

STUDENT PERFORMANCE ANALYTICS

*A Dissertation submitted in partial fulfillment of the requirement for the award of the degree of
MASTER OF COMPUTER APPLICATIONS*

Of

VISVESVARAYA TECHNOLOGICAL UNIVERSITY



By

Ms. ANJANA ROSAMMA

4SN23MC003

Under the Guidance of

Internal Guide

Dr . SHASHIDHAR KINI K

External Guide

Mr. VIPIN RAO



DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

SRINIVAS INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi,

Recognized by Govt. of Karnataka)

Mangaluru -574143, Karnataka

AUGUST 2025

STUDENT PERFORMANCE ANALYTICS

A Dissertation submitted in partial fulfillment of the requirement for the award of the degree of
MASTER OF COMPUTER APPLICATIONS

Of

VISVESVARAYA TECHNOLOGICAL UNIVERSITY



By

Ms. ANJANA ROSAMMA

4SN23MC003

Under the Guidance of

Internal Guide

Dr . SHASHIDHAR KINI K

Head of Department
Department of MCA
SIT, Mangaluru

External Guide

Mr. VIPIN RAO

Founder and CEO
Dregal i Private Limited
Mangalore, Karnataka



DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

SRINIVAS INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi,

Recognized by Govt. of Karnataka)

Mangaluru -574143, Karnataka

SRINIVAS INSTITUTE OF TECHNOLOGY

(NAAC Accredited)

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Valachil, Mangalore-574143



**(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi,
Recognized by Govt.of Karnataka)**

CERTIFICATE

This is to certify that, Ms. ANJANA ROSAMMA bearing USN 4SN23MC003 has successfully completed her final semester project work entitled “STUDENT PERFORMANCE ANALYTICS” as a partial fulfillment for the award for Master of Computer Applications degree, during the academic year 2024-2025 under the joint supervision.

Dr. Shashidhar Kini K

Internal Project Guide
Head of Department
Department of MCA
SIT, Mangalore

Mr. Vipin Rao

External Project Guide
Dregal i Private Limited
Mangalore, Karnataka

Dr. Shashidhar Kini K

Head of Department
Department of MCA
SIT, Mangalore

Dr. Shrinivasa Mayya D

Principal
SIT, Mangalore

Date:

Examiners

Signature

1.

2.



Dregal i Private Limited

REGD. OFFICE: 1-5-19-T-13 63/25, No 409, Kamala Towers, Fourth Floor, Near Urwa market, Urwa Mangaluru 575006, Karnataka
Ph: 8951626455

Date: 03-09-2025

CERTIFICATE OF PROJECT COMPLETION

This is to certify that Mr./Ms. **ANJANA ROSAMMA** (USN:4SN23MC003), a bona fide student of **SRINIVAS INSTITUTE OF TECHNOLOGY, VALACHIL**, has completed the Project Work titled "**STUDENT PERFORMANCE ANALYTICS**" at **DREGAL I PRIVATE LIMITED** from 26th May 2025 to 02nd September, 2025 as a part of curriculum.

The intern is professionally sound, hardworking. A dedicated and a motivational whose contribution has helped in realization of organizational goals and objectives.

During the service period, the intern have been found sincere, reliable, trustworthy and open to all challenges.

We wish you all the very best in your future endeavors.

Sincerely,

For DREGAL | PRIVATE LIMITED

George 91
DIRECTOR



**Vipin Rao,
Founder & CEO,
Dregal i Private Limited**

Recognised By:



DECLARATION

I, Ms. **Anjana Rosamma** student of 4th sem MCA, **Srinivas Institute of Technology Valachil** bearing the USN **4SN23MC003** hereby declare that I have carried out the project entitled "**Student Performance Analytics**" under the supervision of External Guide **Mr. Vipin Rao** and Internal Guide **Dr. Shashidhar Kini K** Professor and Head , Department of MCA and submitted in partial fulfillment for the award of the degree Master of Computer Application during the year 2024-2025, from Visvesvaraya Technological University Jnana Sangama, Belagavi.

The work which is being presented in the project report submitted to the Department of MCA from Srinivas Institute of Technology, Valachil, Mangalore, is an authentic record of project work.

Date:

Place: Mangalore

Ms. Anjana Rosamma

Signature:

ACKNOWLEDGEMENT

The successful completion of this project report would not have been possible without the support and assistance of many individuals and organizations. I feel immensely blessed to have gotten this during the course of my project program. I would like to take this opportunity to offer my earnest admiration to each and every one of them.

Thanks to our internal guide, **Dr. Shashidhar Kini K**, Head of the Department MCA, for his guidance and advice during this project. I will never forget his timely support, guidance, encouragement, and more until my project get accomplished.

I express my sentiment of gratitude to **Mr. Vipin Rao**, who has been a continuous source of inspiration as my project guide. Without his constant guidance and suggestions, this report would have been nowhere near completion. My gratitude for his trust and generosity goes beyond words.

I sincerely thank our Head of the Department of MCA **Dr. Shashidhar Kini K**, for his timely suggestions and kind support toward the completion of the project.

I am also grateful to our principal **Dr. Shrinivasa Mayya D**, for encouraging and supporting me throughout the project period.

I also thank our Management who helped me directly and indirectly in completion of my project and our all teaching and non-teaching staff Department of MCA for their timely suggestions and kind support towards the completion of the project.

I thank my parents for their blessings and the constant support that they have been for me also thanks to my friends, classmates who helped me directly or indirectly to accomplish my work. There are not enough words to express how grateful I am to them for everything they have done for me.

INDEX

CHAPTER NUMBER	TITLE	PAGE NUMBER
1	INTRODUCTION	1-4
	1.1 OVERVIEW AND BACKGROUND	1
	1.2 PROBLEM STATEMENT	2
	1.3 OBJECTIVES	2
	1.4 COMPANY PROFILE	3
2	LITERATURE SURVEY	5-11
	2.1 EXISTING SYSTEM	6
	2.2 PROPOSED SYSTEM	6
	2.3 FEASIBILITY STUDY	7
	2.4 TOOLS AND TECHNOLOGIES USED	8
3	SOFTWARE REQUIREMENT SPECIFICATION	12-15
	3.1 PURPOSE	12
	3.2 SCOPE	13
	3.3 FUNCTIONAL REQUIREMENTS	13
	3.4 NON-FUNCTIONAL REQUIREMENTS	14
	3.5 HARDWARE AND SOFTWARE REQUIREMENTS	14
	3.5.1 SOFTWARE REQUIREMENTS	14
	3.5.2 HARDWARE REQUIREMENTS	15
4	SYSTEM DESIGN	16-20
	4.1 INTRODUCTION	16
	4.2 SCOPE	16
	4.3 SOFTWARE PRODUCT ARCHITECTURE	17
	4.4 DATA FLOW DIAGRAM	19

5	DETAILED DESIGN	21-23
	5.1 USE CASE DIAGRAM	21
	5.2 ACTIVITY DIAGRAM	22
	5.3 ER DIAGRAM	23
6	IMPLEMENTATION	24-28
	6.1 INTRODUCTION	24
	6.2 IMPLEMENTATION DETAILS	24
	6.3 ADMIN INTERFACE	26
7	TESTING	30-32
	7.1 INTRODUCTION	29
	7.2 TEST CASES	32
8	CONCLUSION	33
9	FUTURE ENHANCEMENT	34
Appendix A	BIBLIOGRAPHY	35
Appendix B	ADMIN MANUAL	36-39

LIST OF FIGURES

FIGURE NUMBER	FIGURES	PAGE NUMBER
4.1	ARCHITECTURAL DIAGRAM	17
4.2	DATA FLOW DIAGRAM	19
5.1	USE CASE DIAGRAM	21
5.2	ACTIVITY DIAGRAM FOR ADMIN	22
5.3	ER DIAGRAM	23
6.1	LOGIN PAGE	27
6.2	STUDENT PAGE	28
6.3	VIEW STUDENT TABLE PAGE	29



SRINIVAS INSTITUTE OF TECHNOLOGY

Mangaluru - 574 143.

INSTITUTION'S INNOVATION COUNCIL
Srinivas Institute Of Technology
IC202116073



CERTIFICATE

This is to certify that

ANJANA ROSAMMA

has participated and presented a paper titled,

STUDENT PERFORMANCE ANALYTICS

at the Two Days International Conference on

Recent Innovations in Computer Science

conducted on 13th and 14th May 2025 held at SIT, Mangaluru.

Dr. Sandeep Bhat
Convener ICRICS - 2025

Dr. Shrinivasa Mayya D
Principal

ABSTRACT

Student performance is among the most significant factors that shapes academic success and helps guide future learning strategies. This project focuses on developing a Student Performance Analytics web application using the MERN stack (MongoDB, Express.js, React.js, Node.js). The application is designed to systematically collect, analyze, and visualize academic records and feedback, providing a secure and user-friendly platform for students, teachers, and administrators. The goal is to support data-driven decision-making while making it easier to monitor academic progress.

The system is powered by a MongoDB database, which efficiently manages student details, courses, and performance data in a structured way. To ensure data protection, the application incorporates JWT-based authentication and password encryption. Students can register, log in, view their performance records, and receive personalized feedback. At the same time, teachers and administrators are able to add results, analyze academic trends, and generate detailed performance reports. An admin dashboard is also provided for effective management of student records, courses, and overall performance tracking.

A lot of stress has been put on security, reliability, and usability. The program has undergone extensive testing across devices to ensure smooth functionality. The interface is fully responsive, offering a clean, interactive, and intuitive user experience. With built-in charts and reports, academic progress can be easily visualized, making the data more meaningful and actionable.

Ultimately, the Student Performance Analytics system is more than just a tool for tracking grades. It empowers students to understand their strengths and areas for improvement, while giving educators the insights they need to guide learning more effectively. By bridging the gap between students and teachers, the system encourages collaboration and continuous improvement in academic performance.

CHAPTER 1**INTRODUCTION****1.1 OVERVIEW AND BACKGROUND**

The Student Performance Analytics Web Application is a straightforward yet robust platform crafted with the MERN stack (MongoDB, Express.js, React.js, and Node.js). Its primary goal is to simplify the management of academic and attendance data for teachers, students, and administrators all in one convenient location. Rather than relying on paper records or disorganized spreadsheets, this system consolidates everything into a secure and user-friendly application. Teachers can easily input grades, update attendance, and monitor each student's progress. In the meanwhile, students may access their accounts to monitor their attendance, performance history, and visual charts, which provides them with a clear picture of their development and motivates them to keep improving.

Administrators have access to a dashboard that allows them to manage student data, track overall performance, generate reports, and ensure the system runs smoothly. Standout features of the application is its data visualization capability. Instead of just being a collection of numbers, the information is presented in easy-to-understand charts and graphs. For example, a bar chart can illustrate attendance trends, a pie chart can highlight strengths by subject, and a trend graph can show how a student's performance has evolved over time. This makes it easier for teachers to see where their pupils are succeeding and where they may require additional a considerable amount of time guidance. Additionally, the system has intelligent search and filtering capabilities.

This makes it simple for teachers and administrators to locate student information by name, topic, USN, or even time period. Compared to going through several files, this saves a significant amount of time and work. The platform's architecture guarantees security and scalability, protecting data from unwanted access, and it is built to expand as additional students and records are added. The application for student performance analytics is a complete solution for improving educational results, not just a tool for preserving records.

By turning unstructured data into valuable insights, this tool acts as a smart assistant in the educational journey. With its real-time updates, secure login, and easy-to-understand visual reports, teachers, students, and administrators can work together to boost academic performance and foster a more effective learning atmosphere.

1.2 PROBLEM STATEMENT

Keeping track of student performance and attendance manually can really bog things down, leading to inefficiencies, mistakes, and inconsistencies. Inputting, computing, and assembling data takes a lot of time for educators and administrators, which only increases the possibility of mistakes and delays in report generation. Traditional methods of record-keeping make it tough to quickly spot strengths and weaknesses in different subjects or to identify students who might be at risk and need timely help. Because of this, important academic trends and insights can slip through the cracks, resulting in missed chances to enhance student outcomes.

Additionally, when performance records are kept manually or scattered across various sources, students usually lack visibility into their own progress. This makes it harder for them to track their attendance, understand their academic standing, or take the necessary steps to improve. Without a centralized, automated, and transparent system, both students and teachers struggle with effective communication, monitoring, and decision-making. That's why there's a pressing need for a digital solution that brings all this data together, automates performance tracking, and offers real-time visual insights to help with better academic planning and outcomes.

1.3 OBJECTIVES

- To offer a centralized system for managing and monitoring the student academic information.
- To enable teachers and administrators to add, edit, and track student achievement.
- To provide students with access to their own grades, attendance, and advancement.
- The development of reports and graphics to aid in analysis and decision-making.
- To enhance academic results and encourage the early identification of weak areas.
- To reduce manual effort and paperwork by automating the process of storing, retrieving, and analyzing student performance records.
- To ensure secure login and authentication so that only authorized users (students, teachers, administrators) can access and manage the data.
- To provide search and filtering options that allow quick access to student records by name, USN, subject, or time period.
- To support scalability and flexibility so the system can be expanded with more features or handle a growing number of students in the future.

1.4 COMPANY PROFILE

ORGANIZATION STRUCTURE

Since its founding in 2023, Dregali Pvt. Ltd. has quickly made a name for itself as a reliable supplier of customisable management software that doesn't require any code. Dregali provides scalable, inventive, and user-friendly digital solutions that are customized to meet specific needs, with a strong emphasis on empowering both businesses and students. Our main product, Durgasoft, makes it possible for businesses in a variety of sectors, including professional services, healthcare, education, taxation, and inventories, to digitize their operations without the need for technological know-how. Instead of being constrained by rigid, one-size-fits-all software, clients are able to select and customize the precise tools they require. Dregali provides students with project-assistance programs and internships in addition to business solutions. These internships are intended to give students practical industry exposure so they may work on real-world projects and put their theoretical knowledge to use. Under the direction of our team of skilled software engineers, students can also use these projects for their research, academic reports, and professional growth.

VISION

Our goal at Dregali Pvt. Ltd. is to revolutionize the way that companies and students use technology by offering training and platforms that are tailored to their requirements rather than making them adjust to software.

Our objective is to become the most trusted technology partner for SMEs and the most reliable source of project support for students, enabling both to thrive in a quickly changing digital economy.

MISSION

Give companies access to scalable, no-code software that simplifies processes. Encourage students' academic and professional development by providing internships and on-the-spot project support. Provide affordable, sector-neutral solutions that help SMEs close the technological divide. By enabling companies and individuals to install their own management systems without the requirement for programming expertise, you can foster creativity. Add value by combining state-of-the-art technologies with extensive industry understanding.

Fundamentally, Dregali is dedicated to creating a culture of creativity, flexibility, and mentoring, making sure that every solution and internship experience is technically sound, adaptable, and prepared for the future.

SERVICES OFFERED BY THE DREGALI SOLUTIONS

Through its internship programs and Durga soft platform, Dregali offers a whole range of services. Workflows, forms, and dashboards may be customized with no-code management software without the need for technical expertise. Industry-Specific Modules: Accounting, taxation, inventory, task tracking, patient management, student management, attendance, and more. Student internships and project support offer practical experience, real-world projects, and mentored assistance with academic project reports. Enterprise-grade encryption and backup systems for data security and compliance. Cloud-based solutions offer real-time updates and secure hosting for access from any location. Training & Support Staff training, ongoing assistance, and customized onboarding. Dregali guarantees that both clients and students have access to cutting-edge resources, solutions supported by research, and practical instruction by fusing technology with business logic and hands-on training.

COMPANY PROCESS AND CULTURE

Our culture at Dregali Pvt. Ltd. is centered on customer success, mentorship, and innovation. In order to improve our services, we continuously assess industry trends, emerging technology, and customer and student input. We firmly believe in continuous improvement. Market responsiveness is the ability to quickly adjust to changing academic and corporate needs. Client-Centric Design Developing adaptable solutions and internships that are grounded in practical issues. Hands-on Approach Giving companies and students projects and tools that they can set up, test, and implement right away. Innovation at the Core Motivating groups and interns to think creatively and creatively in order to produce more intelligent solution.

CHAPTER 2

LITERATURE SURVEY

Recently, there has been a growth in the sharp increase in the need for automated systems to track student progress and handle academic data. As data-driven decision-making becomes more and more important in education, researchers and developers have been putting in a lot of effort to create web-based systems that can efficiently collect, analyze, and report on student performance data. These systems are designed to eliminate human labor, cut down on mistakes, and offer an easy-to-use method of tracking academic achievement. These systems' potential has grown even more in tandem with the aid of contemporary technologies such as artificial intelligence and cloud computing , and machine learning, which enable automation, scalability, and predictive analysis. Several research have shown the benefits of combining visualization tools with academic data. For example, Kumar et al. created an academic monitoring system that digitally tracked student attendance and grades, enabling teachers to automatically generate reports and pinpoint students who may be struggling.

In a similar vein, Sharma and Verma developed a web-based application that integrated student records with visual dashboards, showcasing performance trends by subject and attendance data. Their system underscored the importance of real-time monitoring and quick access to valuable insights for educators and administrators. By using predictive analytics to assess student performance, Anderson et al. made a significant contribution. They were able to identify pupils who were at danger prior to the announcement of final results by utilizing algorithm for machine learning to predict academic outcomes. This method highlighted the significance of blending traditional record-keeping with cutting-edge data science techniques to foster personalized learning and timely interventions.

The importance of developing systems that do more than simply store academic data is well demonstrated by these research. By combining analytics, visualization, and prediction, we can create a strong framework that will save instructors a great deal of time while simultaneously improving student assistance. In today's educational environment, these instruments are necessary for raising student achievement, honing teaching methods, and encouraging openness among all parties.

2.1 Existing System

Most student performance management systems out there today are stuck in the past, relying on spreadsheets, manual record-keeping, or basic database tools that only churn out static reports. Without automation, interactive dashboards, or advanced features like visualization and predictive analysis, teachers often struggle to quickly identify trends or provide timely support for at-risk students. Teachers still manually calculate figures to generate reports in many schools, which is time-consuming and prone to errors. It can be challenging to gather all of the data for a useful analysis because it is often scattered across numerous registers or files. Additionally, our current tools hardly ever provide real-time insights, so teachers are forced to evaluate student performance at the end of a term. Since students don't have direct access to their progress data, it limits transparency and their chances for self-improvement. These issues clearly show the need for a centralized, automated, and user-friendly system that can simplify record management, ensure accuracy, and deliver valuable academic insights. Moreover, traditional systems lack predictive features that could help spot poor performance or irregular attendance early on—both crucial for timely interventions. With most current systems not allowing shared access or communication, collaboration between parents, teachers, and students is minimal. On top of that, manual records and basic tools don't provide enough protection against data loss or unauthorized access, raising concerns about the security of sensitive student information. All in all, the limitations in scalability and advanced analytics of these systems make them ill-suited to meet the growing demands of modern educational institutions.

2.2 Proposed System

The proposed Student Performance Analytics system is designed as a centralized platform built on the MERN stack, making it easy to organize, store, and evaluate student data efficiently. With features like interactive dashboards, infographics, attendance tracking, and performance visualization, educators can effectively monitor student progress, identify the areas that require improvement and make informed decisions to enhance learning outcomes. To promote transparency and personal growth, the system also allows students to access their academic records, which include grades, attendance, and detailed subject reports. Administrators can save a considerable amount of time and effort by automating the creation of reports. Cutting-edge data visualization technologies facilitate trend identification and offer prompt insights for academic planning. Additionally, the platform can accommodate an increasing number of students and courses without experiencing any performance issues because of its scalable architecture.

2.3 FEASIBILITY STUDY

The system is built on solid technical foundations, utilizing modern and popular technologies. The frontend is crafted with React.js, which promotes modularity and reusability, ensuring a smooth user experience with quick rendering thanks to the Virtual DOM. On the backend, Node.js paired with Express.js creates a lightweight and scalable environment that efficiently manages multiple requests at once. For data storage and management, we rely on MongoDB, a NoSQL database celebrated for its flexible schema design, making it perfect for academic datasets where attributes can differ. Moreover, Mongoose ORM streamlines database operations and schema modeling, which helps reduce the complexity of development. Libraries like Chart.js and Recharts are useful for visualization since they provide interactive graphs and charts that enhance analytical skills. A primary focus is security, which is addressed with JWT authentication, bcrypt.js for hashing passwords, and Helmet.js for protecting HTTP headers, all of which guarantee the confidentiality and integrity of data. The system may be set up on reasonably priced noisy platforms such as Heroku, AWS, or MongoDB Atlas that facilitate high availability and scalability. The program's web-based design ensures cross-platform compatibility by operating fluidly in contemporary browsers on Windows, Linux, and mobile devices.

Operational Feasibility:

This system fits right into the daily routines of schools, causing hardly any disruption to what's already in place. With its intuitive design, teachers and administrators can jump in without needing a tech degree. They can easily register students, update grades, track attendance, and check out analytics dashboards. Plus, students and parents can view progress reports from anywhere, which really boosts transparency and teamwork in education. By automating monotonous processes such as creating reports, comparing performances, and tracking attendance, the system saves a ton of time and cuts down on human errors. It offers real-time data updates and keeps tabs on historical performance, helping teachers spot long-term trends in student success. Being web-based means it's accessible anytime, anywhere with an internet connection, making it perfect for both in-person and remote learning. This way, academic monitoring continues smoothly, even during breaks or unexpected events.

Economical Feasibility:

This project really shines when it comes to cost-effectiveness, thanks to its use of free and open-source technologies that cut down on software licensing fees. The upfront costs are pretty minimal, mainly just development time and a few infrastructure expenses for cloud hosting. Plus, services like the free tiers of MongoDB Atlas and Heroku help keep those operational costs low. From an organizational standpoint, this system can lead to significant long-term savings by cutting down on manual paperwork, reducing administrative burdens, and lessening the need for physical record-keeping. By spotting at-risk students early on, the platform empowers schools to boost overall academic performance and lower dropout rates, which in turn enhances the institution's reputation and helps retain students. On top of that, the system's ability to generate reports more efficiently and accurately lightens the load for faculty, allowing them to concentrate more on teaching instead of getting bogged down by administrative tasks. The return on investment (ROI) becomes clear as the system aids in making better academic decisions, streamlining operations, and ultimately improving educational outcomes across the board.

Schedule Feasibility:

The Student Performance Analytics system is definitely schedule feasible. Its development can be wrapped up within a reasonable timeframe thanks to agile and modular practices. The project is broken down into smaller phases like requirements gathering, frontend work, backend integration, database design, testing, and deployment. Each of these can be completed in quick iterations, ensuring steady and measurable progress. The MERN stack, with its widespread support and reusable libraries/frameworks, really speeds up the development process and simplifies the coding. Tools like GitHub/GitLab for version control and project management platforms like Jira or Trello turn it into a breeze to track tasks and deadlines. Since the system has a modular architecture, features like student management, performance analytics, and chart visualization can be developed in parallel by different team members. This further reduces the total time needed. All in all, this project can realistically be finished within academic or institutional timelines without causing any delays.

2.4 TOOLS AND TECHNOLOGIES USED

Frontend (React.js):

The frontend of the Student Performance Analytics system is crafted using React.js, a robust JavaScript library that's perfect for building user interfaces. With its component-based architecture, React.js allows us to break the application down into reusable and modular pieces, like the student form, performance form, student table, performance table, and chart dashboards.

This approach not only accelerates development but also enhances the system's maintainability and scalability. For deployment, we can utilize affordable cloud platforms like Heroku, AWS, or MongoDB Atlas, which ensure high availability and scalability. Since the application is web-based, it guarantees cross-platform compatibility and runs smoothly on modern browsers across Windows, Linux, and mobile devices. This setup boosts the system's efficiency and responsiveness, especially when dealing with large datasets of student information and performance records. Features like state management and props in React are instrumental in managing data flow between components, allowing for seamless updates whenever students' marks, attendance, or subjects are added or changed. When it comes to data visualization, React works beautifully with libraries such as Chart.js and Recharts, which we use in this project to create interactive charts that showcase subject-wise averages, attendance trends, and performance distributions. These visual aids make it much easier for teachers and administrators to interpret raw academic data. In summary, React.js delivers a dynamic, interactive, and user-friendly interface that significantly improves the usability of the Student Performance Analytics system, making it a breeze for users to input data, view student performance, and analyze results effectively.

Backend(Node.js, Express.js):

The backend of the Student Performance Analytics system is built using Node.js and Express.js, creating a solid foundation for handling server-side tasks. Node.js is a lightweight and efficient runtime environment that allows developers to use JavaScript for backend work. Its non-blocking, asynchronous design means the application can juggle multiple requests at once without slowing down, which is crucial for real-time tasks like storing student records, fetching performance data, and updating attendance. Express.js serves as the backend web framework that complements Node.js. It provides powerful tools for managing HTTP requests, routing, and middleware, making API development smoother. In this project, Express is responsible for creating the routes that connect the database to the frontend, including those for adding new student information, updating performance records, retrieving attendance stats, and generating analytics reports. This organized routing ensures that all the system's components can communicate seamlessly. Thanks to the combination of Node.js and Express.js, the backend can effectively manage database connections, API handling, and business logic. For instance, when a teacher uses the React frontend to input a new performance record, Express routes the request to the backend, where Node.js processes it and saves it in MongoDB. Similarly, when performance charts are displayed, the backend efficiently pulls the necessary data from the database and sends it to the frontend for presentation.

Additionally, tasks such as data validation, authentication, and error management are handled by the middleware functions in Express. This not only boosts the system's security but also enhances its reliability. It ensures that sensitive student information is protected during transactions and that only valid data is stored in the system. In summary, the backend, built on Node.js and Express.js, offers a scalable, secure, and efficient framework for the Student Performance Analytics system. It guarantees smooth integration between the frontend interface and the database, all while ensuring consistent performance across various operations.

Database (MongoDB):

The database layer of the Student Performance Analytics system is built using MongoDB, a document-oriented NoSQL database that excels in efficiency, scalability, and flexibility for managing academic data. Unlike traditional relational databases, MongoDB's schema-less design allows data to be stored in documents that look like JSON. Because of this capability, it is ideal for managing dynamic and diverse data, including performance indicators, student profiles, and attendance records. The system's flexibility allows it to readily accommodate future needs, such as adding new fields or changing current data formats, without requiring significant structural adjustments. MongoDB is essential to maintaining the organization and accessibility of all student-related data. Student information such as name, USN, class, and subject, for instance, is kept in one collection, while performance information such as attendance, total and attainable grades, and monthly statistics are tracked in another. References help maintain the links between students and their performance histories, making it easy to integrate and retrieve data across different collections. To manage database operations, the system utilizes Mongoose ORM (Object Relational Mapping), which acts as a connector between Node.js/Express.js and MongoDB. With Mongoose, developers can ensure data consistency, enforce validation rules, and define schemas for collections. For instance, the Performance schema includes fields like topic, total marks, marks earned, and attendance, while the Student schema outlines fields such as name, USN, and class. Even though MongoDB is schema-less, this schema modeling ensures that the information kept in the database maintains a consistent structure.

MongoDB shines when it comes to handling large volumes of unstructured and semi-structured data, making it an excellent choice for educational analytics where data formats can vary. With its powerful indexing and querying capabilities, teachers can effortlessly access insights like average grades by topic, attendance trends, or track individual student progress—all while ensuring efficient and reliable performance, even as the dataset grows. In summary, MongoDB serves as the backbone of the Student Performance Analytics system, offering real-time analytics, scalability

for increasing records, and flexibility in data management. When used alongside Mongoose ORM, it ensures a well-organized, maintainable database layer that supports seamless backend integration and enhances the overall reliability of the system.

Authentication & Security (JWT,bcrypt.js):

The Student Performance Analytics system relies heavily on authentication and security to ensure that only those with permission may access important academic data. This solution creates a safe environment for user management by using JSON Web Tokens (JWT) for authentication and bcrypt.js for password encryption. These technologies work together to safeguard user credentials and stop unwanted access, creating a solid security foundation. When a user signs in, their information is compared to the database, which uses bcrypt.js to store the password in an encrypted manner. Even if an attacker manages to get into the database, It will be extremely challenging for them to retrieve the original password since this hashing approach ensures that the password is never kept in plain form. Additionally, Bcrypt employs salting, which strengthens security by guaranteeing that every hash is distinct and successfully thwarting brute-force and rainbow table assaults.

The password is securely stored using bcrypt.js, and when a user signs in, their credentials are validated against the database. Even if the database were to be compromised, attackers would struggle to retrieve the original password due to this hashing method, which ensures that the password is never kept in a readable format. The salting feature of Bcrypt further enhances security by making each hash distinct, thwarting brute-force and rainbow table attacks. Furthermore, JWT and bcrypt.js work together to improve security while still offering a smooth user experience. This token-based strategy improves the system's efficiency, scalability, and statelessness by doing away with the requirement for server-side session storage. By using security best practices like token expiration and password strength verification, it also reduces risks and protects important academic data. To wrap it up, JWT and bcrypt.js are the backbone of the Student Performance Analytics system's authentication and security. They lay down a solid groundwork for secure logins, encrypted storage of credentials, and controlled access to various system features. This setup not only protects data integrity but also shields against common online threats, boosting trust in the system's reliability.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 PURPOSE

The Student Performance Analytics System is all about creating a reliable and efficient digital space for managing student information, academic performance, and attendance records. Traditional methods, like paper records or basic spreadsheets, can be a real hassle they take up a lot of time, are prone to human error, and don't really help in generating useful insights. This makes it tough for educators and administrators to effectively track student progress or make timely decisions. By bringing automation into the mix for data storage, retrieval, and visualization, this system tackles those challenges head-on, ensuring that record management is faster, more accurate, and well-organized. Instead of sifting through piles of paperwork or scattered files, teachers and administrators can pull up all student-related data in just seconds, which saves them a ton of time and effort. This shift to digital not only streamlines the process but also reduces the chances of data duplication, misplacement, or inconsistencies. Another key feature of this system's features is its ability to provide actionable insights through academic analytics. It goes beyond just storing records; it analyzes performance data and presents it in easy-to-understand charts, graphs, and dashboards. These visual tools help educators quickly spot strengths and weaknesses in different subjects, track performance trends over time, and notice attendance patterns. For instance, if a student is consistently struggling in a specific subject or has irregular attendance during a specific month, the system flags these issues for prompt intervention. In this manner, kids who are in danger may be recognized early on and receive the additional help they require before their academic performance suffers. The Student Performance Analytics System is fundamentally much more than a record-keeping system. It's an effective instrument that encourages openness, responsibility, and proactive decision-making, which increases productivity and raises educational standards. By ensuring that records are handled appropriately and allowing for insightful analysis, this method helps to create an environment where every student gets the attention and support they need to thrive in their academic pursuits.

3.2 SCOPE

The Student Performance Analytics System is all about making the management of student records, exam results, and attendance a whole lot easier. By doing away with the outdated manual record-keeping techniques, this system improves accuracy and dependability while also reducing effort. It offers teachers and administrators interactive dashboards and visual reports, making it super simple to keep an eye on academic progress and spot where students shine or need a little extra help. The system also prioritizes security with role-based authentication, so only authorized folks like administrators, teachers, and students can access or update information. Plus, it's designed to grow with the institution, ready to handle new classes, subjects, or advanced analytics features as needed. In short, this system doesn't just manage student performance data effectively; it also paves the way for long-term growth and smarter decision-making for the institution.

3.3 FUNCTIONAL REQUIREMENTS

The Student Performance Analytics System is crafted to fulfill a number of essential tasks that simplify the management of academic data. It must make it simple for instructors and administrators to add, edit, and arrange student information in an understandable and systematic manner, including names, USN, classrooms, and subjects. Together with monthly attendance records, the system should also enable the recording of exam results, including subject-specific scores, overall marks, and percentages. To facilitate effective monitoring, it ought to automatically calculate important performance metrics such as averages, attendance rates, and score distributions. Another crucial aspect is the ability to display this information in a way that's easy to understand and visually appealing. Dashboards, charts, and tables should be at hand to help teachers and administrators quickly spot students who are thriving, as well as those who might need a little extra support. To ensure that only those who are authorized users may access or modify any data, the system must also provide secure authentication and login. Additionally, it should facilitate role-based access, which would enable administrators to monitor the system as a whole, teachers to manage performance records, and students to view their outcomes.

3.4 NON-FUNCTIONAL REQUIREMENTS

The Student Performance Analytics System is defined not just by its functions but also by its effectiveness. First off, performance is key: the system needs to manage multiple user requests at the same time without any lag, so that students, teachers, and administrators can access information instantly. Security is also important because the platform manages valuable academic data. Features that assist prevent unwanted usage and guarantee data privacy include encrypted passwords, secure authentication via JWT, and regulated role-based access. Another crucial component is usability; the system should have an easy-to-use interface. It should be simple for teachers to record performance data, and it should be easy for students to browse and view their findings. Incorporating charts, tables, and dashboards makes the system engaging and simplifies the understanding of performance trends. Scalability is also an important non-functional requirement, ensuring the system can expand as data volume and user numbers grow. Whether it's more students enrolling, new subjects being added, or additional analytical features being introduced, the system should adapt smoothly without needing major overhauls. Moreover, reliability is essential, meaning that data must always be consistent, accurate, and accessible without crashes or data loss. Finally, maintainability and flexibility are crucial, allowing future developers or administrators to update the code, add new features, or fix issues efficiently. These non-functional requirements ensure that the Student Performance Analytics System remains effective, secure, and sustainable over time.

3.5 HARDWARE AND SOFTWARE REQUIREMENTS

3.5.1 SOFTWARE REQUIREMENTS

- Operating System: Windows 10 or above / Linux / macOS
- Frontend: React.js, HTML5, CSS3, JavaScript
- Backend: Node.js with Express.js
- Database: MongoDB with Mongoose ORM
- Authentication & Security: JWT (JSON Web Token), bcrypt.js
- Development Tools: Visual Studio Code, Postman
- Runtime Environment: Node.js LTS version
- Browser Support: Google Chrome, Microsoft Edge, Mozilla Firefox

3.5.2 HARDWARE REQUIREMENT

- Processor: Intel i3 or above / AMD equivalent
- RAM: Minimum 4 GB (8 GB recommended)
- Hard Disk: At least 250 GB (SSD preferred for faster performance)
- Display: 14-inch or above, resolution 1366×768 or higher
- Network: Stable internet connection (minimum 2 Mbps)
- Additional: Keyboard, mouse, and basic peripherals

3.5.3 TOOLS AND PLATFORMS USED

- Frontend : HTML, CSS, JavaScript, React.js
- Backend: Node.js ,Express.js
- Database : Mongo DB

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

System design is all about mapping out the architecture, modules, components, interfaces, and data flow of an application to meet specific needs. For the Student Performance Analytics Project, the system is thoughtfully crafted to ensure it's scalable, secure, and user-friendly. This platform empowers teachers to record, manage, and assess student performance and attendance, while using captivating visual insights, students may actively explore their academic path. The architecture of the system is founded on the MERN Stack model, which combines MongoDB, Express.js, React.js, and Node.js. This layered approach guarantees seamless communication between the frontend, backend, and database. The frontend, developed with React.js, offers a dynamic and responsive user interface that supports real-time updates. Meanwhile, the backend, driven by Node.js and Express.js, handles API requests, business logic, and secure interactions between the client and the database. With a customizable structure that makes scalability simple, MongoDB serves as the database, holding attendance, test scores, and student records information, and other academic data. The design's emphasis on data visualization stands out. With the use of the system's charts and graphs, instructors and administrators may quickly see trends in student performance, attendance patterns, and subject-specific strengths and weaknesses. The modular design of the architecture ensures that each component operates independently while remaining well-integrated, simplifying maintenance and paving the way for future enhancements.

4.2 SCOPE

The Student Performance Analytics system provides an all-inclusive platform for handling student academic data, such as attendance, grades, subjects, and personal information. It empowers teachers to effectively track student progress, identify strengths and weaknesses, and generate visual reports that support informed decision-making. Moreover, Students have access to their performance histories, promoting transparency and personal growth. By leveraging the MERN stack, the system ensures scalability, real-time updates, and secure data management. Because of its adaptability, educational institutions of all sizes whether they are colleges, universities, or schools looking for data-driven strategies to improve student outcomes can use it to great advantage.

4.3 SOFTWARE PRODUCT ARCHITECTURE

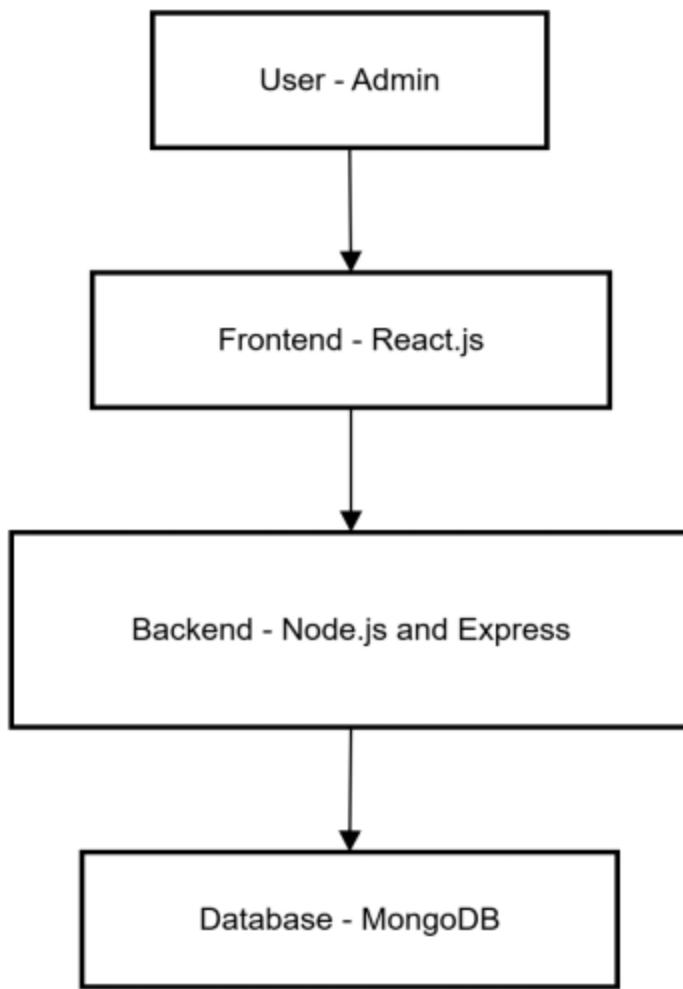


Fig 4.1 Architectural Diagram

View Layer (User Layer)

For end users, the magic happens in the View Layer, sometimes referred to as the User Layer. It is the component of the system that permits direct communication between them and the Student Performance Analytics System. React.js and is intended to be aesthetically pleasing, responsive, and incredibly user-friendly. Students are able to see their individual performance information, including grades, attendance, and progress in different subjects, by logging in here. To make this information easier to understand, it is presented in both tables and visually appealing graphs. This layer is also used by administrators and teachers to add performance information, update records, register new students, and analyze data using user-friendly dashboards. The view layer not only makes navigation a breeze but also enhances usability by turning raw data into an interactive and easy-to-read format, making it the most visible and user-focused part of the system.

Application Layer

The foundation of the Student Performance Analytics System is basically the Application Layer, also known as the Service Layer. It handles all of the business logic and makes sure that the database and user interface communicate effectively. Constructed with Node.js and Express.js, this layer provides services and APIs that manage user requests such as creating analytical reports, adding performance records, registering students, and logging in. It ensures secure authentication and authorization for a range of roles, including administrators, teachers, and students, verifies inputs, and applies rules for managing academic data. For instance, when a teacher submits marks or attendance details, the application layer processes that request, verifies the information, and securely stores it in the database. Likewise, when a student asks for their progress report, this layer retrieves the necessary data, performs calculations like percentages or averages, and sends it back in a well-organized format for display in charts or tables. By managing workflows, integrating chart libraries, and ensuring data consistency, the application layer truly acts as the decision-making heart of the system.

Access Layer (Database Layer)

The foundation of the Student Performance Analytics System is the Access Layer, sometimes referred to as the Database Layer, which handles all data management duties. This layer, which is based on MongoDB, provides a flexible schema that is not constrained by inflexible structures and can readily adjust to various academic records. This makes it ideal for handling a wide range of data, such as attendance records, grades, subjects, and student information. Maintaining data integrity is among its main duties, making certain that the data entered by administrators or teachers is correctly stored and consistent across the system. This layer also serves as a crucial link between the application logic and the persistent storage. When the application layer sends a request—like retrieving a student's attendance for a specific month or calculating overall class performance—the database layer processes that query and returns the necessary data. Additionally, MongoDB's query optimization and indexing capabilities enable it to swiftly retrieve substantial amounts of data, guaranteeing seamless operation even when handling hundreds of students over several academic sessions. Another essential element of this layer is security. It prevents unwanted changes or data leaks by limiting access to sensitive student information according to roles (student, teacher, admin) through integration with authentication mechanisms like JWT and bcrypt.js. Furthermore, even in the hardware case or network failures, data availability and dependability are guaranteed by MongoDB's backup and replication features.

4.4 DATA-FLOW DIAGRAM

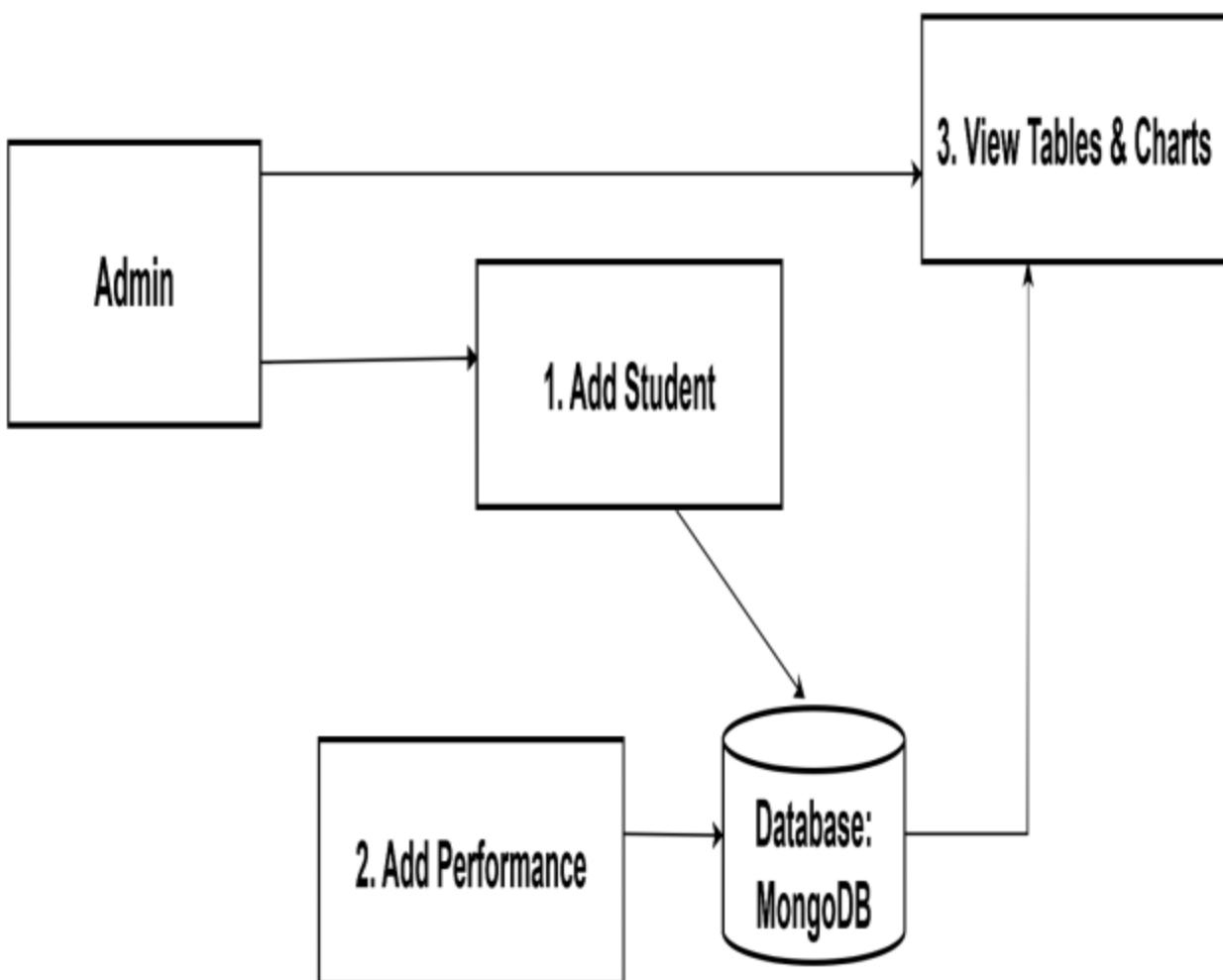


Fig 4.2 Data flow diagram

The Student Performance Analytics system is crafted to maintain a seamless flow of information among students, teachers, administrators, and the database. This guarantees that each exchange results in precise and perceptive scholarly data. Students kick things off by logging into the system through a secure interface that checks their credentials before giving them access to their personalized dashboard. Once they're in, they can explore detailed performance records, including grades in various subjects, monthly attendance history, and visual progress reports. These dashboards not only showcase current results but also highlight trends over time, helping students grasp their strengths, weaknesses, and overall academic journey. Teachers and administrators have a more active role in this system. They handle tasks like registering new students, adding subject details, entering exam scores, recording monthly attendance, and updating academic data as needed. After the information is entered, it gets stored in a centralized database, making it easy to pull up individual performance reports or collective analytics later on.

By comparing student performance across subjects, monitoring attendance trends, and identifying pupils who may need more support, the system also gives teachers the capacity to monitor entire classes. Higher-level dashboards can be utilized by administrators to view statistics for the entire class or subject, evaluate the efficacy of the instruction, and make sure the school is meeting its performance targets. At the core of the entire process is the database, which serves as the central hub for all student-related information. It keeps track of personal details like names, USN, and class information, as well as comprehensive performance records that include subject-specific marks, attendance percentages, and monthly reports. Whenever a student or teacher interacts with the system, the application layer connects with the database to either retrieve the necessary data or update it according to the user's input. This interaction is smooth, making certain that the information displayed on dashboards is always accurate, updated in real time, and accessible only to those with the right permissions. The system's information flow is enhanced by the integration of dashboards and visualization tools. The system converts academic records into easily comprehensible interactive charts, graphs, and comparison tables instead of merely displaying data as static reports or plain numbers. Teachers can examine class performance distributions and attendance trends through comprehensive reports, while Students can quickly see their average scores across various subjects displayed as pie charts or bar graphs. In addition to simplifying decision-making, these visual insights encourage preventative actions like more focused mentoring, remedial education, or improved attendance monitoring. The Student Performance Analytics system promotes a more dynamic, open, and results-oriented learning environment by transforming unstructured data into insightful analytics.

The application layer is the backbone of how information flows between users and the database, housing the system's business logic. It takes care of incoming requests, checks the data for accuracy, applies the necessary rules, and interacts utilizing the database to carry out either write or read tasks. After the database sends back a response, the application layer organizes the data and hands it off to the presentation layer, where it's transformed into an easy-to-understand and interactive format using charts, tables, and graphs. This ongoing cycle of input, processing, storage, and output ensures that students, teachers, and administrators always have access to valuable academic insights, making the system an essential tool for enhancing educational outcomes.

CHAPTER 5

DETAILED DESIGN

5.1 USE CASE DIAGRAM

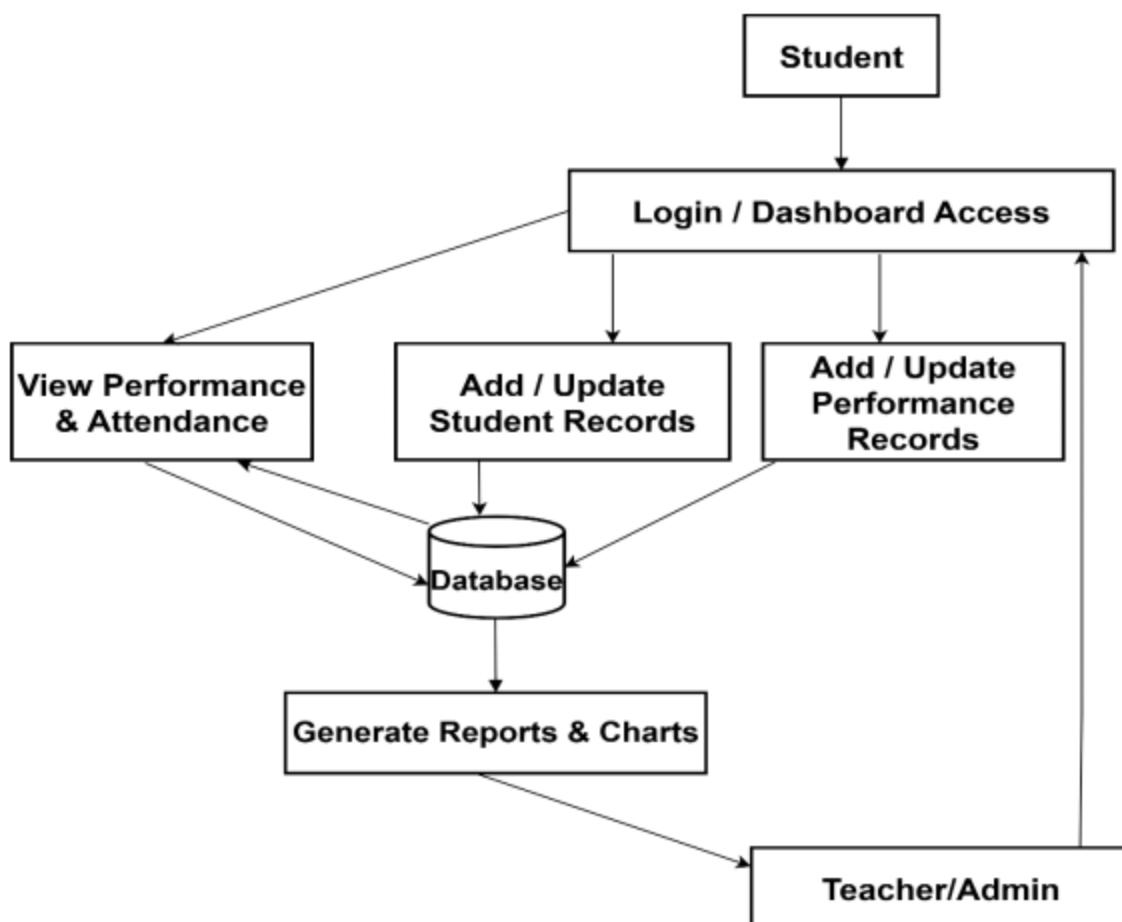
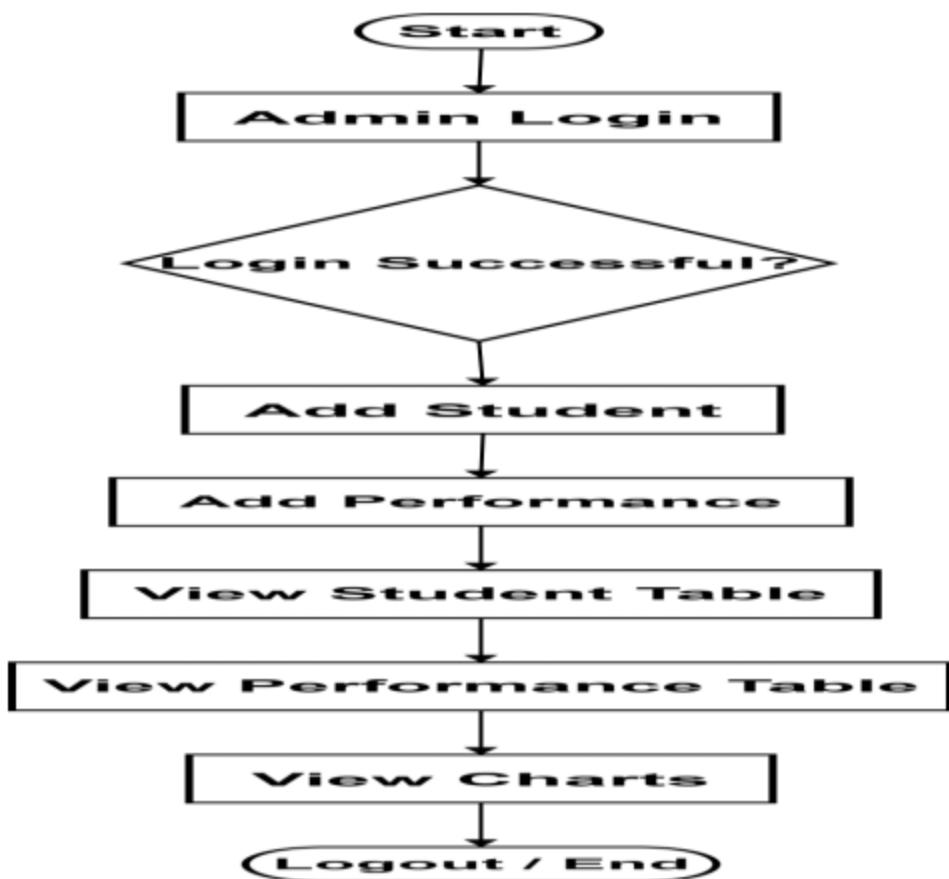


Fig 5.1 use case diagram

The Use Case Diagram clearly and systematically illustrates how various users engage with the Student Performance Analytics System. The Login use case ensuring that only authorized users, such as administrators, instructors, and students, can access the system. Administrators or teachers can enter crucial information like the student's name, USN, class, and subject into the system by going straight to the Add Student use case after successfully logging in. After adding students, the Manage Performance use case comes into play, allowing teachers to record exam scores, attendance, and monthly progress updates. This way, each student's academic journey is kept up-to-date and securely stored in the database.

5.2 ACTIVITY DIAGRAM**Fig 5.2 Acitivity Diagram for Admin**

The administrator can observe how a student uses the Student Performance Analytics System thanks to the Activity Diagram. It all begins when the student safely logs in with their registered login credentials, protecting their academic and personal data. A dashboard that compiles all of their academic information, including grades, attendance, subject performance, and overall progress, greets them when they log in. From this dashboard, students can dive into their performance records, keep an eye on monthly or subject-specific trends, and figure out where they shine and where they might need a little extra help. With handy visual tools like graphs and charts, it's much easier for students to make sense of their data and track their improvements or setbacks over time.

The system is user-friendly for both students and administrators, as the diagram also shows how simple it is to switch between modules, such as viewing attendance, grades, and visual reports. The student logs out at the conclusion of the process, securing their session. This graphic clearly illustrates how the system can give students real-time access, organized information, and captivating visuals, enabling them to take control of their academic journey. It demonstrates how the system balances security, accessibility, and administrator engagement by providing a clear flow from login to logout.

5.3 ER Diagram

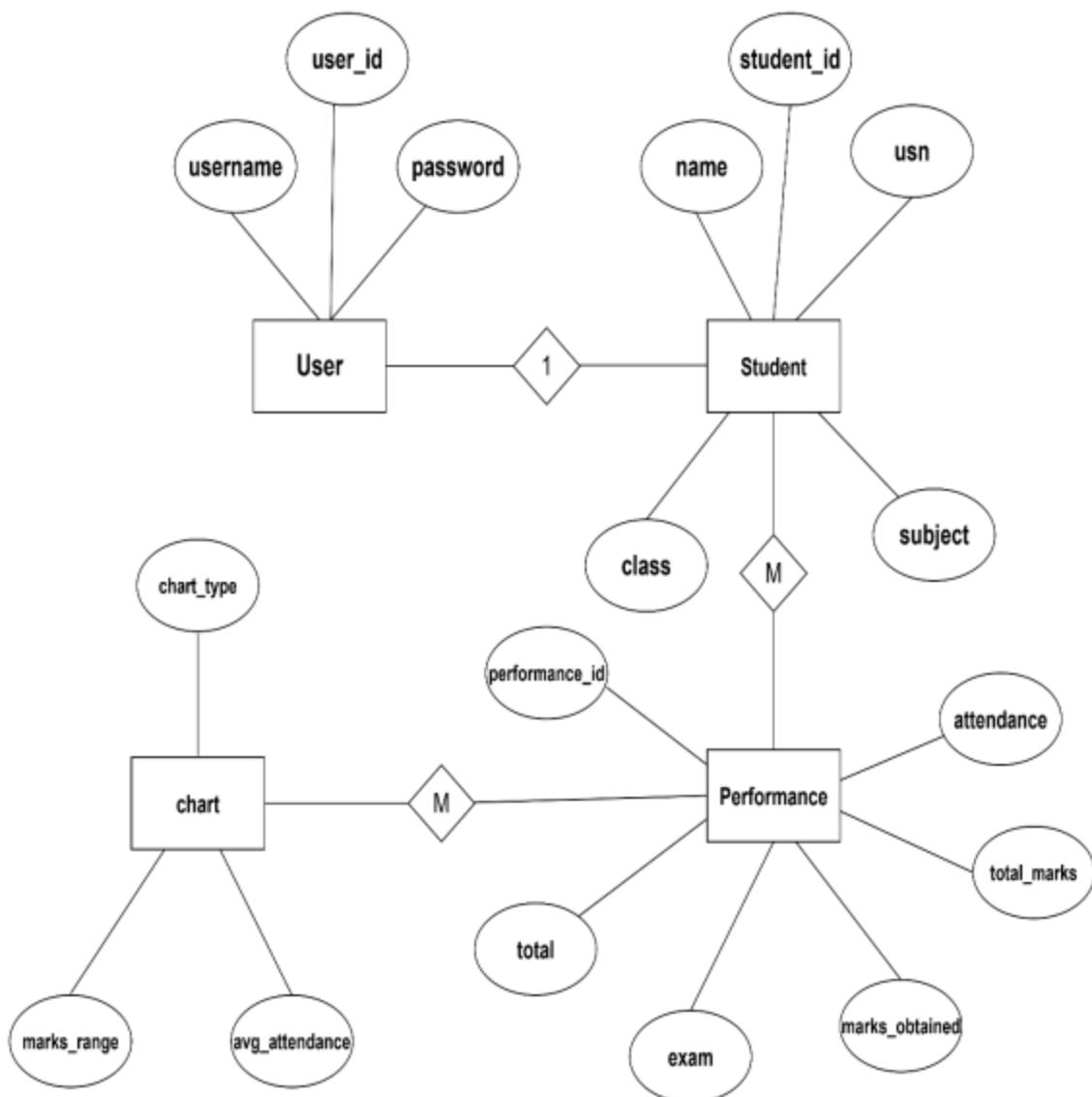


Figure 5.3 ER diagram

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is where the Student Performance Analytics system, crafted with the MERN stack, comes to life as a fully functional application. The frontend, built with React.js, creates an engaging and responsive interface that makes it easy for students, teachers, and administrators to navigate the system. Meanwhile, the backend utilizes Node.js and Express.js, ensuring that business logic is handled smoothly, API communications are secure, and user requests are managed efficiently. MongoDB acts as the central database, providing the flexibility and scalability needed to reliably store academic records, attendance data, and performance metrics. A key priority during implementation is safeguarding data and ensuring secure access. The system employs JWT-based authentication for safe login sessions, bcrypt.js for encrypting passwords, and role-based access control to distinguish user privileges among students, teachers, and administrators. These strategies help protect sensitive academic information from unauthorized access. Additionally, the system is fine-tuned for performance, allowing users to enjoy quick and seamless navigation without any technical hiccups.

Implementation is where the Student Performance Analytics system, crafted with the MERN stack, comes to life as a fully functional application. The frontend, built with React.js, creates an engaging and responsive interface that makes it easy for students, teachers, and administrators to navigate the system. Meanwhile, the backend utilizes Node.js and Express.js, ensuring that business logic is handled smoothly, API communications are secure, and user requests are managed efficiently. MongoDB acts as the central database, providing the flexibility and scalability needed to reliably store academic records, attendance data, and performance metrics. A key priority during implementation is safeguarding data and ensuring secure access. The system employs JWT-based authentication for safe login sessions, bcrypt.js for encrypting passwords, and role-based access control to distinguish user privileges among students, teachers, and administrators. These strategies help protect sensitive academic information from unauthorized access. Additionally, the system is fine-tuned for performance, allowing users to enjoy quick and seamless navigation without any technical hiccups.

6.2 IMPLEMENTATION DETAILS

The Student Performance Analytics system is built using the MERN stack, which combines MongoDB, Express.js, React.js, and Node.js to create a seamless, scalable, and responsive web application.

The admin interface is the frontend, which was primarily created using React.js. Moreover, HTML, CSS, and JavaScript. Teachers, administrators, and every student can communicate with the system efficiently with this configuration. To view their attendance summaries, individual performance reports, and subject-specific scores presented in interactive charts and tables, students must safely log in. Meanwhile, teachers and administrators have the ability to register students, manage academic records, and update performance metrics in real-time. Thanks to React.js's component-based architecture, developers can craft reusable and modular UI elements like login forms, student registration forms, performance input tables, and dashboard components. This method expedites development and streamlines maintenance while guaranteeing consistency across pages. CSS enhances the Admin interface's visual appeal and responsiveness on various devices, while JavaScript introduces interactivity, including form validations, search features, and real-time updates without needing to reload the page.

All server-side processing and database communication are handled by the system's backend, which was constructed using Node.js and Express.js. Node.js offers a lightweight and efficient environment for running JavaScript, while Express.js makes it easier to develop APIs and manage routing. This backend handles requests from Admins, checks data for accuracy, manages authentication, and ensures secure access based on Admin roles. We use JWT (JSON Web Tokens) to authenticate administrators and secure sessions, and bcrypt.js is used to encrypt passwords and prevent unwanted users. Role-based permissions are also managed by the backend, and teachers and administrators have more extensive access to add, edit, and remove records in addition to produce analytics for the entire system, while pupils can only see their own performance data. Every request to the backend is logged with tools like Morgan, which helps with debugging and monitoring the system. For data storage, we use MongoDB, a document-oriented database that's flexible with schemas and can handle complex, semi-structured data. Student profiles, performance records, attendance, and subject details are organized into collections, making queries fast and updates efficient. Mongoose, an Object Data Modeling (ODM) library, helps define schemas and enforce data validation, which simplifies database tasks and ensures everything stays consistent. The system supports CRUD operations (Create, Read, Update, Delete) for student and performance data, allowing teachers to manage records smoothly. Plus, we've integrated charting libraries like Chart.js. The implementation process involves bringing together the frontend, backend, and database layers. When a teacher submits new performance data, the request moves from the React.js frontend to the Node.js/Express.js backend. Here, the data gets processed, the MongoDB database is updated, and the latest information is sent back to the frontend. This smooth flow guarantees that dashboards, tables, and charts always show real-time data.

The implementation process involves bringing together the frontend, backend, and database layers.

When a teacher submits new performance data, the request moves from the React.js frontend to the Node.js/Express.js backend. Here, the data gets processed, the MongoDB database is updated, and the latest information is sent back to the frontend. This smooth flow guarantees that dashboards, tables, and charts always show real-time data. To prevent unwanted access, data loss, or tampering, we have integrated security measures at every stage. We use CORS middleware to securely handle cross-origin requests, and sensitive data, such as passwords and private student information, is encrypted. To further enhance security, we use Helmet middleware to secure HTTP headers and protect against common web vulnerabilities. Testing and validation are crucial during the implementation phase. While Integration testing confirms that information moves smoothly between the frontend, backend, and database, unit testing examines each module separately to make sure everything functions as it should. Real-world situations, like several students accessing the dashboard simultaneously, teachers updating records, and administrators producing reports, are simulated by system testing. To ensure the integrity of the data is maintained and that only administrators with permission can access sensitive features, we also carry out security testing. Another crucial component of implementation is performance optimization we reduce the number of API calls, index database queries for faster retrieval, and optimize React components for effective rendering.

Our system is built to expand and change as needed. Without requiring a total redesign, We can add new features fast like AI-powered predictive analytics, automated alerts for students who might be having trouble, or even integration with other academic tools. The system's modular design allows for the advancement of various components, such as the database, frontend, and backend, independently, which simplifies maintenance. All in all, the Student Performance Analytics system is a fully functional, secure, and interactive platform that simplifies student record management, offers real-time insights into academic performance, and helps educators make well-informed decisions. By utilizing the MERN stack, the system not only boasts technical strength but also operational efficiency, making it a great fit for educational institutions of all sizes.

6.3 ADMIN INTERFACE

6.3.1 Login Page

The login page serves as the initial touchpoint for all Admins, including students, teachers, and administrators. It's crafted to be straightforward, clean, and user-friendly, enabling Admins to securely access the Student Performance Analytics system. To log in, Admins need to input their registered email and password. The login form checks the data in real-time, quickly flagging any incorrect formats or missing information.

Web Tokens)- managed authentication and bcrypt.js-encrypted and verified passwords on the backend for added security. Students can view their performance, attendance history, and visual charts that display their grades by subject and monthly attendance on their personalized dashboard after successfully logging in. Depending on their roles, teachers and administrators are directed to an admin dashboard that offers extra features like adding or updating student records, managing performance entries, generating reports, and accessing analytics. If there are any invalid login attempts, error messages are displayed to provide clarity and guidance for the Admin. The login page is responsive and functions smoothly across numerous gadgets, such as PCs, tablets, and smartphones. There's also a "Forgot Password" feature that allows Admins to reset their password through a secure link sent to their registered email. All in all, the login page blends security, usability, and responsiveness to create a seamless and safe entry point into the system.

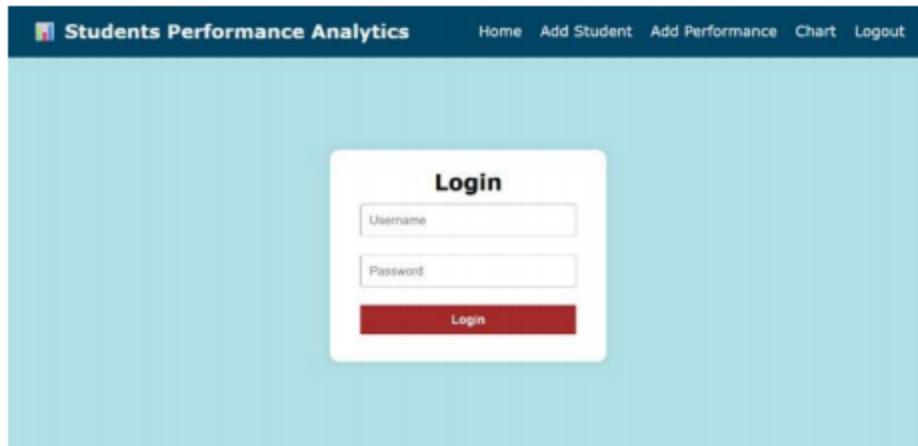


Fig 6.1 Login Page

6.3.2 Student Page

The Student Page is an essential part of the Student Performance Analytics system, mainly used by administrators and teachers who have the authority to manage student records. Its main objective is to enable admins to add new students to the system, ensuring that all essential academic information is accurately captured. When admins access this page, they encounter a user-friendly key student details like Name, USN (University Serial Number), Class, and Subject. To minimize mistakes, input validation is in place to catch issues like empty fields, incorrect USN formats, or duplicate entries. After filling out the form and hitting submit, the data is sent through the backend API to the MongoDB database, where it's securely stored.

Each student record gets a unique identifier, making it easy to retrieve and manage. Besides adding new students, the page also offers features for viewing, updating, or deleting existing records. Admins can quickly search for students using keywords such as name, USN, or class, streamlining the record management process. After each action, feedback messages pop up to confirm successful additions, updates, or deletions, enhancing transparency and usability. The interface is also connected to the dashboard and analytics modules, ensuring that newly added students are automatically included in performance tracking and visual reports. This keeps the system's database synchronized, providing accurate and up-to-date information for both teachers and administrators.

The screenshot shows a web application titled "Students Performance Analytics". The main title is at the top left, and a navigation bar with links for Home, Add Student, Add Performance, Chart, and Logout is at the top right. Below the title, the page has a header "Add Student". It contains four input fields: "Name", "USN", "Class", and "Subject", each with a corresponding text input box. At the bottom of the form are two buttons: a red "Submit" button and a red "View Students" button.

Fig 6.2 Student Page

6.3.3 View Student Table Page

The View Student Table Page is a handy tool for administrators and teachers, giving them access to a detailed list of all student records stored in the system. It showcases information like student names, USNs, classes, and subjects in a well-organized table format, making it super easy to navigate and manage. Admins can quickly find specific students by searching with keywords such as names, USNs, or classes, and they can sort the table by different columns to pinpoint the information they need. The page also allows for editing existing student details or deleting records when necessary. Any changes made are instantly updated in the database, keeping everything accurate and current. This page really boosts the efficiency of student management by offering a clear, organized, and interactive interface for tracking all student records, while also facilitating quick decision-making and data verification.

The screenshot shows a web-based application titled "Students Performance Analytics". The main header includes the title and navigation links for Home, Add Student, Add Performance, Chart, and Logout. Below the header is a sub-header titled "View Students" with a small icon. The main content is a table with the following data:

Name	USN	Class	Subject	Actions
Anu	4SN23MC011	10	Maths	<button>Edit</button> <button>Delete</button>
Suma	4SN23MC010	10A	Maths	<button>Edit</button> <button>Delete</button>
sakshi	4SN23MC012	10A	Maths	<button>Edit</button> <button>Delete</button>

Fig 6.3 View Student Table Page

CHAPTER 7

TESTING

7.1 INTRODUCTION

In order to develop the Student Performance Analytics system, testing is essential. It guarantees that each feature and module operates as planned and satisfies the designated specifications. Testing's primary objective is to recognize and deal with any problems or defects prior to the system's launch for educators, administrators, and students. A thoroughly tested system promises reliability, efficiency, and a seamless experience for administrators. In this project, testing encompasses both individual components and the entire system. Each module, like login, student record management, performance tracking, and chart visualization, goes through unit testing to make sure every function works as it should. Then, integration testing checks that these modules interact smoothly with one another. For instance, it verifies that data entered in the student form is accurately saved in the database and displayed in the performance charts. Finally, system testing assesses the whole application in real-world scenarios, ensuring that tasks like adding students, updating performance records, viewing reports, and generating charts operate without a hitch. Furthermore, the system is put through security testing to ensure that private student data is kept safe. We test authentication methods using JWT and password encryption with bcrypt.js to confirm that unauthorized access is blocked. By conducting thorough testing at every level, the Student Performance Analytics system becomes reliable, secure, and ready for deployment, allowing educators to effectively monitor academic performance while ensuring students can safely access their records.

7.1.1 Unit Testing

Unit testing is all about checking the individual parts of the Student Performance Analytics system to make sure they're doing their job correctly independently. For instance, we test the login and registration forms to ensure they handle inputs properly, while the student and performance modules are examined to confirm that data can be added, updated, and retrieved without a hitch. Additionally, we take a close look at frontend components like charts and dashboards to verify they display the right information. Before we put everything together, Unit testing ensures that every component functions dependably by identifying problems early.

7.1.2 Integration testing

The goal of integration testing in the Student Performance Analytics system is to ensure that various modules function properly together after they are integrated. Integration testing ensures that data flows and communication between components, such as forms, APIs, and database models, occurs without hiccups, while unit Testing confirms that every part works as intended. For example, when a teacher adds a student's performance record, the details should move correctly from the React-based frontend through the Node.js and Express backend to the MongoDB database without any hiccups or data loss. Likewise, if a student's marks or attendance are updated, those changes need to show up instantly not just in the database but also in the analytics dashboards and charts. Integration testing is also essential for verifying that the student module integrates flawlessly with the performance and attendance modules, making certain that only individuals with permission can access records, and validating authentication and session management. By conducting this testing phase, we can uncover any hidden problems in the communication between modules and address them before they become bigger problems during system-level execution.

7.1.3 System testing

System testing is all about putting the Student Performance Analytics application through its paces to make sure it ticks all the right boxes. We simulate real-life situations, like a teacher entering performance data, updating attendance, or pulling up reports, while students log in to check out their own performance and charts. This phase looks at the entire workflow, including how Admins interact with the system, how data is stored, and how it's visualized. The goal is to ensure everything runs smoothly under various conditions, proving that the system is stable, reliable, and ready to go live.

7.1.4 Acceptance testing

Acceptance testing is the final checkpoint for the Student Performance Analytics system, ensuring it aligns with what users expect. This stage involves teachers, students, and administrators actively engaging with the system. Teachers verify that records, reports, and charts are updating correctly, while students check their access to grades, attendance, and graphical reports. Administrators make sure they can manage records, keep an eye on class performance, and generate comprehensive reports. User feedback is crucial here, helping to spot any usability hiccups or missing features. Any issues are sorted out before the system goes live. This phase ensures that the system is user-friendly, dependable, and genuinely helpful. Once it passes this stage, it's all set for real-world use in education.

7.2 TEST CASES

Serial No	Test Case	Expected Outcome	Observed Outcome	Result
1	Admin Login	When a Admin enters the correct USN and password, they should be successfully logged in and redirected to the dashboard.	The Admin logged in successfully and was taken to the dashboard.	Passed
2	Add Student	When an admin enters a student's name, USN, class, and subject, the student record should be saved and displayed in the student table.	The new student record was saved and appeared in the student table.	Passed
3	Add Performance	When an admin/teacher enters subject, exam, total marks, marks obtained, and attendance, the performance record should be saved and shown in the performance table.	The performance record was saved and displayed correctly in the table.	Passed
4	View Student Table	When a Admin/admin navigates to the student table, all student records should be displayed accurately.	All student records were displayed correctly.	Passed
5	View Performance Table	When a Admin/admin navigates to the performance table, all performance records should be displayed accurately.	All performance records were displayed correctly.	Passed
6	Edit/Delete Student	When an admin edits or deletes a student, the changes should be reflected in the database and table.	The edits and deletions were reflected correctly.	Passed
7	Edit/Delete Performance	When an admin/teacher edits or deletes a performance record, the changes should appear in the database and table.	The edits and deletions were applied successfully.	Passed
8	Charts	When a Admin/admin views charts, the system should correctly display subject-wise averages, student-wise attendance, and score distributions.	All charts displayed accurate data for averages, attendance, and score distributions.	Passed

CHAPTER 8

CONCLUSION

The MERN stack—MongoDB, Express, React, and Node.js has been effectively employed in the creation and implementation of the Student Performance Analytics system. In a centralized, safe setting, the platform enables teachers and administrators to effectively manage student records, attendance, and academic performance. Pupils can easily view their own attendance logs, performance data, and visual charts, which aids in understanding their development and pinpointing areas in need of improvement.

The system achieved its main objectives, including easy management of student information, seamless performance tracking, and meaningful data visualization through charts and reports. Features like adding, editing, and deleting student and performance records work as intended, and the interactive dashboard provides quick insights for decision-making.

Sensitive information is safeguarded by security measures like password encryption and JWT-based authentication, and role-based access control stops illegal use of administrative tools.

In conclusion, the Student Performance Analytics system is a reliable and Admin-friendly web application that improves academic monitoring, reduces manual work, and enables data-driven decision-making. With future enhancements such as predictive analytics, mobile accessibility, and advanced reporting, the system has the potential to further empower teachers and students in achieving better educational outcomes.

CHAPTER 9

FUTURE ENHANCEMENT

Even though the Student Performance Analytics system is fully functional and meets its objectives, several improvements can be made to make it even more powerful and Admin-friendly. Using machine learning to integrate predictive analytics is one potential enhancement that might assist in identifying pupils who are at risk of performing poorly early on and enable teachers to offer timely support.

Another improvement is the addition of a mobile application, which would give students, teachers, and administrators easy access to performance data, attendance records, and analytics on the go. Admins could work in places with poor internet connectivity by introducing offline access, with data syncing automatically when the connection is restored.

Advanced search, filtering, and sorting features could be implemented to make it faster and easier to find student records or specific performance data. For big organizations, connectivity to cloud storage services like AWS or Google Cloud could enhance data security, backup, and scalability. Other enhancements could include customizable dashboards, automated report generation, and email/SMS notifications to keep students and parents updated about performance and attendance. The system could also support role-based analytics, allowing department heads or school administrators to access high-level insights without exposing sensitive individual data.

These improvements would make the Student Performance Analytics system smarter, easier to use, and more effective, which would further improve student outcomes and academic decision-making.

APPENDIX A**BIBLIOGRAPHY****PAPERS REFERRED**

- [1] M. Kumar, A. Singh, and P. Sharma, “Development of Web Applications using MERN Stack: A Case Study,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 5, pp. 123–131, 2021.
- [2] S. Patel and R. Mehta, “Admin-Centric Design for Social Networking Web Applications,” *International Conference on Software Engineering and Technology (ICSET)*, pp. 55–62, 2020.
- [3] A. Gupta and V. Reddy, “Scalable Cloud-Based Solutions for Web Applications,” *International Journal of Cloud Computing and Services Science*, vol. 10, no. 4, pp. 211–220, 2019.
- [4] J. Brown, L. White, and K. Johnson, “Data Security and Privacy in Full-Stack Web Applications,” *IEEE International Conference on Information Technology (ICIT)*, pp. 98–104, 2021.
- [5] R. Kaur and T. Malhotra, “Enhancing Admin Engagement in Online Student Management Systems: A Study on Interactive Features,” *International Journal of Emerging Technologies in Engineering Research*, vol. 8, no. 7, pp. 45–52, 2020.

BOOKS REFERRED

- [1] S. Holovaty and J. Kaplan-Moss, *The Definitive Guide to Django: Web Development Done Right*, Apress, 2009.
- [2] M. Cantelon, T. Harter, N. Rajlich, and M. Cantelon, *Node.js in Action*, Manning Publications, 2017.
- [3] A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, O'Reilly Media, 2020.

WEBSITES REFERRED

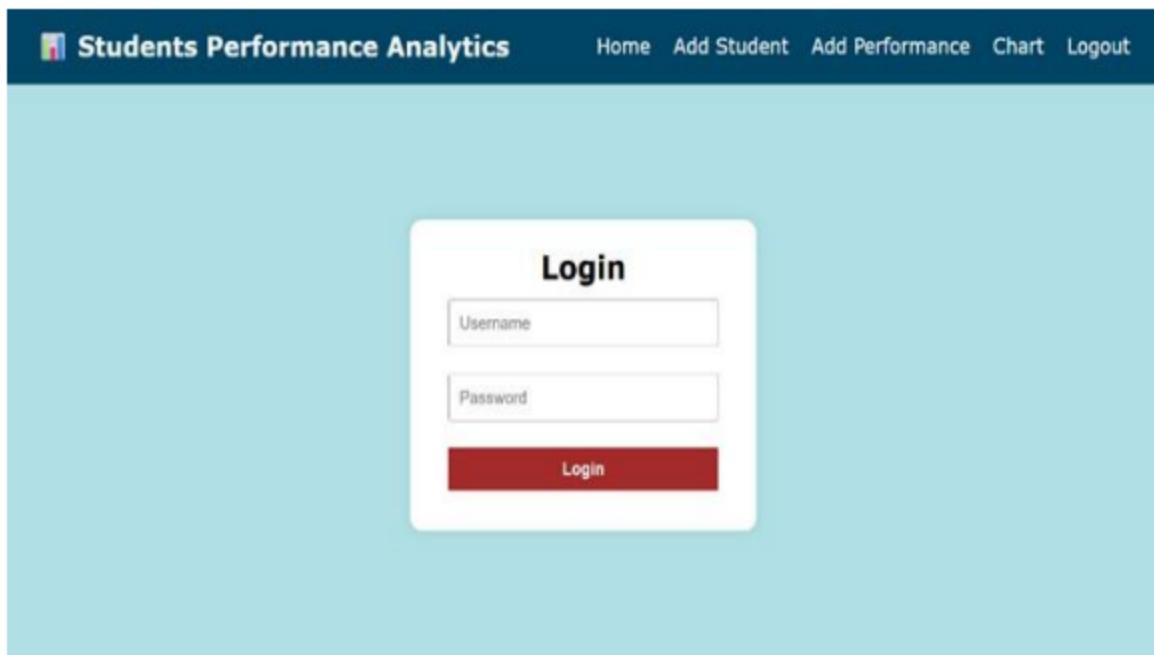
- [1] MongoDB Official Documentation, <https://www.mongodb.com/docs/>
- [2] Node.js Official Documentation, <https://nodejs.org/en/docs/>
- [3] React.js Official Documentation, <https://reactjs.org/docs/getting-started.html>
- [4] Express.js Official Documentation, <https://expressjs.com/en/starter/installing.html>
- [5] Chart.js Documentation, <https://www.chartjs.org/docs/latest/>
- [6] JWT (JSON Web Tokens) Guide, <https://jwt.io/introduction/>

APPENDIX B**ADMIN MANUAL**

This Admin Manual provides step-by-step guidance for using the **Student Performance Analytics** web application. It is designed to help teachers, administrators, and students understand how to operate and navigate the system effectively. The system is developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) and is accessible via any web browser.

Login Page

- The login page allows registered users (Admin, Teacher, or Student) to access their accounts..
- Enter your **Email/Username** and **Password**, then click **Login** to enter the system.

**Add Student Page**

- Purpose: Admin adds new student records.
- Actions: Fill in Name, USN, Class, Subject → Submit → Data stored in database.

The screenshot shows a web-based application titled "Students Performance Analytics". At the top, there is a navigation bar with links for "Home", "Add Student", "Add Performance", "Chart", and "Logout". Below the navigation bar, the title "Add Student" is displayed. The form contains four input fields: "Name", "USN", "Class", and "Subject". Below these fields are two red buttons: "Submit" and "View Students".

Add Performance Page

- Purpose: Admin/Teacher records student marks, attendance, and subject info.
- Actions: Select Student → Enter Subject, Exam, Total Marks, Marks Obtained, Attendance → Submit.

The screenshot shows a web-based application titled "Students Performance Analytics". At the top, there is a navigation bar with links for "Home", "Add Student", "Add Performance", "Chart", and "Logout". Below the navigation bar, the title "Add Student Performance" is displayed. The form includes a dropdown menu for "Select Student", a "Subject" input field, and several other input fields for "Exam (Mid-term / Semester)", "Total Marks", "Marks Obtained", "Attendance (%)", and "Month".

View Performance and Edit / Delete Entries

Open your entry → Click Edit to modify or delete to remove it permanently.

Student Performance Records								
Student Name	USN	Exam	Subject	Total Marks	Marks Obtained	Attendance (%)	Month	Actions
Anu	4SN23MC011	Sem	Maths	100	54	65%	January	<button>Edit</button> <button>Delete</button>
Suma	4SN23MC010	Sem	Maths	100	66	59%	February	<button>Edit</button> <button>Delete</button>
sakshi	4SN23MC012	Sem	Maths	100	79	69%	January	<button>Edit</button> <button>Delete</button>
Arya	4SN23MC013	Sem	English	100	73	75%	January	<button>Edit</button> <button>Delete</button>
theeksha	4SN23MC014	Sem	English	100	8	75%	December	<button>Edit</button> <button>Delete</button>

Charts / Analytics Page

Admins can:

Purpose: Visual representation of student performance and attendance.

Actions: View subject-wise average, student-wise attendance, score distributions

