

# ACKNOWLEDGEMENT

The successful presentation of the **DBMS MINI PROJECT** would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I would like to extend my gratitude to the **MANAGEMENT, KAMMAVARI SANGHAM**, Bengaluru, for providing all the facilities to present the DBMS Mini Project.

I would like to extend my gratitude to **Dr. K. RAMA NARASIMHA**, Principal / Director, K.S.School of Engineering and Management, Bengaluru, for facilitating me to present the Web Technology Mini Project.

I thank **Dr. K Venkata Rao**, Professor and Head, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, for her encouragement.

I would like to thank our Project Guide, **Mrs. Nita Meshram.**, Assistant Professor, Department of Computer Science and Engineering, K. S. School of Engineering and Management, Bengaluru, for their constant guidance and inputs.

I would like to thank all the **Teaching** Staff and **Non-Teaching** Staff of the college for their co-operation.

Finally, I extend my heart-felt gratitude to my family for their encouragement and support without which I would not have come so far. Moreover, I thank all my friends for them invaluable support and cooperation.

**CHANDANA K - 1KG19CS018**

**MANOJ G - 1KG19CS047**

# **ABSTRACT**

Student result management system, is a management information system for education establishments used to manage student data. Student result management system provide capabilities for registering students and the results of student assessment scores; storing individual student information. Information security is a concern, as universities house an array of sensitive personal information, making them potentially attractive targets for security breaches, such as those experienced by retail corporations or healthcare providers. Education institutes can easily use the Student result management system to maintain students' records.

Achieving this objective is difficult using a manual system as the information is scattered, can be redundant, and collecting relevant information may be very time-consuming. All these problems are solved using this project.

# TABLE OF CONTENTS

Chapter No.	Contents	Page No.
	Acknowledgement	I
	Abstract	II
	Table of Contents	III
	List of Figures	V
	List of Tables	V
<b>Chapter 1</b>	<b>INTRODUCTION</b>	1
1.1	OVERVIEW	1
1.2	PROBLEM STATEMENT	1
1.3	DATABSE MANAGEMENT SYSTEM	1
1.4	SQL	1
1.5	HTML	2
1.6	PHP	
<b>Chapter 2</b>	<b>REQUIREMENTS SPECIFICATION</b>	4
2.1	OVERALL DESCRIPTION	4
2.2	SPECIFIC REQUIREMENTS	4
2.3	SOFTWARE REQUIREMENTS	4
2.5	HARDWARE REQUIREMENTS	4
2.6	TECHNOLOGY	4
<b>Chapter 3</b>	<b>DETAILED DESIGN</b>	6
3.1	SYSTEM DESIGN	6
3.2	ENTITY RELATIONSHIP DIAGRAM	6
3.3	RELATIONAL SCHEMA	8
3.4	DESCRIPTION OF TABLES	9
<b>Chapter 4</b>	<b>IMPLEMENTATION</b>	11
4.1	MODULE AND THEIR ROLES	16
4.2	TRIGGERS	16
4.3	STORED PROCEDURES	
<b>Chapter 5</b>	<b>TESTING</b>	17
5.1	SOFTWARE TESTING	17
5.2	MODULE TESTING AND INTEGRATION	17

<b>Chapter 6</b>	<b>SNAP SHOTS</b>	18
6.1	LOGIN PAGE	18
6.2	SIGNUP PAGE	18
6.3	HOME PAGE	19
6.4	ADD STUDENT MARKS PAGE	19
6.5	DELETE STUDENT RECORD PAGE	20
6.6	EDIT STUDENT INFORMATION PAGE	20
6.7	EDIT STUDENT SGPA/CGPA PAGE	21
6.8	STUDENT RESULT PAGE	21
<b>Chapter 7</b>	<b>CONCLUSION</b>	22
<b>Chapter 8</b>	<b>FUTURE ENHANCEMENTS</b>	23
	<b>REFERENCES</b>	24

## LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	E R Diagram of Student Result Management System	7
3.2	Relational Schema Diagram	8
6.1	Login Page	19
6.2	Signup Page	19
6.3	Home Page	20
6.4	Add Student Marks Page	20
6.5	Delete Student Record Page	21
6.6	Edit Student Information Page	21
6.7	Edit Student SGPA/CGPA Marks Page	22
6.8	Student Result Page	22

## LIST OF TABLES

Table No.	Table Name	Page No.
3.3	Description of Tables	9

# Chapter 1

## INTRODUCTION

### 1.1 OVERVIEW

Education institutes can easily use the Student Information Management System to maintain students' records. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant, and collecting relevant information may be very time-consuming. This can be used to maintain students personal information and results, which can only be edited by admins. While students can only view the results. All these problems are solved using this project.

### 1.2 PROBLEM STATEMENT

To make an unified interface for both teachers and students to check the student results, compare with class results. The website Student Result Management System aims to record a considerable number of students records and needs online assistance for managing students' records.

### 1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

### 1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data

access control. SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.[13] Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

## **1.5 HTML**

HTML is a markup language used for structuring and presenting content on the web and the fifth and current major version of the HTML standard. HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web application.

## **1.6 PHP**

PHP is an acronym for “ Hypertext PreProcessor”

PHP is a widely used, open source scripting language

PHP scripts are executed on the server

PHP is free to download and use

### **What is a PHP File?**

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

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

PHP files have extension ".php"

### **PHP Form Handling**

The PHP super global \$\_GET and \$\_POST are used to collect form-data.

### **GET vs. POST**

Both GET and POST create an array (e.g., array( key1 => value1=1, key2 value2, key3 => value3,...)). This array holds key/value pairs, where keys are the names of the form controls and values are the input data from the user.

Both GET and POST are treated as \$\_GET and \$\_POST. These are super global, which means that they are always accessible, regardless of scope and you can access them from any function, class or file without having to do anything special.

\$\_GET is an array of variables passed to the current script via the URL parameters.

\$\_POST is an array of variables passed to the current script via the HTTP POST method.

**When to use GET?**

Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL). GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.

GET may be used for sending non-sensitive data.

**When to use POST?**

Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send. Moreover, POST supports advanced functionality such as support for multi-part binary input while uploading files to server.

**What is MySQL?**

MySQL is a database system used on the web

MySQL is a database system that runs on a server

MySQL is ideal for both small and large applications

MySQL is very fast, reliable, and easy to use



## Chapter 2

### REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

#### 2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

#### 2.2 SPECIFIC REQUIREMENTS

The specific requirements of the Student Result Management System are stated as follows:

##### 2.2.1 SOFTWARE REQUIREMENTS

- ☐ IDE – Visual Studio Code
- ☐ Web Browser – Firefox 50 or later, Google Chrome – 60 or later
- ☐ Database support - MySQL
- ☐ Operating system – Windows
- ☐ Server deployment – Apache server/XAMPP

##### 2.2.2 HARDWARE REQUIREMENTS

- ☐ Processor – Pentium IV or above
- ☐ RAM – 2 GB or more
- ☐ Hard disk – 3 GB or more
- ☐ Monitor – VGA of 1024x768 screen resolution or more
- ☐ Keyboard and Mouse

##### 2.2.3 TECHNOLOGY

- ☐ HTML is used for the front end design. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
- ☐ CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied

- ☐ to any XML document.
- ☐ SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
- ☐ We require a PHP connection between the front end and back end components to write to the database and fetch required data.

## **XAMPP**

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the Apache Friends, and its native source code can be revised or modified by the audience. It consists of Apache HTTP Server, MariaDB, and interpreter for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

### **What is XAMPP?**

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl. XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

### **Components of XAMPP**

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named Apache, a database management system named MariaDB and scripting/ programming languages such as PHP and Perl, X denotes Cross-platform, which means that it can work on different platforms such as Windows, Linux, and macOS.

## Chapter 3

### DETAILED DESIGN

#### 3.1 SYSTEM DESIGN

The web server needs a PHP, to act as a mediator between the client browser and a database. Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic (constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language.

We may create different database user for every aspect of your application with very limited rights to database objects. The most required privileges should be granted only, and avoid that the same user can interact with the same database in different cases. This means that if an intruder gains access to the database using application credentials, they can only effect as many changes as application can.

#### 3.2 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

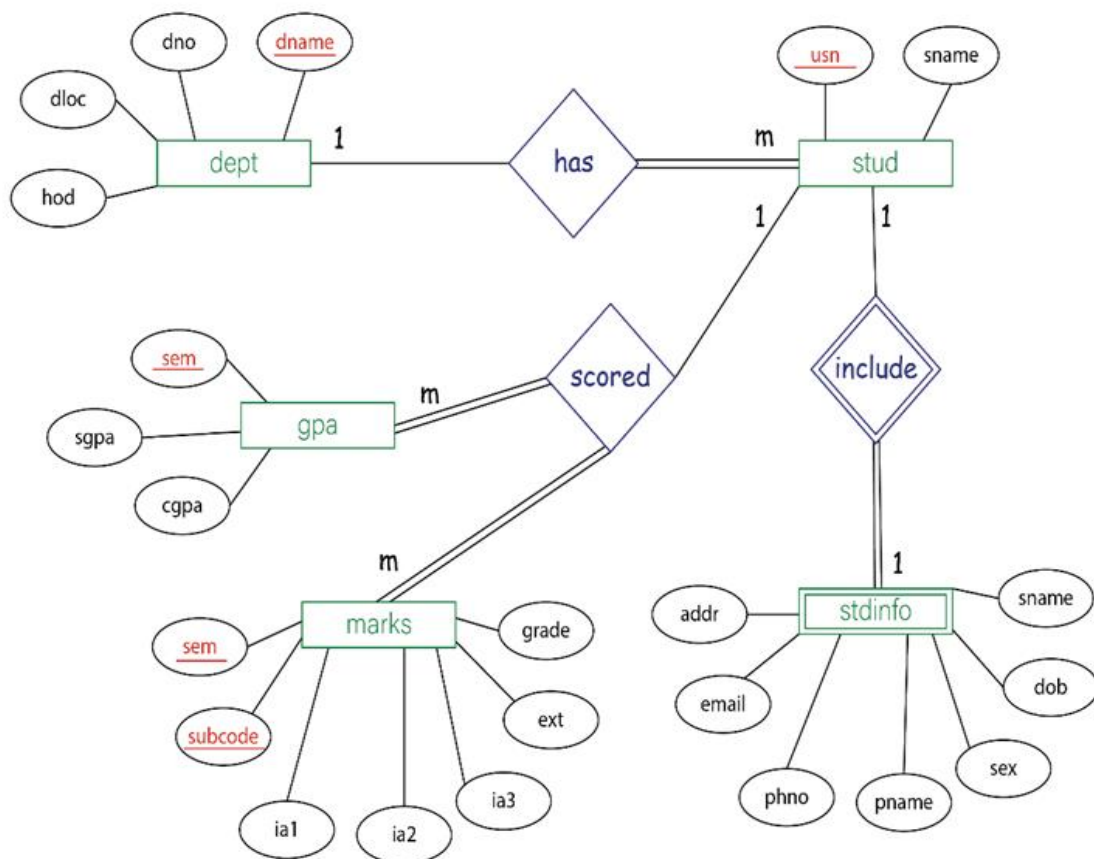
Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field

in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three-schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

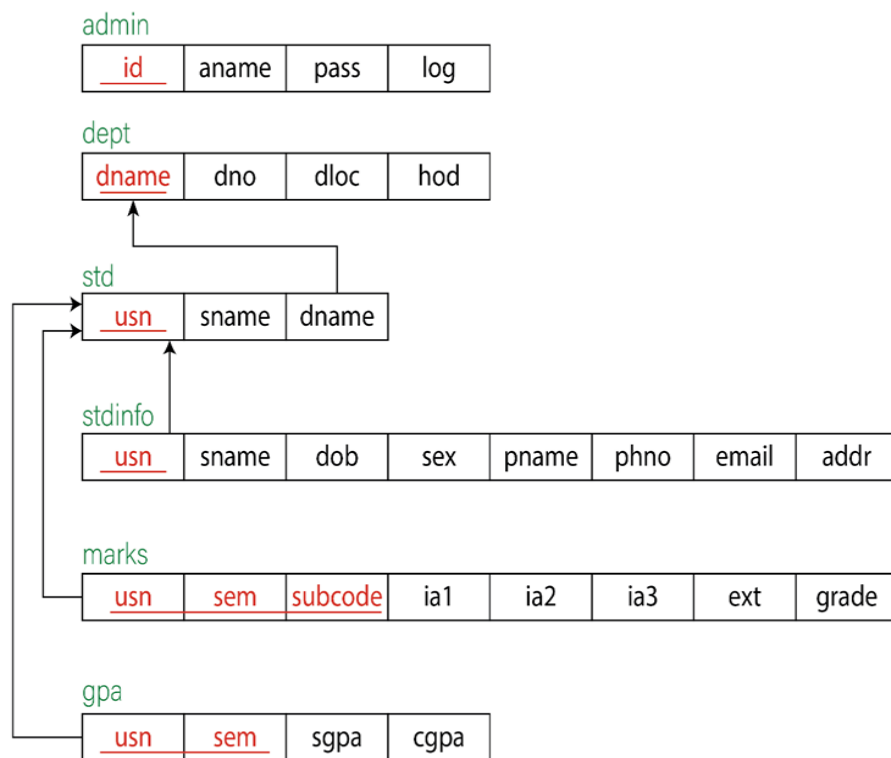
Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional.



**Fig. 3.2: ER diagram of Student Result Management System**

### 3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.



**Fig. 3.3: Relational Schema diagram**

### 3.4 DESCRIPTION OF TABLES

The database consists of six tables:

1. admin: It stores the Admin details.
  - ☐ id: Unique id done by auto increment.
  - ☐ aname: Name of the Admin.
  - ☐ Pass: Password associated with Admin to login into system.
  - ☐ log: Last modified date and time of admin.
2. dept: It stores the department details.
  - ☐ dname: name of the department.
  - ☐ dno: department number.
  - ☐ dloc: location of the department.
  - ☐ hod: head of the department.
3. std: it stores the student details.
  - ☐ usn: university seat number of the student.
  - ☐ sname: name of the student.
  - ☐ dname: foreign key referencing to dname of dept table.
4. stdinfo: it stores the personal information of the student.
  - ☐ usn: foreign key referencing usn of std table.
  - ☐ sname: name of the student.
  - ☐ dob: date of birth of student.
  - ☐ sex: gender of the student.
  - ☐ pname: name of the parent.
  - ☐ phone: student's phone number.
  - ☐ email: student's email address.
  - ☐ addr: student's residential address.
5. marks: It stores the student marks.
  - ☐ usn: foreign key referencing usn of std table.
  - ☐ sem: semester of subject marks being entered.
  - ☐ subcode: subject code of marks being entered.
  - ☐ ia1: ia1 marks of student in respective subject and semester.

- ☐ ia2: ia2 marks of student in respective subject and semester.
  - ☐ ia3: ia3 marks of student in respective subject and semester.
  - ☐ ext: external marks of student in respective subject and semester.
  - ☐ grade: grade obtained by student in respective subject and semester.
6. gpa: It stores the students sgpa/cgpa details.
- ☐ usn: foreign key referencing usn of std table.
  - ☐ sem: semester of sgpa/cgpa being entered.
  - ☐ sgpa: sgpa of the respective semester.
  - ☐ cgpa: cgpa of 8<sup>th</sup> semester

## Chapter 4

## IMPLEMENTATION

### 4.1 MODULES AND THEIR ROLES

#### 4.1.1 Login: Admin login.

```
<?php
session_start();
include("connection.php");
include("function.php");
include("style.php");
if($_SERVER['REQUEST_METHOD']=="POST")
{
$aname=$_POST['aname'];
$pass=$_POST['pass'];
if(!empty($aname) && !empty($pass) && !is_numeric($aname))
{
$query="select * from admin where aname ='$aname'";
$result=mysqli_query($con,$query);
if($result)
{
if($result && mysqli_num_rows($result)>0)
{
$data = mysqli_fetch_assoc($result);
if($data['pass']===$pass)
{
$_SESSION['aname']=$data['aname'];
header("Location: index.php");
die;
}
else
{
echo "invalid admin name or password";
}
}
}
```



```
else
{
echo "invalid admin name or password";
}
}
}else
{
echo "enter valid information!!";
}
}
?>
```

#### **4.1.2 Add Marks: add subject marks**

```
<?php
session_start();
include("connection.php");
include("function.php");
include("style.php");
$udata=check_login($con);
if($_SERVER['REQUEST_METHOD']=="POST")
{
$usn=$_POST['usn'];
$sem=$_POST['sem'];
$subcode=$_POST['subcode'];
$ia1=$_POST['ia1'];
$ia2=$_POST['ia2'];
$ia3=$_POST['ia3'];
$ext=$_POST['ext'];
$grade=$_POST['grade'];
$query="insert into marks(usn,sem,subcode,ia1,ia2,ia3,ext,grade)
values('$usn','$sem','$subcode','$ia1','$ia2','$ia3','$ext','$grade)";
if(mysqli_query($con,$query))
{
echo "inserted into marks table. ";
```

```
}else
{
echo mysqli_error($con);
}
sleep(1);
header("location: index.php");
die;
}
?>
```

#### 4.1.3 Delete stud: delete student records

```
<?php
session_start();
include("connection.php");
include("function.php");
include("style.php");
$data=check_login($con);
if($_SERVER['REQUEST_METHOD']=="POST")
{
$usn=$_POST['usn'];
$query="delete from stud where usn='$usn'";
if(mysqli_query($con,$query))
{
print("records of student '$usn' are deleted from gpa table!!! \r\n");
}else
{
echo mysqli_error($con);
}
$query1="delete from marks where usn='$usn'";
if(mysqli_query($con,$query))
{
print("records of student '$usn' are deleted from marks table!!! \r\n");
}else
{
echo mysqli_error($con);
}
}
```

```
$query="delete from stdinfo where usn='$usn'";
if(mysqli_query($con,$query))
{
print("records of student '$usn' are deleted from stdinfo table!!! \r\n");
}else
{
echo mysqli_error($con);
}
$query="delete from gpa where usn='$usn'";
if(mysqli_query($con,$query))
{
print("records of student '$usn' are deleted from stud table!!! \r\n");
}else
{
echo mysqli_error($con);
}
sleep(1);
header("location: index.php");
die;
}
?>
```

#### **4.1.4 Edit gpa: edit student sgpa/cgpa**

```
<?php
session_start();
include("connection.php");
include("function.php");
include("style.php");
$data=check_login($con);
if($_SERVER['REQUEST_METHOD']=="POST")
{
$usn=$_POST['usn'];
$sem=$_POST['sem'];
$sgpa=$_POST['sgpa'];
$cgpa=$_POST['cgpa'];
if(!empty($usn) && !empty($sem))
{
```

```
if(!empty($sgpa))
{
$query="update gpa set sgpa='$sgpa' where usn='$usn' and sem='$sem'";
if(mysqli_query($con,$query))
{
print("sgpa updated. \n");
}
else
{
echo mysqli_error($con);
}
}
if(!empty($cgpa))
{
$query="update gpa set cgpa='$cgpa' where usn='$usn' and sem=8";
if(mysqli_query($con,$query))
{
print("cgpa updated. \n");
}
else
{
echo mysqli_error($con);
}
}
else{
echo "enter valid usn/sem";
}
sleep(1);
header("location: index.php");
die;
}
?>
```

## 4.2 TRIGGERS

Trigger is used in Student result management system to turn all lowercase alphabets entered to name field of student and parent to uppercase before insertion.

```
CREATE DEFINER=`root`@`localhost`  
TRIGGER `uppercase`  
BEFORE INSERT ON `stdinfo`  
FOR EACH ROW  
BEGIN  
SET NEW.sname=UPPER(NEW.sname);  
SET NEW.pname=UPPER(NEW.pname);  
END ;
```

## 4.3 STORED PROCEDURE

Stored procedure is used select all student details from stdinfo table.

```
CREATE DEFINER=`root`@`localhost`  
PROCEDURE allstud  
BEGIN  
SELECT * FROM stdinfo;  
END
```

## Chapter 5

### TESTING

#### 5.1 SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

Sl no	TEST CASES	EXPECTED RESULT	OBTAINED RESULT	REMARKS
1	Register account for admin	If success, redirect to home page	Redirected to home page	pass
2	Student requesting to view result	Redirect to student result page	Redirected to student result page	pass
3	Enter invalid admin name to login/signup	If invalid, Display error of particular field	Error Message	pass
4	Enter invalid password to login	If invalid, Display error of particular field	Error message	pass

#### 5.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

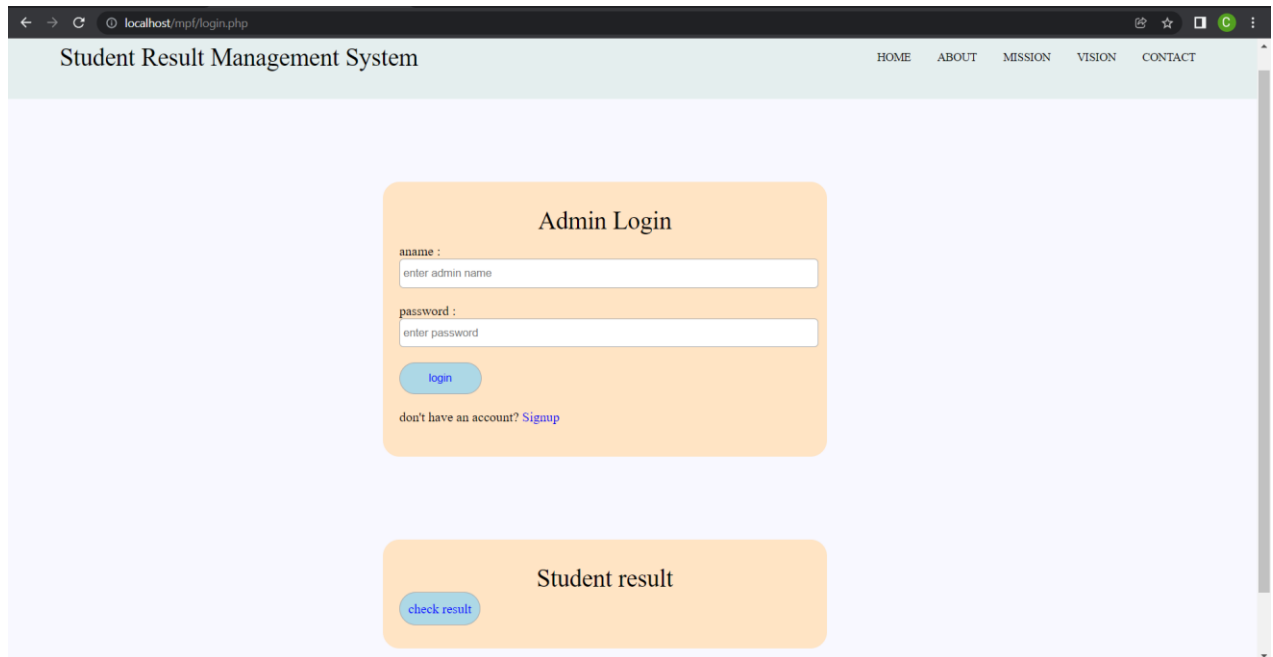
The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

## Chapter 6

### SNAPSHOT

This chapter consists of working screenshots of the project.

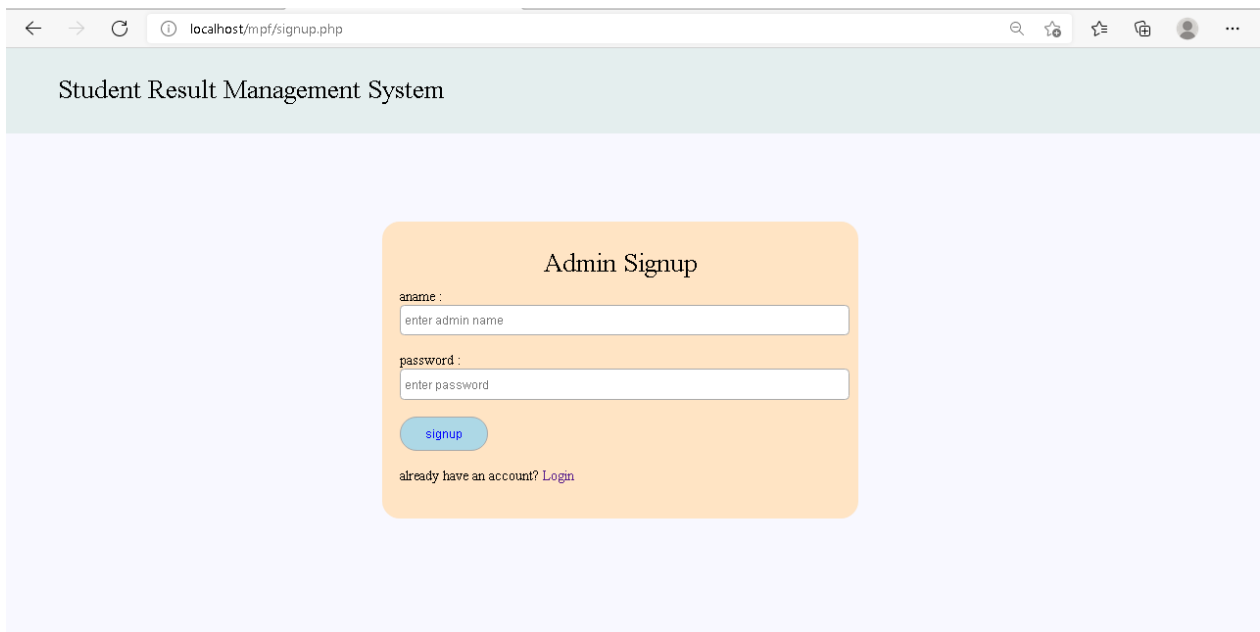
#### 6.1 LOGIN PAGE



The screenshot shows a web browser window with the URL `localhost/mpl/login.php`. The page title is "Student Result Management System". The navigation bar includes links for HOME, ABOUT, MISSION, VISION, and CONTACT. The main content area features an "Admin Login" form with two input fields: "aname : enter admin name" and "password : enter password". Below the fields is a blue "login" button. At the bottom of the form, there is a link: "don't have an account? [Signup](#)". Below the login form is a "Student result" section with a blue "check result" button.

**Fig 6.1: Login page**

#### 6.2 REGISTRATION PAGE



The screenshot shows a web browser window with the URL `localhost/mpl/signup.php`. The page title is "Student Result Management System". The navigation bar includes links for HOME, ABOUT, MISSION, VISION, and CONTACT. The main content area features an "Admin Signup" form with two input fields: "aname : enter admin name" and "password : enter password". Below the fields is a blue "signup" button. At the bottom of the form, there is a link: "already have an account? [Login](#)".

**Fig 6.2: Registration page**

## 6.3 HOME PAGE

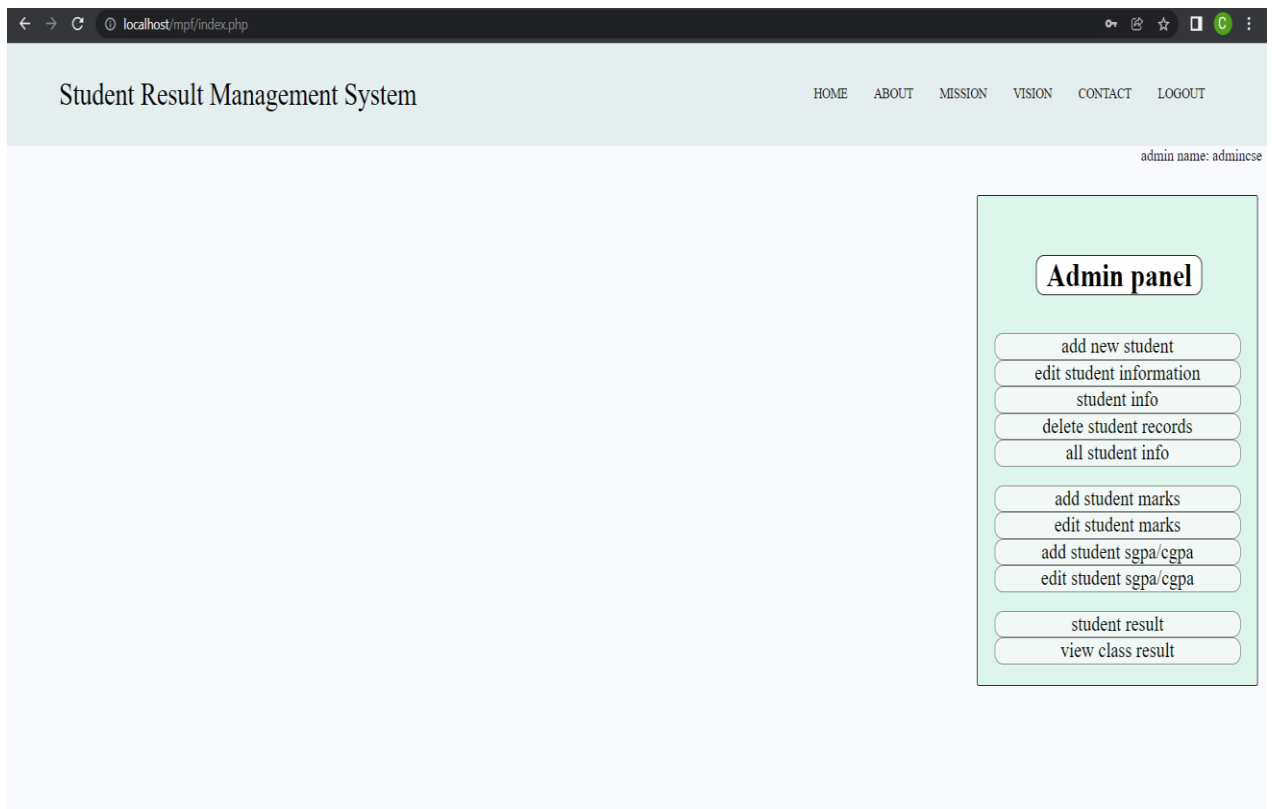


Fig 6.3: Home Page

## 6.4 ADD MARKS PAGE

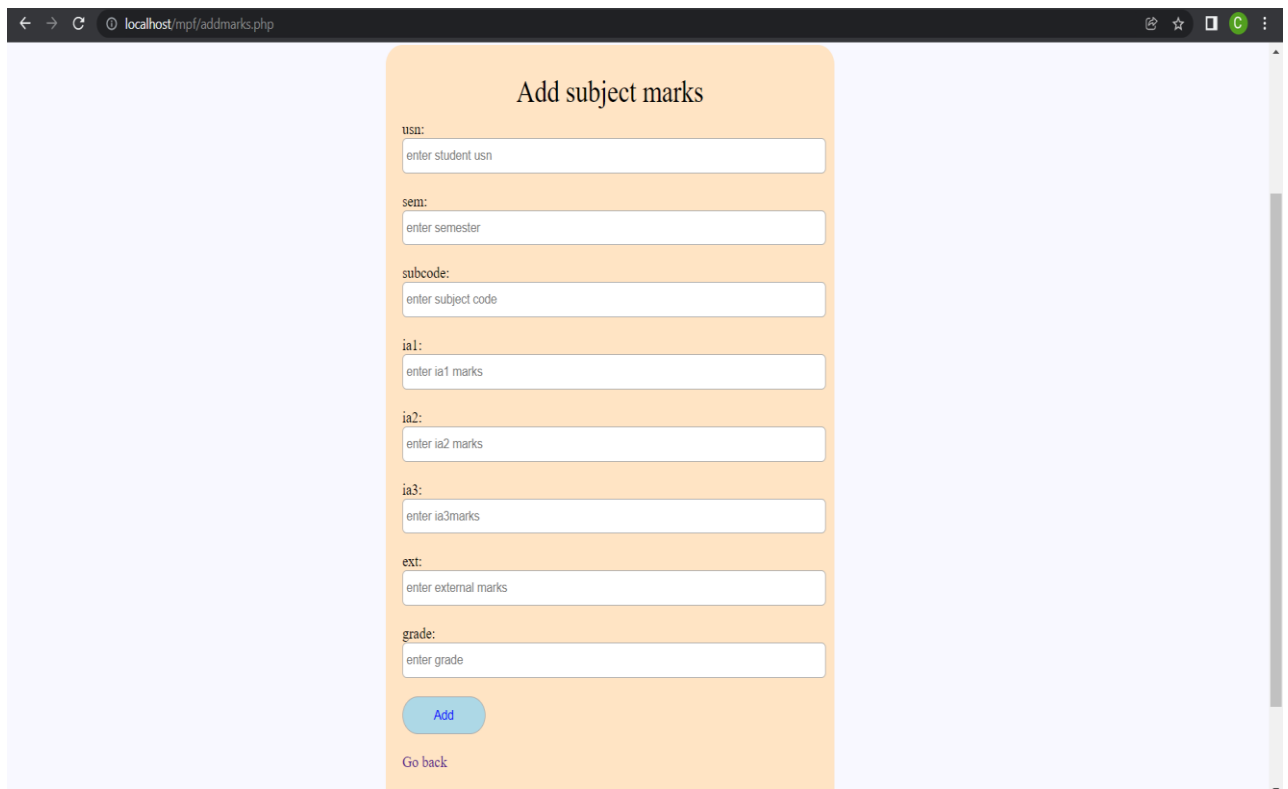
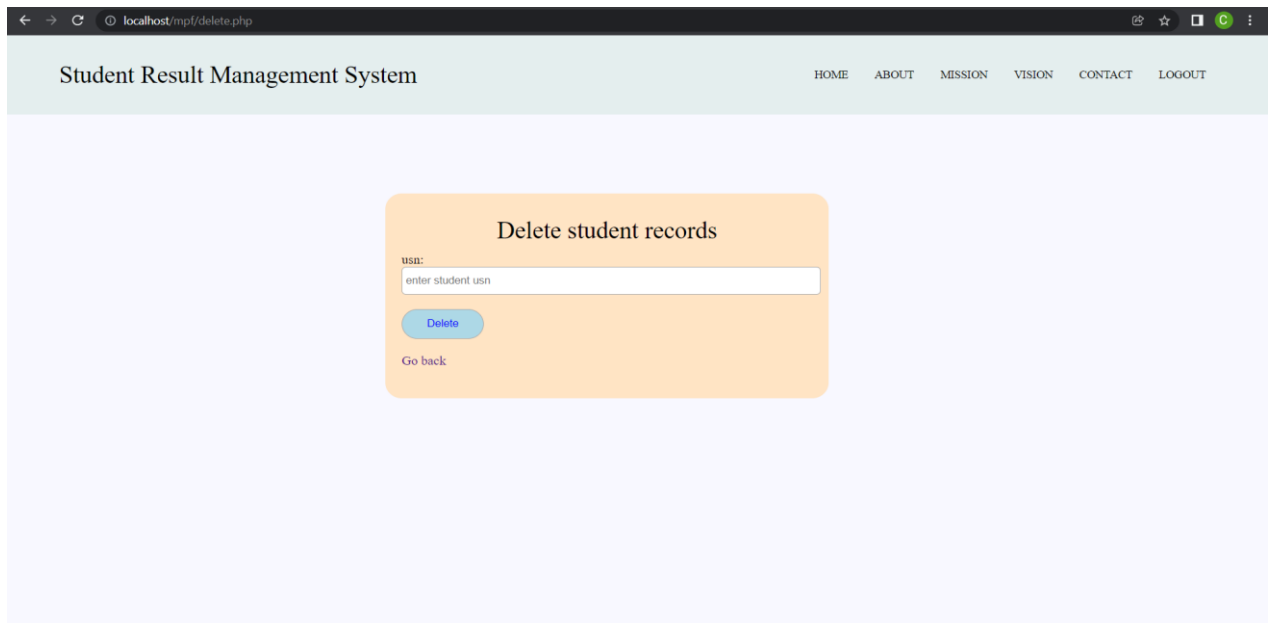


Fig 6.4: Add marks page



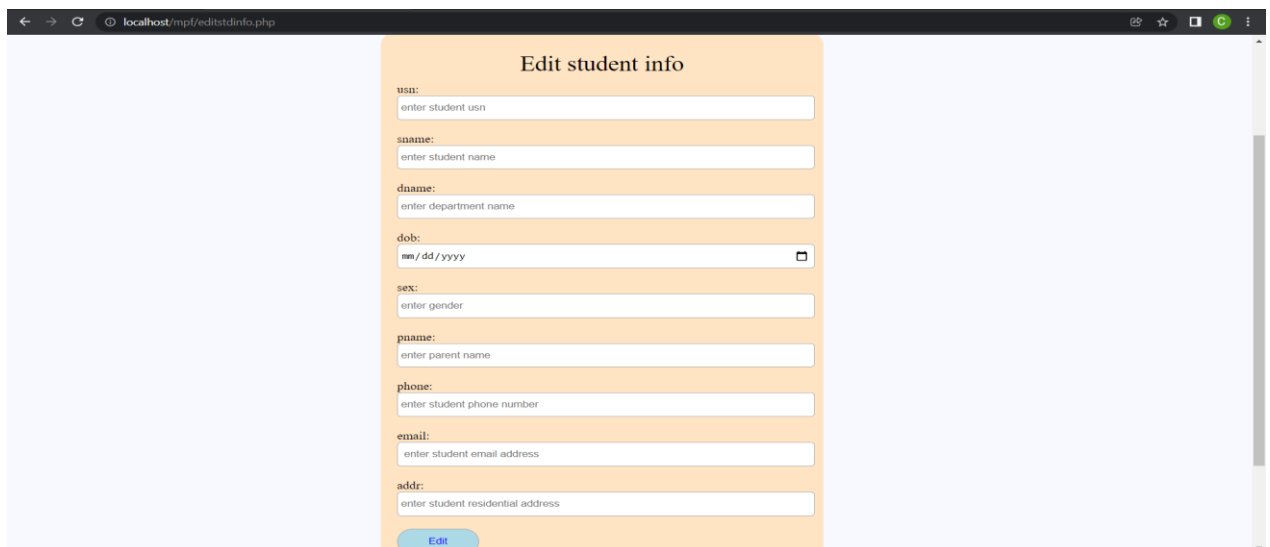
## 6.5 DELETE STUDENT RECORDS PAGE



The screenshot shows a web browser window with the URL `localhost/mpf/delete.php`. The page title is "Student Result Management System". The navigation bar includes links for HOME, ABOUT, MISSION, VISION, CONTACT, and LOGOUT. The main content area features a light orange box titled "Delete student records". Inside this box, there is a label "usr:" followed by a text input field containing the placeholder text "enter student usr". Below the input field is a blue button labeled "Delete". At the bottom of the box is a purple link labeled "Go back".

Fig 6.5: Delete student records page

## 6.6 EDIT STUDENT INFORMATION PAGE



The screenshot shows a web browser window with the URL `localhost/mpf/editstdinfo.php`. The page title is "Student Result Management System". The navigation bar includes links for HOME, ABOUT, MISSION, VISION, CONTACT, and LOGOUT. The main content area features a light orange box titled "Edit student info". Inside this box, there are several form fields: "usr:" with a text input field (placeholder: "enter student usr"), "sname:" with a text input field (placeholder: "enter student name"), "dname:" with a text input field (placeholder: "enter department name"), "dob:" with a date input field (placeholder: "mm / dd / yyyy"), "sex:" with a text input field (placeholder: "enter gender"), "pname:" with a text input field (placeholder: "enter parent name"), "phone:" with a text input field (placeholder: "enter student phone number"), "email:" with a text input field (placeholder: "enter student email address"), and "addr:" with a text input field (placeholder: "enter student residential address"). At the bottom of the box is a blue button labeled "Edit".

Fig 6.6: Edit student information page

## 6.7 EDIT STUDENT SGPA/CGPA PAGE

Fig 6.7: Edit student sgpa/cgpa Page

## 6.8 STUDENT RESULT PAGE

USN:1kg19me124 result							
usn	sem	subcode	ia1	ia2	ia3	ext	grade
1kg19me124	5	18ME51	24	26	27	68	9
1kg19me124	5	18ME52	28	22	15	68	7
1kg19me124	5	18ME53	27	24	21	59	7
1kg19me124	5	18ME54	27	25	27	89	9

USN:1kg19me124 SGPA/CGPA			
usn	sem	sgpa	cgpa
1kg19me124	5	9.00	0.00

Fig 6.8: Student result Page

## Chapter 7

### CONCLUSION

The Website Student result management system records many students records and needs online assistance to manage student's record. The website is user-friendly This is developed using HTML5, CSS, PHP and SQL.

The goals achieved by this project are as follows: admin can register an account and will be able to add new student details, edit student details, delete student details as well as admin can add, edit and delete student's subject marks. On the other hand, students can view their results, and can also compare it with class results.

## Chapter 8

### **FUTURE ENHANCEMENTS**

Future upgrades to this project will implement:

Admins will be able to upload documents, Notifications from institutions can be updated. User interface and user experience can be improved timely, based on the current requirements of users at that instant of time. Can implement a chat box where students can interact with teachers individually. Admins Will be able to contact student or parents directly through website via chat or a voice call. And many things can be improved, enhanced or upgraded in the future.

## REFERENCES

- [1] Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.
- [2] PHP The Complete resference – Steven Holzner
- [3] Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.
- [4] Database Principles , Fundamentals of Design, Implementation and Management, Cengage Learning
- [5] <https://www.w3schools.com>
- [6] <https://www.youtube.com>
- [7] <https://www.google.com>