Mini Project Report on

**Student Result Management System**

Submitted in partial fulfillment of the requirements of the degree of Bachelor in Engineering

By

|  |  |  |
| --- | --- | --- |
| **Khushboo Rathod** | **TE4** | **48** |
| **Vinay Savla** | **TE4** | **52** |
| **Divyaa Thaivalappil** | **TE4** | **62** |
| **Nikhil Vaishya** | **TE4** | **65** |

Under the guidance of

**MS. Vaishali Hirlekar**



**DEPARTMENT OF COMPUTER ENGINEERING SHAH AND ANCHOR KUTCHHI ENGINEERING COLLEGE**

**CHEMBUR, MUMBAI – 400088.**

**University of Mumbai (AY 2023-24)**

**CERTIFICATE**

This is to certify that the Mini Project entitled **“Student Result Management System”** is a bonafide work of **Khushboo Rathod, Vinay Savla, Divyaa Thaivalappil, Nikhil Vaishya** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **“Bachelor of Engineering”** in **“Computer Engineering”.**

###### (MS. Vaishali Hirlekar)

Guide

###### (Prof. UDAY BHAVE) (Prof. BHAVESH PATEL)

Head of Department Principal

# Mini Project Approval

This Mini Project entitled “**Student Result Management System”** by **Khushboo Rathod, Vinay Savla, Divyaa Thaivalappil, Nikhil Vaishya** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

##### Examiners

**1………………………………………**

(Internal Examiner Name & Sign)

###### 2…………………………………………

(External Examiner Name & Sign)

Date:

Place:

**Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of student** | **Class** | **Roll No.** | **Signature** |
| **Khushboo Rathod** | **TE4** | **48** |  |
| **Vinay Savla** | **TE4** | **52** |  |
| **Divyaa Thaivalappil** | **TE4** | **62** |  |
| **Nikhil Vaishya** | **TE4** | **65** |  |
| Date:  Place: |  |  |  |

**Abstract**

The Student Result Management System (SRMS) is a comprehensive software solution

designed to streamline and enhance the management of student academic results within educational institutions. This project report provides an overview of the development and implementation of SRMS, highlighting its key features, functionality, and benefits.

The primary objective of SRMS is to automate and simplify the process of result generation, storage, and retrieval. The system offers a user-friendly interface for administrators, teachers, and students, allowing them to efficiently input, manage, and access student performance data. SRMS is developed with modern web-based technologies, ensuring accessibility from any device with internet connectivity.

**Table of Contents**

### Page Number

Chapter 1: Introduction 1

* 1. [Introduction](#_bookmark0)
  2. [Motivation](#_bookmark1)

[Chapter 2: Literature Review 2](#_bookmark2)

[2.1 Literature survey](#_bookmark3)

[Chapter 3: Problem Definition and Objectives](#_bookmark4) 3

* 1. Proposed System
  2. Features
  3. Database Design
  4. Outputs

### Chapter 4: Future Scope and Conclusion 9

**Chapter 5: References 10**

**List of tables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter** | **Table** | **Table Name** | **Page Number** |
| Chapter 2 | 2.1 | Literature Survey | 2 |

**List of Figures:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter** | **Table** | **Table Name** | **Page Number** |
| Chapter 3 | 3.4 | Outputs | 5 |

**Chapter 1**

## Introduction:

## The educational landscape is continually evolving, with the integration of technology playing a pivotal role in reshaping traditional learning and administrative processes. One such technological advancement is the implementation of Student Result Management Systems (SRMS) within educational institutions. This project report serves as an introduction to SRMS, a comprehensive software solution designed to revolutionize the way academic results are managed and accessed, offering enhanced efficiency, accuracy, and accessibility.

## In the context of modern education, the management of student results is an integral component of the administrative machinery. Academic institutions, ranging from schools to universities, generate and maintain an extensive amount of student performance data. The traditional manual methods of result generation, recording, and dissemination are not only time-consuming but also susceptible to errors, data loss, and inefficiencies.

## The Student Result Management System presented in this report addresses these challenges by providing a holistic and automated approach to result management. SRMS is developed using contemporary web-based technologies, enabling users to access the system from any device with an internet connection. The system is designed with the primary aim of simplifying and streamlining the entire result management process.

## This project report will delve into the fundamental aspects of SRMS, offering insights into its features, functionality, and benefits. It will provide an overview of the development and implementation of SRMS, highlighting the impact it has on various stakeholders within an educational institution. The report will also discuss the significance of SRMS in the context of modern education, emphasizing its role in improving efficiency, data integrity, and communication among administrators, teachers, and students.

## Furthermore, the report will explore the key features and capabilities of SRMS, such as user authentication, result entry, automated result generation, secure data storage, student portals, reporting and analytics, notifications, and data export. These features collectively contribute to a more efficient, accurate, and accessible result management process.

## As we delve deeper into this project report, we will uncover the numerous benefits that SRMS offers to educational institutions. These advantages include enhanced efficiency, reduced administrative workload, improved accuracy and data integrity, greater accessibility, and enriched communication between key stakeholders. Additionally, SRMS equips administrators with valuable insights through data analysis, enabling them to make informed decisions and further improve the educational system.

## In summary, this project report serves as a gateway to understanding the Student Result Management System and its transformative impact on the educational sector. It emphasizes the significance of automating and modernizing result management processes, and it underscores the benefits of SRMS in fostering a more efficient, accurate, and student-centered academic environment. Through the following sections of this report, we will explore the development, implementation, and functionalities of SRMS in detail, shedding light on its role in shaping the future of education.

## 1.2 Motivation:

The motivation behind the development of a Student Result Management System (SRMS) is a critical aspect of any project report. It provides insight into the reasons and objectives that drove the creation of such a system. In a project report, the motivation behind the SRMS can be outlined as follows:

Efficiency Improvement: One of the primary motivations for creating an SRMS is to enhance the efficiency of academic result management. Traditional, paper-based systems are often time-consuming, error-prone, and labor-intensive. The SRMS aims to automate and streamline these processes, reducing administrative workload and the likelihood of human errors.

Data Accuracy and Integrity: Another crucial motivation is to ensure the accuracy and integrity of student performance data. Inaccurate or incomplete results can lead to misunderstandings and disputes. An SRMS aims to minimize such issues by automating result calculations and data storage in a structured and secure manner.

Accessibility and Transparency: With the advent of digital technology, accessibility and transparency are increasingly important in education. The SRMS is motivated by the desire to make academic results accessible to all relevant parties—students, teachers, parents, and administrators. This transparency fosters trust and encourages engagement in the educational process.

Efficient Record Keeping: In addition to managing current results, the SRMS is motivated by the need for efficient, digital record keeping. It allows educational institutions to maintain historical academic data in an organized and easily retrievable manner.

# Chapter 2 Literature Review

### Literature Survey:

We had searched about this type of application on the internet but got one which was not so user-friendly, more efficient and easier so the features added in this project is our own idea. We took the help of the below-mentioned article to build this project.

### Literature Review

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Author/Title/Year** | **Work done/ Algorithm/ Concept/ Idea presented in the paper** | **Remarks** |
| 1 | "STUDENT RESULT MANAGEMENT SYSTEM "International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.9, Issue 4, page no.h182-h185, April-2022 | http://www.jetir.org/papers/JETIR2204722.pdf | This article is about a student result management system. It discusses what the system is and why it was created. It also goes into detail about the system’s features. Some of the important points are that the system is web-based and uses PHP, MySQL, HTML, CSS, and JavaScript. It is also designed to be easy to use for both students and institutions. |
| 2 | Kuznetsov, A. (2020). Building a Node.js Backend With Express. | https://www.smashingmagazine.com/2020/04/nodejs-backend-express-building-rest-api/ | Building a Node.js Backend With Express teaches how to build backend applications using the Node.js runtime environment and the Express framework. This covers a wide range of topics, from creating a new Node.js project with Express to deploying a finished application to production. |
| 3 | Wijesinghe, R. A. U., & Jayarathna, N. A. M. (2020). Building a Student Management System Using ReactJS and ASP.NET Core Web API. In 2020 4th International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 587-591). | https://journals.ieeeauthorcenter.ieee.org/when-your-article-is-published/abstracting-indexing-ai-databases/ | This paper presents the design and implementation of a student management system (SMS) using ReactJS and ASP.NET Core Web API. The SMS allows users to manage student information, such as their personal details, academic records, and attendance. It also allows users to generate reports on student performance. The SMS is designed to be scalable and extensible, and it can be easily adapted to meet the specific needs of different educational institutions. |
| 4 | React Js Documentation, by Facebook | https://react.dev/learn | This page contains all react syntax and usage. |

**Chapter 3**

**Proposed System**

* 1. **Proposed System:**

### The Student Result Management System (SRMS) contains three modules namely Admin, Faculty, and Student. Each of these modules includes profile viewing as a common feature. According to the authorization given to each user, this system adds functionalities for them.

### Admin

### The system administrator manages the overall system using this admin module. The admin can add or delete departments, admin per department, subjects per department, teachers, and students. This will allow them to handle all departments as a common admin. They can also create notices for teachers and students. This is useful for communicating important information to teachers and students, such as changes to the syllabus or important deadlines. They can add or delete subjects related to their department. As mentioned above admins can add admins, which means each department will also have its own admin, which will handle departmental activities. Admin or departmental admin can add subjects year-wise.

### Faculty

### Teachers create tests and add marks for each student using the faculty module. The test creation will include the creation of tests like Internal Assessment 1 (IA1), Internal Assessment 2 (IA2), End Semester Examination (ESE), Term Work (TW) during the whole syllabus, and Practical Examination. This test will give access to add marks for the same. It will ask for maximum marks for the test while creating it. Teachers have access to create tests and upload marks for the same only.

### Student

### Students can view their results and see subject lists using the student module. This is the only feature available for students.

### Features:

### Easy to navigate and use

### Student Result Management System also manage the department, student and faculty details

### Increase efficiency of managing the Student Result

### Time saver

### Database design

MongoDB is a document-oriented database that stores data in JSON-like documents. This makes it a good choice for storing large amounts of unstructured data, such as student records and test scores. In our database, we have a total of six to seven tables:

**Admins**: This table stores information about the administrators of the system. The fields in this table include their username, password, registration number, name, email address, contact number, date of birth, joining year, and department. This is a common table for both admin and departmental admin.

**Departments**: This table stores information about the departments. The fields in this table include the department name and code.

**Faculties**: This table stores information about the faculties. The fields in this table include their username, password, name, gender, designation, department, contact number, email address, date of birth, joining year, and registration number.

**Marks**: This table stores information about the marks for each student in each subject. The fields in this table include the student ID, exam ID, and total marks.

**Notices**: This table stores information about the notices that have been created. The fields in this table include the notice topic, content, from details, notice-for details, and date created.

**Students**: This table stores information about the students. The fields in this table include their username, password, name, contact number, email address, date of birth, current year, subjects, gender, parent details, department, section, batch, and registration number.

**Subjects**: This table stores information about the subjects that are being offered. The fields in this table include the subject name, subject code, total lectures, year, and department.

**Tests**: This table stores information about the tests that have been created by the faculties. The fields in this table include the test name, subject code, department, total marks, year, section, and date created.

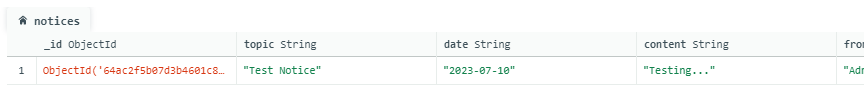


Figure 3.3.2: Notices Table

Figure 3.3.1: Admins Table

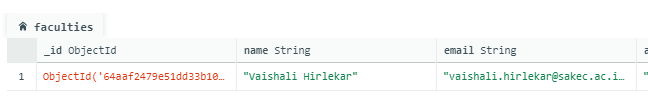
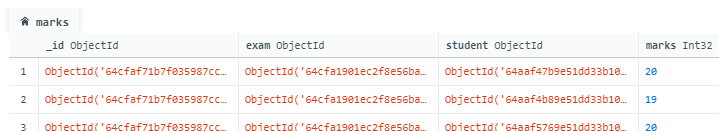


Figure 3.3.3: Faculties Table



Figure 3.3.4: Marks Table

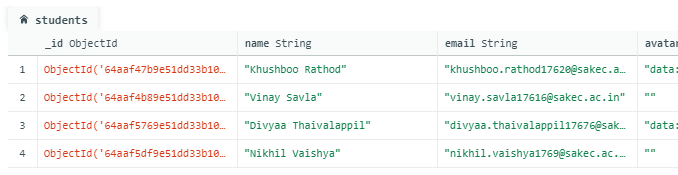
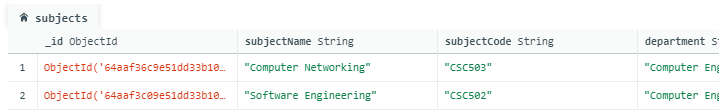


Figure 3.3.7: Subjects Table

Figure 3.3.6: Students Table

Figure 3.3.5: Departments Table

## Output

Figure 3.4.8: Student Test Result Page

Figure 3.4.6: Faculty Create Test Page

Figure 3.4.4: Admin Add Faculty Page

Figure 3.4.2: Login Page

Figure 3.4.1: Homepage

Figure 3.4.7: Student Dashboard

Figure 3.4.5: Faculty Dashboard

Figure 3.4.3: Admin Dashboard

**Chapter 4**

**Future Scope:**

* **Integration with Learning Management Systems (LMS):** The SRMS can be expanded to integrate seamlessly with Learning Management Systems, allowing for a more comprehensive educational platform. This integration can provide a holistic view of student performance by combining academic results with other educational data.
* **Enhanced Data Analytics:** Future developments can focus on more advanced data analytics capabilities. The SRMS can leverage machine learning and artificial intelligence to provide deeper insights into student performance, identify trends, and offer personalized recommendations for improvement.
* **Mobile Application:** Developing a dedicated mobile application for the SRMS can enhance accessibility and convenience for both students and administrators. Mobile apps can provide on-the-go access to academic data and notifications.
* **Gamification and Student Engagement:** The future of SRMS could include gamification elements to motivate and engage students in their academic journey. Features such as achievement badges, leaderboards, and interactive dashboards can make the system more student-friendly.
* **Incorporation of Parent Portals:** Expanding the SRMS to include dedicated parent portals can foster greater parent-teacher communication and involvement in a student's education. Parents can access their child's academic progress and receive regular updates.
* **Multi-Language Support:** To accommodate a diverse student population, future versions of the SRMS can include multi-language support, making it more inclusive and accessible to a wider range of users.

**Conclusion**

## In conclusion, utilizing the MERN (MongoDB, Express.js, React, Node.js) stack for building a Systematic Review Management System (SRMS) offers a comprehensive and efficient solution.

## The integration of these technologies addresses various aspects of SRMS development, ensuring seamless data management, user-friendly interfaces, and robust backend capabilities.

## By combining these technologies, an SRMS gains the ability to streamline the review process, manage data efficiently, and provide an intuitive user experience for both researchers and administrators.

**Chapter 4**

## References

## [1] Appointy Software Inc.

## [2] “Building a Student Result Management System with Node.js and React” by Kirill Konshin

## [3] “Building a Student Result Management System with Node.js and React” by Sarthak Sharma

## [4] “Student Result Management System using React and Node.js” by Hasib Al Rashid

## [5] “Building a Student Result Management System with React and Node.js” by Krunal Vyas