

A  
PROJECT REPORT ON

# **STUDENT RECORD** **MANAGEMENT SYSTEM**

By

**Himanshubhai B Amaliyar (CE-012) (20CEUTG127)**

**Dhirenkumar A Chaudhari (CE-025) (20CEUTG004)**

**B.Tech CE Semester-VI**  
**Subject: CE625 System Design Practice**

**Guided by:**

Prof. Ashish K. Gor,  
Assistant Professor,  
Dept. of Comp. Engg.



**Faculty of Technology**  
**Department of Computer Engineering**  
**Dharmsinh Desai University**



**Faculty of Technology  
Department of Computer Engineering  
Dharmsinh Desai University**

## **CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of

**System Design Practice** and recorded in this journal is the

bonafide work of

**Himanshubhai B Amaliyar (CE-012) (20CEUTG127)**

**Dhirenkumar A Chaudhari (CE-025) (20CEUTG004)**

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2022-2023**.

**Prof. Ashish K. Gor**  
**Assistant Professor**  
Computer Engg. Department  
Faculty of Technology  
D.D.U., Nadiad

**Dr. C. K. Bhensdadia**  
**Professor and HoD,**  
Computer Engg. Department  
Faculty of Technology  
D.D.U., Nadiad

## **INDEX**

### Introduction

|                              |    |
|------------------------------|----|
| Abstract.....                | 04 |
| Introduction & approach..... | 05 |

### Analysis

|                                       |    |
|---------------------------------------|----|
| Problem.....                          | 06 |
| Scope.....                            | 06 |
| System Requirement Specification..... | 07 |
| Use-Case Diagram.....                 | 11 |

### Design

|                                 |    |
|---------------------------------|----|
| Class Diagram.....              | 12 |
| Project Modules.....            | 13 |
| Database Tables Structures..... | 14 |

### Implementation

|                           |    |
|---------------------------|----|
| Tools and Technology..... | 16 |
| Code.....                 | 17 |

### Testing

|                       |    |
|-----------------------|----|
| Testing Strategy..... | 35 |
| Screen Shots.....     | 37 |

### Conclusion & Future Work

|                                  |    |
|----------------------------------|----|
| Limitations and Future Work..... | 50 |
| Conclusion.....                  | 50 |

### Bibliography

|                   |    |
|-------------------|----|
| Bibliography..... | 51 |
|-------------------|----|

# **INTRODUCTION**

## **Abstract:**

The Student Record Management System is a PHP-based project that aims to facilitate the management of student records for educational institutions. This System is designed to have two different sides: the user side and admin side. The user side allows students to access their attendance and results, while the admin side is responsible for managing the courses, students, departments, attendance, subjects, and results.

Overall, the Student Record Management System is a reliable and efficient tool for managing student records. It streamlines the management process, reduces administrative workload, and ensures accuracy and security of student records. This system is suitable for use by educational institutions of various sizes and levels, from primary schools to universities.

## **Introduction & Approach:**

Managing student records is a crucial aspect of educational institutions. It involves the collection, storage, and retrieval of data related to students' academic performance, attendance, and personal information. The traditional method of managing student records involves manual processes, which are often time-consuming, prone to errors, and challenging to track. Moreover, the increasing number of students and the complexity of educational systems have made it difficult to manage student records manually.

To address these challenges, the Student Record Management System has been developed. This system is a web-based application that provides a comprehensive solution for managing student records. It is designed to automate the collection, storage, and retrieval of student data, making the process faster, more accurate, and more efficient. Moreover, it is easy to use and provides a user-friendly interface for both administrators and students.

The admin side has several functionalities, including adding and updating courses, adding and updating students, adding and updating departments, adding and updating attendance records, adding subjects, and managing results. These features enable the admin to have complete control over the management of student records. Moreover, the system provides security measures, such as user authentication and access control, to ensure that only authorized personnel can access and manage the records.

On the other hand, the user side allows students to view their results and attendance records. This feature enables students to keep track of their academic performance and attendance throughout their course of study. The system provides an intuitive user interface that makes it easy for students to navigate and access their records.

Overall, the Student Record Management System is a reliable and efficient tool for managing student records. It simplifies the management process, reduces administrative workload, and ensures accuracy and security of student records. This system is suitable for use by educational institutions of various sizes and levels, from primary schools to universities. In the following sections of this report, we will discuss the features, functionality, and technical aspects of the Student Record Management System in more detail.

## **ANALYSIS**

### **Problem:**

To maintain the student details, history of student's exam scores, attendance, scholastic achievements, course, and performance in extracurricular activities on paper is hard. The administration departments of university are often seen juggling papers and keeping track of documents trapped in files. This way consumes more time and energy.

### **Scope:**

The purpose of this document is to provide a detailed description of the Student Record Management System. Admin can add students, manage students, add subjects, update subjects, add department, manage department, add course, manage course, add results, manage results, add attendance, manage attendance. Student can see their results and their attendance.

## **System Requirement Specification:**

### **Role Specifications:**

- 1.) **Admin:** Add, update and delete students also add and update departments, courses, subjects, results and attendance. Also can change its own login password.
- 2.) **Student:** View results and attendance.

## **Software Requirement Specification:**

### **❖ System Functional Requirements:**

#### **✦ R1. Manage Student Information:**

##### **▪ R1.1: Add Student**

Description: After admin fill the student details and click on add button then student will be added to database.

Input: Student Data

Output: Student Added

##### **▪ R1.2: Update Student**

Description: After admin change the required details and click on Update button then student will be update to database.

Input: Student Data

Output: Student Updated

##### **▪ R1.3: Delete Student**

Description: After admin click on delete button the student will be delete from the database.

Input: Student Id

Output: Student Deleted

#### **✦ R2. Manage Student Result:**

##### **▪ R2.1: Add Student Result**

Description: After admin fill the student result details and click on add button then student result will be added to database.

Input: Student Result

Output: Student Result Added

- **R2.2: Update Student Result**

Description: After admin change the required details of students result and click on Update button then student result will be updated to database.

Input: Student Result Data

Output: Student Result Updated

- ✦ **R3. Manage Student Attendance:**

- **R3.1: Add Student Attendance**

Description: After admin fill the student attendance details and click on add button then student attendance will be added to database.

Input: Student Attendance

Output: Student Attendance Added

- **R3.2: Update Student Attendance**

Description: After admin change the required details and click on Update button then student attendance will be updated to database.

Input: Student Attendance Data

Output: Student Attendance Updated

- ✦ **R4. Manage Departments:**

- **R4.1: Add Department**

Description: After admin fill the department details and click on add button then department and semester will be added to database.

Input: Department Data

Output: Department Added

- **R4.2: Update Department**



Description: After admin change the required details and click on Update button then department and semester will be updated to database.

Input: Department Data

Output: Department Updated

✦ **R5. Manage Subjects:**

▪ **R5.1: Add Subject**

Description: After admin fill the subject details and click on add button then subject will be added to database.

Input: Subject Data

Output: Subject Added

▪ **R5.2: Update Subject**

Description: After admin change the required details and click on Update button then Subject will be updated to database.

Input: Subject Data

Output: Subject Updated

✦ **R6. Manage Subjects Combination (Course):**

▪ **R6.1: Add Subjects Combination**

Description: After admin fill the subject combination details and click on add button then subject combination will be added to database.

Input: Student Data

Output: Student Added

▪ **R6.2: Update Subjects Combination**

Description: After admin change the required details then subject combination will be updated to database.

Input: Subject Combination Data

Output: Subject Combination Updated

✦ **R7. Student View:**

▪ **R7.1: View Result**

Description: After student enter the Roll Id and select department then click on search button then the result will be shown.

Input: Student Roll Id and Department

Output: Result

- **R7.2: View Attendance**

Description: After student enter the Roll Id and select department then click on search button then the attendance report will be shown.

Input: Student Roll Id and Department

Output: Attendance Report

- ✦ **R8. Admin Password:**

- **R8.1: Change Login Password**

Description: Admin can change its own login password

Input: Current password and new password

Output: Password gets changed

## ❖ **Non-Functional Requirements:**

### **R1. Performance:**

The system should be able to handle multiple users simultaneously. The system should have a response time of less than 2 seconds for any user request.

### **R2. Security:**

The system should have secure login and authentication mechanisms. The system should provide access control to ensure that only authorized personnel can access and modify student records.

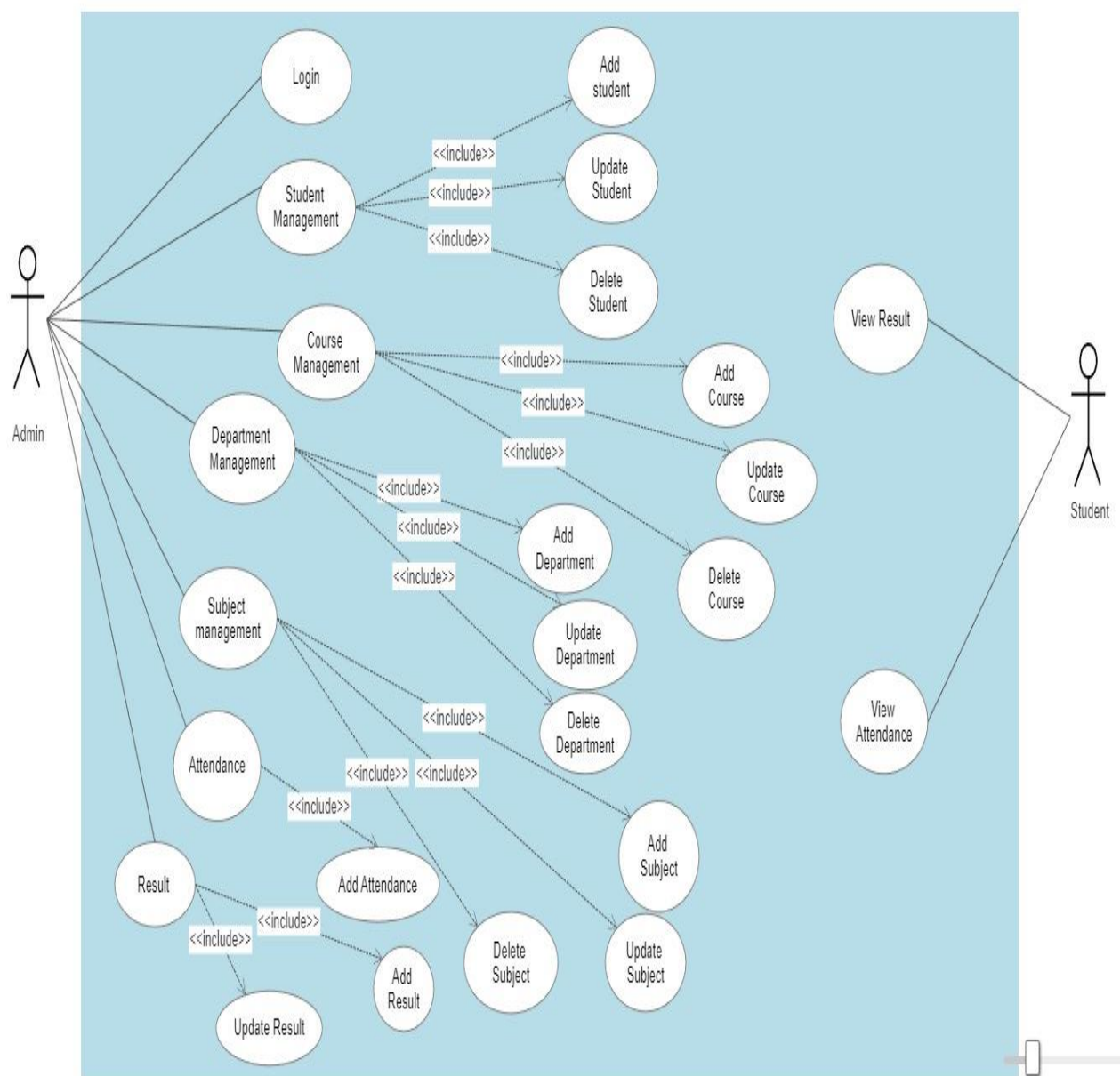
### **R3. Reliability:**

The system should be able to recover data in case of any system failure. The system should provide a backup and restore mechanism to protect against data loss.

#### R4. Usability:

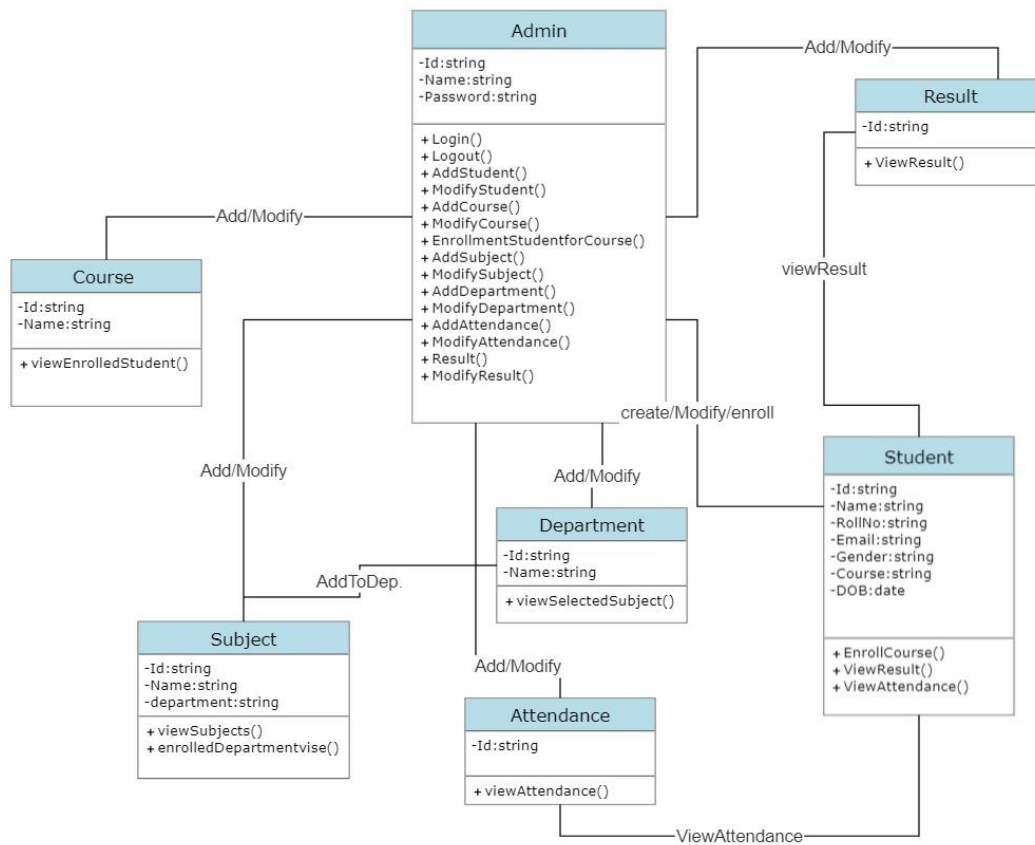
The system should have a user-friendly interface for both administrators and students. The system should provide clear and concise instructions for all user actions.

#### Use-Case Diagram:



# DESIGN

## Class Diagram:



## **Project Modules:**

Our application deals with six modules.

1. Student Module
2. Admin Module
3. Subject Module
4. Subject Combination Module
5. Result Module
6. Attendance Module

Student Module:

- In this student module admin will register the details of the student.
- A student can view his result and attendance.

Admin Module:

- Administrator has the power to add new student and can edit, view and delete a student.
- The administrator can add, edit and delete result for the student.
- The administrator can add, edit and delete attendance for the student.

Subject Module:

- In this module Administrator register all subjects and also provide subject code to each and every subject.
- Administrator edit and delete the subjects.

Subject Combination Module:

- Assign subjects to every branch in semester wise.
- Administrator can edit and delete the subject combination.

Result Module:

- Administrator enters result of the student in semester wise.
- Administrator can also edit and delete the result of the student.

Attendance Module:

- Administrator enters attendance of the student in semester wise.

- Administrator can also edit and delete the attendance of the student.

### **Database Tables Structures:**

#### **tblstudents:**

| Field Name   | Data Type    |
|--------------|--------------|
| StudentId    | Int(11)      |
| StudentName  | Varchar(100) |
| RollId       | Varchar(100) |
| StudentEmail | Varchar(100) |
| Gender       | Varchar(10)  |
| DOB          | Varchar(100) |
| DepartmentId | Int(11)      |
| RegDate      | timestamp    |
| UpdationDate | timestamp    |
| status       | Int(1)       |

#### **tbldepartments:**

| Field Name     | Data Type    |
|----------------|--------------|
| id             | Int(11)      |
| DepartmentName | Varchar(200) |
| Semester       | Int(10)      |
| CreationDate   | Timestamp    |
| UpdationDate   | Timestamp    |

#### **tblsubjects:**

| Field Name   | Data Type    |
|--------------|--------------|
| Id           | Int(11)      |
| SubjectName  | Varchar(100) |
| SubjectCode  | varchar(100) |
| Creationdate | Timestamp    |
| UpdationDate | timestamp    |

#### **tblsubjectscombination:**

| Field Name   | Data Type |
|--------------|-----------|
| Id           | Int(11)   |
| DepartmentId | Int(11)   |
| SubjectId    | Int(11)   |
| Status       | Int(1)    |

|              |           |
|--------------|-----------|
| CreationDate | Timestamp |
| UpdationDate | timestamp |

**tblresult:**

| Field Name   | Data Type |
|--------------|-----------|
| id           | Int(11)   |
| StudentId    | Int(11)   |
| DepartmentId | Int(11)   |
| SubjectId    | Int(11)   |
| marks        | Int(11)   |
| PostingDate  | Timestamp |
| UpdationDate | timestamp |

**tblattendance:**

| Field Name   | Data Type |
|--------------|-----------|
| id           | Int(11)   |
| StudentId    | Int(11)   |
| DepartmentId | Int(11)   |
| SubjectId    | Int(11)   |
| Attendance   | Int(11)   |
| PostingDate  | Timestamp |
| UpdationDate | timestamp |

**admin:**

| Field Name   | Data Type    |
|--------------|--------------|
| id           | Int(11)      |
| UserName     | Varchar(100) |
| Password     | Varchar(100) |
| updationDate | Timestamp    |

# **IMPLEMENTATION**

## **Tools and Technology:**

### **✦ Technology:**

- PHP
- Boot Strap
- CSS
- MySQL
- Java script

### **✦ Tools:**

- Sublime Text Editor
- XAMPP
- MySQL Server



## Codes for major modules:

### ❖ Admin Module

#### ➤ Adding Student

```
<form class="form-horizontal" method="post">

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Full Name</label>
<div class="col-sm-10">
<input type="text" name="fullanme" class="form-control"
id="fullanme" required="required" autocomplete="off">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Roll Id</label>
<div class="col-sm-10">
<input type="text" name="rollid" class="form-control" id="rollid"
maxlength="5" required="required" autocomplete="off">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Email id</label>
<div class="col-sm-10">
<input type="email" name="emailid" class="form-control" id="email"
required="required" autocomplete="off">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Gender</label>
<div class="col-sm-10">
<input type="radio" name="gender" value="Male" required="required"
checked="">Male <input type="radio" name="gender" value="Female"
required="required">Female <input type="radio" name="gender"
value="Other" required="required">Other
</div>
</div>

<div class="form-group">
<label for="date" class="col-sm-2 control-label">DOB</label>
<div class="col-sm-10">
```

```

<input type="date" name="dob" class="form-control" id="date">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Department</label>
<div class="col-sm-10">
<select name="department" class="form-control" id="default"
required="required">
<option value="">Select Department</option>
<?php $sql = "SELECT * from tbldepartments";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{    ?>
<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->DepartmentName); ?>&nbsp;
Semester-<?php echo htmlentities($result->Semester); ?>
</option>
<?php }} ?>
</select>
</div>
</div>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="submit"
class="btn btn-primary">Add</button>
</div>
</div>
</form>

```

## ➤ Edit Student

```

<form class="form-horizontal" method="post">
<?php

$sql = "SELECT
tblstudents.StudentName,tblstudents.RollId,tblstudents.RegDate,tblstudents.Stu
dentId,tblstudents.Status,tblstudents.StudentEmail,tblstudents.Gender,tblstude
nts.DOB,tbldepartments.DepartmentName,tbldepartments.Semester from tblstudents
join tbldepartments on tbldepartments.id=tblstudents.StudentId where
tblstudents.StudentId=:stid";
$query = $dbh->prepare($sql);
$query->bindParam(':stid',$stid,PDO::PARAM_STR);
$query->execute();

```

```

$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{   ?>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Full Name</label>
<div class="col-sm-10">
<input type="text" name="fullanme" class="form-control"
id="fullanme"
value="<?php echo htmlentities($result->StudentName)?>"
required="required" autocomplete="off">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Rooll Id</label>
<div class="col-sm-10">
<input type="text" name="rollid" class="form-control" id="rollid"
value="<?php echo htmlentities($result->RollId)?>" maxlength="5"
required="required" autocomplete="off">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Email id</label>
<div class="col-sm-10">
<input type="email" name="emailid" class="form-control" id="email"
value="<?php echo htmlentities($result->StudentEmail)?>"
required="required" autocomplete="off">
</div>
</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Gender</label>
<div class="col-sm-10">
<?php $gndr=$result->Gender;
if($gndr=="Male")
{
?>
<input type="radio" name="gender" value="Male" required="required"
checked>Male <input type="radio" name="gender" value="Female"
required="required">Female <input type="radio" name="gender"
value="Other" required="required">Other

```

```

<?php }?>
<?php
if($gndr=="Female")
{
?>
<input type="radio" name="gender" value="Male"
required="required">Male <input type="radio" name="gender"
value="Female" required="required" checked>Female <input
type="radio" name="gender" value="Other"
required="required">Other
<?php }?>
<?php
if($gndr=="Other")
{
?>
<input type="radio" name="gender" value="Male"
required="required">Male <input type="radio" name="gender"
value="Female" required="required">Female <input type="radio"
name="gender" value="Other" required="required" checked>Other
<?php }?>

```

```

</div>
</div>

```

```

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Department</label>
<div class="col-sm-10">
<input type="text" name="departmentname" class="form-control"
id="departmentname"
value="<?php echo htmlentities($result->DepartmentName)?>(<?php echo
htmlentities($result->Semester)?>)"
readonly>
</div>
</div>
<div class="form-group">
<label for="date" class="col-sm-2 control-label">DOB</label>
<div class="col-sm-10">
<input type="date" name="dob" class="form-control"
value="<?php echo htmlentities($result->DOB)?>" id="date">
</div>
</div>
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Reg Date: </label>
<div class="col-sm-10">
<?php echo htmlentities($result->RegDate)?>
</div>

```

```

</div>

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Status</label>
<div class="col-sm-10">
<?php $stats=$result->Status;
if($stats=="1")
{
?>
<input type="radio" name="status" value="1" required="required"
checked>Active <input type="radio" name="status" value="0"
required="required">Block
<?php }?>
<?php
if($stats=="0")
{
?>
<input type="radio" name="status" value="1"
required="required">Active <input type="radio" name="status"
value="0" required="required" checked>Block
<?php }?>

</div>
</div>

<?php }} ?>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="submit"
class="btn btn-warning">Update</button>
</div>
</div>
</form>

```

## ➤ Adding Department

```

<form method="post">
<div class="form-group has-success">
<label for="success" class="control-label">Department Name</label>
<div class="">
<input type="text" name="departmentname" class="form-control"
required="required" id="success">
<span class="help-block">Eg- Computer (CE), Information Techno.
(IT),Chemical (ChE) etc</span>
</div>

```

```

</div>
<div class="form-group has-success">
<label for="success" class="control-label">Semester</label>
<div class="">
<input type="number" name="semester" required="required"
class="form-control" id="success">
<span class="help-block">Eg- 1,2,4,5 etc</span>
</div>
</div>
<div class="form-group has-success">

<div class="">
<button type="submit" name="submit"
class="btn btn-success btn-labeled">Submit<span
class="btn-label btn-label-right"><i
class="fa fa-check"></i></span></button>
</div>

</div>
</form>

```

## ➤ Edit Department

```

<form method="post">
<?php
$did=intval($_GET['departmentid']);
$sql = "SELECT * from tbldepartments where id=:did";
$query = $dbh->prepare($sql);
$query->bindParam(':did',$did,PDO::PARAM_STR);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{   ?>

<div class="form-group has-success">
<label for="success" class="control-label">Department Name</label>
<div class="">
<input type="text" name="departmentname"
value="<?php echo htmlentities($result->DepartmentName);?>"
required="required" class="form-control" id="success">
<span class="help-block">Eg- Computer (CE), Information Techno.
(IT),Chemical (ChE) etc</span>
</div>
</div>

<div class="form-group has-success">

```

```

<label for="success" class="control-label">Semester</label>
<div class="">
<input type="number" name="semester"
value="<?php echo htmlentities($result->Semester);?>"
required="required" class="form-control" id="success">
<span class="help-block">Eg- 1,2,4,5 etc</span>
</div>
</div>

<?php }} ?>
<div class="form-group has-success">

<div class="">
<button type="submit" name="update"
class="btn btn-success btn-labeled">Update<span
class="btn-label btn-label-right"><i
class="fa fa-check"></i></span></button>
</div>

</div>

</form>

```

## ❖ Subject Module

### ➤ Adding Subject

```

<form class="form-horizontal" method="post">
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Department</label>
<div class="col-sm-10">
<select name="class" class="form-control" id="default"
required="required">
<option value="">Select Class</option>
<?php $sql = "SELECT * from tbldepartments";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{
?>
<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->DepartmentName); ?>&nbsp;
Semester-<?php echo htmlentities($result->Semester); ?>
</option>

```

```

<?php }} ?>
</select>
</div>
</div>
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Subject</label>
<div class="col-sm-10">
<select name="subject" class="form-control" id="default"
required="required">
<option value="">Select Subject</option>
<?php $sql = "SELECT * from tblsubjects";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{    ?>
<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->SubjectName); ?></option>
<?php }} ?>
</select>
</div>
</div>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="submit"
class="btn btn-primary">Add</button>
</div>
</div>
</form>

```

## ➤ Edit Subject

```

<form class="form-horizontal" method="post">

<?php
$sid=intval($_GET['subjectid']);
$sql = "SELECT * from tblsubjects where id=:sid";
$query = $dbh->prepare($sql);
$query->bindParam(':sid',$sid,PDO::PARAM_STR);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)

```



```

{
foreach($results as $result)
{    ?>
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Subject Name</label>
<div class="col-sm-10">
<input type="text" name="subjectname" value="<?php echo htmlentities($result->SubjectName);?>" class="form-control" id="default" placeholder="Subject Name"
required="required">
</div>
</div>
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Subject Code</label>
<div class="col-sm-10">
<input type="text" name="subjectcode" class="form-control" value="<?php echo htmlentities($result->SubjectCode);?>" id="default" placeholder="Subject
Code" required="required">
</div>
</div>
<?php }} ?>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="Update" class="btn btn-primary">Update</button>
</div>
</div>
</form>

```

## ❖ Subject Combination Module

### ➤ Adding Subject Combination

```

<form class="form-horizontal" method="post">
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Department</label>
<div class="col-sm-10">
<select name="class" class="form-control" id="default"
required="required">
<option value="">Select Class</option>
<?php $sql = "SELECT * from tbldepartments";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{    ?>

```

```

<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->DepartmentName); ?>&nbsp;  
Semester-<?php echo htmlentities($result->Semester); ?>
</option>
<?php }} ?>
</select>
</div>
</div>
<div class="form-group">
<label for="default" class="col-sm-2 control-label">Subject</label>
<div class="col-sm-10">
<select name="subject" class="form-control" id="default"
required="required">
<option value="">Select Subject</option>
<?php $sql = "SELECT * from tblsubjects";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{    ?>
<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->SubjectName); ?></option>
<?php }} ?>
</select>
</div>
</div>
</div>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="submit"
class="btn btn-primary">Add</button>
</div>
</div>
</form>

```

### ➤ Edit Subject Combination

```

<table id="example" class="display table table-striped table-bordered"
cellspacing="0" width="100%">
<thead>
<tr>
<th>#</th>
<th>Department and Semester</th>
<th>Subject </th>

```

```

<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tfoot>
<tr>
<th>#</th>
<th>Department and Semester</th>
<th>Subject </th>
<th>Status</th>
<th>Action</th>
</tr>
</tfoot>
<tbody>
<?php $sql = "SELECT
tbldepartments.DepartmentName,tbldepartments.Semester,tblsubjects.SubjectName,
tblsubjectcombination.id as scid,tblsubjectcombination.status from
tblsubjectcombination join tbldepartments on
tbldepartments.id=tblsubjectcombination.DepartmentId join tblsubjects on
tblsubjects.id=tblsubjectcombination.SubjectId";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $result)
{    ?>
<tr>
<td><?php echo htmlentities($cnt);?></td>
<td><?php echo htmlentities($result->DepartmentName);?> &nbsp;
Semester-<?php echo htmlentities($result->Semester);?></td>
<td><?php echo htmlentities($result->SubjectName);?></td>
<td><?php $stts=$result->status;
if($stts=='0')
{
echo htmlentities('Inactive');
}
else
{
echo htmlentities('Active');
}
?></td>

<td>
<?php if($stts=='0')
{ ?>

```

```

<a href="manage-subjectcombination.php?acid=<?php echo htmlentities($result-
>scid);?>"
onclick="confirm('do you really want to ativate this subject');"><i
class="fa fa-check" title="Acticvate Record"></i>
</a><?php } else {?>

<a href="manage-subjectcombination.php?did=<?php echo htmlentities($result-
>scid);?>"
onclick="confirm('do you really want to deativate this subject');"><i
class="fa fa-times" title="Deactivate Record"></i>
</a>
<?php }?>
</td>
</tr>
<?php $cnt=$cnt+1;}} ?>

</tbody>
</table>

```

## ❖ Result Module

### ➤ Adding Result

```

<form class="form-horizontal" method="post">

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Department</label>
<div class="col-sm-10">
<select name="department" class="form-control clid"
id="departmentid" onChange="getStudent(this.value);"
required="required">
<option value="">Select Department</option>
<?php $sql = "SELECT * from tbldepartments";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{   ?>
<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->DepartmentName); ?>&nbsp;
Semester-<?php echo htmlentities($result->Semester); ?>
</option>
<?php }} ?>
</select>
</div>

```

```

</div>
<div class="form-group">
<label for="date" class="col-sm-2 control-label ">Student Name</label>
<div class="col-sm-10">
<select name="studentid" class="form-control stdid" id="studentid"
required="required" onChange="getresult(this.value);">
</select>
</div>
</div>

<div class="form-group">

<div class="col-sm-10">
<div id="reslt">
</div>
</div>
</div>

<div class="form-group">
<label for="date" class="col-sm-2 control-label">Subjects</label>
<div class="col-sm-1">
<div id="subject">
</div>
</div>
</div>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="submit" id="submit"
class="btn btn-primary">Declare Result</button>
</div>
</div>
</form>

```

## ❖ Attendance Module

### ➤ Adding Attendance

```

<form class="form-horizontal" method="post">

<div class="form-group">
<label for="default" class="col-sm-2 control-label">Department</label>
<div class="col-sm-10">
<select name="department" class="form-control clid"
id="departmentid" onChange="getStudent(this.value);"
required="required">

```

```

<option value="">Select Department</option>
<?php $sql = "SELECT * from tbldepartments";
$query = $dbh->prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
if($query->rowCount() > 0)
{
foreach($results as $result)
{    ?>
<option value="<?php echo htmlentities($result->id); ?>">
<?php echo htmlentities($result->DepartmentName); ?>&nbsp;
Semester-<?php echo htmlentities($result->Semester); ?>
</option>
<?php }} ?>
</select>
</div>
</div>
<div class="form-group">
<label for="date" class="col-sm-2 control-label ">Student Name</label>
<div class="col-sm-10">
<select name="studentid" class="form-control stdid" id="studentid"
required="required" onChange="getattendance(this.value);">
</select>
</div>
</div>

<div class="form-group">

<div class="col-sm-10">
<div id="reslt">
</div>
</div>
</div>

<div class="form-group">
<label for="date" class="col-sm-2 control-label">Subjects</label>
<div class="col-sm-10">
<div id="subject">
</div>
</div>
</div>

<div class="form-group">
<div class="col-sm-offset-2 col-sm-10">
<button type="submit" name="submit" id="submit"
class="btn btn-primary">Declare Attendance</button>
</div>

```

```
</div>
</form>
```

## ❖ Student Module

### ➤ View Result

```
<table class="table table-hover table-bordered">
<thead>
<tr>
<th>#</th>
<th>Subject</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<?php
// Code for result

$query = "select
t.StudentName,t.RollId,t.DepartmentId,t.marks,SubjectId,tblsubjects.SubjectName
e from (select sts.StudentName,sts.RollId,sts.DepartmentId,tr.marks,SubjectId
from tblstudents as sts join tblresult as tr on tr.StudentId=sts.StudentId)
as t join tblsubjects on tblsubjects.id=t.SubjectId where (t.RollId=:rollid
and t.DepartmentId=:departmentid)";
$query= $dbh -> prepare($query);
$query->bindParam(':rollid',$rollid,PDO::PARAM_STR);
$query->bindParam(':departmentid',$departmentid,PDO::PARAM_STR);
$query-> execute();
$results = $query -> fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($countrow=$query->rowCount(>0)
{
foreach($results as $result){
?>

<tr>
<th scope="row"><?php echo htmlentities($cnt);?></th>
<td><?php echo htmlentities($result->SubjectName);?></td>
<td><?php echo htmlentities($totalmarks=$result->marks);?>
</td>
</tr>
<?php
$totlcount+=$totalmarks;
$cnt++;}
?>
<tr>
<th scope="row" colspan="2">Total Marks</th>
```

```

<td><b><?php echo htmlentities($totlcount); ?></b> out of
<b><?php echo htmlentities($outof=($cnt-1)*100); ?></b>
</td>
</tr>
<tr>
<th scope="row" colspan="2">Percentage</th>
<td><b><?php echo htmlentities($totlcount*(100)/$outof); ?>
%</b></td>
</tr>

<?php } else { ?>
<div class="alert alert-warning left-icon-alert" role="alert">
<strong>Notice!</strong> Your result not declare yet
<?php }
?>
</div>
<?php
} else
{?>

<div class="alert alert-danger left-icon-alert" role="alert">
<strong>Oh snap!</strong>
<?php
echo htmlentities("Invalid Roll Id");
}
?>
</div>
</tbody>
</table>

```

## ➤ View Attendance

```

<table class="table table-hover table-bordered">
<thead>
<tr>
<th>#</th>
<th>Subject</th>
<th>Attendance</th>
</tr>
</thead>
<tbody>

<?php
// Code for attendance

$query ="select
t.StudentName,t.RollId,t.DepartmentId,t.attendance,SubjectId,tblsubjects.Subje
ctName from (select

```



```

sts.StudentName,sts.RollId,sts.DepartmentId,tr.attendance,SubjectId from
tblstudents as sts join tblattendance as tr on tr.StudentId=sts.StudentId) as
t join tblsubjects on tblsubjects.id=t.SubjectId where (t.RollId=:rollid and
t.DepartmentId=:departmentid)";
$query= $dbh -> prepare($query);
$query->bindParam(':rollid',$rollid,PDO::PARAM_STR);
$query->bindParam(':departmentid',$departmentid,PDO::PARAM_STR);
$query-> execute();
$results = $query -> fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($countrow=$query->rowCount()>0)
{

foreach($results as $result){

?>

<tr>
<th scope="row"><?php echo htmlentities($cnt);?></th>
<td><?php echo htmlentities($result->SubjectName);?></td>
<td><?php echo htmlentities($totalattendance=$result->attendance);?>
</td>
</tr>
<?php
$totlcount+=$totalattendance;
$cnt++;}
?>
<tr>
<th scope="row" colspan="2">Total attendance</th>
<td><b><?php echo htmlentities($totlcount); ?></b> out of
<b><?php echo htmlentities($outof=($cnt-1)*100); ?></b>
</td>
</tr>
<tr>
<th scope="row" colspan="2">Percentage</th>
<td><b><?php echo htmlentities($totlcount*(100)/$outof); ?>
%</b></td>
</tr>

<?php } else { ?>
<div class="alert alert-warning left-icon-alert" role="alert">
<strong>Notice!</strong> Your attendance not declare yet
<?php }
?>
</div>
<?php
} else
{?>

```

```
<div class="alert alert-danger left-icon-alert" role="alert">
<strong>Oh snap!</strong>
<?php
echo htmlentities("Invalid Roll Id");
}
?>
</div>
</tbody>

</table>
```

# TESTING

## Testing Strategy:

For testing our application, a mixed approach integration testing and regression testing is used.

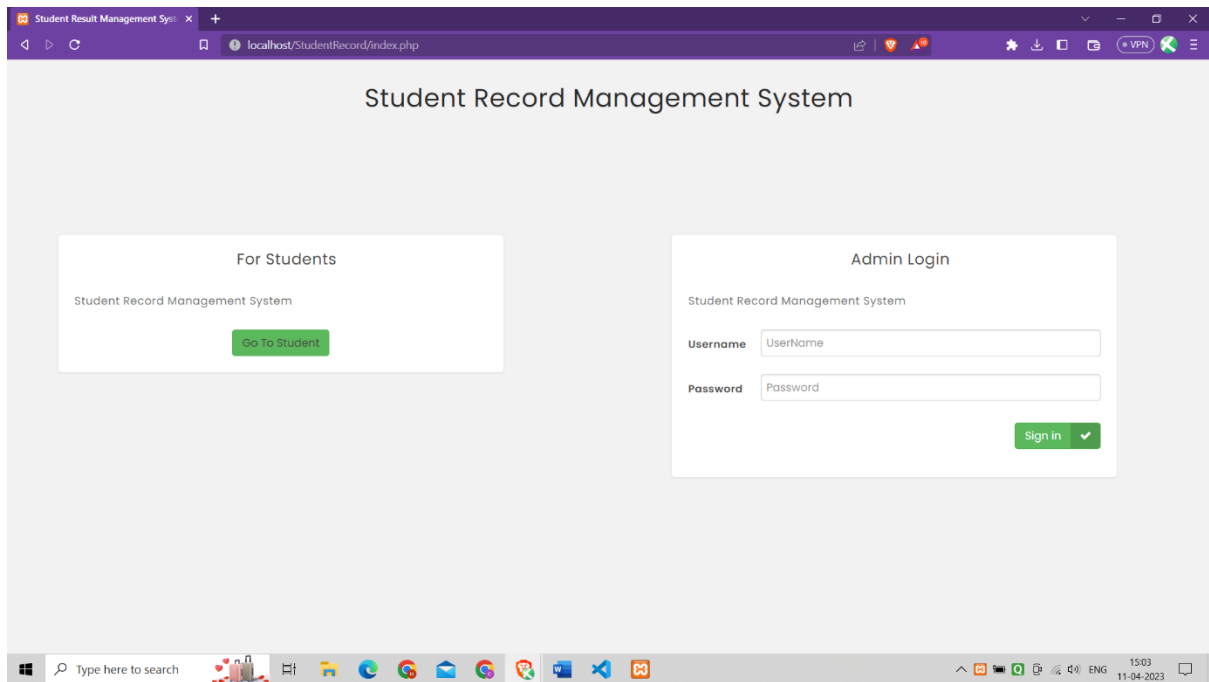
- ❖ **Integration testing:** Each main part of application is tested after each small function is tested first and then combine them and test that main part.
- ❖ **Regression testing:** After main part of application is created, added them in the whole system, and then test the whole system that make sure the whole system is work fine after adding some main part.
- ❖ **Manual Testing:** Manual testing is used to find and fix the bug in our application

| NO | Test Scenario                    | Expected Output                            | Actual Output                  | Status  |
|----|----------------------------------|--|--------------------------------|---------|
| 1  | Entered Admin username, password | Admin Dashboard                            | Admin Dashboard                | Success |
| 2  | Adding New Student               | New student should be added to DB          | New student added              | Success |
| 3  | Update student details           | Updated details should be in DB            | Details changed in DB          | Success |
| 4  | Delete a student                 | Student should be removed from DB          | Student gets removed from DB   | Success |
| 5  | Adding Subject                   | New Subject should be Added                | Subject gets added             | Success |
| 6  | Update Subject                   | Updated Subject details should be in DB    | Details changed in DB          | Success |
| 7  | Adding Department                | New Department should be Added             | Department gets added          | Success |
| 8  | Update Department                | Updated Department details should be in DB | Details changed in DB          | Success |
| 9  | Adding Subject Combination       | New Subject Combination should be Added    | Subject Combination gets added | Success |
| 10 | Update Subject Combination       | Updated Subject Combination                | Details changed in DB          | Success |

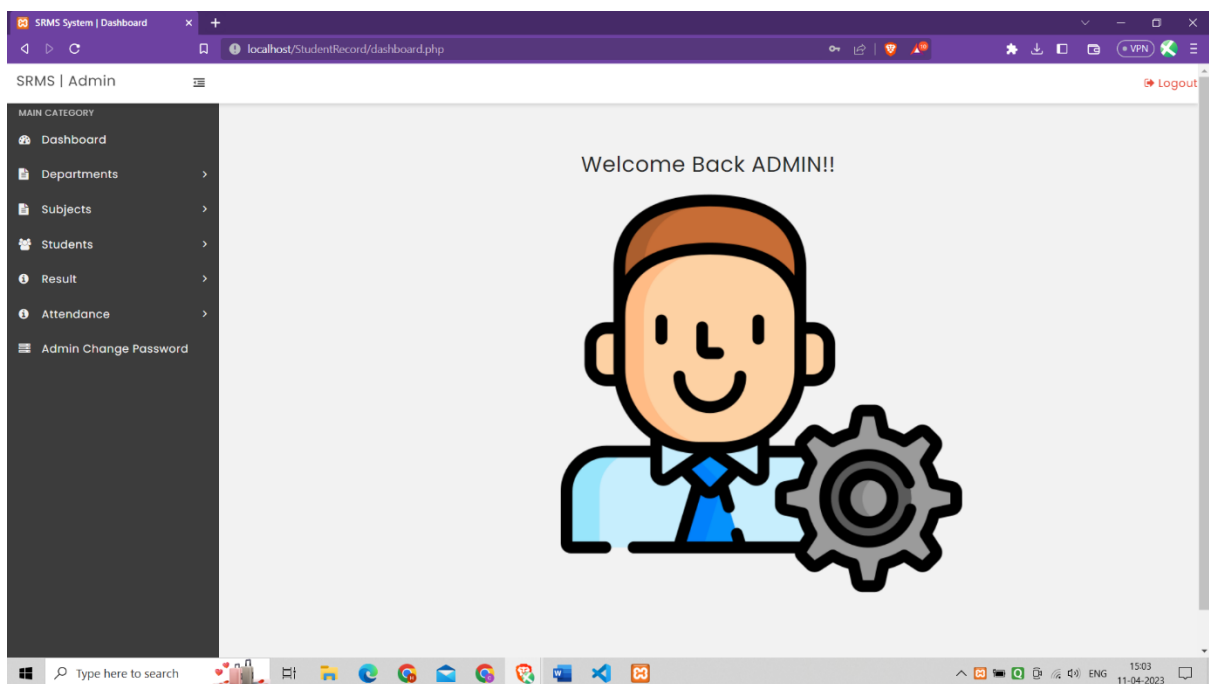
|    |  |  |                          |         |
|----|--|--|--------------------------|---------|
|    |  | details should be in DB                          |                          |         |
| 11 | Add Result   | Result should be added of a student              | Result added             | Success |
| 12 | Update Result  | Updated result data should be in DB              | Details changed in DB    | Success |
| 13 | Redeclaring result of same student                         | Error for redeclaring result should be shown     | Error message appeared   | Success |
| 14 | Add Attendance   | Attendance should be added of a student          | Attendance added         | Success |
| 15 | Update Attendance  | Updated attendance data should be in DB          | Details changed in DB    | Success |
| 16 | Redeclaring attendance of same student                     | Error for redeclaring attendance should be shown | Error message appeared   | Success |
| 17 | View result by student entering roll id and department     | Result should be shown                           | Result showed            | Success |
| 18 | View result by entering wrong details                      | Error message should be shown                    | Error message appeared   | Success |
| 19 | View attendance by student entering roll id and department | Attendance report should be shown                | Attendance report showed | Success |
| 20 | View attendance by entering wrong details                  | Error message should be shown                    | Error message appeared   | Success |

## Screen Shots:

### Index page:



### Admin Dashboard:



## Add Department:

SRMS Admin Create Department

localhost/StudentRecord/create-department.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Create Department

Home / Departments / Create Department

Create Department

Department Name

Eg- Computer (CE), Information Techno. (IT), Chemical (CHE) etc

Semester

Eg- 1,2,4,5 etc

Submit

## Manage Department:

SRMS Admin Manage Department

localhost/StudentRecord/manage-department.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Manage Department

Home / Departments / Manage Departments

View Departments Info

Show 10 entries

Search:

| # | Department Name                    | Semester | Creation Date       | Action |
|---|------------------------------------|----------|---------------------|--------|
| 1 | Computer Engineering               | 1        | 2023-04-02 03:25:23 |        |
| 2 | Computer Engineering               | 2        | 2023-04-02 03:25:29 |        |
| 3 | Chemical Engineering               | 1        | 2023-04-02 03:25:39 |        |
| 4 | Chemical Engineering               | 2        | 2023-04-02 03:25:43 |        |
| 5 | Information Technology Engineering | 1        | 2023-04-02 03:26:08 |        |
| 6 | Information Technology Engineering | 2        | 2023-04-02 03:26:12 |        |
| # | Department Name                    | Semester | Creation Date       | Action |

Showing 1 to 6 of 6 entries

Previous 1 Next

## Update Department:

The screenshot shows a web browser window with the URL `localhost/StudentRecord/edit-department.php?departmentid=1`. The page title is "SRMS | Admin" and there is a "Logout" link in the top right corner. On the left, there is a sidebar menu with the following items: Dashboard, Departments, Subjects, Students, Result, Attendance, and Admin Change Password. The main content area is titled "Update Department" and has a breadcrumb trail: Home / Departments / Update Department. The form is titled "Update Department info" and contains the following fields: "Department Name" with the value "Computer Engineering" and a hint "Eg- Computer (CE), Information Techno. (IT), Chemical (ChE) etc", "Semester" with the value "1" and a hint "Eg- 1,2,4,5 etc", and a green "Update" button with a checkmark icon.

## Add Student:

The screenshot shows a web browser window with the URL `localhost/StudentRecord/add-students.php`. The page title is "SRMS | Admin" and there is a "Logout" link in the top right corner. On the left, there is a sidebar menu with the following items: Dashboard, Departments, Subjects, Students, Result, Attendance, and Admin Change Password. The main content area is titled "Student Admission" and has a breadcrumb trail: Home / Student Admission. The form is titled "Fill the Student info" and contains the following fields: "Full Name", "Roll id", "Email id", "Gender" with radio buttons for "Male", "Female", and "Other" (where "Male" is selected), "DOB" with a date picker showing "dd-mm-yyyy", and "Department" with a dropdown menu showing "Select Department". There is a blue "Add" button at the bottom of the form.

## Manage Student:

SRMS Admin Manage Students

localhost/StudentRecord/manage-students.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Manage Students

Home / Students / Manage Students

View Students Info

Show 10 entries

Search:

| # | Student Name   | Roll Id | Department              | Reg Date            | Status | Action               |
|---|----------------|---------|-------------------------|---------------------|--------|----------------------|
| 1 | Dhruv H Variya | CE001   | Computer Engineering(1) | 2023-04-02 04:46:16 | Active | <a href="#">Edit</a> |
| 2 | Deep Padsala   | CE003   | Computer Engineering(2) | 2023-04-02 23:19:33 | Active | <a href="#">Edit</a> |
| 3 | hima           | CE012   | Computer Engineering(1) | 2023-04-03 16:11:02 | Active | <a href="#">Edit</a> |

Showing 1 to 3 of 3 entries

Previous 1 Next

## Update Student Details:

SMS Admin Edit Student

localhost/StudentRecord/edit-student.php?sid=2

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Student Admission

Home / Student Admission

Fill the Student info

Full Name: Deep Padsala

Roll Id: CE003

Email id: ce003@ddu.ac.in

Gender: ☒ Male ☐ Female ☐ Other

Department: Computer Engineering(2)

DOB: 28-06-2002

Reg Date: 2023-04-02 23:19:33

Status: ☒ Active ☐ Block

Update



## Create Subject:

SRMS Admin Subject Creation

localhost/StudentRecord/create-subject.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Subject Creation

Home / Subjects / Create Subject

#### Create Subject

Subject Name

Subject Code

## Manage Subject:

SRMS Admin Manage Subjects

localhost/StudentRecord/manage-subjects.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Manage Subjects

Home / Subjects / Manage Subjects

#### View Subjects Info

Show  entries

Search:

| # | Subject Name                   | Subject Code | Creation Date       | Updation Date       | Action               |
|---|--------------------------------|--------------|---------------------|---------------------|----------------------|
| 1 | MATHEMATICS I                  | BS101        | 2023-04-02 02:50:09 | 0000-00-00 00:00:00 | <a href="#">Edit</a> |
| 2 | BASIC ELECTRICAL ENGINEERING   | ES101        | 2023-04-02 02:50:28 | 0000-00-00 00:00:00 | <a href="#">Edit</a> |
| 3 | PROBLEM SOLVING PROGRAMMING I  | ES102        | 2023-04-02 02:50:44 | 0000-00-00 00:00:00 | <a href="#">Edit</a> |
| 4 | MATHEMATICS II                 | BS201        | 2023-04-02 02:51:27 | 0000-00-00 00:00:00 | <a href="#">Edit</a> |
| 5 | PROBLEM SOLVING PROGRAMMING II | ES201        | 2023-04-02 02:51:42 | 0000-00-00 00:00:00 | <a href="#">Edit</a> |
| 6 | PHYSICS                        | ES202        | 2023-04-02 02:51:54 | 0000-00-00 00:00:00 | <a href="#">Edit</a> |
| # | Subject Name                   | Subject Code | Creation Date       | Updation Date       | Action               |

Showing 1 to 6 of 6 entries

Previous **1** Next

## Create Subject Combination:

SRMS Admin Subject Combination

localhost/StudentRecord/add-subjectcombination.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Add Subject Combination

Home / Subjects / Add Subject Combination

Add Subject Combination

Department: Select Class

Subject: Select Subject

Add

## Manage Subject Combination:

SRMS Admin Manage Subjects Combination

localhost/StudentRecord/manage-subjectcombination.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Manage Subjects Combination

Home / Subjects / Manage Subjects Combination

View Subjects Combination Info

Show 10 entries

Search:

| # | Department and Semester         | Subject                        | Status   | Action |
|---|---------------------------------|--------------------------------|----------|--------|
| 1 | Computer Engineering Semester-1 | MATHEMATICS I                  | Active   | ✕      |
| 2 | Computer Engineering Semester-1 | BASIC ELECTRICAL ENGINEERING   | Active   | ✕      |
| 3 | Computer Engineering Semester-1 | PROBLEM SOLVING PROGRAMMING I  | Active   | ✕      |
| 4 | Computer Engineering Semester-2 | MATHEMATICS II                 | Active   | ✕      |
| 5 | Computer Engineering Semester-2 | PROBLEM SOLVING PROGRAMMING II | Active   | ✕      |
| 6 | Computer Engineering Semester-2 | MATHEMATICS II                 | Inactive | ✓      |
| 7 | Computer Engineering Semester-2 | PHYSICS                        | Active   | ✕      |
| # | Department and Semester         | Subject                        | Status   | Action |

Showing 1 to 7 of 7 entries

Previous 1 Next

## Add Result:

SRMS Admin Add Result

localhost/StudentRecord/add-result.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Declare Result

Home / Student Result

Department: Computer Engineering Semester-1

Student Name: Select Student

Subjects:

- BASIC ELECTRICAL ENGINEERING  
Enter marks out of 100
- MATHEMATICS I  
Enter marks out of 100
- PROBLEM SOLVING PROGRAMMING I  
Enter marks out of 100

Declare Result

## Manage Result:

SRMS Admin Manage Students

localhost/StudentRecord/manage-results.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Manage Results

Home / Results / Manage Results

View Students Result

Show 10 entries

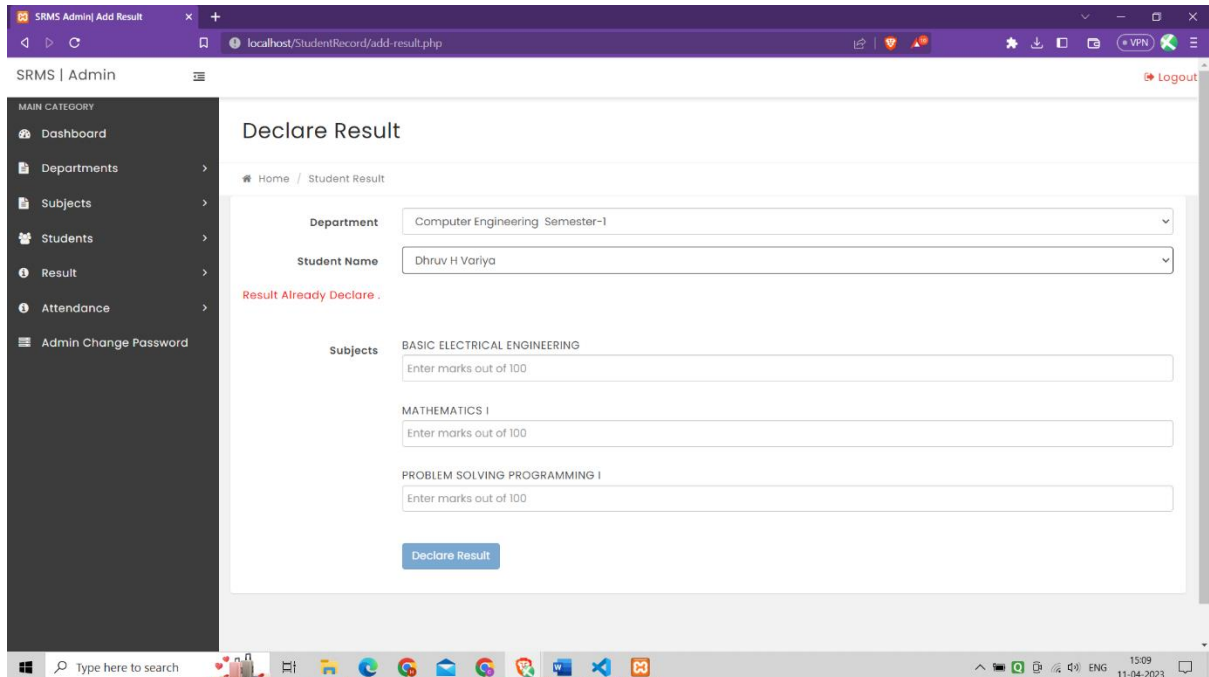
Search:

| # | Student Name   | Roll Id | Department Id | Marks | Subject Name                  | Action               |
|---|----------------|---------|---------------|-------|-------------------------------|----------------------|
| 1 | Dhruv H Variya | CE001   | 1             | 85    | BASIC ELECTRICAL ENGINEERING  | <a href="#">Edit</a> |
| 2 | Dhruv H Variya | CE001   | 1             | 95    | MATHEMATICS I                 | <a href="#">Edit</a> |
| 3 | Dhruv H Variya | CE001   | 1             | 100   | PROBLEM SOLVING PROGRAMMING I | <a href="#">Edit</a> |
| 4 | Deep Padsala   | CE003   | 2             | 56    | MATHEMATICS II                | <a href="#">Edit</a> |
| 5 | Deep Padsala   | CE003   | 2             | 95    | MATHEMATICS II                | <a href="#">Edit</a> |
| 6 | Deep Padsala   | CE003   | 2             | 97    | PHYSICS                       | <a href="#">Edit</a> |
| 7 | hima           | CE012   | 1             | 65    | BASIC ELECTRICAL ENGINEERING  | <a href="#">Edit</a> |
| 8 | hima           | CE012   | 1             | 99    | MATHEMATICS I                 | <a href="#">Edit</a> |
| 9 | hima           | CE012   | 1             | 56    | PROBLEM SOLVING PROGRAMMING I | <a href="#">Edit</a> |

Showing 1 to 9 of 9 entries

Previous Next

## Redeclare Result:



The screenshot shows the 'SRMS Admin Add Result' page in a web browser. The page has a dark sidebar on the left with a 'MAIN CATEGORY' menu containing: Dashboard, Departments, Subjects, Students, Result, Attendance, and Admin Change Password. The main content area is titled 'Declare Result' and includes a breadcrumb 'Home / Student Result'. It features two dropdown menus: 'Department' (set to 'Computer Engineering Semester-1') and 'Student Name' (set to 'Dhruv H Variya'). A red error message 'Result Already Declare .' is displayed. Below this, there are three input fields for subjects: 'BASIC ELECTRICAL ENGINEERING', 'MATHEMATICS I', and 'PROBLEM SOLVING PROGRAMMING I', each with a placeholder 'Enter marks out of 100'. A blue 'Declare Result' button is at the bottom.

SRMS Admin Add Result

localhost/StudentRecord/add-result.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Declare Result

Home / Student Result

Department: Computer Engineering Semester-1

Student Name: Dhruv H Variya

Result Already Declare .

Subjects

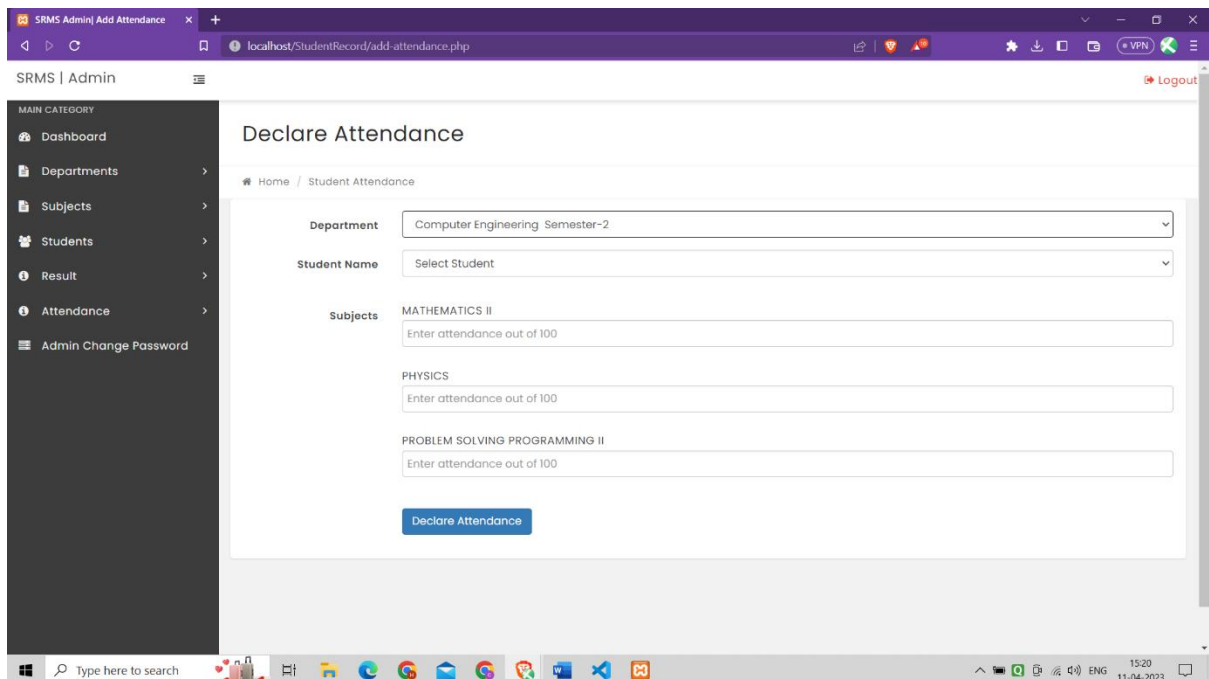
BASIC ELECTRICAL ENGINEERING  
Enter marks out of 100

MATHEMATICS I  
Enter marks out of 100

PROBLEM SOLVING PROGRAMMING I  
Enter marks out of 100

Declare Result

## Add Attendance:



The screenshot shows the 'SRMS Admin Add Attendance' page in a web browser. The page has a dark sidebar on the left with a 'MAIN CATEGORY' menu containing: Dashboard, Departments, Subjects, Students, Result, Attendance, and Admin Change Password. The main content area is titled 'Declare Attendance' and includes a breadcrumb 'Home / Student Attendance'. It features two dropdown menus: 'Department' (set to 'Computer Engineering Semester-2') and 'Student Name' (set to 'Select Student'). Below this, there are three input fields for subjects: 'MATHEMATICS II', 'PHYSICS', and 'PROBLEM SOLVING PROGRAMMING II', each with a placeholder 'Enter attendance out of 100'. A blue 'Declare Attendance' button is at the bottom.

SRMS Admin Add Attendance

localhost/StudentRecord/add-attendance.php

SRMS | Admin

Logout

MAIN CATEGORY

- Dashboard
- Departments
- Subjects
- Students
- Result
- Attendance
- Admin Change Password

### Declare Attendance

Home / Student Attendance

Department: Computer Engineering Semester-2

Student Name: Select Student

Subjects

MATHEMATICS II  
Enter attendance out of 100

PHYSICS  
Enter attendance out of 100

PROBLEM SOLVING PROGRAMMING II  
Enter attendance out of 100

Declare Attendance

## Manage Attendance:

SRMS Admin Manage Students

localhost/StudentRecord/manage-attendances.php

SRMS | Admin

Logout

### Manage Attendance

Home / Attendance / Manage Attendance

View Students Attendance

Show 10 entries Search:

| # | Student Name   | Roll Id | Department Name | Attendance | Subject Name                  | Action |
|---|----------------|---------|-----------------|------------|-------------------------------|--------|
| 1 | Dhruv H Variya | CE001   | 1               | 99         | BASIC ELECTRICAL ENGINEERING  |        |
| 2 | Dhruv H Variya | CE001   | 1               | 98         | MATHEMATICS I                 |        |
| 3 | Dhruv H Variya | CE001   | 1               | 100        | PROBLEM SOLVING PROGRAMMING I |        |
| # | Student Name   | Roll Id | Department Name | Attendance | Subject Name                  | Action |

Showing 1 to 3 of 3 entries

Previous 1 Next

## Admin Password Change:

Admin change password

localhost/StudentRecord/change-password.php

SRMS | Admin

Logout

### Admin Change Password

Home / Admin change password

Admin Change Password

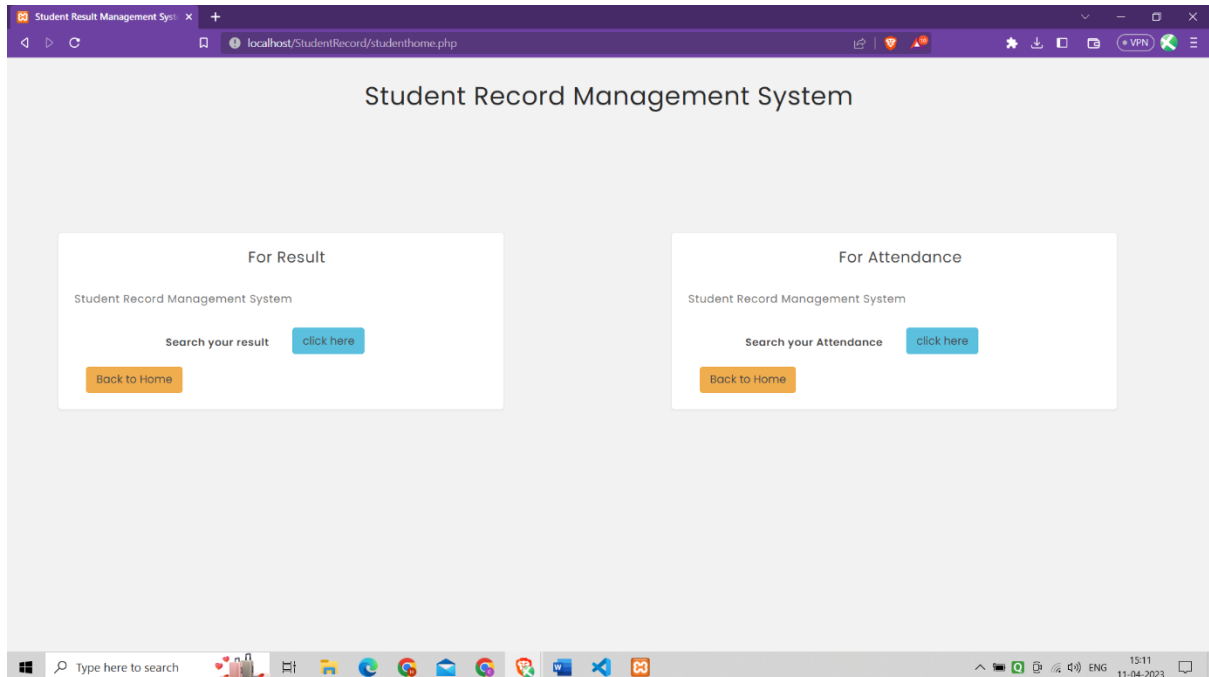
Current Password

New Password

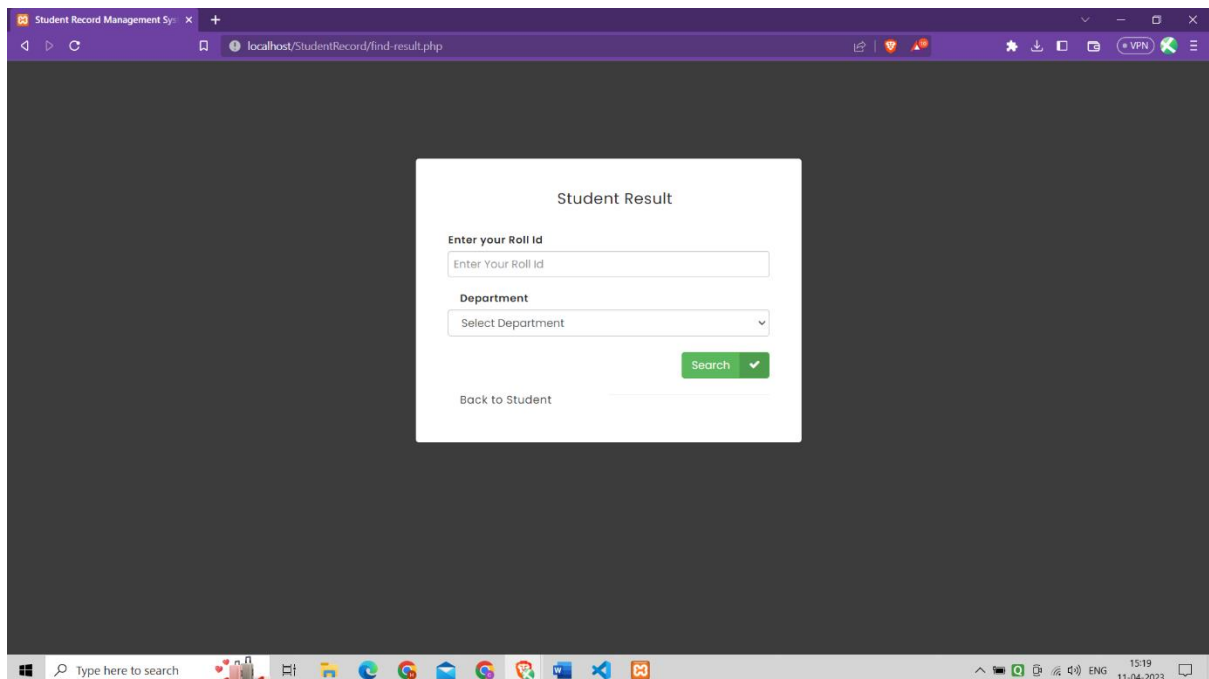
Confirm Password

Change

## Student Homepage:



## Find Result:



## Student Result:

The screenshot shows a web browser window titled "Student Record Management System" with the URL "localhost/StudentRecord/result.php". The page displays the following information:

Student Name : Dhruv H Variya  
Student Roll Id : CE001  
Student Department: Computer Engineering Semester-I

| #           | Subject                       | Marks             |
|-------------|-------------------------------|-------------------|
| 1           | BASIC ELECTRICAL ENGINEERING  | 85                |
| 2           | MATHEMATICS I                 | 95                |
| 3           | PROBLEM SOLVING PROGRAMMING I | 100               |
| Total Marks |                               | 280 out of 300    |
| Percentage  |                               | 93.333333333333 % |

Below the table is a button labeled "Back to Student".

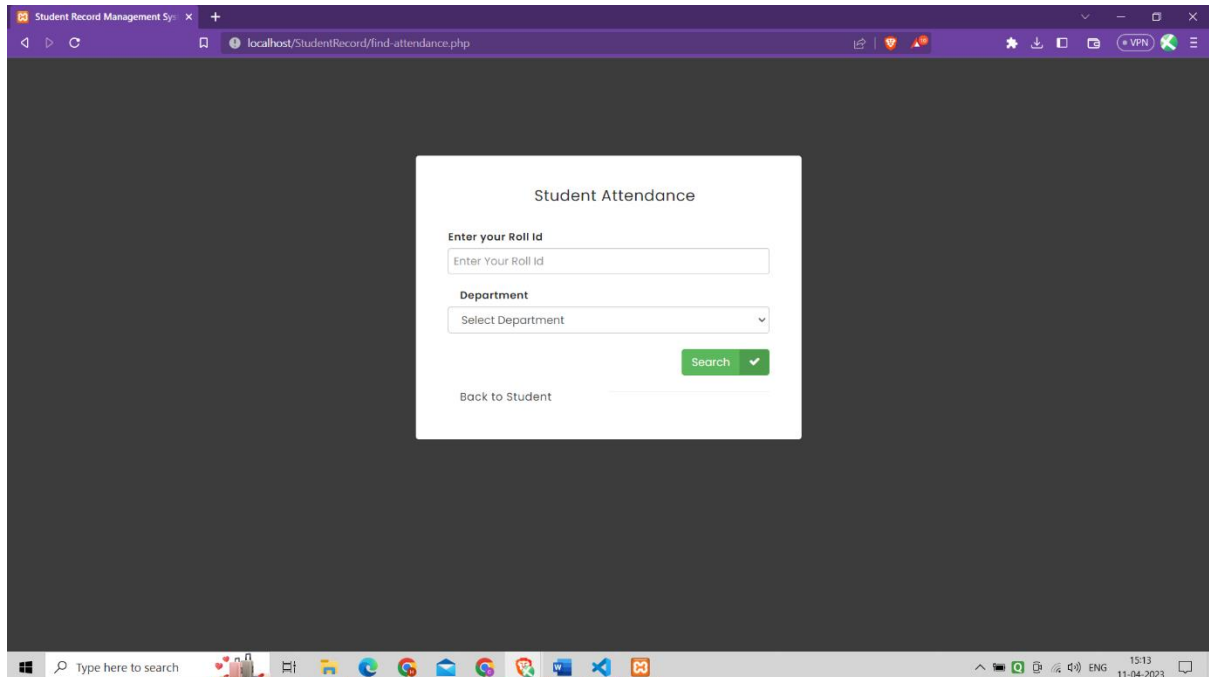
## For Wrong details in searching Result:

The screenshot shows the same web browser window, but with an error message displayed in a red box:

Oh snap! Invalid Roll Id

Below the error message is a button labeled "Back to Student".

## Find Attendance:

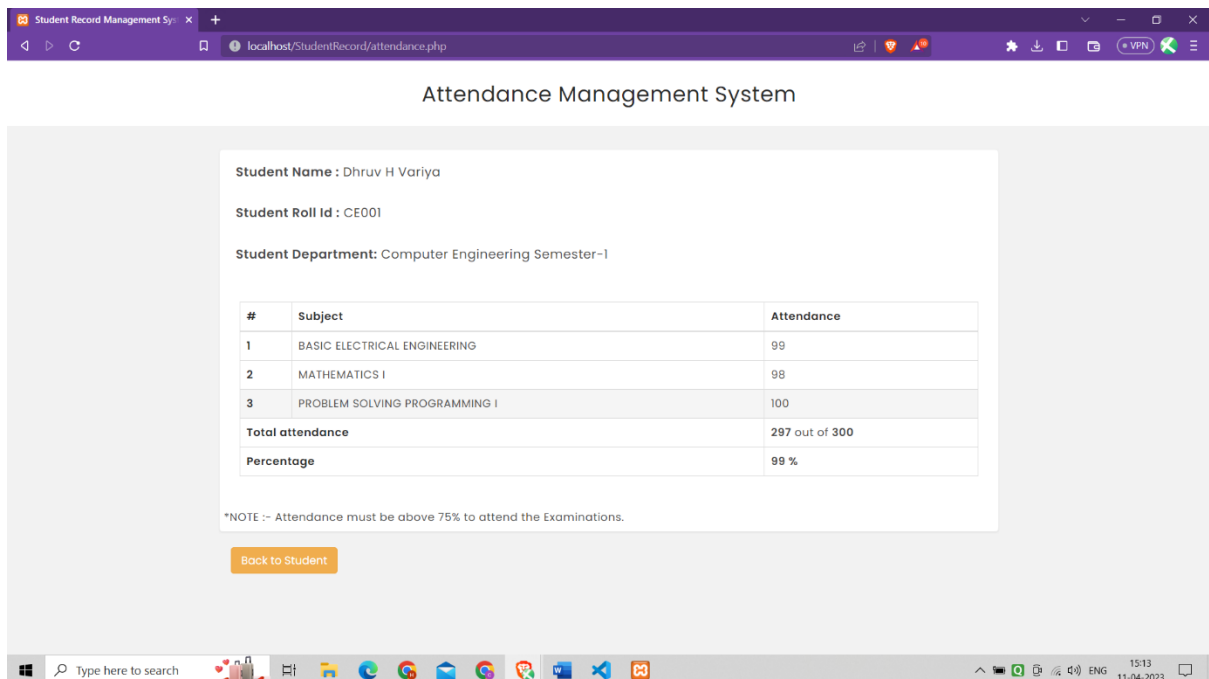


The screenshot shows a web browser window with the title 'Student Record Management System'. The address bar shows 'localhost/StudentRecord/find-attendance.php'. The main content area has a dark grey background with a white form titled 'Student Attendance'. The form contains the following elements:

- A label 'Enter your Roll Id' above a text input field with placeholder text 'Enter Your Roll Id'.
- A label 'Department' above a dropdown menu with the text 'Select Department'.
- A green 'Search' button with a checkmark icon.
- A link 'Back to Student' with a right-pointing arrow.

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right shows the time as 15:13 on 11-04-2023.

## Student Attendance:



The screenshot shows a web browser window with the title 'Student Record Management System'. The address bar shows 'localhost/StudentRecord/attendance.php'. The main content area has a light grey background with a white box containing the following information:

**Student Name :** Dhruv H Variya  
**Student Roll Id :** CE001  
**Student Department:** Computer Engineering Semester-I

| #                       | Subject                       | Attendance            |
|-------------------------|-------------------------------|-----------------------|
| 1                       | BASIC ELECTRICAL ENGINEERING  | 99                    |
| 2                       | MATHEMATICS I                 | 98                    |
| 3                       | PROBLEM SOLVING PROGRAMMING I | 100                   |
| <b>Total attendance</b> |                               | <b>297 out of 300</b> |
| <b>Percentage</b>       |                               | <b>99 %</b>           |

\*NOTE :- Attendance must be above 75% to attend the Examinations.

Below the table is an orange 'Back to Student' button.

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right shows the time as 15:13 on 11-04-2023.



## Not Declared Attendance:

Student Name : hima

Student Roll Id : CE012

Student Department: Computer Engineering Semester-I

**Notice!** Your attendance not declare yet

| # | Subject | Attendance |
|---|---------|------------|
|---|---------|------------|

\*NOTE :- Attendance must be above 75% to attend the Examinations.

[Back to Student](#)

## **CONCLUSION & FUTURE WORK**

### **Limitations and Future Work:**

#### ❖ Limitations

- In this system student can't login with their account.
- Students can only see their result and attendance.
- Every student can see anyone's result and attendance.
- Students can't download their marksheet.

#### ❖ Future Work

- Students are login with their roll no. and password.
- Students can download their result and get some privacy with their result and marksheet.

### **Conclusion:**

The Student Record Management System is an important tool for educational institutions to manage student records efficiently. The system should meet the functional and non-functional requirements specified in this document. The system should be user-friendly, reliable, and secure, and should undergo rigorous testing before deployment.

## ❖ **Bibliography:**

- Websites:

- ✦ <https://github.com/>
- ✦ <https://www.mysql.com/>
- ✦ <https://www.php.net/>
- ✦ <https://getbootstrap.com/>
- ✦ <https://www.apachefriends.org/>