

## CHAPTER 1

# INTRODUCTION

### 1.1 Background

The title of the project is “SJCIT Management System”. This project will handle whole the activities of our college. It provides facilities to keep the records of Student, Teacher, Department and subject with all their required details along with all required operation handling. It has facilities to generate test result given by student for particular subject with respective department.

In this new SJCIT Management System, much of manual processing repetition work has been eliminated data redundancy, save time and storage area..

It is easier to update the student’s details, teacher’s details, faculties details, delete the same using this project. The IA marks of every internals can be entered where the program itself calculates the average of the three IA marks and give the Total which is much easier for the teachers and faculties.

### 1.2 Introduction to Database Management System:

- Data : Data is the information that has been translated into a form that is efficient for processing.
- Database:
  - A database is logically coherent collection of data with some inherent meaning.
  - Database is designed, build and populated with data for specific purpose.
  - Database represents the real world object.
- Database management System (DBMS):

The DBMS is a general purpose software system that facilitates the process of defining, constructing, manipulating and sharing database among various user and applications.

#### Advantages of using DBMS approach:

- Restricting unauthorized access.
- Controlling Redundancy.
- Providing persistent storage for Program objects.
- Providing Storage Structure for efficient Query Processing.

- Providing multiple user interfaces.
- Representing complex relationships among the data ➤ Enforcing Integrity Constraints.

### 1.3 Introduction to SQL

- Structured Query Language (SQL) is comprehensive database language. Hence it has both DDL and DML.
- Data Definition Language (DDL): We can use CREATE, INSERT, DELETE MODIFY statements. We cannot manipulate the data in the table.
- Data Manipulation Language (DML): We can manipulate the data in the record using UPDATE and ALTER statements.
- SQL has several different techniques for writing programs in various Programming languages that include SQL statements to access one or more database.
- SQL has transaction control commands. These are used to specify units of database processing for concurrency control and recovery purpose.

#### Number of Modules

A module is a bounded contiguous group of statements having a single name and that can be treated as a unit. In other words, a single block in a pile of blocks.

#### Guidelines for Modularity

- Make sure modules perform a single task, have a single entry point, and have a single exit point.
- Isolate input-output (I-O) routines into a small number of standard modules that can be shared system-wide.
- Isolate system-dependent functions (e.g., getting date or time) in the application to ease possible future conversions to other computer platforms or to accommodate future operating system revisions.

A module is a bounded contiguous group of statements having a single name and that can be treated as a unit. In other words, a single block in a pile of blocks.

The system after careful analysis has been identified to be presented with the following modules:

**SMS** (SJCIT MANAGEMENT SYSTEM) makes management to get the most updated information always by avoiding manual accounting process. This system has the following functional divisions.

**STUDENT** : The Student module contains the information of student. It consists of student name, gender, date-of-birth, address, phone-no, email-id and note.

**TEACHER** : The Teacher module contains the information of the teacher. It consists of teacher name, gender, date-of-birth, address, degree, salary, marries or not, phone-no, email-id and note.

**FACULTY** : The Faculty module contains the information of the faculties. It consists of faculties name and note.

**SUBJECT** : The Subject module contains the information of the subjects. It consists of subject name, semester and note.

**STUDENT SCORE** : The Student Score module contains the information of the scores of student. It consists of IA marks of 1<sup>st</sup> Internal, IA marks of 2<sup>nd</sup> Internal, IA marks of 3<sup>rd</sup> Internal.

**USER** : The User Module contains the information of the users. It consists of username and note.

## CHAPTER 2

# LITERATURE SURVEY

### 2.1 Existing System Details and Problems

Lack of immediate retrievals: In the conventional system, information is distributed across several files. This might also lead to data redundancy with repetition of the same information in various files. In the event of a complex or nested query, the search has to scan several files, thus making procurement of requested query results very cumbersome.

Maintenance of Accuracy and Reliability issues: With redundancy comes consistency issues as the update of information in a single record should be echoed in all records containing the same information. Also atomicity issues ie, completion of a transaction in totality or nothing at all; has to be maintained. This is difficult in a multi-file system.

Lack of prompt update: Updates associated with a record in a file is to be reflected in all records wherein the particular record is present. This concurrent update poses the problem of time lag. Errors in commit operation to some particular files cause the grave issue of data inconsistency.

Error prone manual calculation: Manual calculations are error prone and relatively immensely time consuming, in spite of which they may result in generation of incorrect information. Verification is another overhead, which can be saved through efficient design and implementation.

### 2.2 Alternative Solutions

Improved manual System: One of the alternative solutions is the improvement of the manual system. Anything, which can be done by using automated methods, can be done manually. But the key question is how to perform a task manually in a sound and optimal manner. Following are some suggestions, which can be useful in manual system.

A more sophisticated register maintenance dedicated to each subsystem can be maintained with centralised control and evolution. Adequate dedicated staff may be maintained so that updates are

made at very moment at the same time. Proper provision for proper work should be put into place. This would require considerable extra work force.

**Batch system:** An alternative solution can be used of computer based batch system for maintaining the information regarding personal details. A batch system refers to a system in which data is processed in a periodical basis. The batch system is able to achieve most of the goals and sub goals. But a batch system data is processed in sequential basis. Therefore batch system is not suggested.

**Online system:** This system (HRDBMS) provides online storage, update and retrieval facility. This system promises very less or no paper work and also provides help to administrate and company personal. In the system everything is stored electronically so constants of paper work are eliminated and multiple files usage is not required. Information can be retrieved very easily without scanning multiple registers this system is been discussed here.

### **2.3 Advantages of Proposed System**

- Very simple and easy to implement
- Security of data
- Ensure data accuracy
- Administrator discretion and control over the entire system.
- Reduces the damages of machine
- Minimizes manual data entry
- Greater efficiency
- User friendly and interactive
- Less time consuming

## **FEASIBILITY STUDY**

Feasibility study is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. It involves preliminary investigation of the project and examines whether the designed system will be useful to the organization. Months or years of effort, thousand for millions of money and untold professional embarrassment can be averted if an in-conceived system is recognized early in the definition phase.

The different types of feasibility are: Technical feasibility, Operational feasibility, Economical feasibility.

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

### **Technical Feasibility**

Technical Feasibility centers on the existing computer system hardware, software, etc. and to some extent how it can support the proposed addition. This involves financial considerations to accommodate technical enhancements. Technical support is also a reason for the success of the project. The techniques needed for the system should be available and it must be reasonable to use. Technical Feasibility is mainly concerned with the study of function, performance, and constraints that may affect the ability to achieve the system. By conducting an efficient technical feasibility we need to ensure that the project works to solve the existing problem area.

Since the project is designed with ASP.NET with C# as Front end and SQL Server 2000 as Back end, it is easy to install in all the systems wherever needed. It is more efficient, easy and user-friendly to understand by almost everyone. Huge amount of data can be handled efficiently using SQL Server as back end. Hence this project has good technical feasibility

### **Operational Feasibility**

People are inherently instant to change and computers have been known to facilitate change. An estimate should be made to how strong a reaction the user staff is likely to have towards the development of the computerized system.

The staff is accustomed to computerized systems. These kinds of systems are becoming more common day by day for evaluation of the software engineers. Hence, this system is

operationally feasible. As this system is technically, economically and operationally feasible, this system is judged feasible

## **Economical Feasibility**

The role of interface design is to reconcile the differences that prevail among the software engineer's design model, the designed system meet the end user requirement with economical way at minimal cost within the affordable price by encouraging more of proposed system. Economic feasibility is concerned with comparing the development cost with the income/benefit derived from the developed system. In this we need to derive how this project will help the management to take effective decisions. Economic Feasibility is mainly concerned with the cost incurred in the implementation of the software. Since this project is developed using PHP and MYSQL Server which is more commonly available and even the cost involved in the installation process is not high.

**Similarly it is easy to recruit persons for operating the software since almost all the people are aware of PHP and MYSQL. Even if we want to train the persons in these area the cost involved in training is also very less. Hence this project has good economic feasibility.**

## **EXISTING SYSTEM:**

Under existing system, teachers do not have the facility to check their student's internals marks online. Students do not have the facility to check their internals marks online and understands where they have make their mistakes. Students need to go to teacher's chamber to collect the internals marks. Students do not have the facility to read study materials online. Placement department have to issue regular news update for candidate registration to take part in the placement process.

## **PROPOSED SYSTEM:**

In this new SJCIT Management System much of manual processing repetition work has been eliminated data redundancy, save time and storage area. Here the students details can easily be updated or deleted by the authorized user. The teachers can easily add the internals marks of the students where each student will be having unique student\_id. There is no need of calculation of

average of the three internals marks since the program will automatically calculates it and it will be assigned for the particular student. The updation and deletion operation is so easy. If the authorized user needs any information he/she can easily search the information where the search portal is provided for each table. We have provided notes where we can add some notes regarding to the performance.

## CHAPTER 3

# SYSTEM SPECIFICATION

### 3.1 Hardware Specification

Processor	:	Intel Pentium V
Clock speed	:	500 MHZ
System bus	:	64bits
RAM	:	4 GB of RAM
HDD	:	40 GB or higher
Monitor	:	SVGA COLOR
Keyboard	:	108 keys
Mouse	:	2 button mouse

### 3.2 Software Specification

Operating System	:	WINDOWS XP/7/8/10
Front End	:	PHP (Visual Studio)
Back End	:	My SQL
Server	:	XAMPP Control panel V3.2.1

## Technology USED

### PHP

PHP is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document.

PHP source code is compiled on-the-fly to an internal format that can be executed by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the webpage is accessed, PHP scripts can also be deployed in executable format using a PHP compiler.

PHP is one of the most popular server side scripting languages running today. It is used for creating dynamic webpages that interact with the user offering customized information. PHP offers many advantages; it is fast, stable, secure, easy to use and open source (free).

PHP code is inserted directly into the HTML that makes up a website. When a visitor comes to the website, the code is executed. Because PHP is a server side technology, the user does not need any special browser or plug-ins to see the PHP in action.

Another key advantage of PHP is its connective abilities. PHP uses a modular system of extensions to interface with a variety of libraries such as graphics, XML, encryption, etc. In addition, programmers can extend PHP by writing their own extensions and compiling them into the executable or they can create their own executable and load it using PHP's dynamic loading mechanism. A huge advantage that PHP offers is its community. Since PHP is an open source project, the PHP community is willing to share. If you're looking for a particular script, chances are another user has already created something similar. Check within the PHP community for availability. Likewise, if you have created a function that others might enjoy, be sure to post the code for others.

A PHP scripting block always starts with <? php and ends with ?>. A PHP scripting block can be placed anywhere in the document.

On servers with shorthand support enabled you can start a scripting block with <? and end with ?>. For maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

## MY SQL

MySQL is an [relational database management system](#)(RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](#)'s daughter and "SQL", the abbreviation for [Structured Query Language](#). The MySQL development project has made its [source code](#) available under the terms of the [GNU General Public License](#), as well as under a variety of [proprietary](#) agreements. MySQL was owned and sponsored by a single [for-profit](#) firm,

the [Swedish](#) company [MySQL AB](#), now owned by [Oracle Corporation](#). For proprietary use, several paid editions are available, and offer additional functionality.

MySQL was created by a Swedish company, [MySQL AB](#), founded by [David Axmark](#), Allan Larsson and [Michael "Monty" Widenius](#). Original development of MySQL by Widenius and Axmark began in 1994.<sup>[27]</sup> The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from [mSQL](#) based on the low-level language [ISAM](#), which the creators considered too slow and inflexible. They created a new [SQL](#) interface, while keeping the same [API](#) as mSQL

MySQL is written in [C](#) and [C++](#). Its SQL parser is written in [yacc](#), but it uses a home-brewed [lexical analyzer](#).

Mysql dump is a logical backup tool included with both community and enterprise editions of MySQL. It supports backing up from all storage engines. MySQL Enterprise Backup is a hot backup utility included as part of the MySQL Enterprise subscription from Oracle, offering native InnoDB hot backup, as well as backup for other storage engines.

Xtra Backup is an open-source MySQL hot backup software program. Features include hot, non-locking backups for InnoDB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of I/O operations per second, etc.

MySQL Fabric is an integrated system for managing a collection of MySQL servers, and a [framework](#) on top of which high availability and database sharding is built. MySQL Fabric is open-source, and supports procedure execution in the presence of failure, providing an execution model usually called *resilient execution*.

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

## CHAPTER 4

# DESIGN

### 4.1 SYSTEM DESIGN

System Design is the most creative and challenging phase in the system life cycle. Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. System design is a solution how to approach the creation of a new system. System design transforms a logic representation of what is required to do into the physical specification. The specification is converted into physical reality during development.

#### Logical design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volume's, frequencies etc.
- Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

#### Physical design

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design Walk through.

- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits , costs , conversion date and system constraints

## Design/Specification activities

- Concept formulation.
- Problem understanding.
- High level requirements proposals.
- Feasibility study.
- Requirements engineering.
- Architectural design.

## Input design

- Authorized User enter his/her user id and password for login to authenticate in this system
- User Enters the Student Details.

While registration information of the Student such as

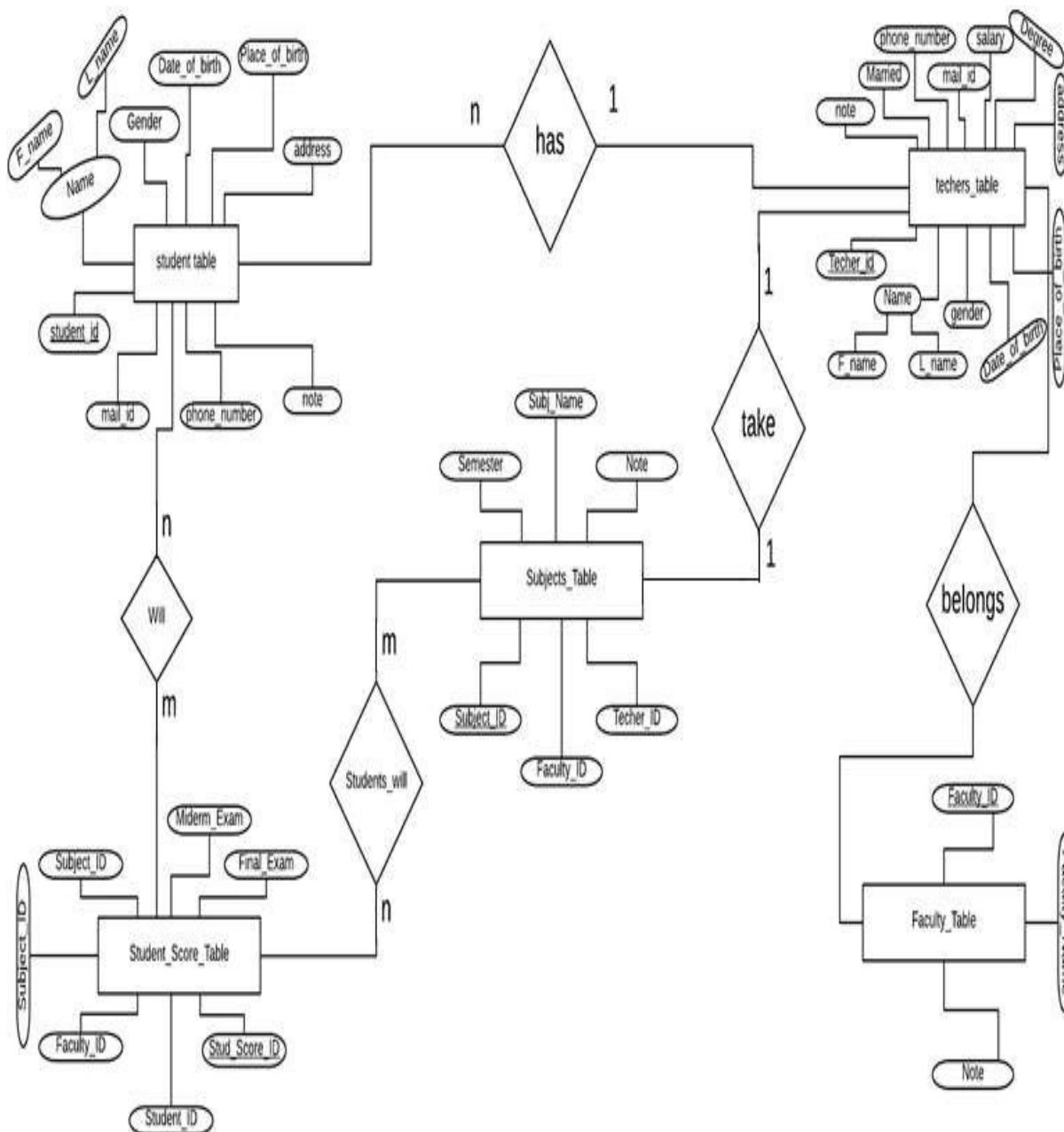
1. Student id
2. Student f\_name
3. Student l\_name
4. Address Information of Student
5. Date-Of-Birth etc...

- Authorized User can create the various Student details in this website.
- Authorized User can create the various Teachers details in this website
- He/she can create Faculties details in this website.
- He/she can enter student IA marks also.
- He/she can also enter the details of new users.

## **Output design**

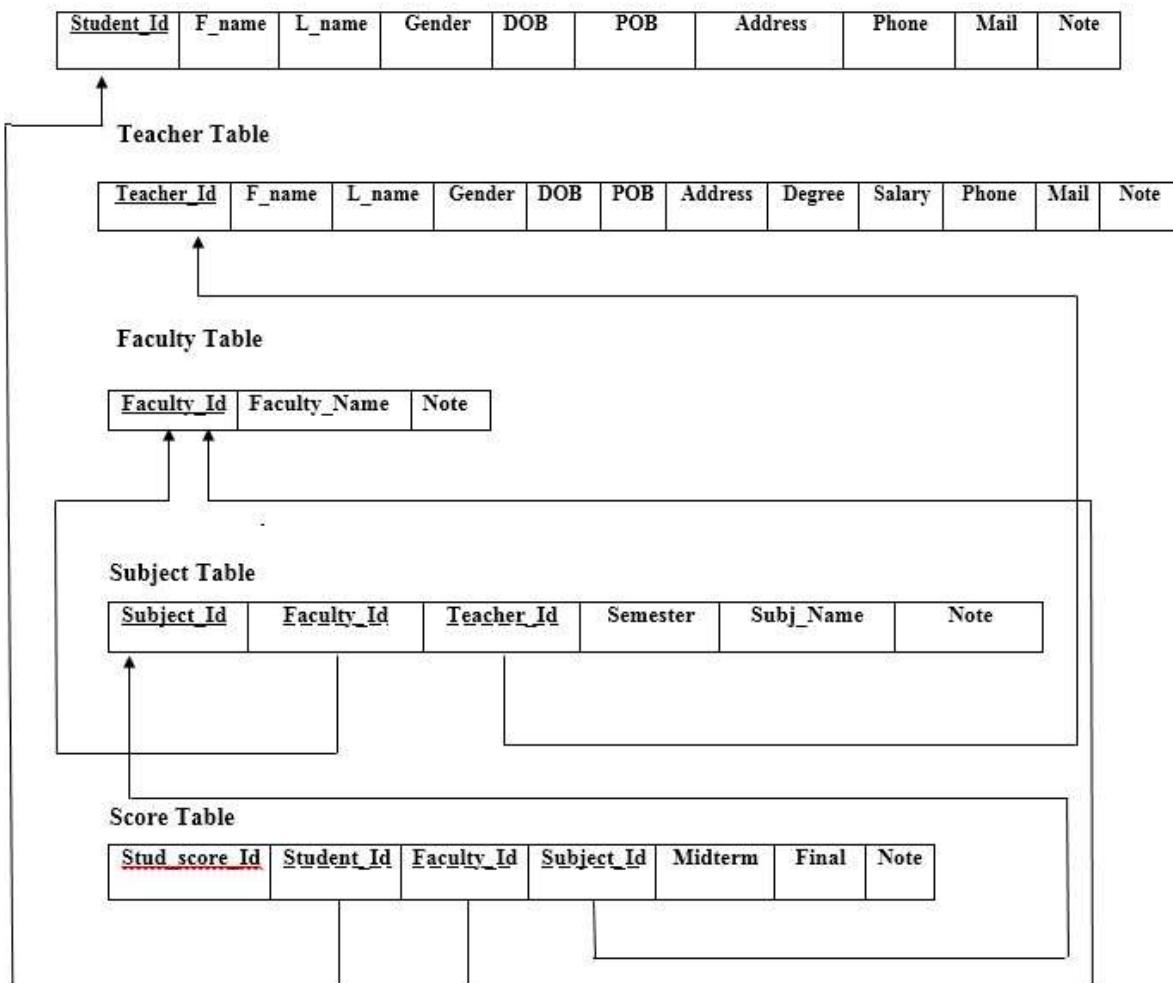
- Authorized User can have his own home page regarding the information about the faculties, staff and student .
- Authorized User get all Students and staff and course details.
- The registered user's data can be stored in centralized database through the system user interface.
- Various types of information can be displayed to the users like teachers information, student information, faculties information, student IA marks etc.....
- The Authorized User can search for the Student information, Teacher information, Faculty information.
- Profile can be updated by the users individually.
- Searching technique which is one of the most unique technique of this project provide the user to easily fetch the information of the students.
- Authorized User can also add another user.
- The Authorized User has all the authority to update, delete the information of the students, teachers, faculties.

#### 4.1 E-R diagram



## 4.2 Schema diagram

## Student table



## CHAPTER 5

# IMPLEMENTATION

### 5.1 Implementation with Screenshot

#### Methodology

This project is modularized as the following:

- 1. Student Entity**
- 2. Teacher entity**
- 3. Faculty's entity**
- 4. Subject entity**
- 5. Student Score entity**

- 1.** **Student entity:** The Student Record Storage and Retrieved by this Module. This Module include student record Information Insert, View, Delete , Update By User friendly Approach.
- 2.** **Teacher entity:** The Teacher Entry and View Teacher Detail by this Module. This includes Teacher Information Storage and maintaining their information. Form validation is already Done by this project.
- 3.** **Faculty's entity:** The common and fixed information related to Department. We can categories each teacher, subject, student and their score. Then Attached to Department to identify or make Search easier.
- 4.** **Subject entity:** The subject entity views the what are subject are there and how that can be allocated for each faculty and which semester and what is the subject name that information should be in this module.
- 5.** **Student Score entity:** By this Module we can insert score for every student i.e. IA1, IA2, IA3. One more Facility provided by this Web Based SJCIT Management System in php is we can also give student their notice related to score.

## 5.1 Code:

**Student\_entry :**

```
<?php
$Id=""; $opr="";
if(isset($_GET['opr']))
    $opr=$_GET['opr'];

if(isset($_GET['rs_id']))
    $id=$_GET['rs_id'];
//-----add data-----
if(isset($_POST['btn_sub'])){
    $f_name=$_POST['fnametxt'];
    $l_name=$_POST['lnametxt'];
    $gender=$_POST['gender'];
    $dob=$_POST['yy']."/".$_POST['mm']."/".$_POST['dd'];
    $pob=$_POST['pobtxt'];
    $addr=$_POST['addrtxt'];
    $phone=$_POST['phonetxt'];
    $mail=$_POST['emailtxt'];
    $note=$_POST['notetxt'];

$sql_ins=mysql_query("INSERT INTO stu_tbl
VALUES(
NULL,
'$f_name',
'$l_name' ,
'$gender',
'$dob',
'$pob',
'$addr',
'$phone',
'$mail',
'$note'
)
");

if($sql_ins==true)
```

```

$msg="1 Row Inserted";
else
    $msg="Insert Error:".$mysql_error();

}

//-----update data----- if(isset($_POST['btn_upd'])){
    $f_name=$_POST['fnametxt'];
    $l_name=$_POST['lnametxt'];
    $gender=$_POST['gender'];
    $dob=$_POST['yy']."/".$_POST['mm']."/".$_POST['dd'];
    $pob=$_POST['pobtxt'];
    $addr=$_POST['addrtxt'];
    $phone=$_POST['phonetxt'];
    $mail=$_POST['emailtxt'];
    $note=$_POST['notetxt'];

    $sql_update=mysql_query("UPDATE stu_tbl SET
                                f_name='$f_name',
                                l_name='$l_name' ,                               gender='$gender',
                                dob='$dob',
                                pob='$pob',                               address='$addr',
                                phone='$phone',
                                email='$mail',                           note='$note'
                                WHERE
                                stu_id=$id
                                ");
    if($sql_update==true)
        header("location:?tag=view_students");
    else
        $msg="Update Fail".$mysql_error();
}
?>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

---

```
<link rel="stylesheet" type="text/css" href="css/style_entry.css" />
<title>:: SJCIT MANAGEMENT SYSTEM .::</title>
</head>
<body>
<?php

if($opr=="upd") {
    $sql_upd=mysql_query("SELECT * FROM stu_tbl WHERE stu_id=$id");
    $rs_upd=mysql_fetch_array($sql_upd);      list($y,$m,$d)=explode('-', $rs_upd['dob']);
?>

<!-- for form Upadte-->

<div id="top_style">
    <div id="top_style_text">
        Students Update </div>
    <!-- end of top_style_text-->
    <div id="top_style_button">
        <form method="post">
            <a href="?tag=view_students"><input type="button" name="btn_view" title="View Students" value="Back" id="button_view" style="width:70px;" /></a>
            </form>
        </div><!-- end of top_style_button-->
    </div><!-- end of top_style-->

<div id="style_informations">
    <form method="post" >
        <div>
            <table border="0" cellpadding="4" cellspacing="0">
                <tr>
                    <td>First Name:</td>
                    <td>
```

```

        <input type="text" name="fnametxt" id="textbox" value="<?php echo
$rs_upd['f_name'];?>"/>
        </td>
    </tr>

    <tr>
        <td>Last Name:</td>
        <td>
            <input type="text" name="lnametxt" id="textbox" value="<?php echo
$rs_upd['l_name'];?>"/>
        </td>
    </tr>

    <tr>
        <td>Gender:</td>
        <td>
            <input type="radio" name="gender" value="Male" <?php
if($rs_upd['gender']=="Male") echo "checked";?>/>Male
            <input type="radio" name="gender" value="Female" <?php
if($rs_upd['gender']=="Female") echo "checked";?>/>Female
        </td>
    </tr>

    <tr>
        <td>Date Of Birth:</td>
        <td>
            <select name="yy" >
                <option>years</option>
                <?php
                    $sel="";
                    for($i=1985;$i<=2015;$i++){
                        if($i==$y){
                            $sel="selected='selected'";
                        }
                        else
                            $sel="";
                    }
                    echo "<option value='$i' $sel>$i </option>";
                }
            </select>
        </td>
    </tr>

```

```

?>

</select>
-
<select name="mm">
    <option>Month</option>
        <?php
            $sel="";
$mm=array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","NOv","Dec");
    $i=0;
    foreach($mm as $mon){
        $i++;
            if($i==$m){
                $sel=$sel="selected='selected'";
            }
            else
                $sel="";
            echo"<option value='$i' $sel> $mon</option>";
        }
    ?>
</select>
-
<select name="dd">
    <option>Date</option>
        <?php
            $sel="";
for($i=1;$i<=31;$i++){
    if($i==$d){
        $sel=$sel="selected='selected'";
    }
    else
        $sel="";
}
    ?>
    <option value=<?php echo $i ;?><?php echo $sel?>>
    <?php
if($i<10)
echo"0".$i ;
else
    echo"$i";

```

```
?>
</option>
<?php
 }?>
</select>
</td>
</tr>

<tr>
    <td>Place Of Birth:</td>
    <td>
        <input type="text"      name="pobtxt"      id="textbox"  value="<?php echo
$rs_upd['pob'];?> "/>
    </td>
</tr>
<tr>
    <td>Address:</td>
    <td>
        <textarea      name="addrtxt"      cols="22"      rows="3">      <?php      echo
$rs_upd['address'];?></textarea>
    </td>
</tr>

<tr>
    <td colspan="2">
        <input type="reset" value="Cancel" id="button-in"/>
        <input type="submit" name="btn_upd" value="Update" id="button-in" />
    </td>
</tr>
</table>
</div>

<div>
<table border="0" cellpadding="4" cellspacing="0">

<tr>
    <td>Phone:</td>
```

```
<td>
<input type="text"    name="phonetxt"    id="textbox"  value="<?php echo
$rs_upd['phone'];?>" />
</td>
</tr>

<tr>
<td>E-mail:</td>
<td>
<input type="text"    name="emailtxt"      id="textbox"  value="<?php echo
$rs_upd['email'];?> "/>
</td>
</tr>

<tr>
<td>Note:</td>
<td>
<textarea    name="notetxt"    cols="22"    rows="5"><?php      echo
$rs_upd['note'];?></textarea>
</td>
</tr>
</table>
</div>
</form>

</div><!-- end of style_informatios -->

<?php
} else
{
?>
<!-- for form Register-->

<div id="top_style">
<div id="top_style_text">
Students Entry
</div><!-- end of top_style_text-->
<div id="top_style_button">
<form method="post">
```

```
<a href="?tag=view_students"><input type="button" name="btn_view" title="View Students" value="View_Students" id="button_view" style="width:120px;" /></a>

</form>
</div><!-- end of top_style_button-->
</div><!-- end of top_style-->

<div id="style_informations">
    <form method="post" >
        <div>
            <table border="0" cellpadding="4" cellspacing="0">
                <tr>
                    <td>First Name:</td>
                    <td>
                        <input type="text" name="fnametxt" id="textbox"/>
                    </td>
                </tr>
                <tr>
                    <td>Last Name:</td>
                    <td>
                        <input type="text" name="lnametxt" id="textbox"/>
                    </td>
                </tr>
                <tr>
                    <td>Gender:</td>
                    <td>
                        <input type="radio" name="gender" value="Male" checked="checked" />Male
                        <input type="radio" name="gender" value="Female"/>Female
                    </td>
                </tr>
                <tr>
                    <td>Date Of Birth:</td>
                    <td>
                        <select name="yy" >
                            <option>Year</option>
                            <?php

```

```
for($i=1985;$i<=2015;$i++){
echo"<option value='".$i."'>".$i."</option>";
}
?>
</select>

-
<select name="mm">
<option>Month</option>
<?php

$mm=array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec");
$i=0;
foreach($mm as $mon){
    $i++;
    echo"<option value='".$i."'>".$mon."</option>";
}
?>
</select>

-
<select name="dd">
<option>Date</option>
<?php
for($i=1;$i<=31;$i++){
?>
<option value="<?php echo $i; ?>">
<?php
if($i<10)
echo"0".$i;           else
echo"$i";
?>
</option>
<?php
}?>
</select>

</td>
</tr>

<tr>
```

```
<td>Place Of Birth:</td>
<td>
<input type="text" name="pobtxt" id="textbox"/>

</td>
</tr>
<tr>
    <td>Address:</td>
    <td>
        <textarea name="addrtxt" cols="22" rows="3"></textarea>
    </td>
</tr>

<tr>
    <td colspan="2">
        <input type="reset" value="Cancel" id="button-in"/>
        <input type="submit" name="btn_sub" value="Register" id="button-in" />
    </td>
</tr>
</table>
</div>

<div>
    <table border="0" cellpadding="4" cellspacing="0">
        <tr>
            <td>Phone:</td>
            <td><input type="text" name="phonetxt" id="textbox" /></td>
        </tr>
        <tr>
            <td>E-mail:</td>
            <td><input type="text" name="emailtxt" id="textbox" /></td>
        </tr>
        <tr>
            <td>Note:</td>
            <td><textarea name="notetxt" cols="22" rows="5"></textarea></td>
        </tr>
    </table>
</div>
```

```
</form>

</div><!-- end of style_informatios -->
<?php
}
?>
</body>
</html>
```

**Subject\_entry :**

```
<?php
$id="";
$opr="";
if(isset($_GET['opr']))
    $opr=$_GET['opr'];

if(isset($_GET['rs_id']))
    $id=$_GET['rs_id'];

if(isset($_POST['btn_sub'])){
    $fa_name=$_POST['factxt'];
    $teach_name=$_POST['techtxt'];
    $semester=$_POST['semestertxt'];
    $sub_name=$_POST['subtxt'];
    $note=$_POST['notetxt'];

$sql_ins=mysql_query("INSERT INTO sub_tbl
VALUES(
NULL,
'$fa_name',
'$teach_name',
'$semester',
'$sub_name',
'$note'
)
");

}
```

```
if($sql_ins==true)
    $msg="1 Row Inserted";
else
    $msg="Insert Error:".mysql_error();

}

//-----update data-----
if(isset($_POST['btn_upd'])){
$fac_id=$_POST['factxt'];
$tea_id=$_POST['techtxt'];
$semester=$_POST['semestertxt'];
$sub_name=$_POST['subtxt'];
$note=$_POST['notetxt'];

$sql_update=mysql_query("UPDATE sub_tbl SET
                                faculties_id='$fac_id' ,
                                teacher_id='$tea_id' ,
                                semester='$semester' ,
                                sub_name='$sub_name' ,
                                note='$note'
                                WHERE sub_id=$id
                                ");
if($sql_update==true)
    header("location:?tag=view_subjects"); else
    $msg="Update Fail!...";
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>::SJCIT MANAGEMT SYSTEM::</title>
<link rel="stylesheet" type="text/css" href="css/style_entry.css" />
</head>
```

```
<body>
```

```
<?php
```

```
if($opr=="upd") {
    $sql_upd=mysql_query("SELECT * FROM sub_tbl WHERE sub_id=$id");
    $rs_upd=mysql_fetch_array($sql_upd);

?>
<div id="top_style">
    <div id="top_style_text">
        Subjects Entry
    </div><!-- end of top_style_text-->
    <div id="top_style_button">
        <form method="post">
            <a href="?tag=view_subjects" ><input type="button" name="btn_view" title="Back" value="Back" id="button_view" style="width:70px;" /></a>

        </form>
    </div><!-- end of top_style_button-->
</div><!-- end of top_style-->

<div id="style_informations">
    <form method="post">
        <div>
            <table border="0" cellpadding="5" cellspacing="0">
                <tr>
                    <td>Facuties's Name</td>
                    <td>
                        <select name="factxt" id="textbox">
                            <option>---- Facuties's Name -----</option>
                        <?php
                            $fac_name=mysql_query("SELECT * FROM faculties_tbl");

while($row=mysql_fetch_array($fac_name)){

if($row['faculties_id']==$rs_upd['faculties_id'])
```

```
                $iselect="selected";
            else
                $iselect="";
            ?>

        <option value="<?php echo $row['faculties_id'];?>" <?php echo
$iselect;?> > <?php echo $row['faculties_name'];?> </option>
        <?php
    }

    ?>
</select>
</td>
</tr>

<tr>
    <td>Teacher's Name</td>
    <td>
        <select name="techtxt" id="textbox">
            <option>---- Teachers's Name ----</option>
            <?php
                $te_name=mysql_query("SELECT * FROM teacher_tbl");
                while($row=mysql_fetch_array($te_name)){
                    if($row['teacher_id']==$rs_upd['teacher_id'])
                        $iselect="selected";
                    else
                        $iselect="";
                    ?>
                    <option value="<?php echo $row['teacher_id'];?>" <?php echo $iselect?> >
                    <?php echo $row['f_name'] ; echo " "; echo $row['l_name'];?> </option>
                }
            <?php
        }

    ?>
</select>
</td>
</tr>

<tr>
```

```

<td>Semester</td>
<td>
<input type="text" name="semestertxt" id="textbox" value="<?php echo $rs_upd['semester'];?>" />
</td>
</tr>

<tr>
<td>Subjects's name</td>
<td>
<input type="text" name="subtxt" id="textbox" value="<?php echo $rs_upd['sub_name'];?>" />
</td>
</tr>

<tr>
<td>Note</td>
<td>
<textarea name="notetxt" cols="23" rows="3"><?php echo $rs_upd['note'];?></textarea>
</td>
</tr>
<body>
<html>
```

## 5.2:Sql file :

```

-- phpMyAdmin SQL Dump
-- version 4.9.0.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Nov 17, 2019 at 08:50 AM
-- Server version: 10.4.6-MariaDB
-- PHP Version: 7.3.8

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
```

---

```
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- 
-- Database: `a`
-- 

DELIMITER $$

-- 
-- Procedures
-- 

CREATE DEFINER=`root`@`localhost` PROCEDURE `insertData` (IN `faculties_name` VARCHAR(50), IN `note` VARCHAR(100)) BEGIN
    insert into facuties_tbl(faculties_name,note) values (faculties_name,note);
END$$

DELIMITER ;

-- -----
-- 

-- Table structure for table `article_tbl`
-- 

CREATE TABLE `article_tbl` (
  `a_id` int(10) UNSIGNED NOT NULL,
  `loca_id` int(10) NOT NULL,
  `title` varchar(100) NOT NULL,
  `content` text NOT NULL,
  `status` char(10) NOT NULL,
  `note` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `article_tbl`
-- 

INSERT INTO `article_tbl` (`a_id`, `loca_id`, `title`, `content`, `status`, `note`)
VALUES
(5, 2, 'goog', 'nice topic', 'Public', 'good');

-- -----
-- 

-- Table structure for table `facuties_tbl`
-- 

CREATE TABLE `facuties_tbl` (
  `faculties_id` int(10) UNSIGNED NOT NULL,
  `faculties_name` varchar(50) NOT NULL,
  `note` varchar(100) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `faculties_tbl`
--

INSERT INTO `faculties_tbl` (`faculties_id`, `faculties_name`, `note`) VALUES
(4, 'CSE', 'Computer Science & Engg');

-- 
-- Table structure for table `location_tb`
-- 

CREATE TABLE `location_tb` (
  `loca_id` int(10) UNSIGNED NOT NULL,
  `l_name` varchar(100) NOT NULL,
  `description` text NOT NULL,
  `note` varchar(150) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `location_tb`
-- 

INSERT INTO `location_tb` (`loca_id`, `l_name`, `description`, `note`) VALUES
(2, 'doddballapur', 'oracle ', 'good');

-- 
-- Table structure for table `stu_score_tbl`
-- 

CREATE TABLE `stu_score_tbl` (
  `ss_id` int(10) UNSIGNED NOT NULL,
  `stu_id` int(10) NOT NULL,
  `faculties_id` int(10) NOT NULL,
  `sub_id` int(10) NOT NULL,
  `ia1` int(11) NOT NULL,
  `ia2` int(11) NOT NULL,
  `ia3` int(11) NOT NULL,
  `note` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `stu_score_tbl`
-- 
```

```
INSERT INTO `stu_score_tbl` (`ss_id`, `stu_id`, `faculties_id`, `sub_id`, `ia1`,  
`ia2`, `ia3`, `note`) VALUES  
(114, 1, 4, 13, 29, 30, 28, ''),  
(115, 2, 4, 14, 24, 20, 21, '');  
  
-- -----  
  
--  
-- Table structure for table `stu_tbl`  
--  
  
CREATE TABLE `stu_tbl` (  
    `stu_id` int(10) UNSIGNED NOT NULL,  
    `f_name` varchar(50) NOT NULL,  

```

```
`note` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `sub_tbl`
-- 

INSERT INTO `sub_tbl` (`sub_id`, `faculties_id`, `teacher_id`, `semester`,
`sub_name`, `note`) VALUES
(13, 4, 14, '5TH', 'M&E', ''),
(14, 4, 15, '5TH', 'CN', ''),
(15, 4, 16, '5TH', 'DBMS', ''),
(16, 4, 17, '5TH', 'ATC', ''),
(17, 4, 18, '5TH', 'J2EE', ''),
(18, 4, 19, '5TH', 'AI', ''),
(19, 4, 19, '5TH', 'AI', '');

-- -----
-- 
-- Table structure for table `teacher_tbl`
-- 

CREATE TABLE `teacher_tbl` (
  `teacher_id` int(10) UNSIGNED NOT NULL,
  `f_name` varchar(30) NOT NULL,
  `l_name` varchar(30) NOT NULL,
  `gender` char(10) NOT NULL,
  `dob` date NOT NULL,
  `pob` varchar(100) NOT NULL,
  `address` varchar(100) NOT NULL,
  `degree` varchar(50) NOT NULL,
  `salary` float NOT NULL,
  `married` char(10) NOT NULL,
  `phone` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `note` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `teacher_tbl`
-- 

INSERT INTO `teacher_tbl` (`teacher_id`, `f_name`, `l_name`, `gender`, `dob`,
`pob`, `address`, `degree`, `salary`, `married`, `phone`, `email`, `note`)
VALUES
(14, 'SRINATH', '', 'Male', '0000-00-00', '', '', '----- Select -----',
---', 0, 'Yes', '', '', ''),
(15, 'VENKATESH', '', 'Male', '0000-00-00', '', '', '----- Select -----',
---', 0, 'Yes', '', '', ''),
```

```

(16, 'PRADEEP', '', 'Male', '0000-00-00', '', '', '----- Select -----
---', 0, 'Yes', '', '', ''),
(17, 'SESHAIH', '', 'Male', '0000-00-00', '', '', '----- Select -----
---', 0, 'Yes', '', '', ''),
(18, 'VINUTHA', '', 'Male', '0000-00-00', '', '', '----- Select -----
---', 0, 'Yes', '', '', ''),
(19, 'AJAY', '', 'Male', '0000-00-00', '', '', '----- Select -----
', 0, 'Yes', '', '', '');

-----
-- Table structure for table `users_tbl`
--



CREATE TABLE `users_tbl` (
  `u_id` int(10) UNSIGNED NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(30) NOT NULL,
  `type` char(10) NOT NULL,
  `note` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `users_tbl`



INSERT INTO `users_tbl` (`u_id`, `username`, `password`, `type`, `note`) VALUES
(8, 'nikhil', 'nikhil@123', 'admin', 'creator'),
(9, 'madhu', 'madhu@123', 'admin', 'creator');

-- Triggers `users_tbl`



DELIMITER $$

CREATE TRIGGER `after_users_insert` AFTER INSERT ON `users_tbl` FOR EACH ROW
BEGIN
insert into users_tbl_backup values(NEW.u_id, NEW.username, NEW.password,
NEW.type, NEW.note);
END
$$

DELIMITER ;


-----
-- Table structure for table `users_tbl_backup`
--



CREATE TABLE `users_tbl_backup` (
  `u_id` int(10) UNSIGNED NOT NULL,

```

---

```
`username` varchar(50) NOT NULL,
`password` varchar(30) NOT NULL,
`type` char(10) NOT NULL,
`note` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `users_tbl_backup`
-- 

INSERT INTO `users_tbl_backup` (`u_id`, `username`, `password`, `type`, `note`)
VALUES
(8, 'nikhil', 'nikhil@123', 'admin', 'creator'),
(9, 'madhu', 'madhu@123', 'admin', 'creator'),
(10, 'raghu', 'raghu@123', 'admin', 'friend');

-- 
-- Indexes for dumped tables
-- 

-- 
-- Indexes for table `article_tbl`
-- 
ALTER TABLE `article_tbl`
ADD PRIMARY KEY (`a_id`);

-- 
-- Indexes for table `facuties_tbl`
-- 
ALTER TABLE `facuties_tbl`
ADD PRIMARY KEY (`faculties_id`);

-- 
-- Indexes for table `location_tb`
-- 
ALTER TABLE `location_tb`
ADD PRIMARY KEY (`loca_id`);

-- 
-- Indexes for table `stu_score_tbl`
-- 
ALTER TABLE `stu_score_tbl`
ADD PRIMARY KEY (`ss_id`);

-- 
-- Indexes for table `stu_tbl`
-- 
ALTER TABLE `stu_tbl`
ADD PRIMARY KEY (`stu_id`);
```

```
-- Indexes for table `sub_tbl`  
--  
ALTER TABLE `sub_tbl`  
    ADD PRIMARY KEY (`sub_id`);  
  
--  
-- Indexes for table `teacher_tbl`  
--  
ALTER TABLE `teacher_tbl`  
    ADD PRIMARY KEY (`teacher_id`);  
  
--  
-- Indexes for table `users_tbl`  
--  
ALTER TABLE `users_tbl`  
    ADD PRIMARY KEY (`u_id`);  
  
--  
-- Indexes for table `users_tbl_backup`  
--  
ALTER TABLE `users_tbl_backup`  
    ADD PRIMARY KEY (`u_id`);  
  
--  
-- AUTO_INCREMENT for dumped tables  
--  
  
--  
-- AUTO_INCREMENT for table `article_tbl`  
--  
ALTER TABLE `article_tbl`  
    MODIFY `a_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;  
  
--  
-- AUTO_INCREMENT for table `facuties_tbl`  
--  
ALTER TABLE `facuties_tbl`  
    MODIFY `faculties_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    AUTO_INCREMENT=8;  
  
--  
-- AUTO_INCREMENT for table `location_tb`  
--  
ALTER TABLE `location_tb`  
    MODIFY `loca_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;  
  
--  
-- AUTO_INCREMENT for table `stu_score_tbl`  
--  
ALTER TABLE `stu_score_tbl`  
    MODIFY `ss_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=117;
```

---

```
--  
-- AUTO_INCREMENT for table `stu_tbl`  
--  
ALTER TABLE `stu_tbl`  
  MODIFY `stu_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;  
  
--  
-- AUTO_INCREMENT for table `sub_tbl`  
--  
ALTER TABLE `sub_tbl`  
  MODIFY `sub_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=20;  
  
--  
-- AUTO_INCREMENT for table `teacher_tbl`  
--  
ALTER TABLE `teacher_tbl`  
  MODIFY `teacher_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=20;  
  
--  
-- AUTO_INCREMENT for table `users_tbl`  
--  
ALTER TABLE `users_tbl`  
  MODIFY `u_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;  
  
--  
-- AUTO_INCREMENT for table `users_tbl_backup`  
--  
ALTER TABLE `users_tbl_backup`  
  MODIFY `u_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;  
COMMIT;  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

### 5.3 Description of Tables :

#### Users\_table :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	u_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	username	varchar(50)	latin1_swedish_ci		No	None		
3	password	varchar(30)	latin1_swedish_ci		No	None		
4	type	char(10)	latin1_swedish_ci		No	None		
5	note	varchar(100)	latin1_swedish_ci		No	None		

#### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	u_id	2	A	No	

#### Partitions

No partitioning defined!

#### Information

##### Space usage

Data	16	KiB
Index	0	B
Overhead		
Effective	16	KiB
Total	16	KiB

##### Row statistics

Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	12
Creation	Nov 17, 2019 at 01:30 PM
Last update	Nov 17, 2019 at 12:33 PM
Last check	Nov 17, 2019 at 12:33 PM

**Teachers\_table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>teacher_id</b>	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>f_name</b>	varchar(30)	latin1_swedish_ci		No	None		
3	<b>l_name</b>	varchar(30)	latin1_swedish_ci		No	None		
4	<b>gender</b>	char(10)	latin1_swedish_ci		No	None		
5	<b>dob</b>	date			No	None		
6	<b>pob</b>	varchar(100)	latin1_swedish_ci		No	None		
7	<b>address</b>	varchar(100)	latin1_swedish_ci		No	None		
8	<b>degree</b>	varchar(50)	latin1_swedish_ci		No	None		
9	<b>salary</b>	float			No	None		
10	<b>married</b>	char(10)	latin1_swedish_ci		No	None		
11	<b>phone</b>	varchar(50)	latin1_swedish_ci		No	None		
12	<b>email</b>	varchar(50)	latin1_swedish_ci		No	None		
13	<b>note</b>	varchar(100)	latin1_swedish_ci		No	None		

**Indexes**

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	teacher_id	6	A	No	

**Partitions**

No partitioning defined!

**Information****Space usage**

Data	16	KiB
Index	0	B
Overhead		
Effective	16	KiB
Total	16	KiB

**Row statistics**

Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	20
Creation	Nov 17, 2019 at 01:30 PM
Last update	Nov 17, 2019 at 12:36 PM
Last check	Nov 17, 2019 at 12:36 PM

**Student\_score\_table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ss_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	stu_id	int(10)			No	None		
3	faculties_id	int(10)			No	None		
4	sub_id	int(10)			No	None		
5	ia1	int(11)			No	None		
6	ia2	int(11)			No	None		
7	ia3	int(11)			No	None		
8	note	varchar(100)	latin1_swedish_ci		No	None		

## Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ss_id	2	A	No	

## Partitions

No partitioning defined!

## Information

Space usage		
Data	16	KiB
Index	0	B
Overhead		
Effective	16	KiB
Total	16	KiB

Row statistics	
Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	119
Creation	Nov 17, 2019 at 01:30 PM
Last update	Nov 17, 2019 at 12:37 PM
Last check	Nov 17, 2019 at 12:37 PM

**Subject\_table:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>sub_id</b>	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>faculties_id</b>	int(10)			No	None		
3	<b>teacher_id</b>	int(10)			No	None		
4	<b>semester</b>	varchar(10)	latin1_swedish_ci		No	None		
5	<b>sub_name</b>	varchar(100)	latin1_swedish_ci		No	None		
6	<b>note</b>	varchar(100)	latin1_swedish_ci		No	None		

## Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	sub_id	7	A	No	

## Partitions

No partitioning defined!

## Information

Space usage		
Data	16	KiB
Index	0	B
Overhead		
Effective	16	KiB
Total	16	KiB

Row statistics	
Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	20
Creation	Nov 17, 2019 at 01:30 PM
Last update	Nov 17, 2019 at 12:39 PM
Last check	Nov 17, 2019 at 12:39 PM

**Student\_table :**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	stu_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	f_name	varchar(50)	latin1_swedish_ci		No	None		
3	l_name	varchar(50)	latin1_swedish_ci		No	None		
4	gender	char(10)	latin1_swedish_ci		No	None		
5	dob	date			No	None		
6	pob	varchar(100)	latin1_swedish_ci		No	None		
7	address	varchar(100)	latin1_swedish_ci		No	None		
8	phone	varchar(50)	latin1_swedish_ci		No	None		
9	email	varchar(70)	latin1_swedish_ci		No	None		
10	note	varchar(100)	latin1_swedish_ci		No	None		

**Indexes**

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	stu_id	4	A	No	

**Partitions**

No partitioning defined!

**Information****Space usage**

Data	16	KiB
Index	0	B
Overhead		
Effective	16	KiB
Total	16	KiB

**Row statistics**

Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	16
Creation	Nov 17, 2019 at 01:30 PM
Last update	Nov 17, 2019 at 12:40 PM
Last check	Nov 17, 2019 at 12:40 PM

**Faculty\_table :**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>faculties_id</b>	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>faculties_name</b>	varchar(50)	latin1_swedish_ci		No	None		
3	<b>note</b>	varchar(100)	latin1_swedish_ci		No	None		

**Indexes**

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	faculties_id	1	A	No	

**Partitions**

No partitioning defined!

**Information**

Space usage		
Data	16	KiB
Index	0	B
Overhead		
Effective	16	KiB
Total	16	KiB

Row statistics	
Format	dynamic
Collation	latin1_swedish_ci
Next autoindex	12
Creation	Nov 17, 2019 at 01:30 PM
Last update	Nov 17, 2019 at 12:41 PM
Last check	Nov 17, 2019 at 12:41 PM

## 5.4 Screenshots

Table	Action	Rows	Type	Collation	Size	Overhead
article_tbl	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 Kib	-
faculties_tbl	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 Kib	-
location_tb	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 Kib	-
stu_score_tbl	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 Kib	-
stu_tbl	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 Kib	-
sub_tbl	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 Kib	-
teacher_tbl	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16 Kib	-
users_tbl	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 Kib	-
users_tbl_backup	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 Kib	-
<b>9 tables</b>	<b>Sum</b>	<b>27</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>144 Kib</b>	<b>0 B</b>

Fig 1: Back end view in localhost and tables present in SMS

The screenshot shows the phpMyAdmin interface for a database named 'a'. The left sidebar lists various databases and tables. The main area displays the 'stu\_tbl' table structure. The table has 10 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	stu_id	int(10)	latin1_swedish_ci	UNSIGNED	No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	f_name	varchar(50)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	l_name	varchar(50)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	gender	char(10)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
5	dob	date			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
6	pob	varchar(100)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
7	address	varchar(100)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
8	phone	varchar(50)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
9	email	varchar(70)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
10	note	varchar(100)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Below the table structure, there are buttons for 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', etc. There is also a section for 'Indexes' with one entry:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<a href="#">Edit</a> <a href="#">Drop</a>	PRIMARY	BTREE	Yes	No	stu_id	4	A	No	

At the bottom, there is a button to 'Create an index on' with a dropdown for 'columns'.

Fig 2 : All Attributes Present in student table

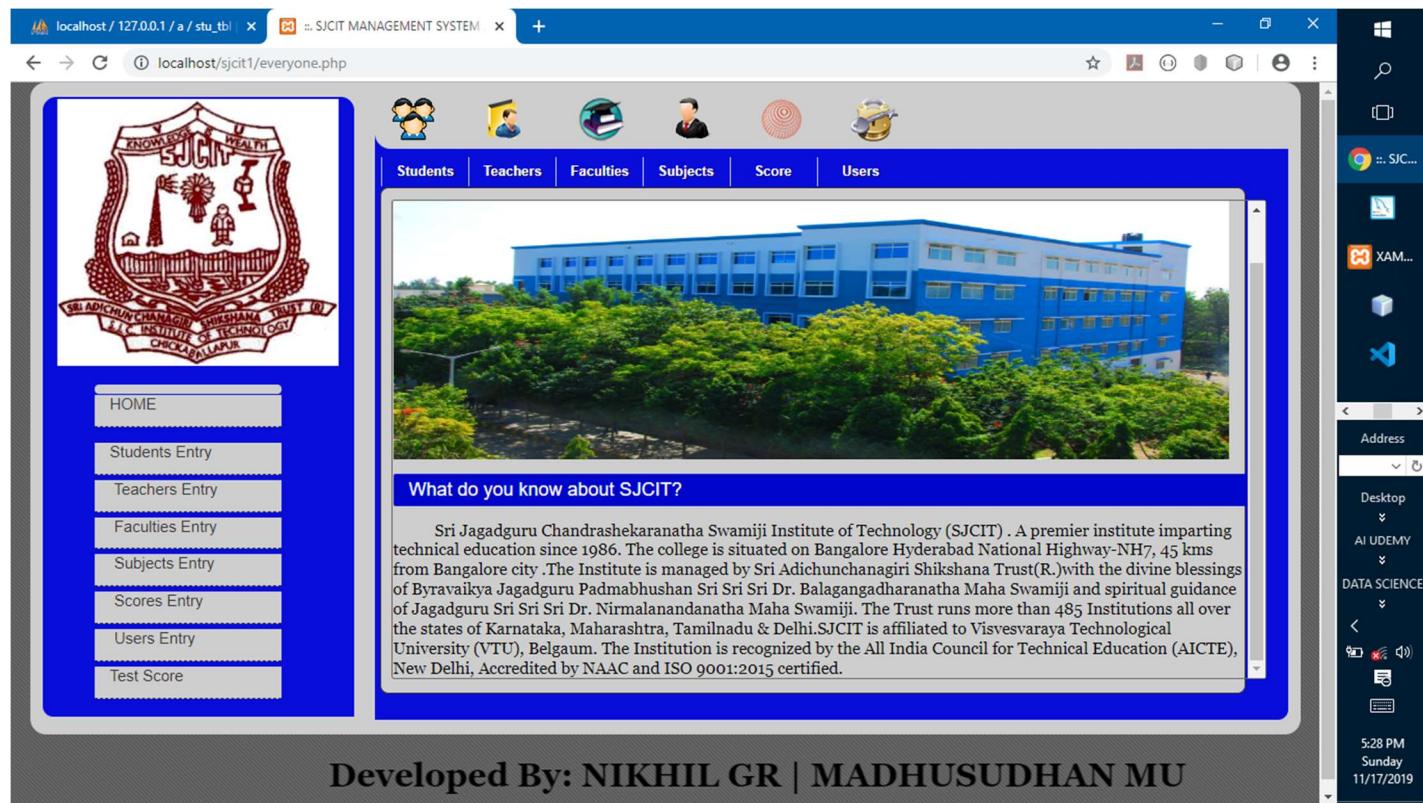
## SJCIT Management System

The screenshot shows the phpMyAdmin interface for a database named 'a'. The left sidebar lists various tables and databases. The main area displays the 'stu\_tbl' table with 4 rows of data. The columns are: stu\_id, f\_name, l\_name, gender, dob, pob, address, phone, and email. The data is as follows:

stu_id	f_name	l_name	gender	dob	pob	address	phone	email
1	NIKHIL	GR	Male	1999-03-09	DODDABALLAPUR	DODDABALLAPUR	7899979749	nikhilgubbi@
2	MADHUSUDHAN	MU	Male	1999-09-28	DODDABALLPUR	DODDABELAVANGALA	9113992200	madhukohli@
3	RAGHAVENDRA	DC	Male	1999-09-14	DODDABALLAPUR	DODDABALLAPUR	8892465672	raghuharsh@
4	MANJU	H	Male	1999-02-02	DODDABALLAPUR	DODDABALLAPUR	9986037547	manjusdupu

Fig 3 : The Values That Are Present in Student Table

## SJCIT Management System



**Developed By: NIKHIL GR | MADHUSUDHAN MU**

Fig 4 : Home page of SJCIT Management System

The screenshot shows a web browser window for the SJCIT Management System. The title bar reads "localhost / 127.0.0.1 / a / stu\_tbl" and ":: SJCIT MANAGEMENT SYSTEM". The main content area displays the "Teachers Entry" form. At the top of the form is a logo of Sri Adichunchanagiri Shikshana Trust (SJCIT) Institute of Technology, Chickballapur. Below the logo is a navigation menu with icons for Students, Teachers, Faculties, Subjects, Score, and Users. The "Teachers Entry" tab is selected. The form itself has fields for First Name, Last Name, Degree (with a dropdown menu showing "Select"), Salary, Gender (radio buttons for Male and Female), Married status (radio buttons for Yes and No), Date Of Birth (with dropdown menus for Year, Month, and Date), Phone, E-mail, Address, and Note. There are "Register" and "Cancel" buttons at the bottom. A "View Teachers" link is located in the top right corner of the form area. On the left side of the screen, there is a vertical sidebar with links: HOME, Students Entry, Teachers Entry (which is highlighted in blue), Faculties Entry, Subjects Entry, Scores Entry, Users Entry, and Test Score. The background of the entire interface is blue.

Fig 5 : Teacher's Entry To Enter the New Teacher details

The screenshot shows a web-based management system for SJCIT. The header includes a logo of the college, the name "SJCIT MANAGEMENT SYSTEM", and a search bar. Below the header is a navigation menu with icons for Students, Teachers, Faculties, Subjects, Score, and Users. The main content area is titled "View Teachers" and contains a table with the following data:

No	Teacher Name	Gender	Date of Birth	Place of Birth	Address	Degree	Salary	Married
1	SRINATH	Male	0000-00-00			Select	0	Yes
2	VENKATESH	Male	0000-00-00			Select	0	Yes
3	PRADEEP	Male	0000-00-00			Select	0	Yes
4	SESHAIH	Male	0000-00-00			Select	0	Yes
5	VINUTHA	Male	0000-00-00			Select	0	Yes
6	AJAY	Male	0000-00-00			Select	0	Yes

Below the table is a large empty space, likely a placeholder for a map or another module.

**Developed By: NIKHIL GR | MADHUSUDHAN MU**

Fig 6 : View of teacher Entry

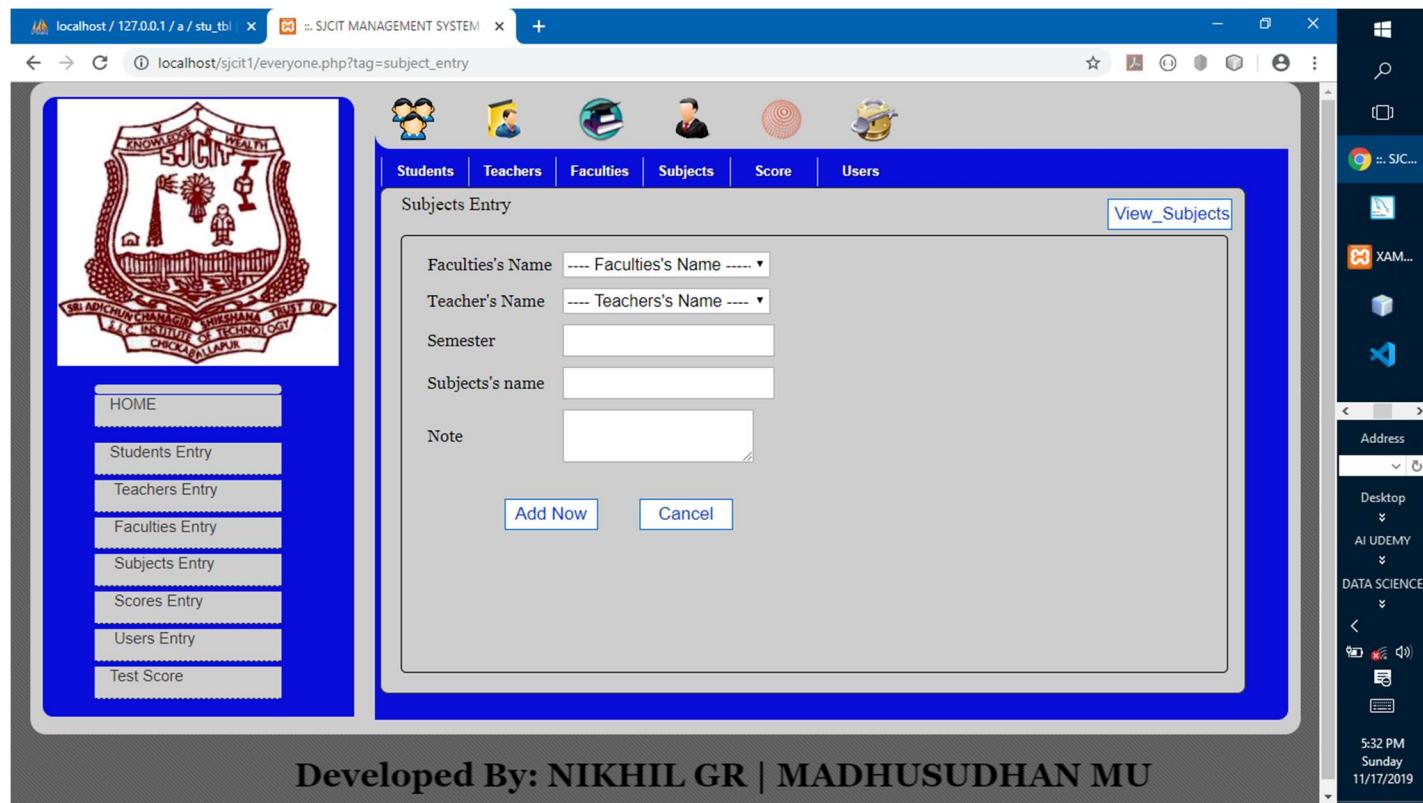


Fig 7 : Subject entry

The screenshot shows a web browser window titled "localhost / 127.0.0.1 / a / stu\_tbl" and ":: SJCIT MANAGEMENT SYSTEM". The main content area displays a table of subjects with columns: No, Faculties Name, Teachers Name, Semester, Subject Name, Note, and Operation. The table contains 6 rows of data. A sidebar on the left lists various administrative entries like Home, Students Entry, Teachers Entry, etc. The top navigation bar includes links for Students, Teachers, Faculties, Subjects, Score, and Users. The right side of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time.

No	Faculties Name	Teachers Name	Semester	Subject Name	Note	Operation
1	4	14	STH	M&E		
2	4	15	STH	CN		
3	4	16	STH	DBMS		
4	4	17	STH	ATC		
5	4	18	STH	J2EE		
6	4	19	STH	AI		

Developed By: NIKHIL GR | MADHUSUDHAN MU

Fig 8 : View of Subject

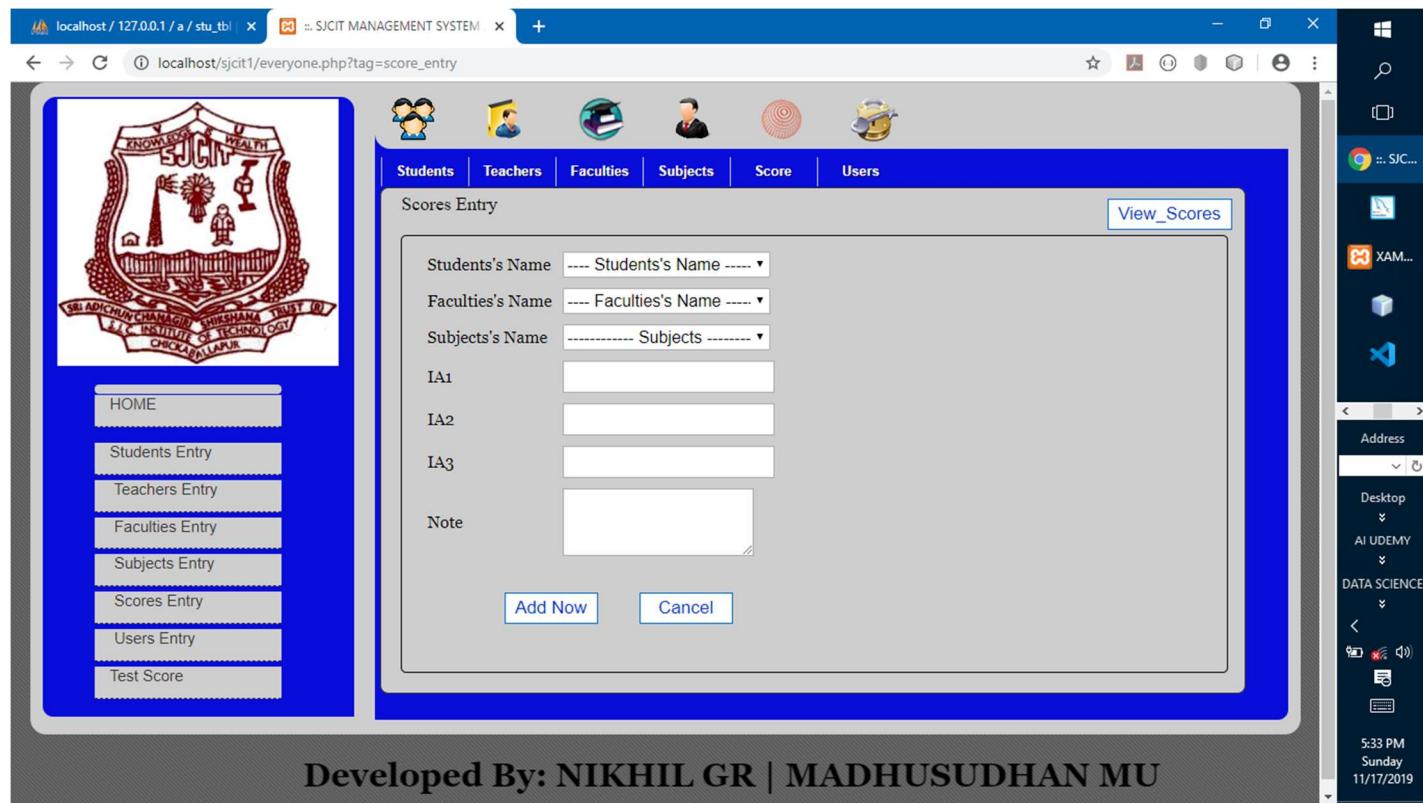


Fig 9 : Score entry

The screenshot shows a web-based management system interface. At the top, there are two tabs: 'localhost / 127.0.0.1 / a / stu\_tbl' and 'SJCT MANAGEMENT SYSTEM'. The main window title is 'localhost/sjcit1/everyone.php?tag=view\_scores'. The interface includes a header with icons for Students, Teachers, Faculties, Subjects, Score, and Users. Below the header is a navigation menu on the left with links: HOME, Students Entry, Teachers Entry, Faculties Entry, Subjects Entry, Scores Entry, Users Entry, and Test Score. The central area displays a table titled 'View Scores' with the following data:

Students ID	Faculties ID	Subject ID	IA1	IA2	IA3	Note
1	4	13	29	30	28	
2	4	14	25	24	22	

At the bottom of the screen, it says 'Developed By: NIKHIL GR | MADHUSUDHAN MU'.

Fig 10 : View of score of each Student

The screenshot shows a web-based management system for SJCIT. The interface includes a header with icons for Students, Teachers, Faculties, Subjects, Score, and Users. On the left, there's a sidebar with links for Home, Students Entry, Teachers Entry, Faculties Entry, Subjects Entry, Scores Entry, Users Entry, and Test Score. The main content area displays a table titled 'View Scores' showing test results for two students: NIKHIL GR and MADHUSUDHAN MU. The table includes columns for Student Name, Sex, Date of Birth, and scores for M&E (IA1, IA2, IA3, Total) and CN (IA1, IA2, IA3, Total). A large gray area below the table is currently empty.

Students Name	Sex	Date of Birth	M&E				CN				DBM			
			IA1	IA2	IA3	Total	IA1	IA2	IA3	Total	IA1	IA2	L	
NIKHIL GR	Male	1999-03-09	29	30	28	<b>29</b>				<b>0</b>				
MADHUSUDHAN MU	Male	1999-09-28				<b>0</b>	25	24	22	<b>24</b>				

Fig 11 : Test\_Score of Each Student

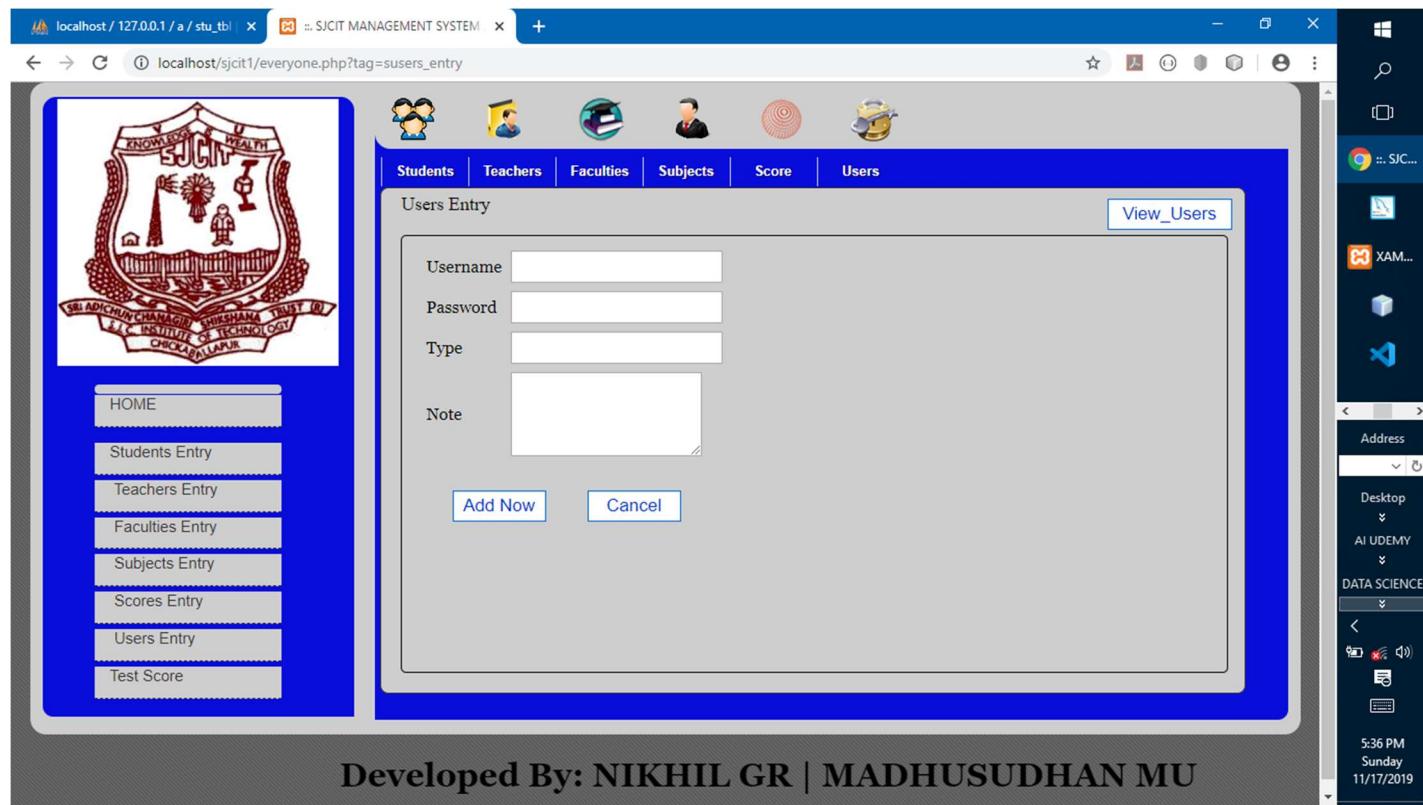


Fig 12 : Admin page it has the rights to access , delete and update

The screenshot shows a web browser window titled "SJCIT MANAGEMENT SYSTEM" with the URL "localhost/sjcit1/everyone.php?tag=view\_users". The page displays a table of users with the following data:

No	Users Name	Password	Type	Note	Operation
1	nikhil	nikhil@123	admin	creator	(Edit) (Delete)
2	madhu	madhu@123	admin	creator	(Edit) (Delete)

The left sidebar contains a navigation menu with links: HOME, Students Entry, Teachers Entry, Faculties Entry, Subjects Entry, Scores Entry, Users Entry, and Test Score. The top navigation bar includes icons for Students, Teachers, Faculties, Subjects, Score, and Users.

**Developed By: NIKHIL GR | MADHUSUDHAN MU**

Fig 13 : View of user

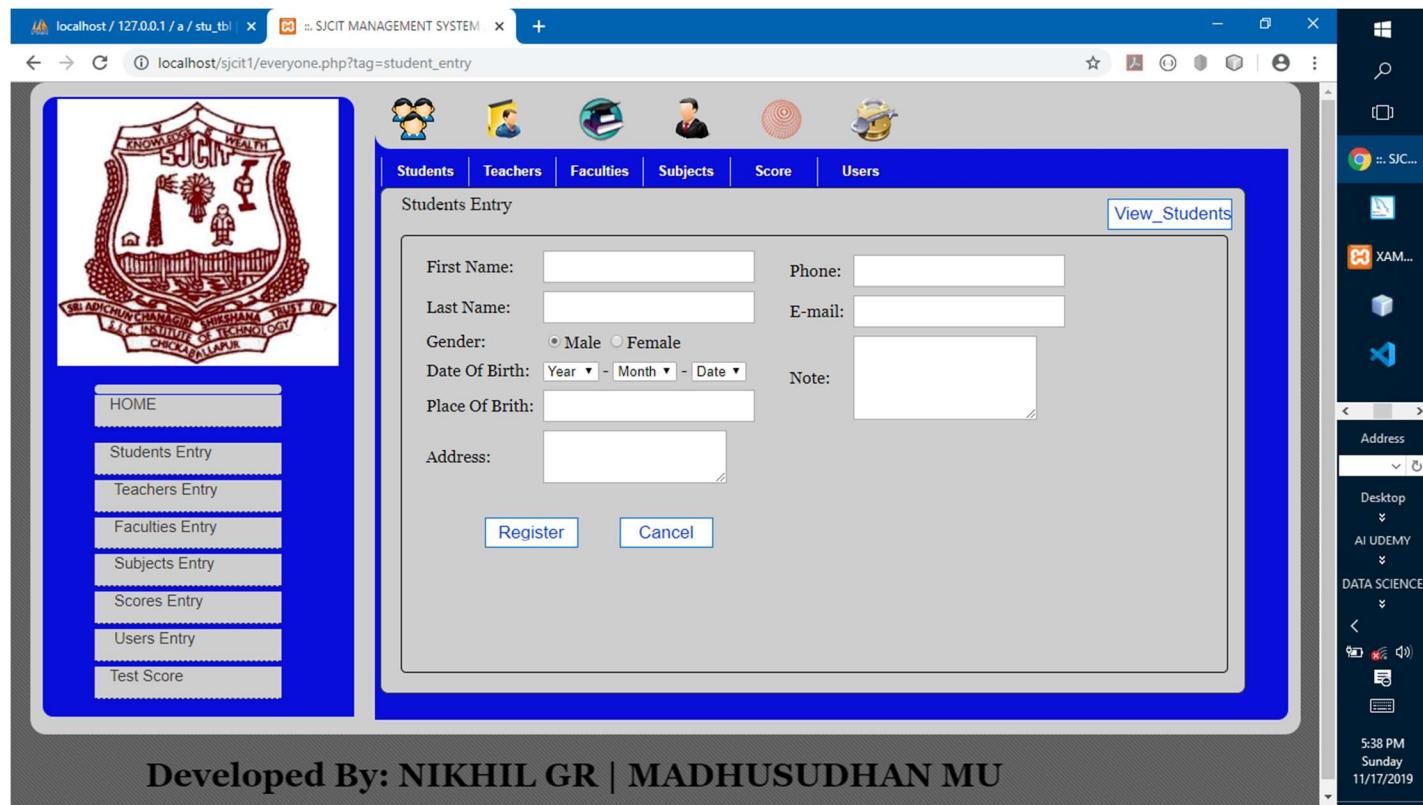


Fig 14 : Student Entry

The screenshot shows a web-based management system for SJCIT. The main window displays a table of student records with columns for No, Student Name, Gender, Date of Birth, Place of Birth, Address, Phone, and Email. The records listed are:

No	Student Name	Gender	Date of Birth	Place of Birth	Address	Phone	Email
1	NIKHIL GR	Male	1999-03-09	DODDABALLAPUR	DODDABALLAPUR	7899979749	nikhilgubb...
2	MADHUSUDHAN MU	Male	1999-09-28	DODDABALLPUR	DODDABELAVANGALA	9113992200	madhukoh...
3	RAGHAVENDRA DC	Male	1999-09-14	DODDABALLAPUR	DODDABALLAPUR	8892465672	raghuharsh...
4	MANU H	Male	1999-02-02	DODDABALLAPUR	DODDABALLAPUR	9986037547	manjusdup...

The left sidebar contains links for navigating the system: HOME, Students Entry, Teachers Entry, Faculties Entry, Subjects Entry, Scores Entry, Users Entry, and Test Score. The top navigation bar includes icons for Students, Teachers, Faculties, Subjects, Score, and Users.

Developed By: NIKHIL GR | MADHUSUDHAN MU

Fig 15 : View of Student Entry

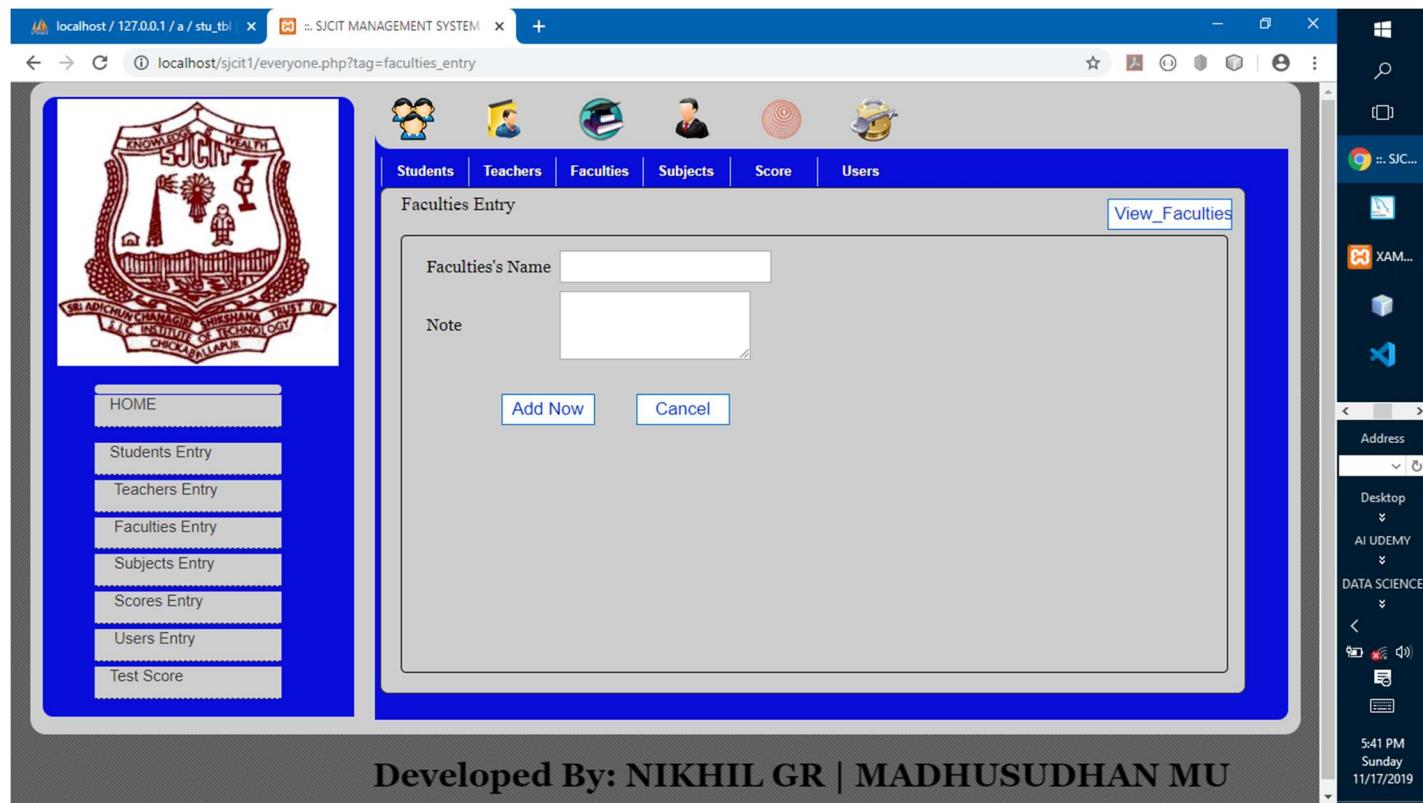


Fig 16 : Faculties Entry

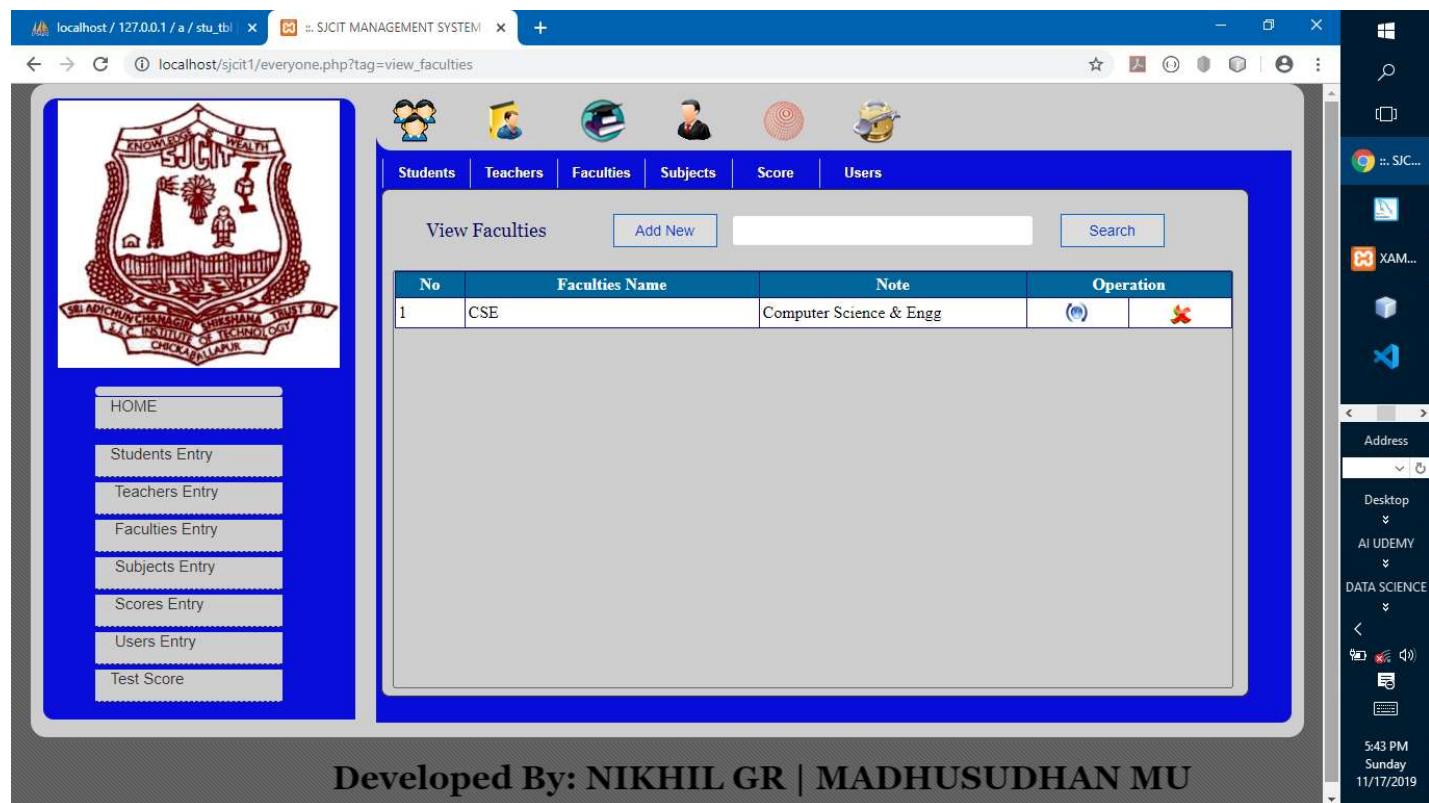


Fig 17 : View of faculties

## CHAPTER 6

# CONCLUSION AND FUTURE ENHANCEMENT

### 6.1 Conclusion

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in HTML & CSS and PHP web based application and no some extent MySQL workbench and MYSQL, but also about all handling procedure related with “SJCIT Management System”. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

#### Benefits:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows:-

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for him that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.

- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency

## 6.2 Future Enhancement

- It can be implemented to upload files with an huge amount of size with the support of various file formats.
- This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
- A console for the data center may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.
- Moreover, it is just a beginning; further the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow based applications.

## REFERENCES

### Websites

<http://www.tcs.com>  
<http://www.codeguru.com>  
<http://www.1001tutorials.com>  
<http://www.projectabstract.com>

### Books

- Software Engineering - A Practitioner's Approach
- Database management system,Ramakrishnan and Gehrke,3<sup>rd</sup> Edition,2014,McGraw Hill
- Andy Harris, 2003 PHP/MySQL Programming for the Absolute Beginners Premier Press, 25 Thomson Place, Boston, MA 02210
- Michael Kofler 2004, The Definitive Guide to MySQL, Second Edition Apress, Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY, 10010
- Julie C Meloni, 2002, Teach Yourself PHP, MySQL™ and Apache in 24 Hours Los Altos, California

- Rasmus Lerdorf, Peter MacIntyre, Kevin Tatroe, 2006, Programming PHP, 2nd Edition  
1005 Gravenstein Highway North, Sebastopol, CA 95472.
- Matt Rutledge, 2004, PHP Game Programming, 25 Thomson Place Boston, MA 02210