

# **STUDENT RESULT MANAGEMENT SYSTEM**

**Designed and developed by  
Mr. Sachin Nagnath Isamantri**

**PRN :- 2021420070**

**Third Year Bachelor of Science  
(Computer Science)**

**Under the esteemed Guidance of  
Prof. Neehit Tiwari (Assistant Professor)**

**Department of Computer Science  
VPM's B. N. Bandodkar College of Science  
(Autonomous)**

**THANE (W) - 400601**

**Maharashtra**

**YEAR -2023-24**

## **Certificate of Completion**

This certifies that **Mr. Sachin Nagnath Isamantri** has successfully completed the project titled “Student Result Management System.”

The project involved the development of a comprehensive Student Result Management System using PHP , MySQL , HTML , CSS , and JavaScript. Throughout the project he has demonstrated an exceptional understanding of software architecture , database management , and web development principles. From conceptualization to execution , they have exhibited unwavering dedication , attention to detail , and problem-solving prowess in every phase of the project lifecycle.

We commend for their exemplary performance , innovative approach , and commitment to excellence throughout the duration of this project. Their dedication and professionalism have been instrumental in the successful completion of the Student Result Management System.

Mrs. Smita Waikar  
(Anam Cara Tutorial)

## **Abstract**

The Student Result Management System (SRMS) is a sophisticated web-based application meticulously crafted to address the intricate needs of managing student academic records. Leveraging a potent combination of PHP, MySQL, HTML, CSS, and JavaScript, SRMS offers a comprehensive solution for educational institutions seeking to streamline their result management processes.

Through an intuitive user interface, administrators, teachers, and students can effortlessly access and update academic data securely, fostering a collaborative and efficient environment. SRMS boasts a plethora of features, including seamless student registration, precise result entry, automated grade calculation, and robust reporting capabilities. By amalgamating cutting-edge technologies with pedagogical expertise, SRMS serves as a catalyst for enhancing transparency, accuracy, and efficacy in managing student results.

Furthermore, SRMS prioritizes accessibility by embracing responsive design principles, ensuring optimal user experience across a myriad of devices and platforms.

## **Acknowledgment**

I would like to express my heartfelt gratitude to the VPM's B. N. Bandodkar College of Science for providing me with the opportunity to pursue this project. The college's nurturing environment and resources have been instrumental in the successful completion of the “ Student Result Management System .”

I extend my sincere appreciation to my project guide, Mr. Neehit Tiwari, for his invaluable guidance, unwavering support, and expertise throughout the project's development. His mentorship has been pivotal in shaping the project and enhancing my understanding of the subject matter.

I would also like to acknowledge the faculty and staff at B. N. Bandodkar College of Science for their continuous encouragement and assistance during this project. Lastly, I thank my family and friends for their encouragement and patience during the project's journey.

The successful completion of this project would not have been possible without the collective support and guidance I have received from these individuals and institution.

## **Declaration**

I , Mr. Sachin Nagnath Isamantri , a student of VPM's B. N. Bandodkar College of Science , solemnly declare that the project titled "Student Result Management System" is an authentic work carried out by me under the guidance of Mr. Neehit Tiwari , my internal project guide , as a part of my academic curriculum at VPM's B. N. Bandodkar College of Science.

I affirm that this project is the result of my own efforts and has not been submitted elsewhere for any other purpose. Any external sources of information used in this project have been duly acknowledged through proper citations.

Furthermore , I take full responsibility for the accuracy and authenticity of the content presented in this project. In case of any discrepancies found later , I willingly accept the consequences as per the rules and regulations of VPM's B. N. Bandodkar College of Science.

Mr. Sachin Nagnath Isamantri  
VPM's B. N. Bandodkar College of Science

## Index

Sr. No.		Table of Contents	Page No.
<b>1</b>		<b>Introduction</b>	<b>1-9</b>
	<b>1.1</b>	Background	
	<b>1.2</b>	Feasibility Study	
	<b>1.3</b>	Objectives	
	<b>1.4</b>	Scope	
	<b>1.5</b>	Purpose	
	<b>1.6</b>	Advantages	
	<b>1.7</b>	Achievements	
<b>2</b>		<b>Survey of Technologies</b>	<b>10-11</b>
<b>3</b>		<b>Requirement Analysis</b>	<b>12-17</b>
	<b>3.1</b>	Problem Definition	
	<b>3.2</b>	Requirement Specification	
	<b>3.3</b>	Planning and Scheduling	
	<b>3.4</b>	Software and Hardware Requirement	
	<b>3.5</b>	Event Table	
<b>4</b>		<b>System Design</b>	<b>18-31</b>
	<b>4.1</b>	Basic Module	
	<b>4.2</b>	Data Design	
	<b>4.3</b>	Data Integrity and Constraints	
	<b>4.4</b>	Logic Diagrams	
	<b>4.5</b>	Security Issues	
	<b>4.6</b>	Test Case Design	
<b>5</b>		<b>Implementation and Testing</b>	<b>32-44</b>
	<b>5.1</b>	Testing Approach	
	<b>5.2</b>	Coding Details and Code Efficiency	

<b>6</b>		<b>Results</b>	<b>45-46</b>
<b>7</b>		<b>Conclusion and Future Work</b>	<b>47-48</b>
	<b>7.1</b>	Future Enhancement	
	<b>7.2</b>	Conclusion	
	<b>7.3</b>	System Maintenance	
<b>8</b>		<b>References</b>	<b>49</b>

# **Introduction**

## **1.1 Background :-**

The Student Result Management System (SRMS) is a vital tool developed for efficient management of student academic records. Designed as a part of the academic curriculum at B. N. Bandodkar College of Science, SRMS is a web-based application utilizing PHP, MySQL, HTML, CSS, and JavaScript. It offers a user-friendly interface for administrators, teachers, and students to access and update academic information securely. SRMS streamlines processes such as student registration, result entry, grade calculation, and reporting, enhancing transparency and efficiency in result management within educational institutions.

## **1.2 Feasibility Study**

- **Technical Feasibility**

Technical feasibility refers to the assessment of whether a Proposed project or system can be developed and Implemented using existing technology, resources, and skills.

### **A] Software Requirements:-**

The system will utilize PHP for server-side scripting, webpage designing and login system . MySQL will be used to store the login/signup data along with students result data .



### **B] Hardware Requirements:-**

A reliable server will host the web application, ensuring optimal performance, while stable and secure network infrastructure will facilitate seamless user-system interaction.

### **C] Database Management:-**

Efficient database schema design will structure student information, courses, and results. Normalization principles will minimize redundancy, and regular backup procedures will prevent data loss with a robust recovery mechanism in place.

- **Economical Feasibility**

The economic feasibility analysis evaluates whether the Student Result Management System project is financially viable. It considers factors such as development costs, operational expenses, and component costs to determine the economic sustainability of the proposed system.

### **A] Development Costs:-**

Development costs encompass expenses associated with designing, coding, testing, and deploying the system. These include salaries for developers, costs for software tools and frameworks, as well as any third-party services required during the development phase.

### **B] Operational Costs:-**

Operational costs involve ongoing expenses to keep the system running efficiently. This includes server hosting fees, maintenance costs, and potential expenses for system updates or upgrades. Operational costs are crucial for long-term sustainability and should be carefully managed to ensure financial viability.

### **C] Component Costs:-**

Component costs include expenses related to software and hardware components used in the system. This encompasses licenses for PHP frameworks, database management system costs, server hardware, and any other essential components integral to the system's functionality.

- **Operational Feasibility**

Operational feasibility evaluates the practicality of implementing the Student Result Management System within the existing operational framework. It assesses factors such as user acceptance, training requirements, and integration with the previous system to ensure a smooth and effective transition.

### **A] User Acceptance:-**

Ensuring user acceptance is fundamental to the operational success of the system. The interface and functionalities will be designed with user-friendly features, and user feedback will be actively solicited during development to address concerns and optimize the system for seamless integration into daily operations.

## **B] User Training:-**

Efficient user training programs will be implemented to familiarize administrators, teachers, and students with the new system. Training materials, workshops, and user support mechanisms will be provided to ensure a quick and smooth adaptation to the Student Result Management System, minimizing disruptions in daily tasks.

## **C] Integration with Previous System:-**

Integrating the new system with the existing infrastructure is a critical aspect of operational feasibility. Compatibility with legacy systems and databases will be considered, and a well-defined integration plan will be established to facilitate a seamless transition. This approach aims to leverage existing data and functionalities while introducing enhanced features and efficiency.

- **Scheduling and Timeline**

Scheduling feasibility refers to the assessment of whether a Proposed project, such as implementing a Student's Result Management System, can be completed within a reasonable Time frame. It involves evaluating the availability of resources, time constraints, and potential conflicts with other ongoing projects or commitments. Considering all the aspects for scheduling and timeline of this Project like developing the project, testing , deployment , User training , maintenance , etc. And concurrently I also Have to handle my studies

simultaneously developing this Project . I will try to handover this system(fully operational) To the client in roughly 1 month .

**1.3 Objectives :-** The objective of the Student Result Management System (SRMS) project is to develop a comprehensive web-based application that simplifies and enhances the management of student academic records within educational institutions. Key objectives include:

- **Efficiency:** Streamlining the process of recording, updating, and accessing student results to save time and resources for administrators, teachers, and students.
- **Accuracy:** Ensuring the accuracy of academic data through standardized result entry, automated grade calculation, and error-checking mechanisms.
- **Transparency:** Facilitating transparent communication and access to academic information for all stakeholders, promoting accountability and trust within the educational community.
- **Accessibility:** Providing a user-friendly interface that is accessible across various devices and platforms, ensuring seamless interaction for users regardless of location or technological proficiency.

**1.4 Scope:-** The scope of the Student Result Management System (SRMS) project encompasses the following key aspects:

- **Functional Scope:**

- Student Registration: Allow students to register and create accounts within the system.
- Result Entry: Enable teachers/administrators to input and update student results for various subjects and assessments.
- Grade Calculation: Automatically calculate grades based on predefined criteria and weightage.
- Data Security: Implement robust security measures to protect sensitive student information and ensure compliance with privacy regulations.

- **Technical Scope:**

- Development Platform: Utilize PHP, MySQL, HTML, CSS, and JavaScript to develop a web-based application accessible across different browsers and devices.
- Responsive Design: Ensure compatibility and optimal user experience on desktops, laptops, tablets, and smartphones.
- Database Management: Design and implement a relational database schema to store and manage student records, results, and other pertinent data.
- Integration: Integrate third-party libraries or APIs for

additional functionalities such as email notifications or authentication mechanisms.

- **Project Constraints:**

- **Timeframe:** Complete development, testing, and deployment within the allotted project timeline.
- **Budget:** Adhere to budgetary constraints and optimize resource allocation to meet project goals.
- **Resource Availability:** Utilize available human and technical resources efficiently to execute project tasks effectively.

By defining the scope of the SRMS project, stakeholders can clearly understand the project's boundaries, deliverables, and constraints, thereby ensuring successful project execution and alignment with organizational objectives.

**1.5 Purpose:-**The purpose of the Student Result Management System (SRMS) project is to address the challenges associated with traditional paper-based result management systems within educational institutions. The key purposes of implementing SRMS are as follows:

- **Efficiency Improvement:**

- Streamline result management processes to reduce manual effort and administrative burden on teachers and staff.

- Accelerate the speed of result processing and dissemination, enabling timely feedback to students and parents.
- **Accuracy Enhancement:**
  - Ensure the accuracy and consistency of student academic records by minimizing errors associated with manual data entry and calculation.
  - Implement validation checks and automated grading mechanisms to uphold the integrity of academic assessments.
- **Accountability:**
  - Foster accountability among educators and administrators by enabling them to track and monitor student performance trends over time.
- **Adaptability and Scalability:**
  - Design the SRMS to be adaptable to the evolving needs and requirements of educational institutions, with scalability to accommodate future growth and technological advancements.
  - Ensure compatibility with existing infrastructure and seamless integration with other educational management systems or platforms.

By fulfilling these purposes, the SRMS project aims to revolutionize result management processes, optimize educational outcomes, and contribute to the overall advancement of educational institutions.

## 1.6 Advantages

- **Streamlined Processes:** Automates result management tasks, such as data entry, calculating and reducing manual effort and improving operational efficiency.
- **Accuracy and Consistency:** Minimizes errors associated with manual data entry and calculation, ensuring the accuracy and consistency of student academic records.
- **Adaptability and Scalability:** Designed to adapt to the evolving needs of educational institutions, with scalability to accommodate future growth and technological advancements.
- **Time and Cost Savings:** Reduces the time and resources required for result management tasks, leading to cost savings and increased operational efficiency.

## 1.7 Achievement

As the developer of the Student Result Management System (SRMS), notable achievements include the successful design, development, and deployment of a streamlined and efficient system that garnered positive feedback from users.



## **Survey of Technologies**

### **Technologies used in this project:-**

- **VS Code**
- **XAMPP**
- **PHP**
- **HTML**
- **CSS**
- **JAVASCRIPT**

### **Visual Studio Code:-**

Visual Studio Code (VS Code) is a lightweight yet powerful source code editor developed by Microsoft. It supports various programming languages and offers features such as syntax highlighting, code completion, debugging, and Git integration. Researchers can use VS Code to write survey scripts, analyze data, or develop custom survey applications.

### **XAMPP:-**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists of Apache HTTP Server, MySQL database, and interpreters for PHP and Perl. Researchers often use XAMPP to set up a local server environment for testing survey applications or hosting survey data.

### **PHP (Hypertext Preprocessor):-**

PHP is a server-side scripting language widely used for web

development. It is particularly suitable for handling form submissions, processing survey responses, and interacting with databases. Researchers can use PHP to create dynamic survey forms, validate user input, and store survey data in a database.

### **HTML (Hypertext Markup Language):-**

HTML is the standard markup language for creating web pages and web applications. Researchers use HTML to structure survey forms, define the layout and appearance of survey questions, and create user interfaces for collecting survey responses.

### **CSS (Cascading Style Sheets):-**

CSS is a style sheet language used for describing the presentation of HTML documents. Researchers use CSS to style survey forms and customize the visual appearance of survey elements such as text, buttons, and input fields. CSS allows for the creation of visually appealing and user-friendly survey interfaces.

### **JavaScript:-**

JavaScript is a versatile programming language commonly used for adding interactivity and dynamic behavior to web pages. Researchers can use JavaScript to implement advanced survey features such as conditional logic, real-time validation, and interactive elements like sliders or dropdown menus.

# **Requirement Analysis**

## **3.1 Problem Definition :-**

Identify the challenges and shortcomings of the current result management system, such as manual data entry, lack of transparency, and inefficiencies in result processing. Determine the need for a comprehensive solution to streamline result management processes, enhance accuracy, and improve accessibility to academic information.

## **3.2 Requirement Specification:**

Define the functional requirements of the SRMS, including features such as student registration, result entry, grade calculation, reporting, and user management. Specify non-functional requirements related to performance, security, usability, and scalability to ensure that the SRMS meets the needs and expectations of stakeholders.

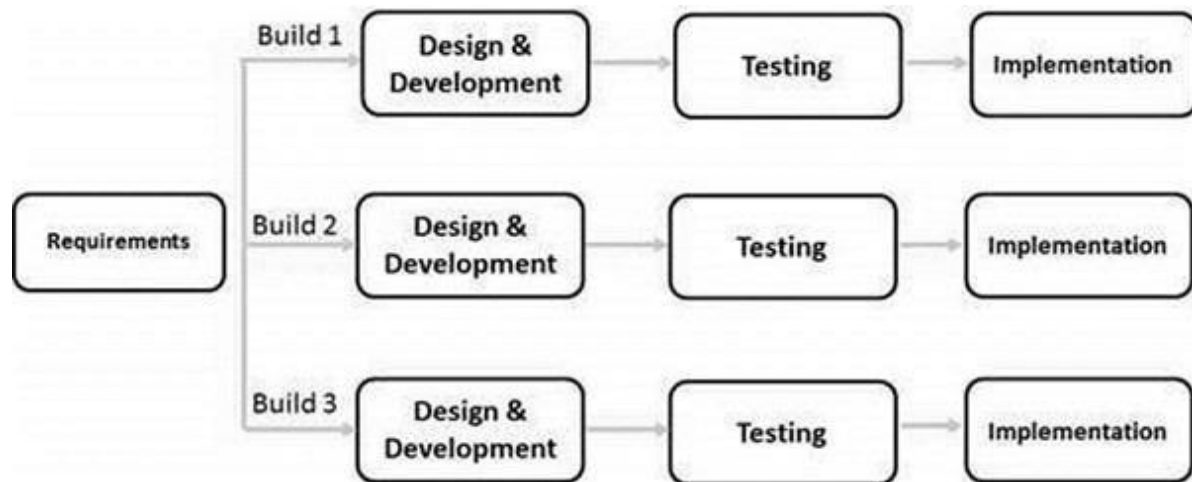
## **3.3 Planning and scheduling:-**

The Software Development Life Cycle (SDLC) outlines the systematic process involved in developing software, from initial planning to deployment and maintenance. For the Student Result Management System, a well-structured SDLC ensures a methodical and organized approach to deliver a robust and effective solution.

## **Selected SDLC Model: Iterative Model**

The Iterative Model is well-suited for the Student Result Management System, as it allows for flexibility in

accommodating changes, feedback, and continuous improvement throughout the development process. This iterative approach ensures a dynamic and responsive system tailored to the evolving needs of the educational institution.



- **Requirements Gathering:**

Understand the needs of the educational institution, administrators, teachers, and students. Gather and document specific requirements for the Student Result Management System to establish a clear project scope.

- **System Design:**

Develop a comprehensive system design, including database schema, user interface, and overall system architecture. Ensure that the design aligns with the identified requirements and provides a scalable and user-friendly solution.

- **Implementation (Coding):**

Begin coding based on the design specifications. Utilize PHP for server-side scripting and MySQL as the database management system. Incorporate a chosen PHP framework for efficient development.

- **Testing:**

Conduct thorough testing, including unit testing, integration testing, and user acceptance testing. Identify and rectify any defects, ensuring the system meets quality standards and functions as expected.

- **Deployment:**

Deploy the Student Result Management System in a controlled environment. Monitor the initial rollout for any unforeseen issues and address them promptly. Ensure a smooth transition from development to live operation.

- **User Training:**

Implement user training programs to familiarize administrators, teachers, and students with the system. Provide support materials and resources to facilitate a smooth adaptation to the new platform.

- **Evaluation and Feedback:**

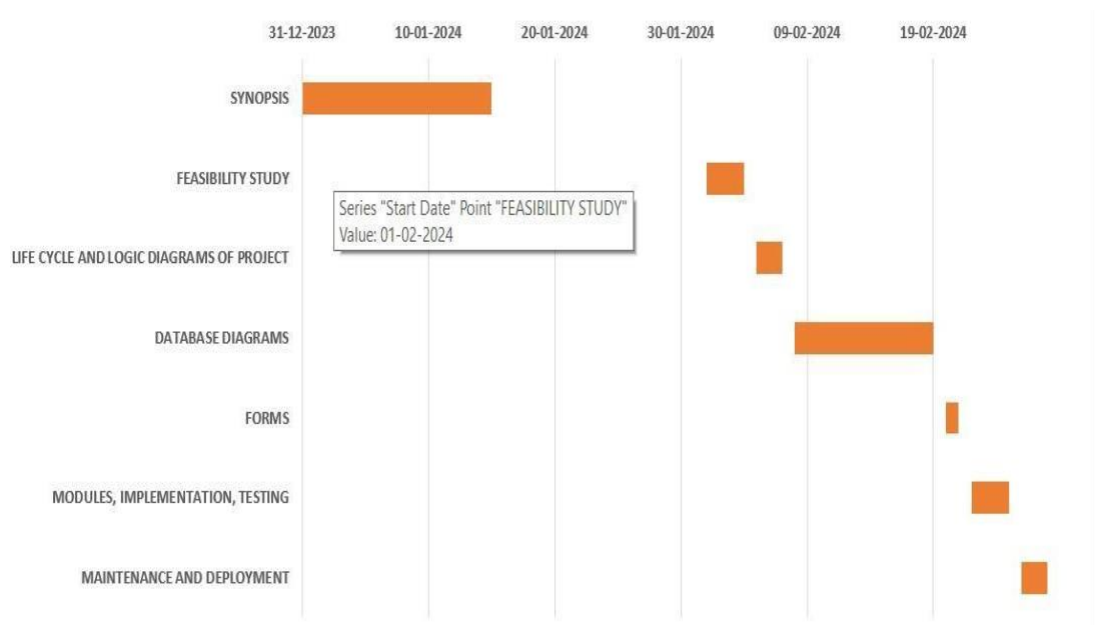
Collect feedback from users and stakeholders after the system is in use. Use this feedback to make iterative improvements and enhancements to address any issues or additional requirements.

- **Maintenance and Updates:**

Continuously monitor and maintain the system, addressing any emerging issues or updates. Implement periodic updates to enhance functionality, security, and performance as needed.

- **GANTT Chart:-**

A Gantt chart is a visual project management tool used to plan, schedule, and track tasks and activities within a project. It displays project tasks as horizontal bars on a timeline, with each bar representing a specific task, its start and end dates, and its duration. Gantt charts provide a clear and organized way to visualize project timelines, task dependencies, and progress, making them a valuable resource for project managers and teams to manage and communicate project details effectively.



### **3.4 Software and Hardware Requirements:-**

- **Software Requirements:-**

- 1. Operating System : Windows 10 and higher**

- 2. Front-end : HTML CSS,JAVASCRIPT**

- 3. Back-end: MYSQL,PHP**

- **Hardware Requirements: -**

- 1. Memory : Minimum 4 GB RAM**

- 2. Hard disk : Minimum 512 GB**

- 3. Processor : Minimum 64-bit processor**

## Event Table

<b>Event</b>	<b>Source</b>	<b>Description</b>	<b>Trigger</b>	<b>Destination</b>
Admin Registration	Admin	Admin register himself into the system	Registered Successfully	Admin
Login	Admin/Teacher	Admin/Teacher will login into the system	Logged in successfully	Admin
Adding Student's data	Admin	Admin stores the data of students into the system	Student's Data Added to the system	Admin
Adding Results	Teacher	Teacher adds results of students into the system	Student's Result Added	Admin
Updating Results	Teacher	Teacher updates results of the students	Student's Result Updated	Admin
Deleting Results	Teacher	Teacher Deletes results of students	Student's Result Deleted	Admin
Checking Result	Student	Students checks results	Result checked Successfully	Admin
Taking Printout of Results	Student	Students take printout of results	Result Printed successfully	Admin



# System Design

## 4.1 Basic Module :-

Basic module within the system design for an Student Result Management System include:

- **Module Name:** User Authentication

**Purpose:** To authenticate and authorize users to access the system.

### Components:

#### 1. Registration:

- Allows users to create an account by providing necessary information.
- Captures and stores user details securely.

#### 2. Login:

- Enables users to log in with their credentials (username and password).
- Verifies user identity and grants access to the system.

### Interactions:

- This module interacts with the database to store user data (username and password) securely.
- It interfaces with the frontend for user registration and login.
- It integrates with other modules to ensure

authenticated users can access their respective dashboards and features.

### **Security Considerations:**

- Encryption of user passwords.
- Protection against brute force attacks.
- Secure session management to prevent unauthorized access.

### **Error Handling:**

- Handling invalid login attempts.
- Guiding users through the registration and password reset processes.

This module forms the basis for user management and access control . It ensures that only authorized users can use the system and access their respective features.

- **Module Name:** Admin Panel

**Purpose:** To provide administrators with tools to manage and monitor the Student Result Management System.

### **Components:**

1. Student Management:
  - Admins can add, edit, or remove the students.
2. Class Management:
  - Admins can create, edit, and delete classes.

### 3. Result Management:

- Add , Update or delete the results of students.
- Taking printout of results.

### **Interactions:**

- The admin panel interacts with the database to manage user accounts, classes, and academic records.
- It communicates with the backend to update system configurations.
- It may have its dedicated web interface or can be integrated with the main web application.

### **Security Considerations:**

- Access control and authentication for admin users.
- Logging and auditing of admin actions.
- Ensuring sensitive configurations are accessible only to authorized admins.

## **4.2 Data Design:-**

Database design refers to the process of creating a detailed data structure for a database. It involves defining the tables, columns, relationships, constraints, and other attributes necessary to organize and store data efficiently and effectively. The goal of database design is to ensure data

integrity, accuracy, and accessibility while optimizing performance and minimizing redundancy.

### **Records:-**

#### **admin\_login**

This table is used to store signup/login data.

Sr. No.	Field Name	Data Type	Size	Description
1	userid	varchar	30	Primary Key
2	Password	varchar	255	Unique Key
3	salt	varchar	64	Not Null

#### **class**

This table is used to store classes of students.

Sr. No.	Field Name	Data Type	Size	Description
1	id	int	11	Primary Key
2	name	varchar	30	Unique Key

## **result**

This table is used to store results of students.

Sr. No.	Field Name	Data Type	Size	Description
1	id	int	11	Primary Key
2	name	varchar	30	Not Null
3	rno	int	11	Not Null
4	name	varchar	30	Not Null
5	P1	int	3	Not Null
6	P2	int	3	Not Null
7	P3	int	3	Not Null
8	P4	int	3	Not Null
9	P5	int	3	Not Null
10	Marks	int	3	Not Null
11	Percentage	float		Not Null

## **students**

This table is used to store student's data.

Sr. No.	Field Name	Data Type	Size	Description
1	name	varchar	30	Primary Key
2	rno	int	11	Not Null
3	name	varchar	30	Foreign Key

### **4.3 Data Integrity:-**

Data integrity and constraints are essential aspects of a well-designed database system, ensuring the accuracy and reliability of stored data. In the context of your Student Result Management System, here are some key data integrity and constraints considerations:

- **Entity Integrity:**

- Ensure each entity (table) has a primary key that uniquely identifies records.
- This guarantees that each student, class, and academic record is uniquely identifiable.

- **Referential Integrity:**

- Use foreign keys to establish relationships between tables.

- **Domain Integrity:**

- Define data types and constraints for each attribute to ensure data falls within acceptable ranges.
- This prevents invalid data from being stored (e.g., storing text in a date field).

- **Constraints:**

#### **1. Primary Key Constraint:**

- Ensure the primary key of each table is unique and not null.
- For example, a student ID or class ID should be

unique.

## **2. Foreign Key Constraint:**

- Enforce referential integrity by ensuring foreign keys point to valid primary keys in related tables.
- For instance, a foreign key in an academic record should point to an existing student and class.

## **3. Check Constraint:**

- Define rules for attribute values to restrict what can be stored.

## **4. Unique Constraint:**

- Ensure that specific columns contain unique values, but they can allow null values.
- Useful for fields like email addresses or usernames to prevent duplication.

By implementing data integrity and constraints, you ensure that the data in your Student Result Management System remains accurate, consistent, and free from errors. This is crucial for maintaining the reliability of academic records and ensuring that the system operates as expected.

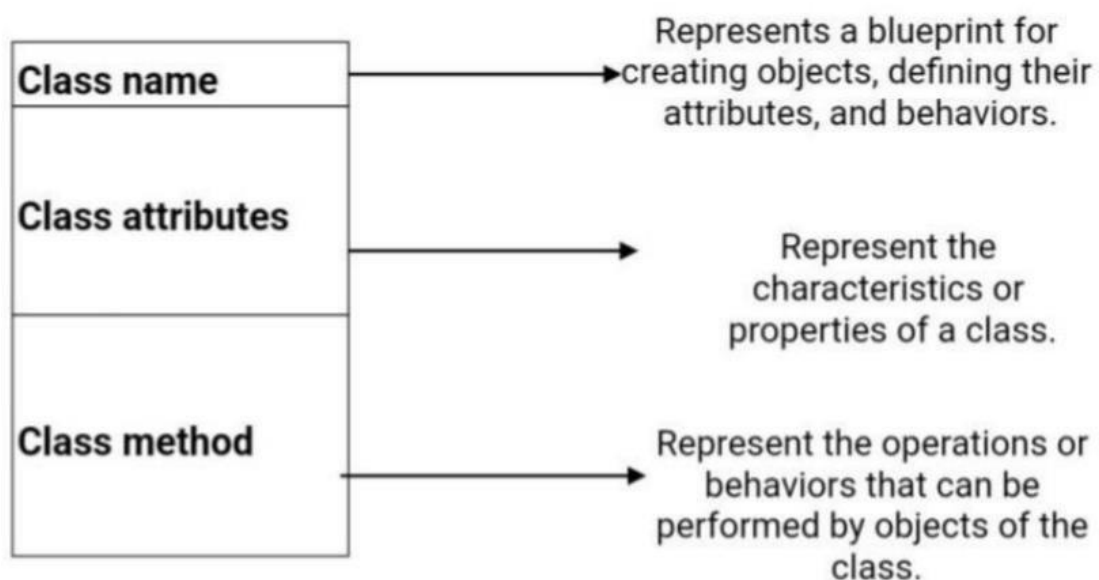
## **4.4 Logic Diagram**

A logical diagram provides a graphical view of the structure of an information system helps you analyses the structure of your data system through entities relationships, in which primary identifiers migrate along one-to-many relationships to become foreign

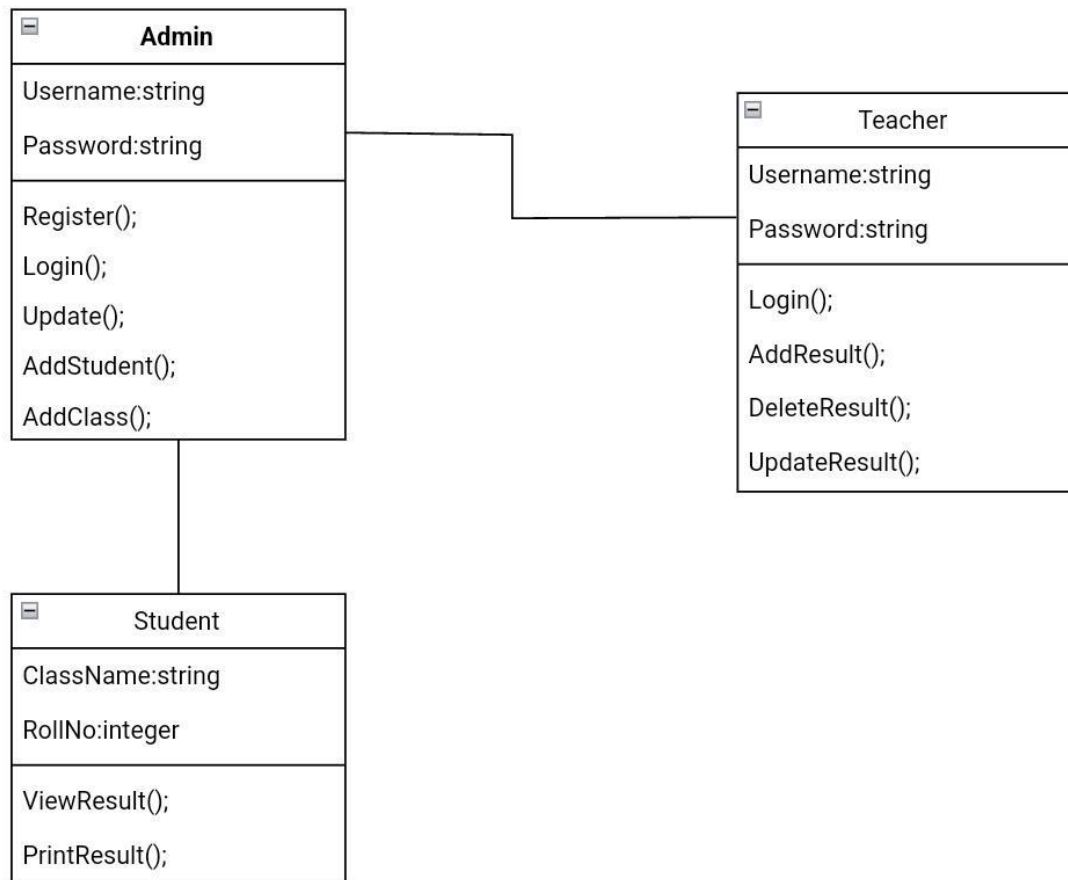
identifiers, and many-to-many relationships can be replaced by intermediate entities. Logical diagram is used to ensure the client understands the proposed system.

- **Class Diagram :-**

A class diagram is a type of structural diagram in UML (Unified Modeling Language) that represents the structure and relationships of classes within a system. It visually illustrates the classes in a system, along with their attributes and methods with other classes. Class diagrams for “Student Result Management System” will look like :







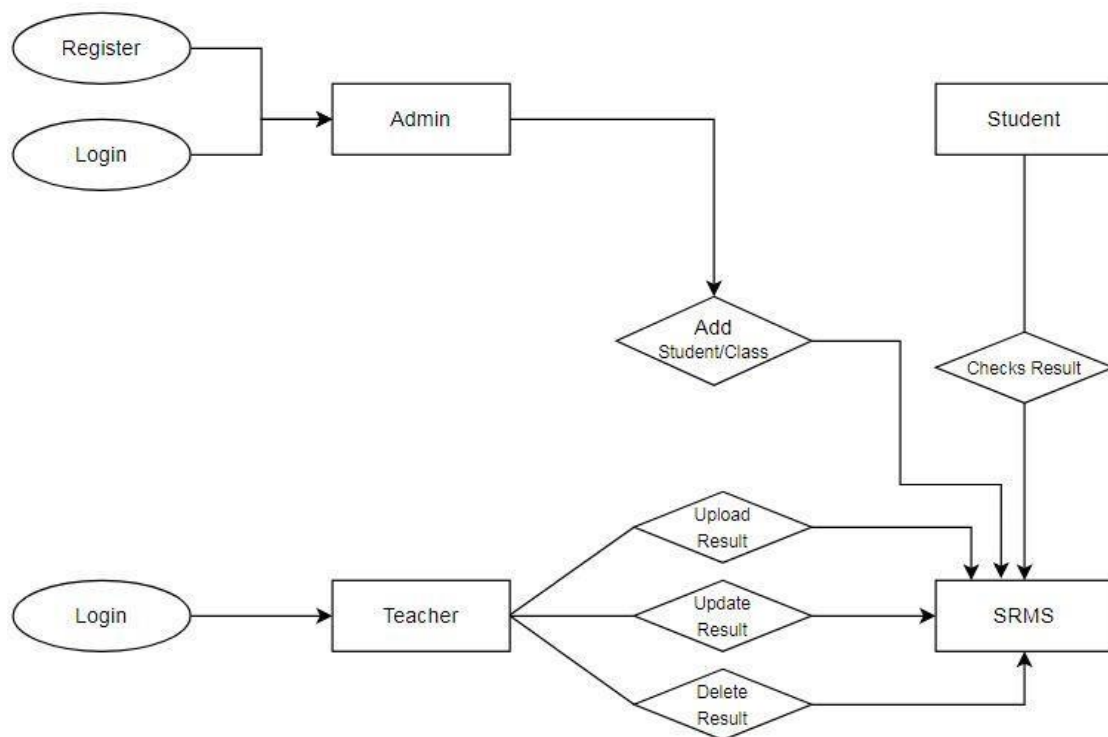




- **ER Diagram :-**

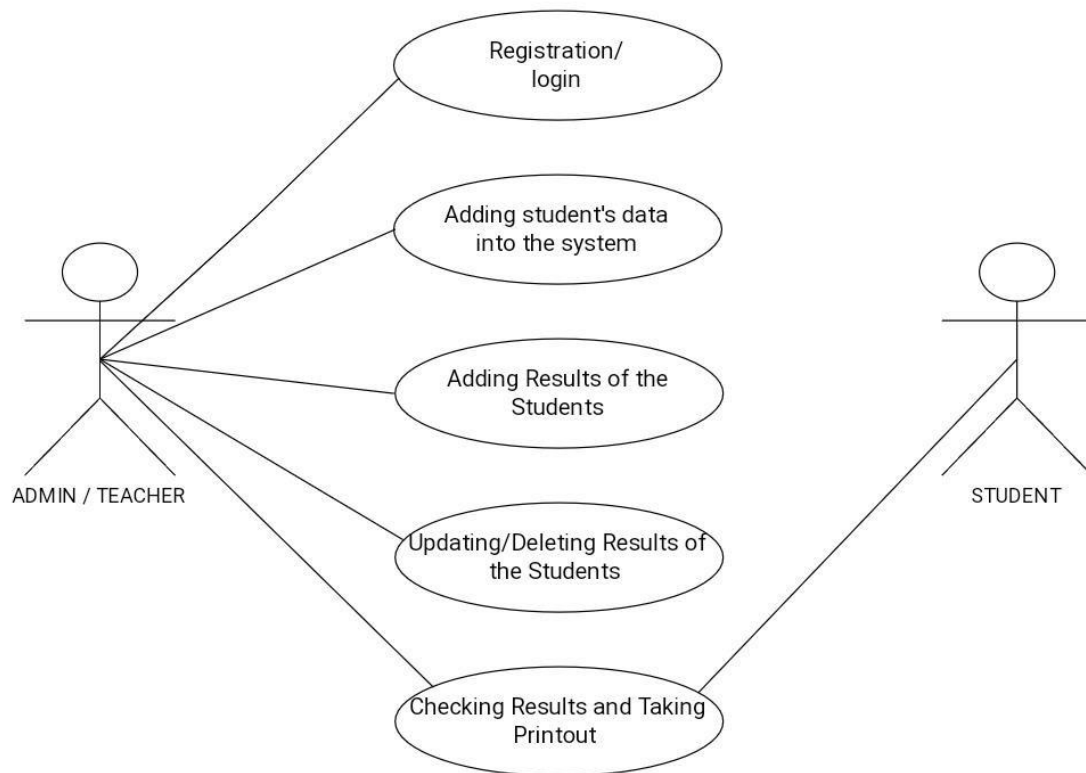
An Entity-Relationship (ER) diagram is a type of data model used to visualize the logical structure of a database. It represents entities (things of interest) and their relationships in a system . They help in identifying entities, their attributes, and the relationships between them, which is crucial for creating an efficient and well-structured database schema.

Symbol	Name	Description
	Rectangle	Represent Entity
	Diamond	Represent Relation
	Attribute	Represent Attribute
	Connection	Represent Link between Attributes , Entities and Relation



- **Use Case Diagram:-**

A Use Case Diagram is a visual representation that illustrates the interactions between different actors (users or external systems) and the system under consideration. It helps to capture the functional requirements of the system by showing the different ways in which users interact with the system and the corresponding actions performed. Here's a simplified textual representation of the Use Case Diagram for the "Student Result Management System" project:



## 4.5 Security Issues

1. **Unauthorized Access:** Ensure that only authorized users have access to specific functionalities and data within the system by implementing robust authentication and access control mechanisms.
2. **Data Integrity:** Implement measures to prevent tampering with student records or results, such as encryption and digital signatures, to maintain data integrity.
3. **Injection Attacks:** Protect against SQL injection and other forms of injection attacks by validating and sanitizing user input before processing it to prevent malicious code execution.
4. **Sensitive Data Exposure:** Encrypt sensitive data, such as student grades and personal information, both at rest and in transit, to prevent unauthorized disclosure and maintain confidentiality.
5. **Cross-Site Scripting (XSS):** Validate and sanitize user input to prevent XSS attacks, which could be used to execute malicious scripts in users' browsers and compromise the security of the system.

## 4.6 Test Case Design

Test Case Id	Description	Steps/Action	Expected Result	Pass/Fail
TC-01	To verify that users can log in successfully with valid credentials.	Navigate to the login page, enter valid username and password and click on the login button.	User should be logged in and directed to the dashboard or appropriate landing page.	Pass
TC-02	To verify that users cannot log in with invalid credentials.	Navigate to the login page , enter invalid username or password and click on the login button.	User should not be logged in, and an error message indicating invalid credentials should be displayed.	Fail
TC-03	To verify that new students can be added to the system.	Navigate to the student management section , click on the "Add Student" button , fill in the required student details and click on the "Save" or "Submit"	New student should be added successfully, and their details should be displayed in the student list.	Pass

		button.		
TC-04	To verify that grades for students can be entered accurately.	Navigate to the result management section , select a student and Enter grades for various subjects. Click on the “Save” or “Submit” button.	Grades should be saved successfully for the selected student.	Pass
TC-05	To verify that students can view their results.	Navigate to the results section and view the grades for various subjects.	Student should be able to view their grades accurately.	Pass

# **Implementation and Testing**

## **Testing Approach :-**

Software Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. It can also provide an objective, independent view of the software to allow the business to appreciate and understand the risk of software implementation. The basic purpose of testing is to detect the errors that may be present in the program. Testing as the process of executing a program with the intent of finding errors.

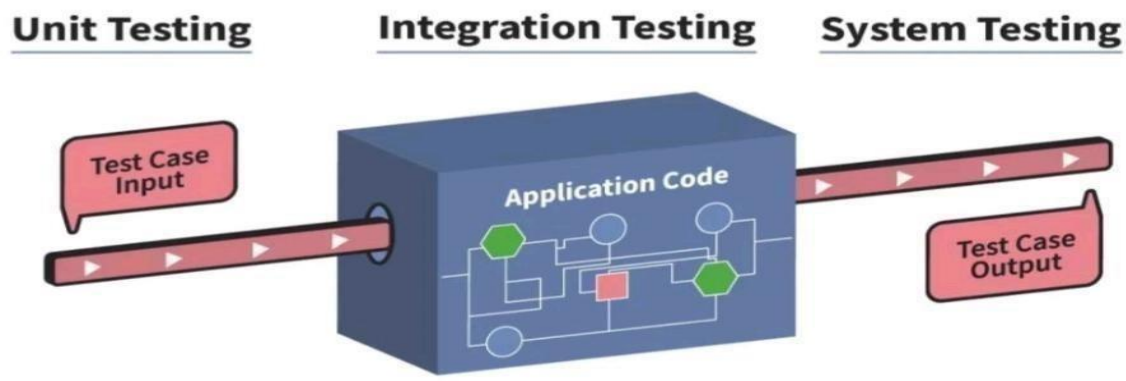
## **The Box Approach :-**

Software testing methods are traditionally divided into white-box and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

## **White-box Testing :-**

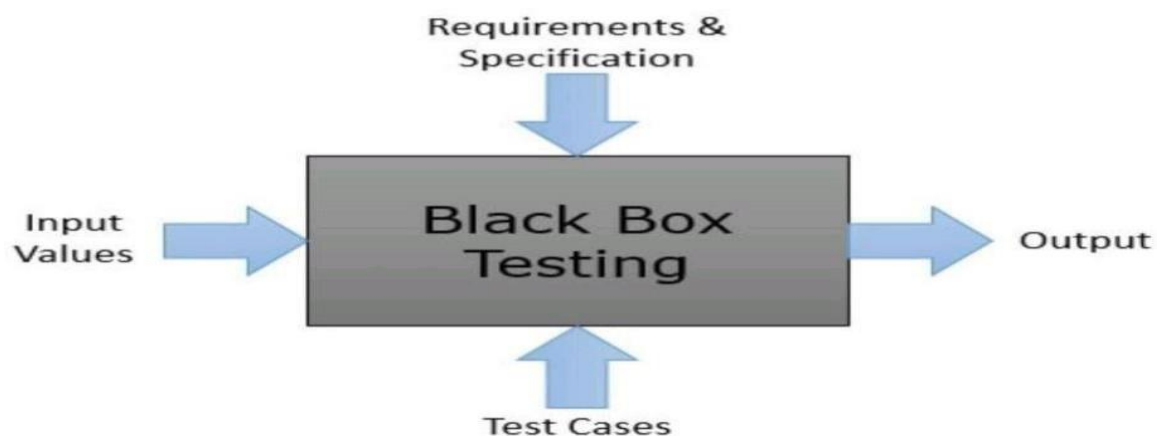
White box testing also known as clear testing, glass testing, and transparent box testing and structural testing. In white box testing an internal perspective of the system, as well as programming skills, are used to design test case. The testers choose inputs to exercise paths through the code and determine the appropriate outputs. While white box testing can be applied at the unit, integration and system levels of the

software testing process, it usually done at the unit level.



### **Black-box Testing :-**

Black box testing treats the software as a “black box”, examining functionally without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black box Testing methods include : equivalence portioning, boundary value analysis, all pairs testing state transition tables, decision table testing, fuzz testing , model –based testing, use-case testing, exploratory testing and specification-based testing.





## **Levels of Testing :**

The levels of testing are as follows :

- 1) Unit Testing
- 2) Integration Testing
- 3) System Testing
- 4) Acceptance Testing

### **1. Unit Testing :-**

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as “Module Design”. This testing carried out during programming stage itself. In this testing each module is found to be working satisfactorily as regard to the expected output from the module. All textboxes are having validation by which they will not remain empty and all work properly as expected. To store mobile number then the length must be 10 digits and only digit is allowed.

### **2. Integration Testing :-**

Integration testing is systematic testing for construction the program structure while at the same time conducting tests to uncover errors associated with in the Interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of cause is complicated by the vast expense of the entire program. In Integration testing I test the system

combining all modules. All the customer data including his id, name, package that assign to member and total amount show together .

### **3. System Testing :-**

It is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then goal will be successfully achieved.

### **4. Acceptance Testing:-**

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

## **Coding details and code efficiency**

### **Code Efficiency Considerations:**

- **Data Structures:** Use appropriate data structures like dictionaries or lists to efficiently store and manage user and student data.
- **Modularization:** Organize code into modular functions and classes to improve readability and maintainability.

- **Error Handling:** Implement robust error handling to gracefully manage exceptions and unexpected inputs.
- **Algorithm Optimization:** Optimize algorithms for tasks like calculating averages or searching for student records to ensure efficient performance, especially with large datasets.

## Coding Details

- **Sign up code :-**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Signup Page</title>
  <link rel="stylesheet" href="css/signup.css">
  <link      rel="stylesheet"      href="./font-awesome-
4.7.0/css/font-awesome.css">
  <link
href="https://fonts.googleapis.com/css?family=Roboto"
rel="stylesheet">
</head>
<body>
  <div class="title">
    <span>Student Result Management System</span>
  </div>
```

```
<div class="main">
  <div class="signup">
    <form action="signup_process.php" method="post"
name="signup" onsubmit="return validateForm()">
      <fieldset>
        <legend class="heading">Signup</legend>
        <input      type="text"      name="username"
id="username"          placeholder="Username"
autocomplete="off" required>
        <input type="password" name="password"
id="password"          placeholder="Password"
autocomplete="off" required>
        <input type="submit" value="Signup">
      </fieldset>
    </form>
    <div class="switch-login">
      <p>Already      have      an      account?      <a
href="login.php">Login</a></p>
    </div>
  </div>
</div>
```

```
<div class="contact-us">
  <p>Contact Us:</p>
  <ul>
    <li><a
href="mailto:sachinisamantri@gmail.com">Email      :
sachinisamantri@gmail.com</a></li>
    <li><a
href="https://api.whatsapp.com/send/?phone=919324209
```

```
076&text&type=phone_number&app_absent=0">Whats  
App : 9324209076</a></li>
```

```
</ul>
```

```
</div>
```

```
<script>
```

```
function validateForm() {
```

```
    var                                username                                =
```

```
document.getElementById("username").value;
```

```
    var                                password                                =
```

```
document.getElementById("password").value;
```

```
    // Check if username is empty
```

```
    if (username.trim() === "") {
```

```
        alert("Please enter a username.");
```

```
        return false;
```

```
    }
```

```
    // Check if password is empty
```

```
    if (password.trim() === "") {
```

```
        alert("Please enter a password.");
```

```
        return false;
```

```
    }
```

```
    // Check if password is at least 8 characters long
```

```
    if (password.length < 8) {
```

```
        alert("Password must be at least 8 characters  
long.");
```

```
        return false;
```

```
    }
```

```

// Check if password contains at least one symbol
if (!/[!@#$$%^&*(),.?":{ }|<>]/.test(password)) {
    alert("Password must contain at least one symbol.");
    return false;
}

// Check if password contains at least one numeric
character
if (!/\d/.test(password)) {
    alert("Password must contain at least one numeric
character.");
    return false;
}

// If all validations pass, return true to submit the form
return true;
}
</script>
</body>
</html>

```

- **Login Page code :-**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Index Page</title>

```

```

<link rel="stylesheet" href="css/login.css">
<link rel="stylesheet" href="./font-awesome-
4.7.0/css/font-awesome.css">
<link
href="https://fonts.googleapis.com/css?family=Roboto"
rel="stylesheet">
</head>
<body>
  <div class="title">
    <span>Anam Cara Tutorial</span>
  </div>
  <div class="main">
    <div class="login">
      <form action="login_process.php" method="post"
name="login">
        <fieldset>
          <legend class="heading">Admin
Login</legend>
          <input type="text" name="userid"
placeholder="Username" autocomplete="off">
          <input type="password" name="password"
placeholder="Password" autocomplete="off">
          <input type="submit" value="Login">
          <div class="switch-login">
            <p>Don't have an account? <a
href="signup.php">Signup</a></p>
          </div>
        </fieldset>
      </form>
    </div>

```

```

<div class="search">
    <form action="./student.php" method="get">
        <fieldset>
            <legend class="heading">For
Students</legend>

            <?php
                include('init.php');

                $class_result=mysqli_query($conn,"SELECT `name`
FROM `class`");
                echo '<select name="class">';
                echo '<option selected disabled>Select
Class</option>';
                while($row =
mysqli_fetch_array($class_result)){
                    $display=$row['name'];
                    echo '<option
value="'. $display. ">'. $display. '</option>';
                }
                echo '</select>'
            ?>
            <input type="text" name="rn"
placeholder="Roll No">
            <input type="submit" value="Get Result">
        </fieldset>
    </form>
</div>
</div>

```



```

<div class="contact-us">
  <p>Contact Us:</p>
  <ul>
    <li><a
href="mailto:sachinisamantri@gmail.com">Email :
sachinisamantri@gmail.com</a></li>
    <li><a
href="https://api.whatsapp.com/send/?phone=919324209
076&text&type=phone_number&app_absent=0">Whats
App : 9324209076</a></li>
  </ul>
</div>
</body>
</html>

```

- **Dashboard Code :-**

```

<?php
  Include("init.php");

  $no_of_classes=mysqli_fetch_array(mysqli_query($conn,"SE
LECT COUNT(*) FROM `class`"));

  $no_of_students=mysqli_fetch_array(mysqli_query($conn,
"SELECT COUNT(*) FROM `students`"));

  $no_of_result=mysqli_fetch_array(mysqli_query($conn,
"SELECT COUNT(*) FROM `result`"));
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/home.css">
  <link
href=https://fonts.googleapis.com/css?family=Roboto
rel="stylesheet">
  <link rel="stylesheet" href="/css/font-awesome-
4.7.0/css/font-awesome.css">
  <link rel="stylesheet" href="normalize.css">
  <title>Dashboard</title>
  <style>
    .main{
      Border-radius: 10px;
      Border-width: 5px;
      Border-style: solid;
      Padding: 20px;
      Margin: 7% 20% 0 20%;
    }
  </style>
</head>
<body>
  <div class="title">
    <a href="dashboard.php"></a>
    <span class="heading">Dashboard</span>
    <a href="logout.php" style="color: white"><span

```

```

class="fa fa-sign-out fa-2x">Logout</span></a>
</div>
<div class="nav">
  <ul>
    <li class="dropdown"
onclick="toggleDisplay('1')">
      <a href="" class="dropbtn">Classes &nbsp;
        <span class="fa fa-angle-down"></span>
      </a>
      <div class="dropdown-content" id="1">
        <a href="add_classes.php">Add Class</a>
        <a href="manage_classes.php">Show
Classes</a>
      </div>
    </li>
    <li class="dropdown"
onclick="toggleDisplay('2')">
      <a href="#" class="dropbtn">Students &nbsp;
        <span class="fa fa-angle-down"></span>
      </a>
      <div class="dropdown-content" id="2">
        <a href="add_students.php">Add
Students</a>
        <a href="manage_students.php">Show
Students</a>
      </div>
    </li>
    <li class="dropdown"
onclick="toggleDisplay('3')">
      <a href="#" class="dropbtn">Results &nbsp;

```

```

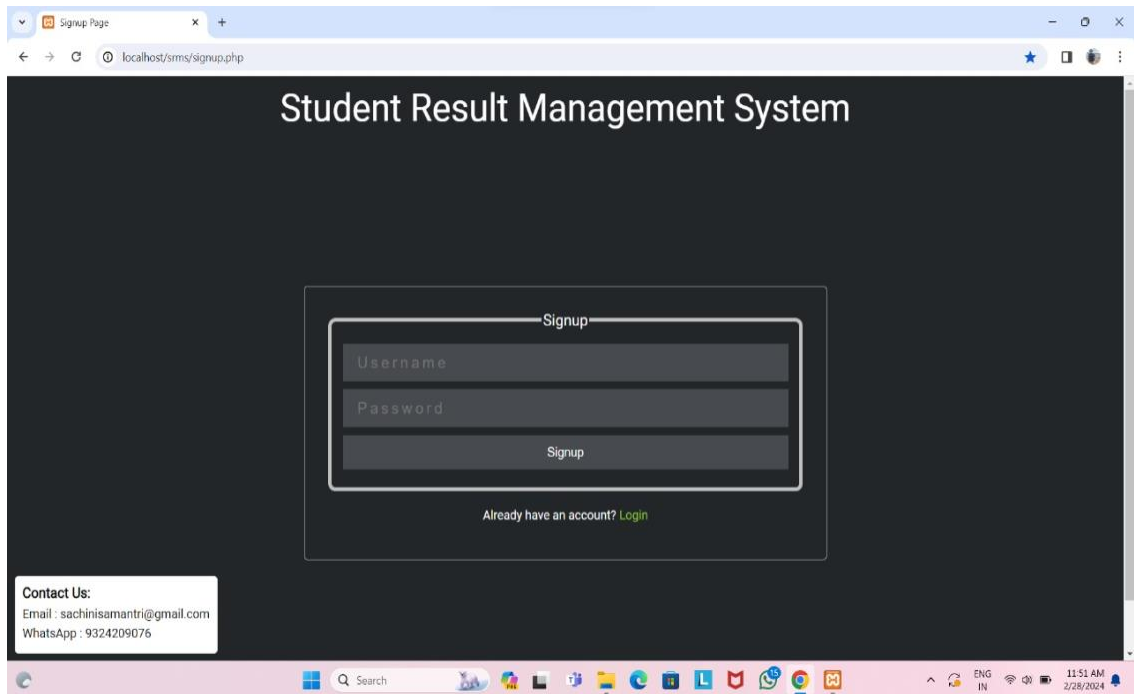
        <span class="fa fa-angle-down"></span>
    </a>
    <div class="dropdown-content" id="3">
        <a href="add_results.php">Add Results</a>
        <a href="manage_results.php">Manage
Results</a>
    </div>
</li>
</ul>
</div>
<div class="main">
    <?php
        Echo '<p>Number of
classes:'. $no_of_classes[0]. '</p>';
        Echo '<p>Number of
students:'. $no_of_students[0]. '</p>';
        Echo '<p>Number of
results:'. $no_of_result[0]. '</p>';
    ?>
</div>
<div class="footer">
</div>
</body>
</html>

<?php
    Include('session.php');
?>

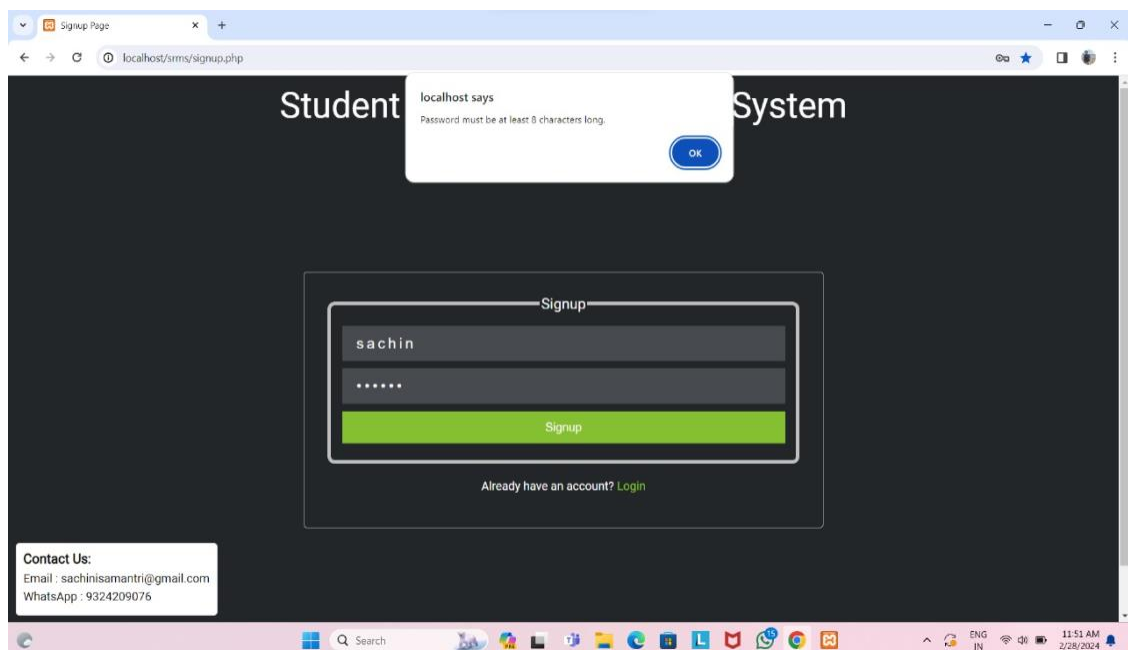
```

# Results

- **Signup Page**



- **Signup Validation**



- Login Page

Index Page

localhost/srms/login.php

## Anam Cara Tutorial

Admin Login

Username

Password

Login

Don't have an account? [Signup](#)

For Students

Select Class

Roll No

Get Result

Contact Us:  
Email : sachinisamantri@gmail.com  
WhatsApp : 9324209076

Search

ENG IN 11:51 AM 2/28/2024

- Dashboard

Dashboard

Logout

Classes Students Results

Number of classes:0

Number of students:0

Number of results:0

Search

ENG IN 11:52 AM 2/28/2024

## Conclusion and Future Work

### Conclusion:

In conclusion, the student result management system presented here provides a solid foundation for managing student information, grades, and user authentication. The system includes features such as user login, student management, result management, and basic security measures like password hashing and access control.

Through the implementation of classes and functions, the system ensures modularity, readability, and maintainability of the codebase. It adheres to coding best practices and includes error handling to manage unexpected scenarios gracefully.

### Future Work:

While the current implementation serves as a functional prototype, there are several areas for future enhancement and expansion :

- **User Interface:** Develop a user-friendly interface using web frameworks like Flask or Django to allow users to interact with the system through a web browser.
- **Database Integration:** Integrate a database system like SQLite or PostgreSQL to store user credentials, student information, and grades persistently.
- **Enhanced Security Measures:** Implement additional security measures such as role-based access control

(RBAC), input validation, and session management to further enhance system security.

- **Reporting and Analytics:** Add reporting and analytics features to generate insights from student data, such as performance trends and class averages.
- **Notification System:** Implement a notification system to alert users about important updates, such as new grades or system announcements.
- **Scalability:** Optimize the system architecture and algorithms to handle large datasets efficiently, ensuring scalability as the user base grows.
- **Integration with Learning Management Systems (LMS):** Integrate with existing LMS platforms to streamline data exchange and provide a seamless experience for educators and students.
- **Mobile Application:** Develop a mobile application to provide on-the-go access to student information and results for both administrators and students.

By addressing these areas for future work, the student result management system can evolve into a comprehensive solution that meets the needs of educational institutions, administrators, teachers, and students alike.



## **Reference**

While developing this project internet was the eternal support. Following are the websites referred by us which helped us in developing our projects

- Wikipedia
- Google (application) for problem solving
- YouTube (application)