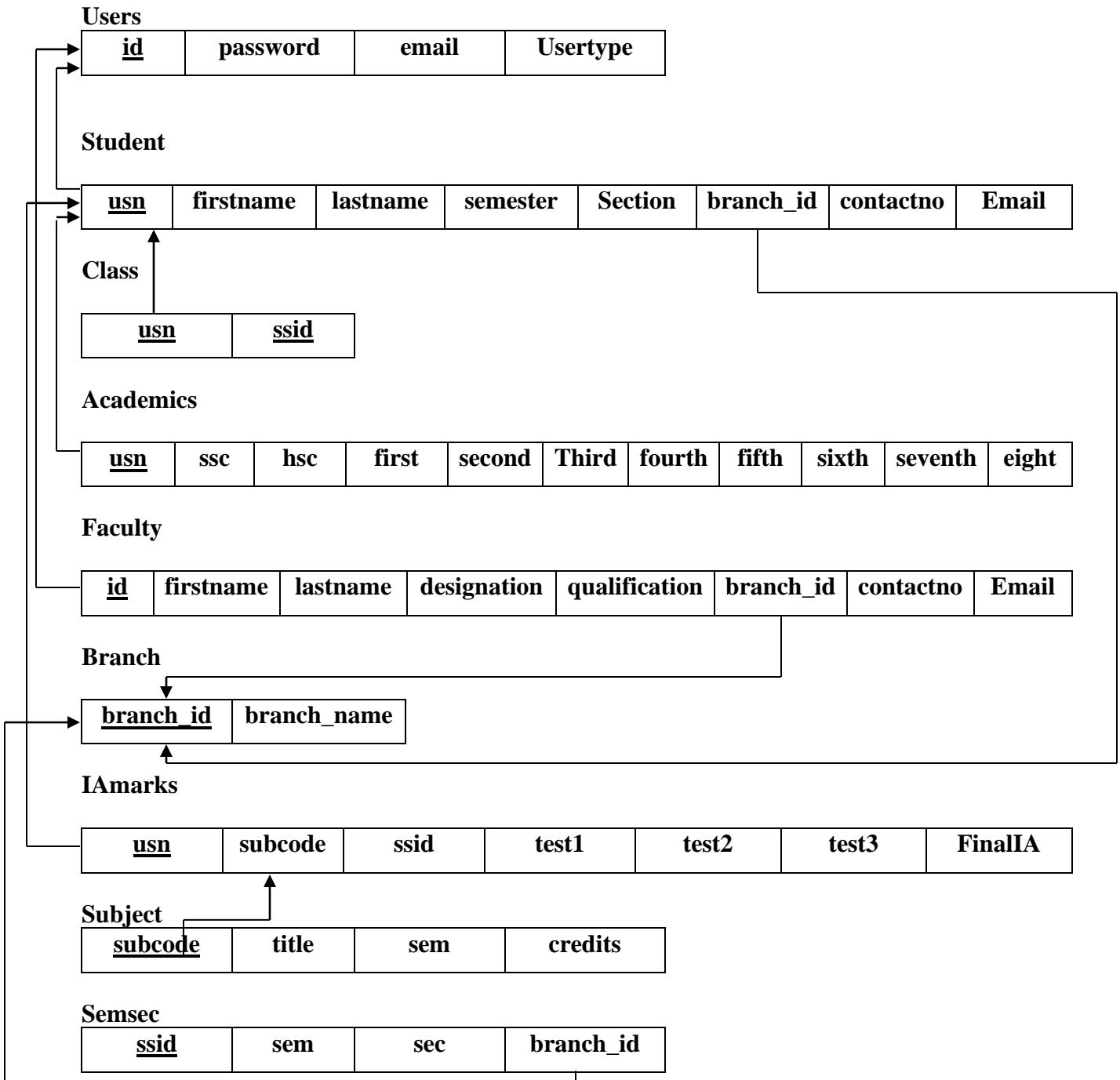


Fig 5.6: The Schema Diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

CHAPTER 6

RESULT AND ANALYSIS

6.1 USER SIGNUP PAGE

The screenshot shows a web browser window titled "Login" with the URL "localhost:3000/login.php". The page features a header with the "SAMBHRAM INSTITUTIONS" logo. Below the header is a navigation bar with links: Home, About us, Admissions, Courses, Placements, Contact Us, and Sambhram Institutions. The main content area has a dark background. On the left, there is a "Login Here" form with fields for Email and password, and a dropdown menu for "I am a:". On the right, there is a section titled "About the Project" featuring a photograph of a multi-story building with a blue roof and multiple entrances. At the bottom of the page, there is a sidebar with options for Faculty, Admin, and buttons for Submit, Signup, Signup As Faculty, and Signup As Student. The footer contains the text "Portal Developed By: Shashwat Shahi and Shubham Chakraborty".

Fig 6.1: The User login page , which acts as the main entry point to the College Login Interface

ONLINE COLLEGE PORTAL

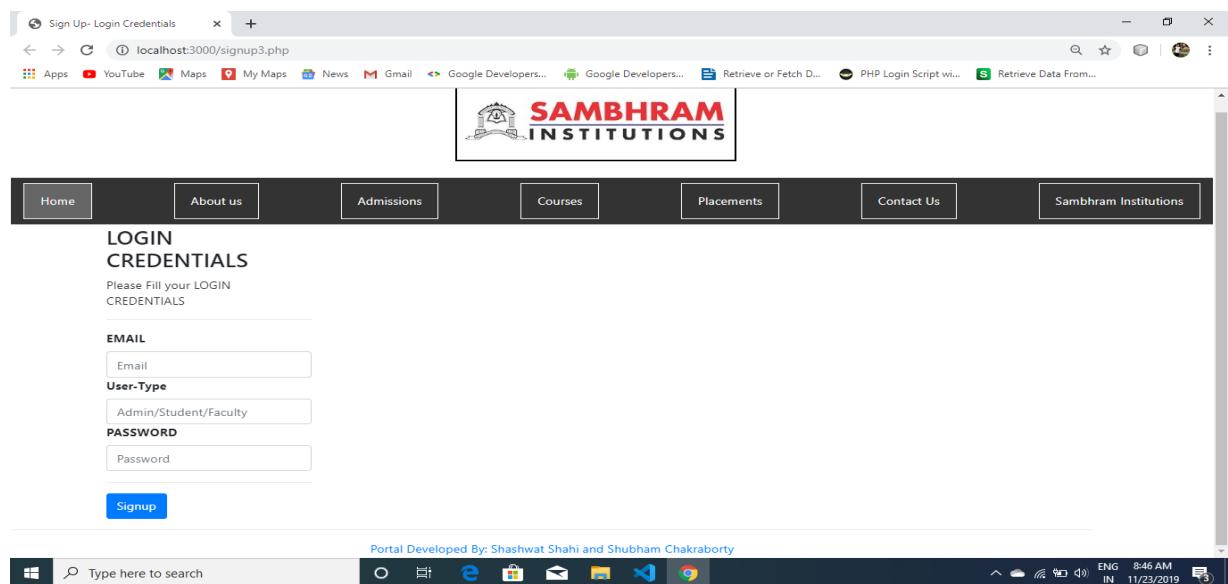


Fig 6.2: USER SIGNUP PAGE

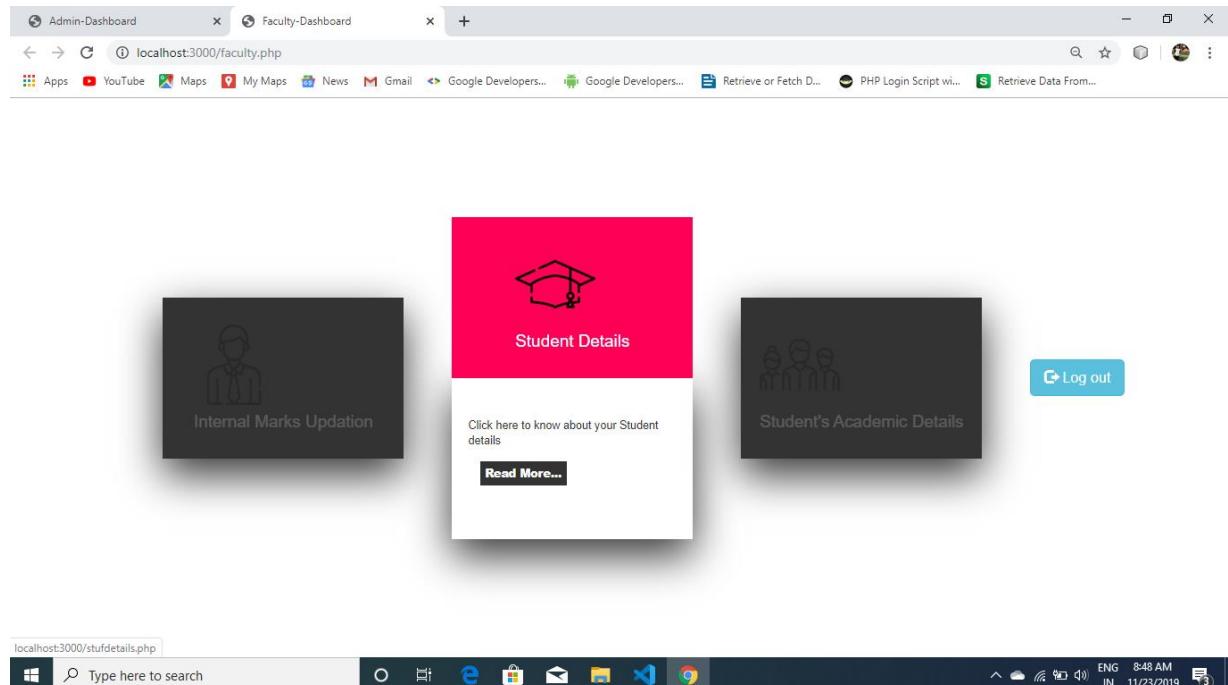


Fig 6.3: The Dashboard for Faculty

ONLINE COLLEGE PORTAL

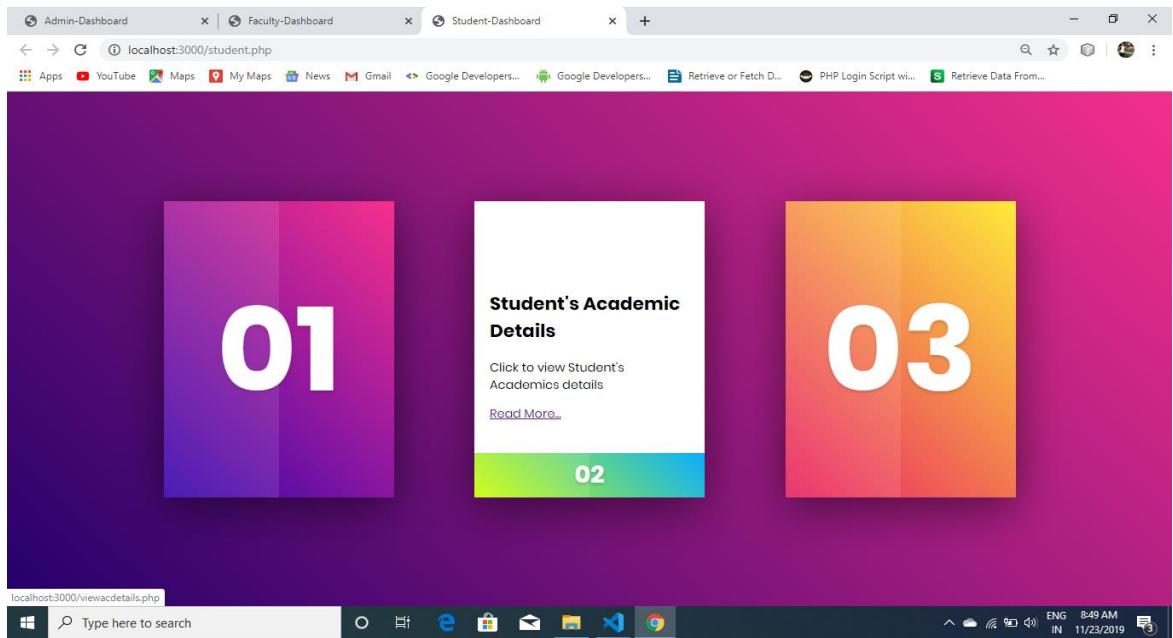


Fig 6.4: The Dashboard for Student

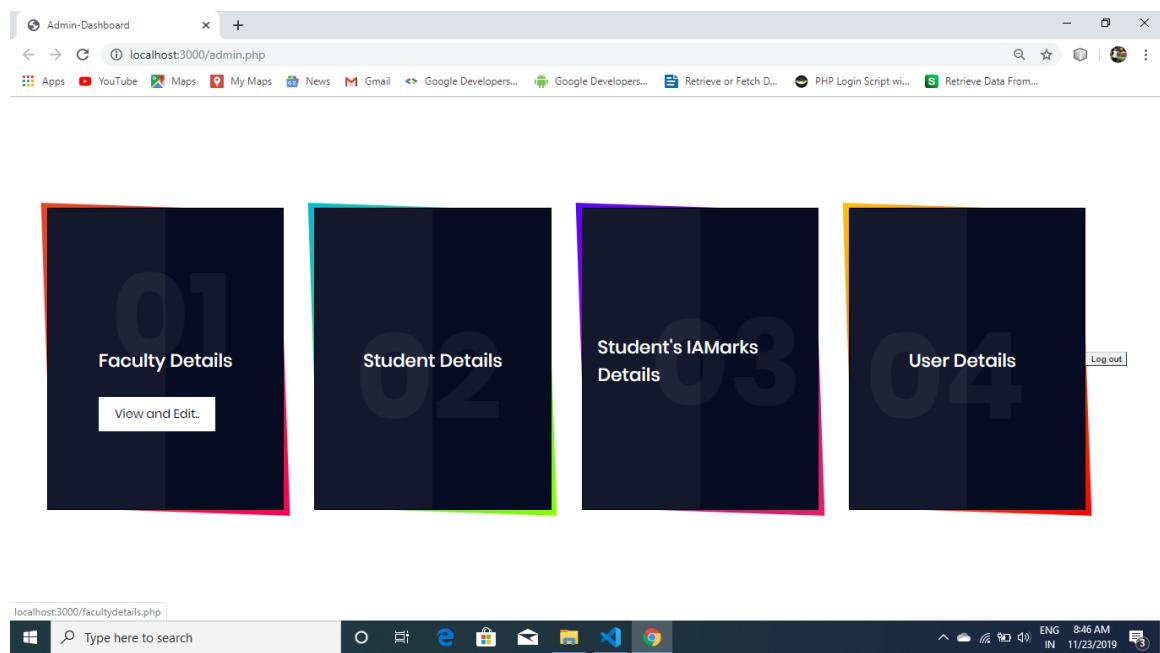


Fig 6.5: The Dashboard for Administrator

ONLINE COLLEGE PORTAL

The screenshot shows a web browser window for 'localhost:3000/studetails.php'. At the top, there's a navigation bar with links for Home, About us, Admissions, Courses, Placements, Contact Us, and Sambhram Institutions. The main content area has a title 'Student Details' and a table showing student information:

USN	First Name	Last Name	Semester	Section	Branch Id	Contact No	Email	Update	Delete
1st17cs148	Shashwat	Shahi	5	b	cse	2147483647	shashwats538@gmail.com	Update	Delete

Below the table are links for 'Back to Dashboard' and 'Logout'. A footer note at the bottom states 'Portal Developed By: Shashwat Shahi and Shubham Chakraborty'.

Fig 6.6: Student Details page(with update and delete permission)

The screenshot shows a web browser window for 'localhost:3000/viewacdetails.php'. The layout is identical to Fig 6.6, with a similar navigation bar and table. However, the 'Update' and 'Delete' buttons are missing from the table row. The table data is as follows:

USN	SSC Marks	HSC Marks	First Sem Marks	Second Sem Marks	Third Sem Marks	Fourth Sem Marks	Fifth Sem Marks	Sixth Sem Marks	Seventh Sem Marks	Eighth Sem Marks
1st17cs148	10	10	11	9	8	9	0	0	0	0

Below the table are links for 'Back to Dashboard' and 'Logout'. A footer note at the bottom states 'Portal Developed By: Shashwat Shahi and Shubham Chakraborty'.

Fig 6.7: Student Academic's Details page (Only View Permission)

ONLINE COLLEGE PORTAL

The screenshot shows a web browser window with the URL `localhost:3000/viewiamarks.php`. The page title is "Internal Marks Details". At the top, there is a navigation bar with links: Home, About us, Admissions, Courses, Placements, Contact Us, and Sambhram Institutions. Below the navigation bar is a logo for "SAMBHRAM INSTITUTIONS". A table displays internal marks data:

USN	Subcode	SSID	Test 1	Test 2	Test 3	Final IA
1st17cs148	1	1	25	25	25	25

Below the table are two buttons: "Back to Dashboard" and "Logout". At the bottom of the page, a footer note reads "Portal Developed By: Shashwat Shahi and Shubham Chakraborty".

Fig 6.8: Student's IAMarks page (Only View Permission)

The screenshot shows a web browser window with the URL `localhost:3000/iadetails.php`. The page title is "Internal Marks Details". The layout is identical to Fig 6.8, featuring a navigation bar, a logo, and a table displaying internal marks data. The table includes additional columns for "Update" and "Delete":

USN	Subcode	SSID	Test 1	Test 2	Test 3	Final IA	Update	Delete
1st17cs148	1	1	25	25	25	25	Update	Delete

Below the table are two buttons: "Back to Dashboard" and "Logout". At the bottom of the page, a footer note reads "Portal Developed By: Shashwat Shahi and Shubham Chakraborty".

Fig 6.9: Student's IAMarks page (With Update and Delete Permission)

Table	Action	Rows	Type	Collation	Size	Overhead
academics	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 Kib	-
branch	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 Kib	-
class	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 Kib	-
faculty	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 Kib	-
iamarks	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 Kib	-
semsec	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 Kib	-
student	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	32 Kib	-
subject	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 Kib	-
userlog	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 Kib	-
users	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	32 Kib	-
10 tables		Sum		28 InnoDB latin1_swedish_ci 192 Kib		0 B

Fig 6.10: Database's Structure

Name	Action	Type	Returns
getacademicsdetails	Edit Execute Export Drop	PROCEDURE	
getfacultydetails	Edit Execute Export Drop	PROCEDURE	
getiamarks	Edit Execute Export Drop	PROCEDURE	
getstudentdetails	Edit Execute Export Drop	PROCEDURE	
getuserdetails	Edit Execute Export Drop	PROCEDURE	

Fig 6.11: Added Stored Procedures

The screenshot shows the phpMyAdmin interface for the 'finalproj' database. The left sidebar lists various databases and tables, including 'finalproj' and its tables like 'branch', 'class', 'faculty', etc. The main panel displays the 'Table structure' for the 'academics' table. The table has 11 columns: 'usn' (varchar(10)), 'ssc' (float), 'hsc' (float), 'first' (float), 'second' (float), 'third' (float), 'fourth' (float), 'fifth' (float), 'sixth' (float), 'seventh' (float), and 'eighth' (float). The primary key is 'usn'. An index 'PRIMARY' is defined on column 'usn'. The 'Indexes' section also shows a warning: 'No partitioning defined!'. Below the table structure, there are tabs for 'Space usage' and 'Row statistics'.

Fig 6.12: Table Structure (academics)

The screenshot shows the phpMyAdmin interface for the 'finalproj' database. The left sidebar lists various databases and tables, including 'finalproj' and its tables like 'branch', 'class', 'faculty', etc. The main panel displays the 'Table structure' for the 'branch' table. The table has 2 columns: 'branch_id' (varchar(3)) and 'branch_name' (varchar(50)). The primary key is 'branch_id'. An index 'PRIMARY' is defined on column 'branch_id'. The 'Indexes' section shows a warning: 'No partitioning defined!'. Below the table structure, there are tabs for 'Space usage' and 'Row statistics'.

Fig 6.13: Table Structure (branch)

Table structure for class:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	usn	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
2	ssid	int(11)			No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	usn	0	A	No	

Information:

- Space usage: 16 KIB
- Format: dynamic
- Collation: latin1_swedish_ci

Fig 6.14: Table Structure (class)

Table structure for faculty:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
2	firstname	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	lastname	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	designation	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	qualification	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
6	branch	varchar(3)	latin1_swedish_ci		No	None			Change Drop More
7	contactno	int(10)			No	None			Change Drop More
8	email	varchar(30)	latin1_swedish_ci		No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	

Information:

- Space usage: 16 KIB
- Format: dynamic
- Collation: latin1_swedish_ci

Fig 6.15: Table Structure (faculty)

The screenshot shows the phpMyAdmin interface for the 'finalproj' database. The left sidebar lists various databases and tables. The main panel displays the structure of the 'iamarks' table. The table has seven columns: 'usn' (varchar(10), primary key, auto-increment), 'subcode' (varchar(10)), 'ssid' (int(10)), 'test1' (int(10)), 'test2' (int(10)), 'test3' (int(10)), and 'finalia' (int(10)). An index is defined on the 'usn' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	usn	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
2	subcode	varchar(10)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	ssid	int(10)			Yes	NULL			Change Drop More
4	test1	int(10)			Yes	NULL			Change Drop More
5	test2	int(10)			Yes	NULL			Change Drop More
6	test3	int(10)			Yes	NULL			Change Drop More
7	finalia	int(10)			Yes	NULL			Change Drop More

Fig 6.16: Table Structure (IAMarks)

The screenshot shows the phpMyAdmin interface for the 'finalproj' database. The left sidebar lists various databases and tables. The main panel displays the structure of the 'semsec' table. The table has four columns: 'ssid' (int(10), primary key, auto-increment), 'sem' (int(1)), 'sec' (varchar(1)), and 'branch' (varchar(3)). An index is defined on the 'ssid' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ssid	int(10)			No	None	AUTO_INCREMENT		Change Drop More
2	sem	int(1)			No	None			Change Drop More
3	sec	varchar(1)	latin1_swedish_ci		No	None			Change Drop More
4	branch	varchar(3)	latin1_swedish_ci		No	None			Change Drop More

Fig 6.17: Table Structure (SemSec)

Table structure for student:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	usn	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
2	fname	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	lname	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	semester	int(1)			No	None			Change Drop More
5	section	varchar(1)	latin1_swedish_ci		No	None			Change Drop More
6	branchid	varchar(3)	latin1_swedish_ci		No	None			Change Drop More
7	contactno	int(15)			No	None			Change Drop More
8	email	varchar(30)	latin1_swedish_ci		No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	usn	1	A	No	
Edit Drop	hasbranch	BTREE	No	No	branchid	1	A	No	

Create an index on: 1 columns [Go](#)

Partitions: [No partitioning defined!](#)

Fig 6.18: Table Structure (Student)

Table structure for subject:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	subcode	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
2	title	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	sem	int(1)			No	None			Change Drop More
4	credits	int(1)			No	None			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	subcode	2	A	No	

Create an index on: 1 columns [Go](#)

Partitions: [No partitioning defined!](#)

Information: [Console](#)

Fig 6.19: Table Structure (Subject)

The screenshot shows the phpMyAdmin interface for the 'finalproj' database. The left sidebar lists various databases and tables. The main panel displays the 'Table structure' for the 'users' table. The table has four columns: 'id' (int(10), primary key, auto-increment), 'email' (varchar(100)), 'usertype' (varchar(100)), and 'password' (varchar(100)). Below the table structure, there are sections for 'Indexes' (with one PRIMARY index on 'id') and 'Partitions' (which is currently empty). The status bar at the bottom shows the URL as 'localhost/phpmyadmin/server_privileges.php?db=finalproj&table=users&checkprvsdb=finalproj&checkprivstable=users&viewing_mode=table'.

Fig 6.20: Table Structure (Users)

CONCLUSION

Since we are entering details of the students electronically in the "ONLINE COLLEGE PORTAL", data will be secured. Using this application, we can retrieve any user's record with a single click. Thus, processing information will be faster. It guarantees accurate maintenance of user. It easily reduces the book keeping task and thus reduces the human effort and increases accuracy speed. The users will be able to access the information with a single click of the system. The system administrator will be able to manage all the system records, view, delete and update them. Each user with different usertypes have different access permissions for each type of record which ensures the security of the records. The maintenance of the records becomes easier.

BIBLIOGRAPHY

- [1] HTML & CSS: Design and Build Web Sites by Jon Duckett
- [2] Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke
- [3] Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon
- [4] <https://www.w3schools.com>
- [5] <https://www.geeksforgeeks.com>
- [6] PL/SQL Programming – Evan Bayross
- [7] Fundamentals of Database Systems 7th Edition by Elmasri , Navathe
- [8] Learning PHP, MYSQLI, JavaScript, CSS & HTML5 by Robin Nixon