



A
PROJECT REPORT ON
STUDENT RESULT MANAGEMENT

SUBMITTED TO

SHIVAJI UNIVERSITY, KOLHAPUR

IN FULFILLMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF COMPUTER APPLICATION

B.C.A .-III (SEM-VI)

SUBMITTED BY

MR. PATIL AKSHAY SANTOSH

MR. CHINAGE SUYASH SANJAY

UNDER GUIDANCE OF :- **MR. LAWAND J.H.**

THE PRINCIPAL

PADMABHUSHAN DR. VASANTRAODADA PATIL MAHAVIDYALAYA, TASGAON.

2023-2024



CERTIFICATE

The project entitled **“Student result management ”** submitted by **Mr. PATIL AKSHAY SANTOSH AND Mr. CHINAGE SUYASH SANJAY**. In partial fulfillment of the requirement of the award of degree **BACHELOR OF COMPUTER APPLICATION (B.C.A.) PART III (SEM-VI)** to **Shivaji University, Kolhapur** has been carried out by them under my guidance and satisfactorily completed during the academic year 2023-2024.

I therefore approve the same and wish them all the best in future career and bright success at every step of life.

Date:

Place: Tasgaon

EXAMINER

H.O.D

(MR. V.T. KUMBHAR)



GUIDE CERTIFICATE

This is to certify that,

Mr. PATIL AKSHAY SANTOSH

Mr. CHINAGE SUYASH SANJAY

has completed the necessary project on “**STUDENT RESULT MANAGEMENT**”,
in a satisfactory manner as a partial fulfillment of the requirement of the
B.C.A.III SEM-VI of Shivaji University, Kolhapur in academic year **2023-2024**.

I therefore approve the same and wish them all the best in the future career and bright
success at every step in their life.

Date:

Place: Tasgaon

GUIDE
(MR. LAWAND J.H)

DECLARATION

To,
The Principal,
Padmabhushan Dr. Vasantodada Patil Mahavidyalaya
Tasgaon, Sangli.

Respected Sir/Madam,

I hereby declare that this report is original in all aspects and the facts prescribed there in are true and not copied from any other project report.

If proved otherwise, I will solely responsible for the consequences arising there from.

Date:

Place: Tasgaon

Signature

Mr. Patil Akshay Santosh

Mr. Chinage Suyash Sanjay

ACKNOWLEDGEMENT

This is a great opportunity to acknowledge and to thanks all those persons without whose support and help this project would have been impossible. I would like to add a few heartfelt words for the people who were part of this project in numerous ways.

We take this opportunity to express our sincere gratitude and indebtedness to the project guide for giving us timely suggestions and directing towards perfection for their guidance, advice and constant encouragement, without which it would have been difficult for us to complete this project work successfully.

We express our sincere thanks to our project guide **Mr. Kumbhar V. T.** (**HOD**) Department of **computer application** who guided us during the entire session of the project. Her evaluations and suggestions have helped us enormously towards the completion of this project.

We also thank our project guide **Mr. Lawand J.H.** for his the motivation and support provided. Finally, we thank to all our colleagues, friends and that completion of our project work **“Student Result Management”**

Thanking You....

Mr. Patil Akshay Santosh

Mr. Chinage Suyash Sanjay

INDEX

1) Introduction to Project.....	
• Introduction	
• Existing System	
• Need and scope of System	
2) Proposed System.....	
• Objectives	
• Requirement Gathering.	
• SRS	
3) System Diagrams.....	
• DFD	
• ERD	
• System Requirements	
• Hardware	
• Software	
4) System Design.....	
• Database Design	
• Input-Output Design	
5) User Guideline.....	
• User Manual	
6) Conclusion and Suggestions.....	
• Conclusion and suggestions	

Chapter - I



Introduction

1.1 Introduction :

Welcome to the “**Student Result Management**” This Project, works as a Website for real estate classified system where admin can add properly details (like Class, Roll No , Marks and Name). and personal details. It developed as a dynamic website, this project developed by using HTML, CSS, JavaScript, MySQL, and PHP to deliver an efficient result to user.

In an year where technology plays an important role in education, our project aims to simplify and enhance the process of managing student results. The utilization of HTML ensures a robust and structured foundation for content, while CSS contributes to a visually appealing and responsive user interface, making the experience seamless across devices.

The integration of JavaScript introduces dynamic functionalities, enabling real-time interactions and providing an engaging user experience. The heart of the system lies in the database management facilitated by MySQL, ensuring secure storage and efficient retrieval of student result data.

PHP, the server-side scripting language, acts as the backbone of the project, facilitating the seamless communication between the user interface and the database. This synergy of technologies not only ensures a user-friendly interface but also guarantees the reliability and accuracy of the result management process.

Our Student Result Management Project isn't just a website; it's a comprehensive solution designed to cater to the diverse needs of educators, administrators, and students. From result input to retrieval, our project empowers educational institutions with a centralized and automated system, promoting transparency, efficiency, and accuracy in managing academic outcomes. Join us on this journey towards a more streamlined and technologically advanced approach to “**student result management**”.

1.2 Existing System :

In the existing system, the management of student results relies heavily on manual processes, making it time-consuming and prone to errors. The absence of a dedicated online platform makes it challenging for educators and students to access and update academic records efficiently. Traditional methods involve paper-based result sheets and manual data entry, leading to delays, inaccuracies, and a lack of real-time information.

Moreover, the existing system lacks the necessary functionalities for data analysis and reporting. Educators face challenges in tracking students' progress over time and identifying trends in academic performance. This limitation hinders informed decision-making and prevents timely interventions to support students who may be struggling.

The absence of a centralized database results in data fragmentation, making it difficult to maintain a comprehensive and organized repository of student information. This fragmentation hampers the overall efficiency of result management and reduces the accessibility of data for relevant stakeholders.

To address these shortcomings, the proposed Student Result Management Project leverages modern web technologies such as HTML, CSS, JavaScript, MySQL, and PHP to create a dynamic and user-friendly website. This digital platform aims to automate result management processes, ensuring accuracy, efficiency, and accessibility. Through the integration of a centralized database, the system will enable real-time updates and retrieval of student results, fostering transparency and accountability in the academic assessment process.

The project's utilization of HTML, CSS, and JavaScript ensures an intuitive and visually appealing user interface, enhancing the overall user experience for both educators and students. The incorporation of PHP facilitates seamless communication between the frontend and backend, while MySQL serves as a robust and scalable database solution, ensuring the secure storage and retrieval of student records.

In conclusion, the transition from the existing manual system to the proposed “**Student Result Management**” Project promises to revolutionize the way academic results are managed. By harnessing the power of web technologies, this project aims to streamline processes, reduce errors, and provide a comprehensive and accessible platform for result management, ultimately enhancing the overall efficiency of academic administration.

1.3 Need and scope of System :

1. Need:

Efficiency Improvement: The existing manual system is time-consuming and prone to errors. The need for an automated system arises to enhance the efficiency of result management, reducing the time spent on manual data entry and result processing.

Data Accuracy: Manual record-keeping increases the likelihood of errors in student results. An automated system is necessary to ensure data accuracy and eliminate discrepancies in academic records.

Real-time Accessibility: The current system lacks real-time accessibility to student results, hindering timely decision-making and interventions. A web-based platform will provide instant access to updated results for educators, students, and administrators.

Transparency and Accountability: Transparency in result management is essential for building trust. An automated system will contribute to transparency by maintaining an organized, centralized database that is easily accessible to authorized users, promoting accountability in the assessment process.

2. Scope:

Result Entry and Processing: The system will allow educators to input and process student results efficiently. Automation ensures quick and accurate recording of grades, reducing the chances of errors associated with manual data entry.

User-friendly Interface: The project will feature an intuitive and user-friendly interface developed using HTML, CSS, and JavaScript. This interface will facilitate easy navigation for educators, students, and administrators.

Centralized Database: A MySQL database will be implemented to store and manage student records in a centralized manner. This will eliminate data fragmentation and ensure a comprehensive repository of academic information.

Authentication and Authorization: The system will incorporate user authentication and authorization mechanisms to ensure that only authorized personnel can access and modify student data. This feature enhances the security and privacy of academic records.

Real-time Updates: PHP scripting will enable seamless communication between the frontend and backend, ensuring real-time updates to student results. This feature supports timely decision-making and intervention strategies.

Chapter - II



Proposed System

2.1 Proposed System :

Our proposed “Student Result Management” System (SRMS) leverages HTML, CSS, JavaScript, MySQL, and PHP to revolutionize result processing. With an intuitive user interface, it automates result calculations, enhances data accuracy, and ensures real-time result availability. Robust security features, including authentication and role-based access, promote data integrity and transparency. The system empowers educators and administrators with data analysis tools for informed decision-making. Customizable reporting functionalities and a centralized database optimize result management. The system prioritizes user training, adopts a scalable architecture, and commits to continuous improvement, ensuring adaptability to evolving educational needs.

2.2 Objectives :

The objectives of our proposed “Student Result Management” System (SRMS) are centered on enhancing efficiency, accuracy, and transparency in result processing. Our primary goals include automating result workflows, improving data accuracy through validation mechanisms, ensuring real-time accessibility for educators and students, enforcing robust security measures, and providing advanced data analysis tools. This system is designed to be flexible, allowing customization to meet specific institutional needs, and prioritizes user training for seamless adoption. Additionally, we aim to build a scalable architecture that can accommodate future growth while committing to a culture of continuous improvement through regular updates and responsiveness to user feedback.

2.4 Requirement Gathering :

The requirement gathering phase for the Student Result Management System (SRMS) is a critical step in understanding and documenting the specific needs of the educational institution. To commence this process, a detailed consultation with key stakeholders, including educators, administrators, and IT personnel, will be conducted. Through collaborative sessions, we aim to identify the existing challenges in the manual result management system and gather insights into the essential functionalities required for an effective SRMS.

These requirements encompass the need for a user-friendly interface, which will be achieved through the use of HTML, CSS, and JavaScript. Additionally, the stakeholders emphasize the importance of automation in result processing, necessitating the development of algorithms for accurate and efficient computation of results. Real-time result availability is a primary requirement, requiring seamless communication between the frontend and backend through PHP scripting.

The security and integrity of student data emerge as top priorities, leading to the requirement for robust authentication mechanisms and role-based access controls. Transparency in result management is a common theme, necessitating the implementation of detailed audit logs and centralized database access for authorized personnel.

Data analysis tools and customizable reporting functionalities are highlighted as crucial requirements to empower educators and administrators with insightful analytics for informed decision-making. Flexibility and adaptability are emphasized, leading to the need for a customizable system architecture that aligns with the unique requirements of each educational institution.

To ensure successful adoption, the requirement gathering process will also include a comprehensive training program for all user groups, covering educators, students, and administrators. Scalability is a key consideration, requiring the system to accommodate future growth and technological advancements.

2.5 Software Requirements Specification (SRS):

1. Introduction:

The Student Result Management System (SRMS) is a comprehensive web-based solution aimed at automating and optimizing the result management processes within educational institutions. This Software Requirement Specification outlines the functional and non-functional requirements of the SRMS, ensuring a robust, secure, and user-friendly platform for educators, students, and administrators.

2. Functional Requirements:

Result Processing Automation: The system shall automate result processing workflows, enabling quick and accurate calculations.

It shall allow educators to input results through an intuitive user interface developed using HTML, CSS, and JavaScript.

Data Accuracy Enhancement: The system shall implement validation checks to ensure the accuracy and integrity of student academic records.

It shall minimize errors associated with manual data entry and result processing.

Real-Time Result Availability: The system shall provide real-time updates to educators and students through PHP scripting. It shall offer a user-friendly interface for instant result retrieval.

Security and Access Control: The system shall include secure authentication mechanisms for user logins. It shall enforce role-based access control to protect sensitive information.

Transparency and Auditability: The system shall generate detailed audit logs to track changes in student records. It shall ensure a centralized database accessible only to authorized personnel.

Data Analysis and Reporting: The system shall incorporate tools for educators and administrators to analyze student performance trends. It shall offer customizable reporting functionalities for data-driven decision-making.

3. Non-Functional Requirements:

Customization and Adaptability: The system shall feature a modular architecture allowing customization to meet specific institutional requirements. It shall be salable to adapt to changes in educational needs and technological advancements.

User Training and Adoption: The system shall provide comprehensive training programs for educators, students, and administrators. It shall offer user-friendly documentation and support resources for seamless adoption.

Scalability: The system architecture shall be designed to handle increased data loads efficiently.

It shall remain compatible with emerging technologies to ensure long-term scalability.

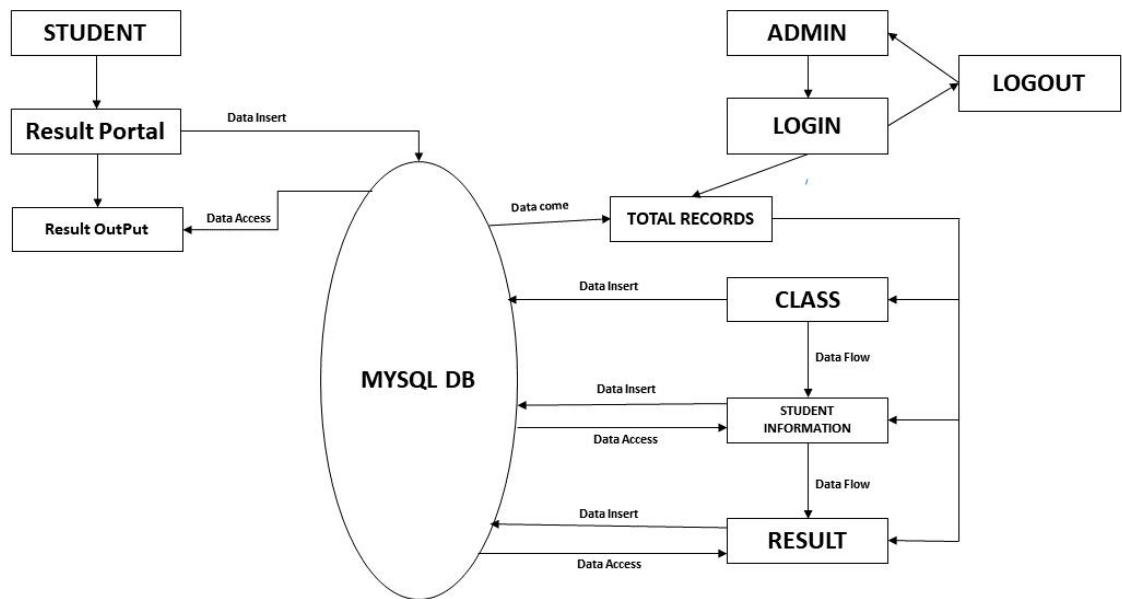
Continuous Improvement: The system shall actively seek and incorporate user feedback for continuous improvement. Regular updates and maintenance shall be conducted to stay responsive to evolving educational requirements.

Chapter III

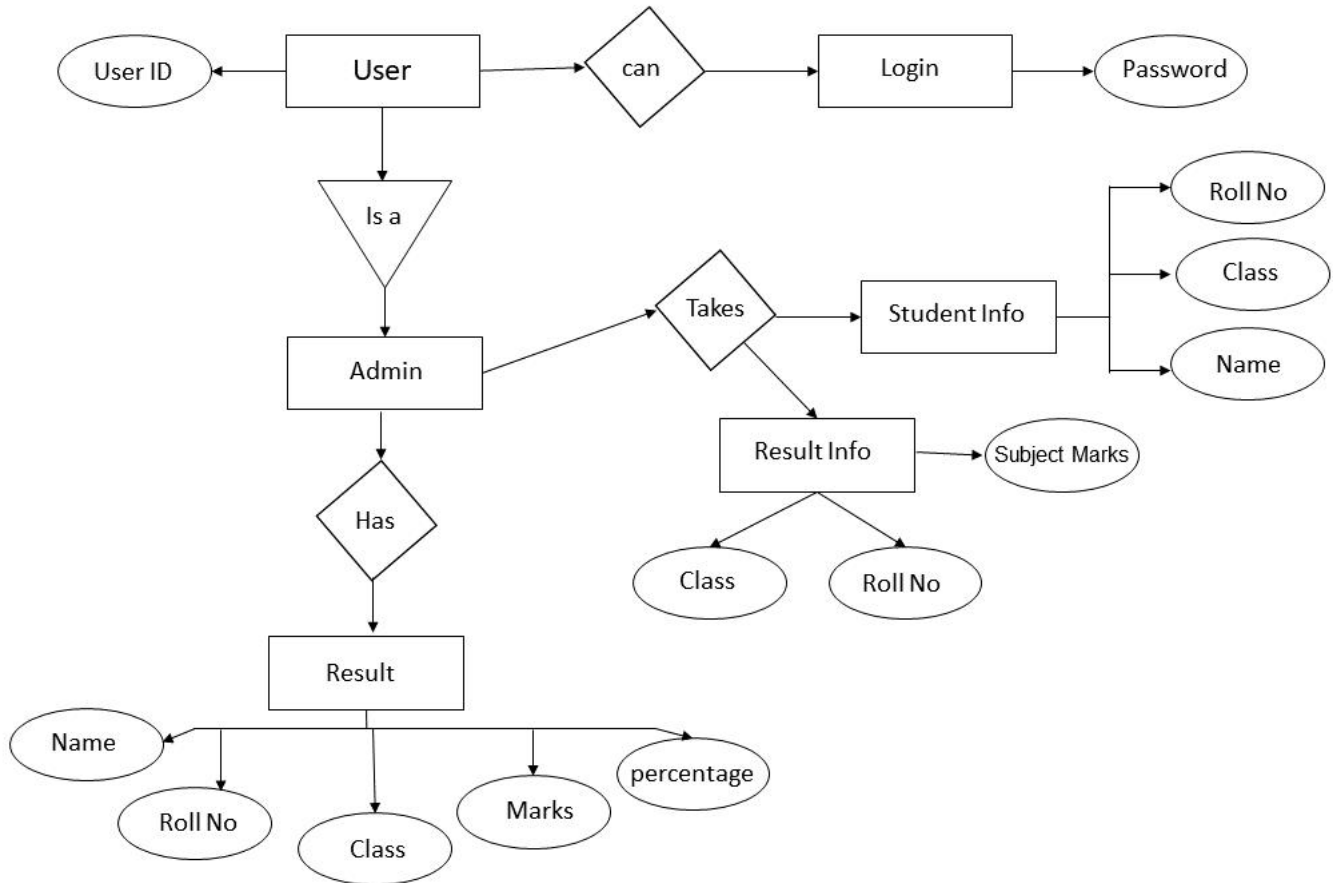


System Diagram

3.1 Data Flow Diagram :



3.2 Entity Relationship Diagram :



3.3 System Requirement :

Operating System : Windows 11 , Linux

Developing Tool : PHP , MYSQL , HTML , CSS , JAVASCRIPT

3.4 Hardware Requirement :

Processor : i3 11th Gen 2GHz Processor

RAM : 8 GB RAM

HDD : 500MB or above

Monitor : 15" Color Monitor

Keyboard : 108 standard

Mouse : Optical

3.5 Software :

Visual Studio Code : It is a source-code editor that can be used with a variety of programming languages, including C, C#, C++, Go, Java, JavaScript, Node.js, Python, Rust, and Julia.

It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine.

XAMPP : It is a free and open-source cross-platform web server. XAMPP is simply a local host or server that is used to test clients or websites before publishing them to a remote web server. The XAMPP server software on a local computer provides an appropriate environment for testing MYSQL, PHP, Apache, and Perl projects.

web browser : It is an application software to explore www (World Wide Web). It provides an interface between the server and the client and it requests to the server for web documents and services. It works as a compiler to render HTML which is used to design a webpage. Whenever we search for anything on the internet, the browser loads a web page written in HTML, including text, links, images, and other items such as style sheets and JavaScript functions. Google Chrome, Microsoft Edge, Mozilla Firefox, and Safari are examples of web browsers.

Chapter IV



System Design

4.1 Database Design

- **Database Name: sms2024**

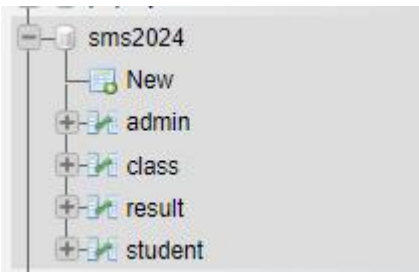
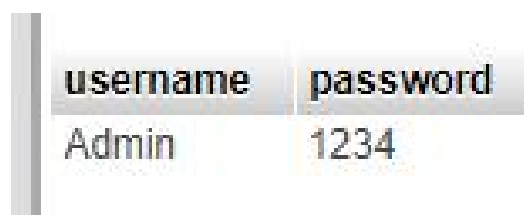


Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> class	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> result	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> student	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	15	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

1) Table Name: Admin

Description: - To store admin username and password.

Field Name	Data Type	Description
Username	Varchar(20)	To store username
Password	Varchar(20)	To store admin's password



2) Table Name: Class**Description: - To store class and year.**

Field Name	Data Type	Description
Year	Int(255)	To store the year in the form of integers.
Class	Varchar(20)	To store the student Class.

year	class
0	
2024	BSC
2024	BCA
2024	MBA

3) Table Name: Student**Description: - To store student class , roll number and student name .**

Field Name	Data Type	Description
class	Varchar(255)	To store student class.
roll	Varchar(255)	To store the roll number of the student.
name	Varchar(255)	To store the name of student.

class	roll	name
BSC	A1	Akshay
BSC	A2	Amit
BCA	201	student 1
MBA	50	Akki

4) Table Name: Result

Description: - To store student marks .

Field Name	Data Type	Description
class	Varchar(255)	To store student class.
roll	Varchar(255)	To store the roll number of the student.
S1	Int(255)	To store the marks of subject 1.
S2	Int(255)	To store the marks of subject 2.
S3	Int(255)	To store the marks of subject 3.
S4	Int(255)	To store the marks of subject 4.
S5	Int(255)	To store the marks of subject 5.
S6	Int(255)	To store the marks of subject 6.
S7	Int(255)	To store the marks of subject 7.

class	roll	name
BSC	A1	Akshay
BSC	A2	Amit
BCA	201	student 1
MBA	50	Akki

4.2 Input-Output Design

Student-Result-panel Page :-

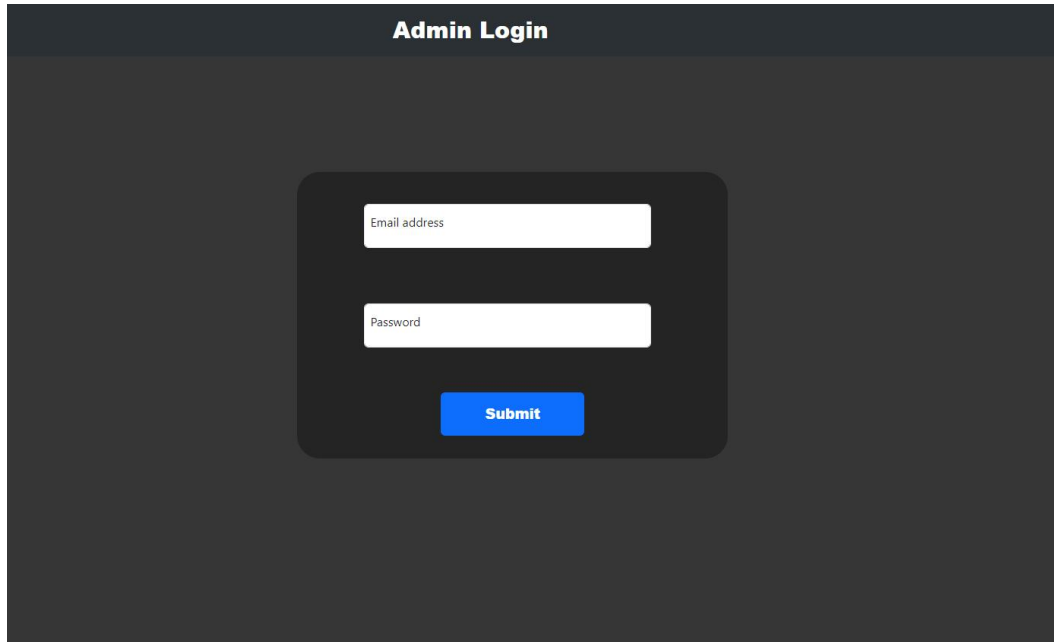
This page is used by student for view the result / marks of subject by entering the Roll no and Class in the form .

When student enter the class and roll number in form and do submit then net page be apper. In appering page student can view their result Online.

Subject	Marks
PHP	55
Python Programming	44
JAVA Programming	22
R Programming	33
MySQL DB	55
Web Development	44
SoftwareDevelopmenten	11

Admin-Login-Page :

In this page admin can login by user ID and Password for create an student result .



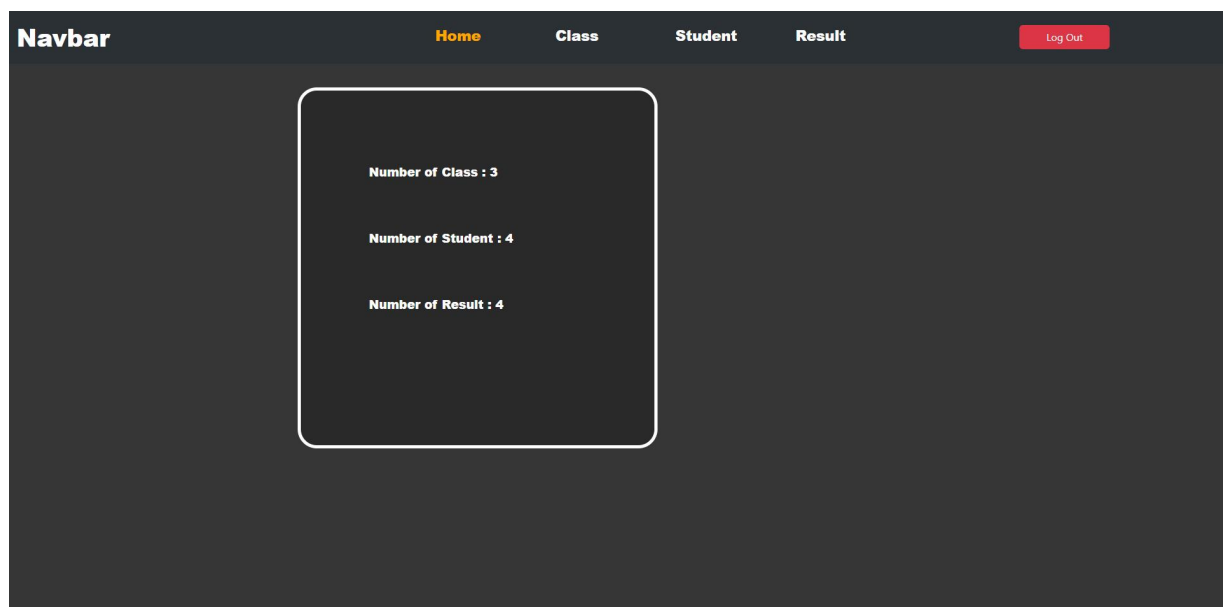
The image shows a dark-themed login page titled "Admin Login". It features a central white box containing two input fields: "Email address" and "Password". Below these fields is a blue "Submit" button.

Dashboared-page :

In this page there was a navigation bar which contains the five different menu in It .

Home , Class , Student , Result , LogOut.

Home - In home menu admin can see the Total number of class , student , result available in database.



The image shows a dashboard page with a dark theme. At the top is a "Navbar" with the following items: "Home" (highlighted in orange), "Class", "Student", "Result", and a "Log Out" button (highlighted in red). The main content area is a white box containing three statistics:

- Number of Class : 3
- Number of Student : 4
- Number of Result : 4

Class - In this menu admin can insert the class name and pass out year of student .

The screenshot shows the 'Class' menu interface. At the top is a dark navbar with the text 'Navbar' on the left and navigation links 'Home', 'Class' (highlighted in orange), 'Student', and 'Result' in the center. A red 'Log Out' button is on the right. The main content area is dark gray and contains a light gray rounded rectangle with two white input fields: 'Enter Year' and 'Enter Class'. Below these fields is a blue 'Submit' button.

Student - In student menu admin can insert the student information like class , roll number and name of the student .

Then submit data to the MySQL server .

The screenshot shows the 'Student' menu interface. At the top is a dark navbar with the text 'Navbar' on the left and navigation links 'Home', 'Class', 'Student' (highlighted in orange), and 'Result' in the center. A red 'Log Out' button is on the right. The main content area is dark gray and contains a light gray rounded rectangle with three white input fields: 'Select Class' (with a dropdown arrow and 'SELECT CLASS' text), 'Enter Roll No.', and 'Enter Name'. Below these fields is a blue 'Submit' button.

Result - In this menu admin can create an result of the student by inserting the marks of the subject , roll number .

LogOut - This option can used by the admin for logout the account . When admin logout then they go the admin-login page.

The screenshot displays the 'Result' management interface of a web application. At the top, a dark navigation bar (Navbar) contains links for 'Home', 'Class', 'Student', 'Result' (highlighted in orange), and a 'Log Out' button. The main content area features a form for entering student results. The form includes a 'Select Class' dropdown menu with 'SELECT CLASS' as the placeholder text. Below this are eight text input fields for subjects: 'Enter Roll No.', 'PHP', 'Python Programming', 'JAVA Programming', 'R Programming', 'MySql DB', 'Web Development', and 'SoftwareDevelopment'. A blue 'Submit' button is positioned at the bottom of the form.

Sampel souce code

1) Index.php :

```

        <?php
ini_set('display_errors','Off');
    $host = "localhost";
    $user="root";
    $password = "";
    $DB="sms2024";

    $conn = mysqli_connect($host,$user,$password,$DB);

    if(!$conn){
        die("Connection Error".mysqli_connect_error());
    }

    $sel=$_POST['sel'];
    $roll=$_POST['Roll'];
    $name=$_POST['Name'];

    $sql = "SELECT * FROM student WHERE   roll = '$roll' ";
    $result = $conn->query($sql);

    if ($result->num_rows == 1) {
        // Login successful
        $_SESSION['class']=$sel;
        $_SESSION["roll"] = $roll;

        // $_SERVER['name']=$name;
        // echo "<script>alert('Sucess User ')</script>";

        echo "<i class='col12'>This Student   is avalabe</i> ";
    }
    else {
        $sql="INSERT INTO student (` class`,`roll`,`name`) VALUE ('$sel','$roll','$name')";
        $res=mysqli_query($conn,$sql);
        //  echo "<script>alert('Invalid User ')</script>";
        // header("Location:  admin.php");
    }

```

```

?>
<!doctype html>
<html lang="en">
  <head>
    <style>
      .col12{
        color:red;
      }
    </style>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
  </head>
  <link rel="stylesheet" href="/css/Class.css">
  <body>
    <!-- *****Navbar
***** -->

    <nav class="navbar navbar-expand-lg bg-body-tertiary" data-bs-theme="dark">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Navbar</a>

        <h1 id="create_class">CREATE ____CLASS</h1>

        <ul class="men11">
          <li class="m1"><a href="/dashboard.php" class="navbar-brand m11 ">Home</a></li>
          <li class="m1"><a href="/Class.php" class="navbar-brand m11 ">Class</a></li>
          <li class="m1"><a href="/Student.php" class="navbar-brand m11 col">Student </a></li>
          <li class="m1"><a href="/Result.php" class="navbar-brand m11">Result</a></li>
        </ul>
      </div>
      <button type="button" class="btn btn-danger" OnClick="next()">Log Out</button>
    </nav>
    <script>
      function next(){

```

```

        window.location.href = "./logout.php";
    }
</script>

<!-- *****navbar end
/***** -->

<div class="form">
    <form action="" method="POST">
        <ul class="f1">
            <li class="f11">
                <?php
                    $query ="SELECT class FROM class";
                    $result = $conn->query($query);
                    if($result->num_rows> 0){
                        $options= mysqli_fetch_all($result, MYSQLI_ASSOC);
                    }
                ?>
                <div class="input-group">
                    <span class="input-group-text">Select Class </span>
                    <select aria-label="First name" class="form-control" name="sel" required>
                        <option>SELECT CLASS</option>
                        <?php
                            foreach ($options as $option) {
                                ?>
                                <option><?php echo $option['class']; ?> </option>
                                <?php
                                    }
                                ?>
                                </select>
                                <!-- <input type="text" aria-label="First name" class="form-control" name="Roll"
required> -->
                                <!-- <input type="text" aria-label="Last name" class="form-control"> -->
                            </div>
                        </li>
                    <li class="f11">
                        <div class="input-group">
                            <span class="input-group-text">Enter Roll No. </span>
                            <input type="text" aria-label="First name" class="form-control" name="Roll" required>
                            <!-- <input type="text" aria-label="Last name" class="form-control"> -->

```

```

    </div>
  </li>
  <li class="f11">
    <div class="input-group">
      <span class="input-group-text">Enter Name </span>
      <input type="text" aria-label="First name" class="form-control" name="Name"
        required>

      <!-- <input type="text" aria-label="Last name" class="form-control"> -->
    </div>
  </li>
</li>
<li class="f11">
  <button type="submit" class="btn btn-primary ad" name="submit">Submit</button>
</li>
</ul>
</form>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>

</body>
</html>

```

2) Studentresult.php :

```

<?php
    $host = "localhost";
    $user="root";
    $password = "";
    $DB="sms2024";

    $conn = mysqli_connect($host,$user,$password,$DB);

    if(!$conn){

        die("Connection Error".mysqli_connect_error());
    }
    // echo "yes";

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $roll = $_POST["roll"];
        $class = $_POST["class"];

        // You should properly sanitize and validate user inputs before using them in queries to
        prevent SQL injection.

        $sql = "SELECT * FROM result WHERE roll = '$roll' AND class = '$class'";
        $result = $conn->query($sql);

        if ($result->num_rows == 1) {
            // Login successful
            $_SESSION["roll"] = $roll;
            $_SESSION["class"] = $class;

            $student = "SELECT * FROM student WHERE class = '$class'   AND
            roll = '$roll'   ";
            $res = $conn->query($student);

            if($res->num_rows == 1)
            {
                $_SESSION["roll"] = $roll;

```



```

$_SESSION["class"] = $class;
// $_SESSION["name"] = $name;
// echo"Yes";

$stu="SELECT*FROM student WHERE roll ='$roll'";
$res1=$conn->query($stu);

while($row = mysqli_fetch_assoc($res1) ){
    echo"<br>Name : &#160 &#160 &#160 &#160 &#160".$row["name"]."
<br> <br>Class : &#160 &#160 &#160 &#160 &#160 &#160".$row["class"]."<br> <br>Roll No . :  &#160
".$row['roll'];
}
$marks="SELECT*FROM result WHERE roll ='$roll'";
$res2=$conn->query($marks);

while($rows = mysqli_fetch_assoc($res2) ){
    ?>
<style>
    button{
        position: relative;
        top: 4cm;
        right: 2cm;
    }
    body{
        overflow-y: hidden;
        overflow-x: hidden;
    }
    table{
        position: relative;

        top: 4cm;
        text-align: center;
        font-size: 0.8cm;
    }
</style>
<?php
echo " <table style='float:right'>
<tr>

```

```

        <th>Subject</th>
        <th>Marks</th>
    </tr>
    <tr>
        <td>PHP</td>
        <td>". $rows['s1']."</td>
    </tr>
    <tr>
        <td>Python Programming</td>
        <td>" . $rows['s2']."</td>
    </tr>
    <tr>
        <td>JAVA Programming</td>
        <td>".$rows['s3'] . "</td>
    </tr>
    <tr>
        <td>R Programming</td>
        <td>".$rows['s4']."</td>
    </tr>
    <tr>
        <td>MySQL DB</td>
        <td>".$rows['s5']."</td>
    </tr>
    <tr>
        <td>Web Development</td>
        <td>". $rows['s6']."</td>
    </tr>
    <tr>
        <td>Software Developmen</td>
        <td>".$rows['s7']."</td>
    </tr>
</table>";
?>
<button onclick="window.print()">Print Result</button>

```

```

<h3><?php
    $total_marks =
    $rows['s1']+$rows['s2']+$rows['s3']+$rows['s4']+$rows['s5']+$rows['s6']+$rows['s7'];
    $marks = 100*7;

```

```

        $percentage = ($total_marks / $marks)*100 ;
        echo "percentage"." &#160 ". $percentage."%";

        if($percentage > 36.00){
            echo "<br><br> Result:  &#160 &#160"."Pass";
        }
        else{
            echo "Fale";
        }
        ?></h3>

<?php
    }
    //      echo $roll."<br>".$class."<br>".$res['name'];
    }
    else{
        echo "Not Avalavbel";
    }
    }
    else{
        echo "Not Avalavbel";
    }
}

else {
    echo "Not Avalavbel";
}

?>

```

3) Admin.php :

```

<!doctype html>
<html lang="en">
  <head>
    <!-- <link rel="stylesheet" href="./style.css"> -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
    <link rel="stylesheet" href="./styleadmin.css">
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
integrity="sha384-l7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8O
Dewa9r" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"
integrity="sha384-
BBtl+eGJRgqQAUMxJ7pMwbEyER4l1g+O15P+16Ep7Q9Q+qzX6gSbd85u4mG4QzX+"
crossorigin="anonymous"></script>
  </head>
  <body>
    <!-- *****Navbar***** -->
    <nav class="navbar navbar-expand-lg bg-body-tertiary" data-bs-theme="dark">

      <div class="container-fluid">
        <a class="navbar-brand" href="#">Admin Login</a>

      </div>
    </nav>
    <!-- *****navbar end***** -->
    ***** -->
    <!-- *****Form***** -->
    ***** -->
    <div class="form1">
      <form action="/login.php" method="POST">
        <div class="mb-3 form-floating">

```

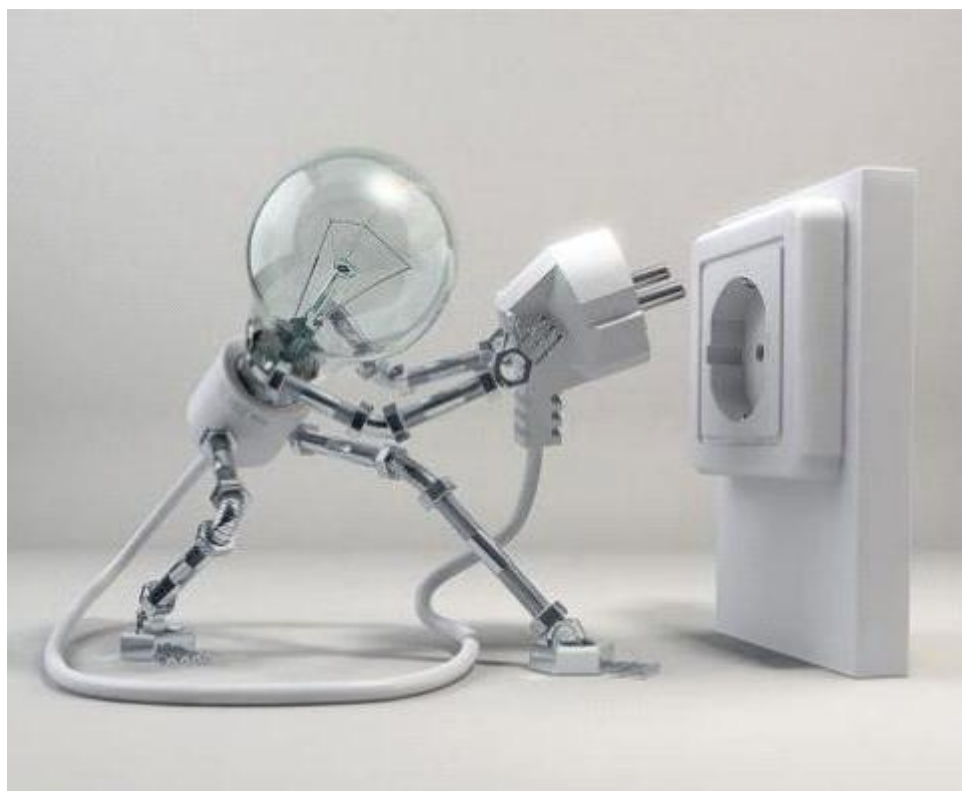
```

        <label for="exampleInputEmail1" class="form-label">Email address</label>
        <input type="text" class="form-control" id="exampleInputEmail1" aria-
describedby="emailHelp" name='username'>
        <div id="emailHelp" class="form-text">We'll never share your email with anyone
else.</div>
    </div>
    <div class="mb-3 form-floating">
        <label for="exampleInputPassword1" class="form-label">Password</label>
        <input type="password" class="form-control" id="exampleInputPassword1"
name='password'>
    </div>

    <button type="submit" class="btn btn-primary" name='submit'>Submit</button>
</form>
</div>
<!-- *****form end
***** -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
</body>
</html>

```



User Guideline

5.1 User Manual

Student Result Management Project, works as a Website for real estate classified system where admin can add properly details (like Class, Roll No , Marks and Name).and personal details. It developed as a dynamic website, this project developed by using HTML, CSS, JavaScript, MySQL, and PHP to deliver an efficient result to user.

For admin we provide different option to use the website .The admin is a main user of the dashboard . He has all the rights to do in dashboard .

- **Admin-Login-Page :**

In this page admin can login by user ID and Password for create an student result .

- **Dashboared-page :**

In this page there was a navigation bar which contains the five different menu in It .
Home , Class , Student , Result , LogOut.

I) Home -

In home menu admin can see the Total number of class , student , result available in database.

II) Class -

In this menu admin can insert the class name and pass out year of student .

III) Student -

In student menu admin can insert the student information like class , roll number and name of the student . Then submit data to the MySQL server .

IV) Result -

In this menu admin can create an result of the student by inserting the marks of the subject , roll number .

V) LogOut -

This option can used by the admin for logout the account . When admin logout then they go the admin-login page.

For Student we provide option to use the website .In this student is a viewers who can see there result online . But they not having any permission to edit data and information .

Student-Result-panel Page :-

This page is used by student for view the result / marks of subject by entering the Roll no and Class in the form .

When student enter the class and roll number in form and do submit then net page be apper.

In appering page student can view their result Online.



Conclusion And Suggestion

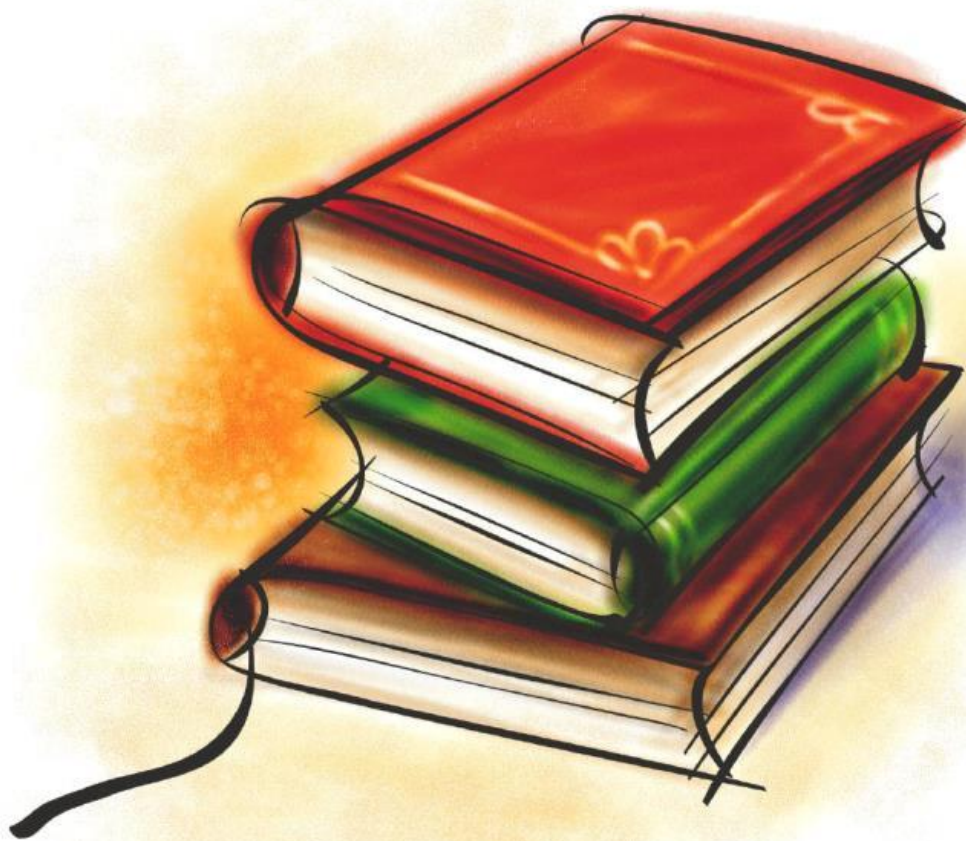
5.1 Conclusion :

The present “ **STUDENT RESULT MANAGEMENT** ” is useful for small levels school to keep the student information like class , student roll number and show exam result online to student .

This “ **STUDENT RESULT MANAGEMENT** ” is developed as simple as possible to school . Being user friendly ,where user does not find any difficulty in using it.

Our “ **STUDENT RESULT MANAGEMENT** ” project successfully gives the following functionalities :

- Keep the student information.
- Create student result .
- User friendly.
- Provide result portal for the student .



Bibliography

Bibliography:

- www.google.com
- www.w3school.com