

**COUNCIL FOR TECHNICAL
EDUCATION AND VOCATIONAL
TRAINING (CTEVT)**

ACME COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER
ENGINEERING



Major Project Final Report
On
STUDENT MANAGEMENT SYSTEM
[Subject Code: EG 3205CT]

Submitted by:

Gopal Khadka

Sagar Kushwaha

Sumit Pratap Sah

Amit Neupane

Kathmandu, Nepal

July 2024

ACKNOWLEDGEMENT

We would like to extend our heartfelt appreciation to everyone who contributed to the successful completion of this project. Our sincere thanks go to our project team members for their dedication and collaboration throughout the project. Each member played a significant role in shaping the outcome. Special thanks to our supervisor, Er. Jagdish Bhatta, for the guidance and valuable feedback, which enriched our work.

We'd like to express our heartfelt gratitude to CTEVT and Acme College of Engineering for affording us the opportunity to collaborate on the “**STUDENT MANAGEMENT SYSTEM**”.

ABSTRACT

Efficient management of academic institutions is crucial for ensuring smooth administrative operations and enhancing the overall educational experience. This project presents the design and implementation of a comprehensive Student Management System (SMS) aimed at automating various academic and administrative processes. The system integrates modules for student information management, course registration, attendance tracking, faculty management and timetable scheduling.

By using a centralized database, the SMS ensures real-time data accessibility and accuracy, reducing administrative workload and minimizing errors. The system features a user-friendly interface, accessible to students, faculty, and administrative staff, enabling seamless interaction and information exchange. Advanced functionalities such as automated notifications, reporting tools, and analytics provide insights for data-driven decision-making and strategic planning.

The SMS is designed to be scalable and adaptable, catering to the unique needs of different educational institutions. Security measures, including role-based access control and data encryption, ensure the confidentiality and integrity of sensitive information. This project highlights the potential of the Student Management System to transform the administrative landscape of academic institutions, fostering a more organized, transparent, and effective educational environment.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1 BACKGROUND THEORY	1
1.2 PROBLEM STATEMENT	1
1.3 OBJECTIVE.....	2
1.4 SCOPE AND APPLICATION OF PROJECT	2
1.5 FEATURES OF PROJECT.....	2
CHAPTER 2: LITERATURE REVIEW	2
2.1 EXISTING SYSTEM.....	3
2.2 PROPOSED SYSTEM.....	3
CHAPTER 3: DESIGN AND METHODOLOGY	4
3.1 SYSTEM DESIGN	4
3.1.1 ER DIAGRAM	4
3.1.2 USE CASE DIAGRAM.....	4
3.1.3 DFD DIAGRAMS	5
3.1.4 FLOWCHART.....	6
3.2 PROCESS MODEL	7
3.3 TOOLS USED.....	8
3.4 DATA SOURCE	8
CHAPTER 4: RESULT AND ANALYSIS.....	9
CHAPTER 5: CONCLUSION	9
5.1 FUTURE ENHANCEMENT.....	9
REFERENCES.....	9

LIST OF ABBREVIATIONS

HTML - Hypertext Markup Language

CSS - Cascading Style Sheets

JS - JavaScript

IDE - Integrated Development Environment

REST - Representational State Transfer

VSCode - Visual Studio Code

CRUD - Create, Read, Update, Delete

API - Application Programming Interface

UI - User Interface

UX - User Experience

SQL - Structured Query Language

DBMS - Database Management System

CSV - Comma-Separated Values

HTTPS - Hypertext Transfer Protocol Secure

LIST OF FIGURES

Fig 3.1.1: ER Diagram of Student Management System.....	4
Fig 3.1.2: Use case diagram of Student Management System.....	5
Fig 3.1.3.1: DFD Level 0	5
Fig 3.1.3.2: DFD Level 1	5
Fig 3.1.3.3: DFD Level 2	6
Fig 3.1.4: Flowchart of Student Management System.....	6
Fig 3.2: Prototyping Process Model.....	7

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND THEORY

Manual student management tasks such as tracking attendance, recording grades, and maintaining student data can be time-consuming and error-prone. To overcome these challenges, there is a growing need for automated systems that improve efficiency, accuracy, and user experience. These systems can use various technologies, such as databases, dashboards, and automated record-keeping features, to streamline operations.

In this project, the focus is on developing a comprehensive Student Management System using Django, which integrates automated functionalities for attendance tracking, grade management, and feedback. By reducing the manual workload, this system can enhance productivity for both students and staff while ensuring more accurate record-keeping.

The proposed system leverages Django's web framework for creating a user-friendly interface and backend, making it easy for admins, staff, and students to access and manage information. This report will cover the technical aspects of the system, including the architecture, database management, and user roles. It will also highlight the system's potential benefits, such as saving time, reducing human errors, and simplifying administrative tasks. Finally, it will discuss the challenges encountered during development and suggest potential future enhancements to further improve the system's efficiency and usability.

1.2 PROBLEM STATEMENT

Recording student attendance manually at the beginning of every lecture, section, or lab is a time-consuming and inefficient process, particularly when dealing with large groups of students. This often leads to delays in starting the class, disrupts the flow of lessons, and can even cause interruptions during exams. Furthermore, the manual attendance sheets are vulnerable to damage or loss as they are passed among students and staff. When the number of students is large, instructors may randomly call names, leading to unfair and inaccurate attendance records. Additionally, managing and tracking attendance for a large number of students increases the likelihood of human error, making it difficult for staff to accurately monitor student attendance rates.

1.3 OBJECTIVE

- Develop a student management system that employs digital system to keep and track all the student records.
- Eliminate the need for manual attendance recording.
- Create a user-friendly interface that is intuitive and easy to navigate for teachers, administrators, and students involved in student management.

1.4 SCOPE AND APPLICATION OF PROJECT

This student management system project has extensive and widespread scope, especially in this digital education era where basic tasks like course registration, student information tracking, student attendance etc. can be quite easy and simple but time consuming and needs manual interference.

The application of this project can be in various areas:

- Data analysis can be easily done on the student's data to predict the student score and other hidden insights.
- Teachers can be involved through real time access to student's attendance and performance.

1.5 FEATURES OF PROJECT

The features of this project can be listed below:

- User authentication: Users need to be authenticated to use this system. Access control mechanisms also have been implemented to provide specific privileges to the different types of the user.
- Profile Management: Users can manage and manipulate their respective profiles through this system.
- Student Management: Admins can add, update, and delete student records, managing all student information in one place.
- Attendance Monitoring: Attendance for students is automatically tracked and stored, reducing manual errors and saving time.

CHAPTER 2: LITERATURE REVIEW

The integration of technology into educational systems has become essential for improving administrative processes and enhancing operational efficiency. Traditional student management systems, which often rely on manual methods for handling data, attendance, and academic records, are increasingly seen as inefficient and error-prone. Automation in these areas has therefore become a key focus, as it simplifies workflows, reduces human error, and allows educational institutions to manage large volumes of data more effectively.

The use of web-based systems, such as those built with Python (Django), is rapidly growing in educational settings for managing students, staff, courses, and feedback.

These systems offer real-time access to data, facilitate easy updates, and support various user roles (admin, staff, and students), which aligns with the goals of our project to automate and streamline administrative tasks. By integrating a Django-based platform, our project aims to reduce manual efforts and improve the overall management of student and staff information.

Furthermore, the application of data analytics within educational systems has shown significant promise in enhancing decision-making processes. The use of summary charts and performance reports within our system will allow staff and administrators to track academic progress, attendance trends, and leave records more efficiently. This data-driven approach not only supports informed decision-making but also helps improve student outcomes by identifying areas for improvement.

2.1 EXISTING SYSTEM

In traditional student management systems, administrative tasks such as managing student records, attendance, and academic performance are typically handled manually. Attendance is usually recorded on paper or via sign-in sheets, where students physically mark their presence or use ID cards to track attendance. This manual process is time-consuming, error-prone, and often requires a significant amount of effort from both staff and students to maintain and update records.

The existing systems may also involve spreadsheets or basic software where instructors input attendance data manually, leading to potential inaccuracies and delays. These systems lack integration and real-time tracking, making it difficult to monitor student attendance efficiently. In cases of large class sizes, this method becomes even more cumbersome and prone to mistakes, making it challenging for staff to maintain accurate and up-to-date attendance records.

2.2 PROPOSED SYSTEM

The proposed system is a comprehensive Django-based Student Management System designed to automate and streamline student data management, attendance tracking, and academic performance monitoring. Instead of relying on manual methods, attendance and other student records will be automatically tracked through the system. Staff and administrators will be able to add, update, and manage student, course, and staff data in real-time.

This automated system eliminates the need for physical attendance sheets, reducing administrative effort and minimizing errors. Attendance records are updated instantly as students check in, and performance data such as grades and feedback can be easily monitored. Additionally, the system generates real-time reports and summary charts for both students and staff, providing valuable insights into attendance trends, academic performance, and leave status.

CHAPTER 3: DESIGN AND METHODOLOGY

3.1 SYSTEM DESIGN

The system design for the student management website involves the development of a frontend using HTML along with Bootstrap for CSS and JavaScript functionalities to create an intuitive and responsive user interface. Python, supported by the Django framework, will drive the backend logic, managing user requests and integrating face recognition technology.

SQLite will be utilized to store student information and attendance records efficiently. User authentication mechanisms will ensure secure access, while data retrieval methods will enable efficient handling of student data for tasks such as report generation and profile updates.

3.1.1 ER DIAGRAM

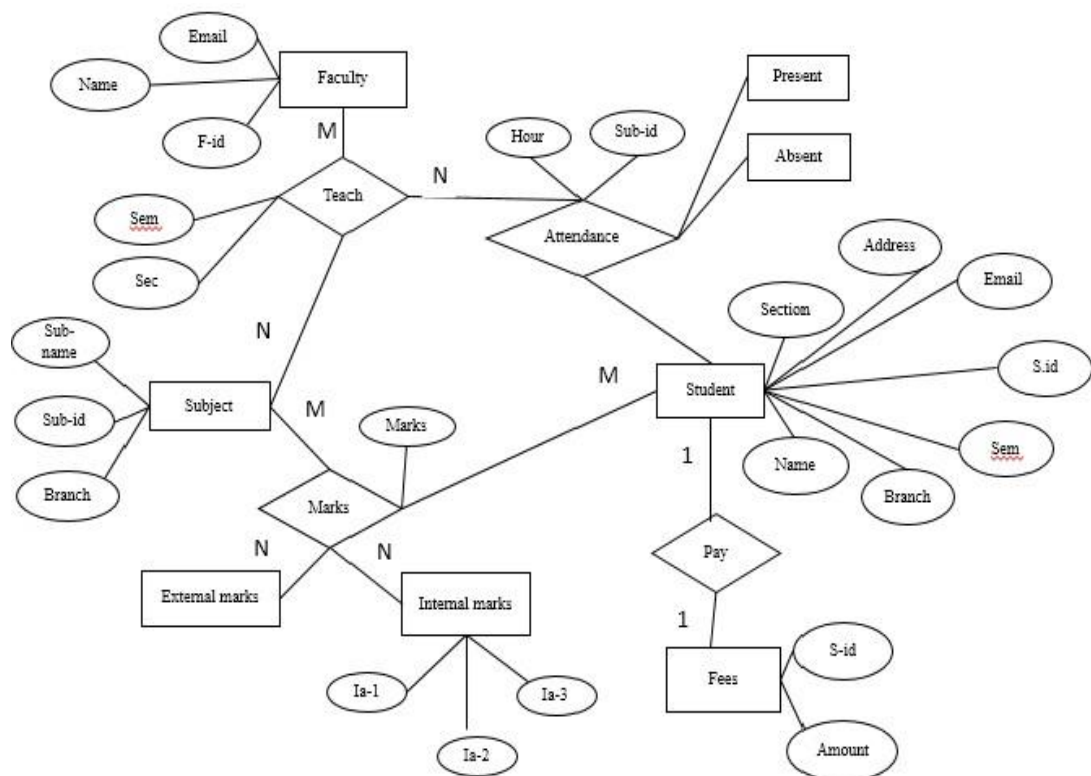


Fig 3.1.1: ER Diagram of Student Management System

3.1.2 USE CASE DIAGRAM

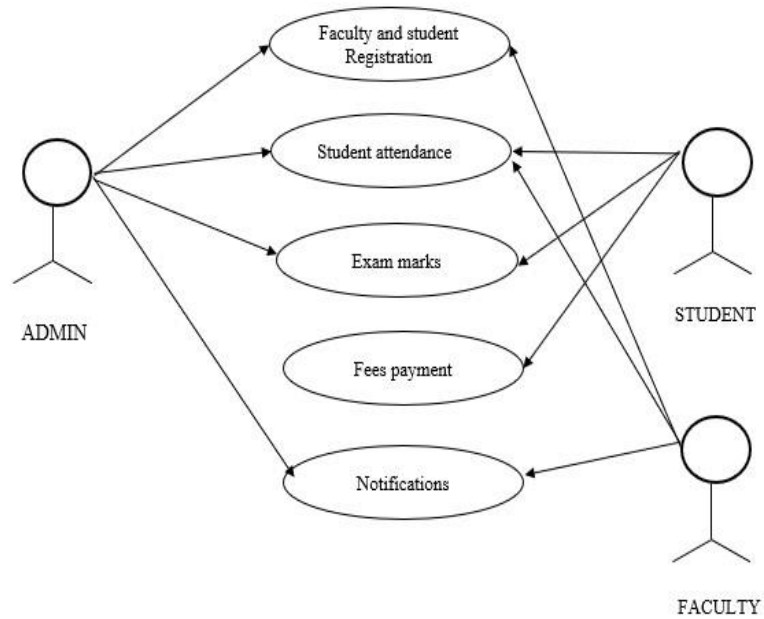


Fig 3.1.2: Use case diagram of Student Management System

3.1.3 DFD DIAGRAMS

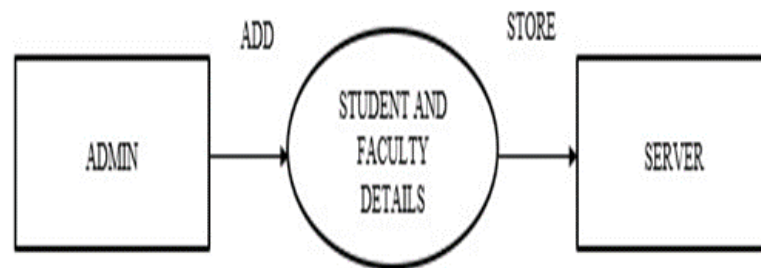


Fig 3.1.3.1: DFD Level 0

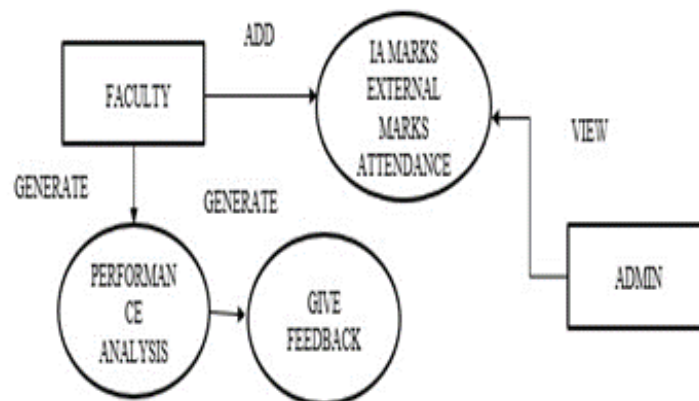


Fig 3.1.3.2: DFD Level 1

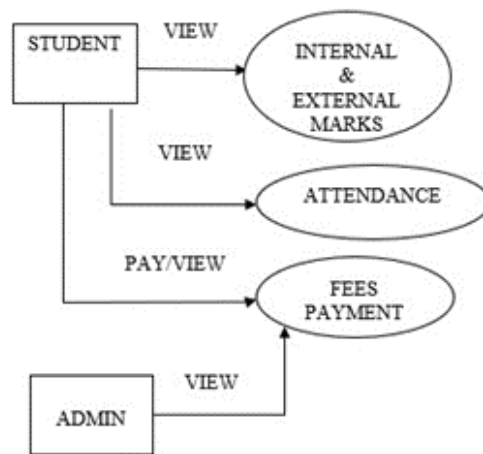


Fig 3.1.3.3: DFD Level 2

3.1.4 FLOWCHART

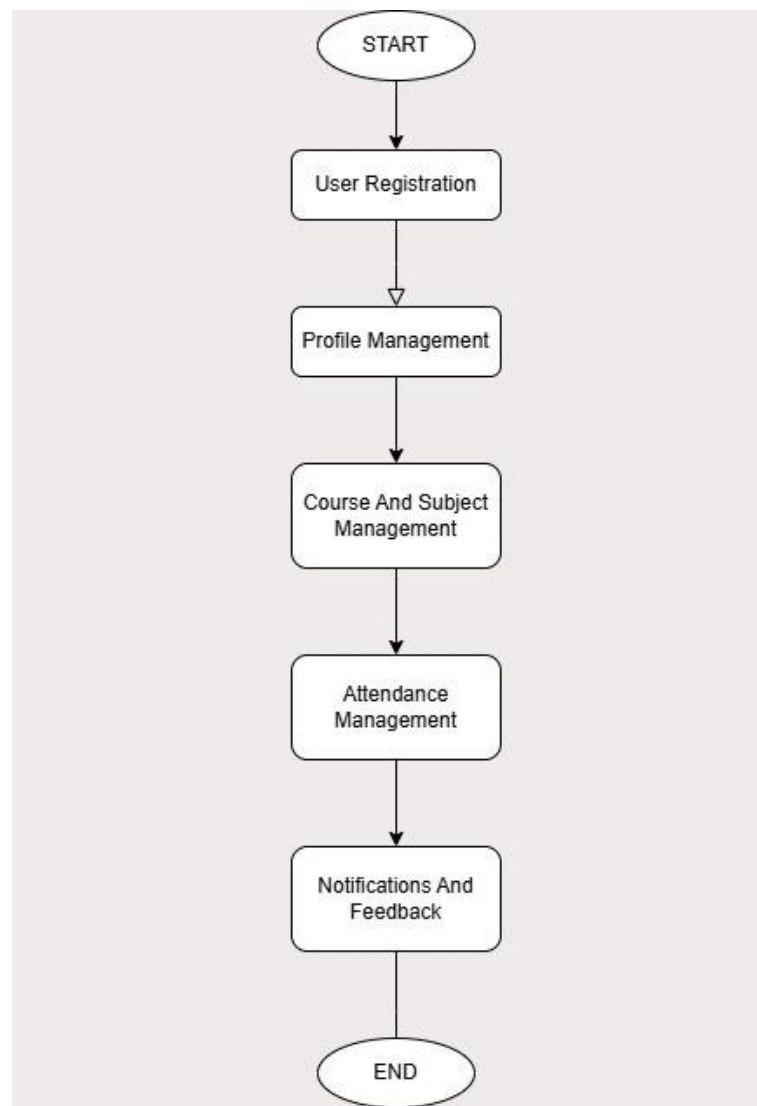


Fig 3.1.4: Flowchart of Student Management System

3.2 PROCESS MODEL

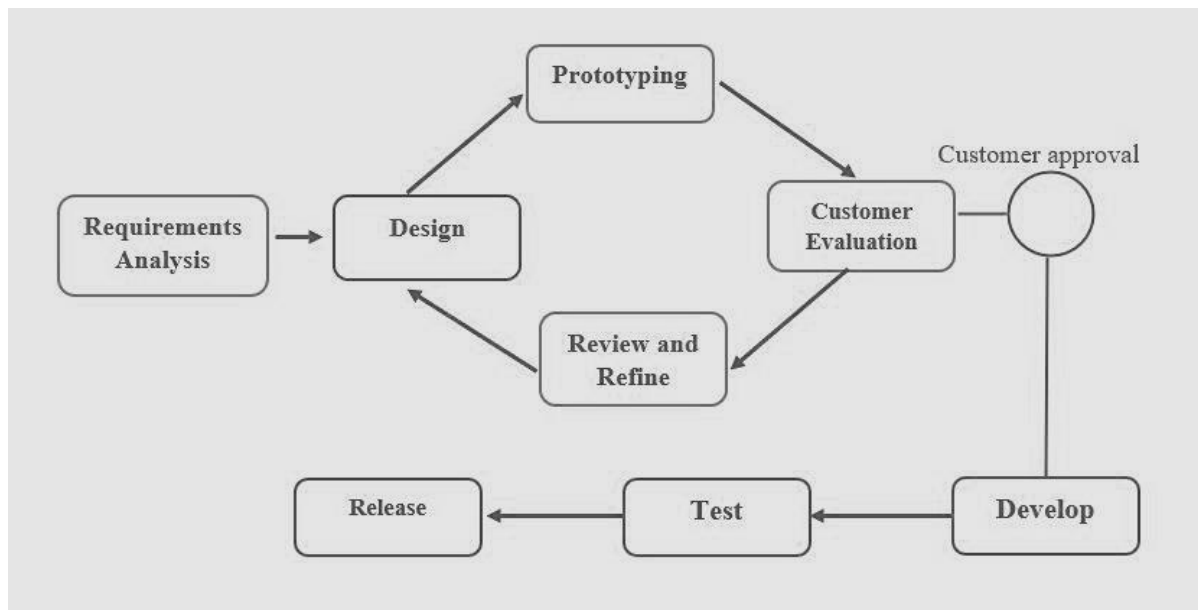


Fig 3.2: Prototyping Process Model

Step-1: Requirements gathering and analysis:

Requirement analysis is the first step in developing a prototyping model. During this phase, the system's desires are precisely defined. During the method, system users are interviewed to determine what they expect from the system.

Step-2: Quick design:

The second phase could consist of a preliminary design or a quick design. During this stage, the system's basic design is formed. However, it is not a complete design. It provides the user with a quick overview of the system. The rapid design aids in the development of the prototype.

Step-3: Build a Prototype:

During this stage, an actual prototype is intended to support the knowledge gained from quick design. It is a small low-level working model of the desired system.

Step-4: Initial user evaluation:

The proposed system is presented to the client for preliminary testing at this stage. It is beneficial to investigate the performance model's strengths and weaknesses. Customer feedback and suggestions are gathered and forwarded to the developer.

Step-5: Refining prototype:

If the user is dissatisfied with the current model, you may want to improve the type that responds to user feedback and suggestions. When the user is satisfied with the upgraded model, a final system based on the approved final type is created.

Step-6: Implement Product and Maintain:

The final system was fully tested and distributed to production after it was developed to support the original version. To reduce downtime and prevent major failures, the programmer is run on a regular basis.

3.3 TOOLS USED

Visual Studio Code (VSCode):

Visual Studio Code is a popular and versatile source code editor developed by Microsoft. It provides a plethora of features that streamline the coding process, including syntax highlighting, debugging capabilities, intelligent code completion, and Git integration. VSCode supports various programming languages and extensions, making it a preferred choice for many developers for its flexibility and efficiency.

SQLite:

SQLite is a lightweight and self-contained relational database management system. Unlike traditional database management systems that operate as separate server processes, SQLite functions as a library that applications can link to. It is widely used in embedded systems and mobile applications due to its simplicity, speed, and ease of integration. SQLite databases are stored in a single file, making them portable and suitable for small to medium-scale applications.

Python with Django:

Python is a high-level, versatile programming language known for its readability and vast ecosystem of libraries and frameworks. Django is a high-level Python web framework that simplifies the development of web applications by providing built-in features for tasks like URL routing, database handling, and user authentication.

Bootstrap:

Bootstrap is a front-end framework developed by Twitter that simplifies the process of creating responsive and visually appealing websites. It includes a collection of pre-designed CSS and JavaScript components, such as flex, buttons, forms, and navigation bars, that can be easily integrated into web projects.

3.4 DATA SOURCE

Student Information Database:

The system will source student information from a SQLite database, which will store details such as student names, identification numbers, contact information, and attendance records. This database serves as the central repository for student data, enabling efficient management and retrieval of information.

CHAPTER 4: RESULT AND ANALYSIS

The project successfully developed a Django-based Student Management System, automating key tasks like student data management, attendance tracking, and performance monitoring. Using Python and Django for backend logic, and HTML with Bootstrap for the frontend, the system provides a user-friendly interface for all users. Attendance is automatically tracked and stored in a SQLite database, while user authentication ensures secure access.

The system effectively automated administrative tasks, improved data accuracy, and enhanced operational efficiency, meeting its objectives and providing a secure, user-friendly platform for educational institutions.

CHAPTER 5: CONCLUSION

The Django-based Student Management System will efficiently manage and streamline various administrative tasks. The integration of different modules reduces the time and effort needed for handling student data, attendance, and academic performance. By automating processes, the system ensures that student and staff information is collected and stored dynamically, eliminating paperwork and minimizing manual effort. This not only benefits the students and staff but also significantly reduces the administrative workload, making the management process more efficient and effective.

5.1 FUTURE ENHANCEMENT

The future enhancement of the system can be done by introducing online examination module to conduct online examination. Further, the faculty can upload subject notes and videos of their class lectures on this site and students who has missed those classes can view those videos.

REFERENCES

1. Isbudeen Noor Mohamed, Ahmad Tasnim Siddiqui, Syed Ajaz, S. Mohamed Idhris, "Student Information Report System with SMS (SIRS)", in proc. 2016 International Conference on Computer Communication and Informatics (ICCCI -2016), Jan. 07 – 09, 2016, Coimbatore, INDIA.
2. Kartiki Datakar, "Online College Management System," International Journal of Computer Science and Mobile Computing, Vol. 5, Issue 4, April 2016.

3. Lalit Joshi, "A Research Paper on College Management System", International Journal of Computer Applications, Vol.122, Issue 11, July 2015.
4. Zhibing Liu, Huixia Wang, Hui Zan "Design and implementation of online college management system." 2010 International symposium on intelligence information processing and trusted computing. 978-0-7695- 41969/10 IEEE.
5. S.R.Bharamagoudar, "Web-Based Student Information Management System", International Journal of Advanced Research in Computer and Communication Engineering, Vol.2, Issue 6, June 2013.
6. Gateways to portals research. International Journal of web portals, 1(1), 1–15.