



PROJECT REPORT ON

SCHOOL/COLLEGE MANAGEMENT SYSTEM

EDUSTATION

PREPARED BY ANSHU JOSHI
(0032032110)

Submitted To

**Geetanjali College of Computer
Science & Commerce (BBA),
Rajkot-Gujarat**

**Saurashtra University,
Rajkot-Gujarat**

CERTIFICATE HERE

PROJECT PROFILE

- **TITLE:** EduStation | School/College Management System
- **CREATED BY:** Anshu Joshi (032032110)
- **GUIDED BY:** Prof. Pranav Trivedi
- **FRONT END:** HTML5, CSS3, JavaScript
- **BACK END:** Django, Python, SQLite
- **TIME DURATION:** 28 Days
- **COURSE:** Bachelor's in Computer Application
- **SEMESTER:** VIth

DECLARATION

I, the student of Geetanjali Group of Colleges, Rajkot undersigned here, declare that the School Management System is my own work and therefore it has not been submitted to any other University be it locally or internationally, for the purpose of attaining a degree or for an examination. In instances where I have used or quoted external material, these sources have been accredited and indicated clearly in the complete reference list. I fully understand that the use of external material without giving due credit to the source or author may and will be considered as plagiarism.

ACKNOWLEDGEMENTS

I would like to thank the faculties of the institute for making this work extremely easy, without whom I would not have reached this far. I would also like to convey my profound sense of gratitude to my supervisor **Prof. Pranav Trivedi** sir for his constructive academic advice and guidance, constant encouragements and valuable suggestions rendered throughout this project work. I really benefited from his excellent and professional supervision. My sincere thanks go to all my Friends, Course mates and School managers who helped, inspired and gave me mental support at different stages in different moments on this project. Above all I want to extend my thanks to my dearly friends and parents for their encouraging words and moral support.

And, as you are the user of this project, I request you to please give your much precious opinions & feedbacks. In case of any bugs or processing errors, please feel free to retort me, I'll be pleasant enough to solve them for you, Thank You...

Yours Sincerely,

Anshu C. Joshi

(0032032110)

ABSTRACT

In recent years India has seen a steady increase of ICTs in educational institutions. Most schools that are equipped, utilize second-hand or refurbished computers. The integration of ICTs in learning and teaching practice has been limited, although the introduction of computer studies as a school study subject has begun to change the way in which we value ICTs in educational institutions. The main purpose of the Strategy for Information and Communication Technology Implementation in Education is to provide the prospects and trends of integrating information and communication technology (ICT) into the general educational activities. One such activity is the reporting of Students' progress and achievements. Most schools in India still use the manual system of Student Management that involves management of every school. This has proven to be a challenge for busy management that are unable to collect manage students during open day. It is for this reason that the Students' School Management System has been development. The system has been implemented using the waterfall methodology.

EduStation, Student Management System helps headmasters to get the most accurate information to make effective decisions. Teachers and students to gain time saving administrative tools, students gain immediate access to their grades and progress. EduStation SMS is equipped with features that makes it possible to generate schedules and reports in minutes and to retrieve attendance records, grade checks, report cards, transcripts and form letters in just a few clicks.

EduStation SMS helps teachers to complete grade book, track student attendance, input class notes, create lesson plans, and detailed reports and communicate with other staff members, students, and parents via an e-mail. It also helps students to access assignments and tests, view attendance records, grades, report cards, and progress reports all online. They also can communicate through mail and forums with teachers and other students online.

EduStation SMS is a web-based application developed in Django, one of the most popular frameworks of Python Programming Language and SQLite, a minified version of another popular SQL Service, MySQL. To implement this application, schools and institutes do not need expensive hardware and software, all they are required with is a high-speed internet connection with minimum average speed of 5Mbps. It will also work as a centralized database and application that schools & institutes can easily access from anywhere through a standard internet accessible system. This system can be easily customized for individual institutes as per their needs

I N D E X

Sr. No.	Topics	Page No.
1	Chapter – I: Industry Introduction 1.1 Introduction to School/College Management System 1.2 The Problem Statement 1.3 Objectives of School/College Management System 1.4 Scope of School/College Management System	1
2	Chapter – II: Literature Review 2.1 Necessary Designs used in School/College Management Systems 2.2 Existing Systems and their features & limitations	5
3	Chapter – III: Problem Identification & Definition, Process Modification 3.1 Problem Identification 3.2 Problem Definition 3.3 Project Purpose 3.4 Project Management 3.5 Process Modification 3.6 Advantages of Proposed System	7
4	Chapter – IV: Industrial Process / Product & Problem Analysis 4.1 Whole Industrial Process 4.2 Problem Study 4.3 Project Plan	12
5	Chapter – V: System Analysis & Design, Technology Study 5.1 System Analysis & Design 5.2 System Specifications 5.3 Technology Study 5.4 ER & DFD Diagrams 5.5 Database Design	15
6	Chapter – VI: Implementation & Testing 6.1 Implementation of the System 6.2 Implementation of Programming Languages 6.3 Test Cases	27
7	Chapter – VII: System Results 7.1 General 7.2 Administrator 7.3 Staff 7.4 Student	31
8	Chapter – VIII: Conclusion & Future Works 8.1 Summary of EduStation 8.2 Future Improvisation	44
9	Chapter – IX: References 9.1 List of Bibliographies	45
10	Chapter – X: Appendices 10.1 List of Appendices	46

Chapter One: Industry Introduction

1.1 Introduction to School/College Management System

A school/college management system is a software application designed to automate and streamline the administrative and academic processes of educational institutions. The system provides a comprehensive and centralized platform for managing various aspects of a school/college, including student information, staff management, class schedules, and academic records.

The need for an efficient and effective school/college management system has become increasingly important as educational institutions have grown in size and complexity. In the past, school/college management was done manually, which was time-consuming and prone to errors. With the advent of technology, school/college management systems have become more sophisticated and are now able to automate many of the tasks that were previously done manually.

The school/college management system offers a range of features that enable schools/colleges to manage their operations more efficiently. Some of the key features include:

1. **Student information management:** This allows schools/colleges to store and manage student records, including personal information, contact details, and academic performance.
2. **Staff management:** The system provides tools for managing staff information, including contact details, qualifications, and employment history.
3. **Class scheduling and timetable management:** The system allows schools/colleges to create and manage class schedules and timetables.
4. **Attendance tracking:** The system enables schools/colleges to keep track of student attendance and generate reports.
5. **Examination management:** The system allows schools/colleges to create and manage examination schedules and results.
6. **Financial management:** The system provides tools for managing school/college finances, including fee collection, budgeting, and expenses.
7. **Communication:** The system allows schools/colleges to communicate with students, parents, and staff through various channels, including email and SMS.

Overall, a school/college management system is a powerful tool that can help schools/colleges to improve their operations, increase efficiency, and provide better services to students, parents, and staff.

1.2 The Problem Statement

The problem statement in a school/college management system is focused on addressing the challenges and inefficiencies faced by educational institutions in managing their day-to-day operations. These challenges include:

1. **Manual and time-consuming processes:** Many school/college management tasks are still done manually, which is time-consuming and prone to errors. This includes tasks such as student record-keeping, class scheduling, and examination management.
2. **Lack of centralized information:** Schools/Colleges often have multiple systems and databases for managing different aspects of their operations, which makes it difficult to access and share information. This can lead to inconsistencies and errors in data, and makes it harder to make informed decisions.
3. **Inadequate communication and collaboration:** Schools/Colleges often have difficulty communicating and collaborating with students, parents, and staff. This can lead to misunderstandings, delays, and a lack of transparency in school/college operations.
4. **Inefficiency in financial management:** Schools/Colleges often face difficulties in managing their finances, including collecting fees, budgeting, and tracking expenses.
5. **Difficulty in providing personalized support:** Schools/Colleges often struggle to provide personalized support to students, such as tracking academic progress and providing guidance.

The problem statement of a school/college management system is to address these challenges by providing a comprehensive, centralized, and user-friendly platform that automates and streamlines school/college management tasks and enables efficient communication and collaboration among the stakeholders.

1.3 Objectives of School/College Management System

The objectives of a school/college management system are to:

1. **Automate and streamline administrative and academic processes:** The system aims to automate and streamline tasks such as student record-keeping, class scheduling, examination management, and financial management. This will save time and reduce errors, allowing schools/colleges to focus on more important tasks.
2. **Provide a centralized platform for managing and sharing information:** The system aims to provide a centralized platform for storing and managing all school/college information, including student records, staff details, and academic schedules. This will allow schools/colleges to access and share information more easily and reduce inconsistencies in data.

3. **Improve communication and collaboration among stakeholders:** The system aims to improve communication and collaboration among students, parents, staff, and administration. By providing tools for communication, the system will enable stakeholders to share information, ask questions, and provide feedback more easily.
4. **Enhance financial management:** The system aims to provide a platform for schools/colleges to manage their finances efficiently, including fee collection, budgeting, and expenses.
5. **Personalized support:** The system aims to provide personalized support to students and help them to achieve academic success by tracking their academic progress, providing guidance, and providing opportunities for feedback and interaction.
6. **Reporting and Data analysis:** The system aims to provide a platform for schools/colleges to generate reports and analyze data, which will help in decision making and in identifying areas that need improvement.
7. **Improved security:** The system aims to provide a secure platform for storing and managing sensitive information, such as student records and financial details.

Overall, the objectives of a school/college management system are to improve the efficiency and effectiveness of school/college operations and to provide better services to students, parents, and staff.

1.4 Scope of School/College Management System

The scope of a school/college management system encompasses the various functionalities and features that the system provides to automate and streamline the administrative and academic processes of educational institutions. The scopes of a school/college management system can include:

1. **Student Information Management:** The system allows schools/colleges to store and manage student records, including personal information, contact details, and academic performance. This includes features such as enrollment, attendance tracking, grade management, and report generation.
2. **Staff Management:** The system provides tools for managing staff information, including contact details, qualifications, and employment history. This includes features such as staff scheduling, payroll, and performance evaluation.
3. **Class Scheduling and Timetable Management:** The system allows schools/colleges to create and manage class schedules and timetables. It also allows for automatic generation of timetable and allows to manage the classroom and facility reservation.
4. **Examination Management:** The system allows schools/colleges to create and manage examination schedules and results. This includes features such as question paper generation, online examination and results analysis.

5. **Financial Management:** The system provides tools for managing school/college finances, including fee collection, budgeting, and expenses. This includes features such as fee management, financial reports, and budget forecasting.
6. **Communication:** The system allows schools/colleges to communicate with students, parents, and staff through various channels, including email, SMS, and other digital messaging systems.
7. **Reporting and Data Analysis:** The system provides a platform for schools/colleges to generate reports and analyze data, which will help in decision making and in identifying areas that need improvement.
8. **Security:** The system aims to provide a secure platform for storing and managing sensitive information, such as student records and financial details.

These are some of the scopes of a school/college management system, but the specific features and functionalities may vary depending on the specific needs of the institution and the system used.

Chapter Two: Literature Review

2.1 Necessary Designs used in School/College Management Systems

There is a significant body of literature on school/college management systems, which covers a wide range of topics such as their design, implementation, and evaluation. Some of the key themes that are addressed in the literature include:

1. **User-centered design:** Many studies have focused on the importance of user-centered design in the development of school/college management systems. This includes research on user-centered design methodologies and the use of user research and usability testing to inform design decisions.
2. **System functionality:** There is a large amount of literature that focuses on the specific functionalities and features of school/college management systems. This includes research on student information management, staff management, class scheduling, examination management, and financial management.
3. **Implementation and adoption:** Many studies have focused on the implementation and adoption of school/college management systems. This includes research on the challenges and barriers to implementation, as well as best practices for system rollout and adoption.
4. **Evaluation and impact:** There are a significant amount of literature that focuses on the evaluation and impact of school/college management systems. This includes research on the effectiveness of different systems in improving school/college operations, as well as the impact of systems on students, staff, and administrators.
5. **Integration with other systems:** Some of the literatures have also focused on the integration of school/college management systems with other systems such as Student Information Systems (SIS), Learning Management Systems (LMS) and other enterprise resource planning (ERP) systems.
6. **Cloud-based and Mobile systems:** With the growing popularity of cloud-based and mobile systems, there is an increasing amount of literature on the implementation and benefits of these systems in school/college management.

Overall, the literature on school/college management systems highlights the importance of user-centered design, the need for comprehensive system functionality, and the challenges and barriers to implementation and adoption. The literature also provides valuable insights into the evaluation and impact of these systems on school/college operations and the people who use them.

2.2 Existing Systems and their features & limitations

There are many existing school/college management systems available on the market, each with their own set of features and limitations. Some of the most popular systems include:

1. **Blackbaud:** Blackbaud is a comprehensive school/college management system that offers a range of features such as student information management, staff management, class scheduling, examination management, financial management, and communication tools. One of its key features is the integration with Blackbaud's student information system (SIS) and learning management system (LMS). However, it can be quite expensive and its implementation can be complex.
2. **SchoolTime:** SchoolTime is a cloud-based school/college management system that offers features such as student information management, staff management, class scheduling, examination management, financial management, and communication tools. It also provides features for parent-teacher communication and mobile access. However, it may lack some advanced features and customization options compared to other systems.
3. **Fedena:** Fedena is an open-source school/college management system that offers features such as student information management, staff management, class scheduling, examination management, financial management, and communication tools. It's free to use and offers customization options but it may require some technical expertise to set up and maintain.
4. **CampusVue:** CampusVue is a comprehensive school/college management system that offers features such as student information management, staff management, class scheduling, examination management, financial management, and communication tools. It also provides features for student self-service and mobile access. However, it can be quite expensive and its implementation can be complex.
5. **Eduma:** Eduma is a Learning Management System (LMS) that offers features such as student information management, class scheduling, examination management, financial management, and communication tools. It's a WordPress based LMS, it can be easily customizable and it's free to use. However, it may lack some advanced features and customization options compared to other systems.

Overall, the features of existing school/college management systems vary, and it's important to carefully evaluate the features and limitations of each system before making a decision. It's also important to consider the specific needs and resources of the school/college in order to choose the best system for their needs.

Chapter Three: Problem Identification & Definition, Process Modification

3.1 Problem Identification

1. Inadequate communication between administrators, teachers, and parents.
2. Difficulty in tracking student progress and identifying areas where they need improvement.
3. Limited resources and budget constraints affecting the quality of education.
4. Difficulty in managing student data and ensuring its security and privacy.
5. Inefficient scheduling and allocation of resources such as classrooms and equipment.
6. Lack of integration between different systems and software used by the school.
7. Limited ability to customize the system to meet the specific needs of the school or district.
8. Difficulty in generating accurate and meaningful reports for decision making.
9. Inadequate technology infrastructure and support for online and remote learning.
10. Difficulty in tracking and managing student attendance and punctuality.

3.2 Project Definition

A school/college management system is a software solution that helps educational institutions to manage and organize a few core administrative tasks and functions. It typically includes the following features:

1. **Student Information Management:** Store and organize student records, including personal information, contact details, and basic academic history.
2. **Class Scheduling:** Create and manage class schedules, including room assignments and teacher assignments.
3. **Grade Tracking:** Track and record student grades, including test scores and homework assignments.
4. **Student Attendance:** Track student attendance, including absences and tardiness.
5. **Communication:** Facilitate basic communication between teachers and administrators.
6. **Reporting:** Generate basic reports such as attendance records, grade reports and basic student information reports.
7. **User Management:** Manage user accounts and roles, including teachers, administrators, and students.

This type of system is typically used by smaller educational institutions or as a starting point for larger institutions that want to automate their basic processes before implementing more advanced features.

3.3 Project Purpose

A school/college management system serves several purposes, including:

1. **Automation of administrative tasks:** The system automates tasks such as student registration, class scheduling, grade tracking, and report card generation, reducing the need for manual data entry and increasing efficiency.
2. **Improved communication and collaboration:** The system facilitates communication and collaboration between teachers, administrators, and parents, allowing them to easily share information and stay informed about student progress.
3. **Increased data security and privacy:** The system provides a secure and centralized database of student information, ensuring that sensitive data is protected and can only be accessed by authorized personnel.
4. **Better decision making:** The system generates accurate and meaningful reports that can be used to make informed decisions about student progress and the overall operation of the school or college.
5. **Enhanced student tracking:** The system allows for the tracking of student attendance, punctuality, and performance which enables the school to identify areas where students need additional support.
6. **Flexibility and customization:** The system can be customized to meet the specific needs of the school or college, allowing it to be tailored to their unique requirements.
7. **Improved resource management:** The system enables efficient scheduling and allocation of resources such as classrooms and equipment.
8. **Online and remote learning:** The system can support online and remote learning, which are becoming more prevalent in recent years.

3.4 Project Management

Project management of a school/college management system can be understood in the context of the software development life cycle (SDLC). The SDLC is a methodology used to develop software systems, and it typically includes the following phases:

1. **Planning:** Identify the need for a school/college management system, define the project objectives, scope, and deliverables. This phase is similar to the initiation phase of project management.
2. **Analysis:** Gather and analyze requirements for the system, including the needs and expectations of users and stakeholders. Identify the key features and functionalities required, and create a detailed project plan. This phase is similar to the planning phase of project management.

3. **Design:** Create a detailed design of the system, including the user interface, database structure, and system architecture. This phase is also similar to the planning phase of project management.
4. **Implementation:** Coordinate the development and testing of the system, including the selection of software vendors or developers, and the management of any subcontractors. This phase is similar to the execution phase of project management.
5. **Testing:** Conduct testing of the system to ensure it meets the requirements and functionalities, validate the system performance and usability, and fix any bugs or defects. This phase also similar to the monitoring and controlling phase of project management.
6. **Deployment:** Roll out the system to the school or college and provide training and support to users. This phase is similar to the closure and implementation phase of project management.
7. **Maintenance:** Monitor the system's performance and make necessary updates, fix defects, and implement changes to ensure the system continues to meet the needs of the school or college. This phase is similar to the monitoring and controlling phase of project management.

As you can see, the project management process and the SDLC are closely related and work together to ensure the successful development and implementation of a school/college management system. Project management of a school/college management system requires a team of experts with different skill sets. The team should include project manager, software developers, IT experts, system analysts, and subject matter experts from the school or college. A good communication and collaboration are vital for successful project management. Additionally, it is important to have a clear project plan, set milestones and regularly review the progress against the plan, to ensure the project is on track.

3.5 Process Modification

Process modification of a school/college management system refers to making changes to the existing processes and procedures used to manage the system. This can include changes to the way data is entered and stored, changes to the way reports are generated, or changes to the way users interact with the system. The process modification typically includes the following steps:

1. **Identifying the need for modification:** Identify areas where the existing processes and procedures are not meeting the needs of the school or college or where there is room for improvement.
2. **Analysis:** Gather and analyze the requirements for the modification, including the needs and expectations of users and stakeholders. Identify the key processes and procedures that need to be added or modified.

3. **Design:** Create a detailed design of the modification, including changes to the system processes and procedures.
4. **Implementation:** Coordinate the implementation of the modification, including training and support for users.
5. **Testing:** Conduct testing of the modification to ensure it meets the requirements and functionalities, validate the system performance and usability, and fix any bugs or defects.
6. **Deployment:** Roll out the modification to the school or college and provide training and support to users.
7. **Maintenance:** Monitor the system's performance and make necessary updates, fix defects, and implement changes to ensure the system continues to meet the needs of the school or college.

It's important to note that process modification should be planned and executed carefully to minimize disruptions to the system's normal operation and to ensure that the modifications don't negatively impact the existing functionalities. Also, the process modification should be aligned with the system modification, to ensure the system and the process are working together seamlessly.

3.6 Advantages of Proposed System

A proposed school/college management system can offer several advantages that can improve the overall functioning of an educational institution. Some of these advantages are:

1. **Streamlined administration:** The system automates tasks such as student registration, class scheduling, grade tracking, and report card generation, reducing the need for manual data entry, and increasing efficiency.
2. **Improved communication and collaboration:** The system facilitates communication and collaboration between teachers, administrators, and parents, allowing them to easily share information, and stay informed about student progress. This can lead to better coordination and teamwork among the stakeholders.
3. **Increased data security and privacy:** The system provides a secure and centralized database of student information, ensuring that sensitive data is protected and can only be accessed by authorized personnel. This can help to comply with regulations and protect student's data.
4. **Better decision making:** The system generates accurate and meaningful reports that can be used to make informed decisions about student progress and the overall operation of the school or college. This can help to identify areas of improvement and make data-driven decisions.

5. **Enhanced student tracking:** The system allows for the tracking of student attendance, punctuality, and performance which enables the school to identify areas where students need additional support. This can help to improve student's performance and reduce dropout rates.
6. **Flexibility and customization:** The system can be customized to meet the specific needs of the school or college, allowing it to be tailored to their unique requirements. This can help to make the system more user-friendly and meet the institution's specific needs.
7. **Improved resource management:** The system enables efficient scheduling and allocation of resources such as classrooms and equipment. This can help to reduce waste and increase resource utilization.
8. **Online and remote learning:** The system can support online and remote learning, which are becoming more prevalent in recent years. This can help to adapt to the changing learning environment and support students who are not able to attend in person.
9. **Cost-effective:** A proposed school/college management system can be cost-effective as it can automate tasks, reduce the need for manual data entry, and increase efficiency, this can lead to cost savings for the institution.
10. **Scalability:** A good proposed school/college management system can be scalable, meaning it can adapt to the changing needs of the institution as it grows or evolves. This can help to ensure the system will continue to meet the institution's needs in the future.

Chapter Four: Industrial Process, Product & Problem Analysis

4.1 The Whole Process

A project process for a school/college management system typically includes the following steps:

1. **Initiation:** Identify the need for a school/college management system and define the project objectives, scope, and deliverables.
2. **Planning:** Develop a detailed project plan, including timelines, resource requirements, and project milestones. This step includes identifying the stakeholders, gathering the requirements, selecting the appropriate methodology, and creating a project charter.
3. **Execution:** Coordinate the development and testing of the system, including the selection of software vendors or developers, and the management of any subcontractors. This step includes the development of the system, testing and debugging, and creating the necessary documentation.
4. **Monitoring and controlling:** Track the progress of the project, identify and address any issues that arise, and make any necessary adjustments to the project plan. This step includes monitoring the project schedule, budget, and quality, and controlling the scope, risks, and issues.
5. **Closure:** Close out the project by conducting a final review and evaluation, and documenting any lessons learned for future reference. This step includes completing the project deliverables, conducting a final project review, and closing out the project.
6. **Implementation:** Roll out the system to the school or college, and provide training and support to users. This step includes the deployment of the system, providing the necessary support and training for the users, and monitoring the system's performance.

It is important to note that the project process should be flexible and adaptable to the unique requirements and constraints of each project, and it should be aligned with the SDLC (Software Development Life Cycle) methodology.

4.2 Problem Study

A problem study of a school/college management system involves identifying and analyzing the issues and challenges related to the existing system in use. This process typically includes the following steps:

1. **Identifying the problem:** Gather feedback from users, including teachers, administrators, and parents, to understand the issues and challenges they are facing with the current system.

2. **Analysis:** Analyze the data collected and identify the root cause of the problem.
3. **Identifying potential solutions:** Based on the analysis, identify potential solutions to address the problem.
4. **Evaluating solutions:** Evaluate the potential solutions and determine the best course of action.
5. **Implementing solutions:** Implement the chosen solution and monitor its effectiveness.
6. **Review and maintenance:** Review the system regularly to ensure it is meeting the needs of the school or college and make any necessary adjustments.

The problem study of a school/college management system can help to identify areas where the existing system is not meeting the needs of the school or college, and can lead to the implementation of solutions that can improve the overall functioning of the institution. This process can be an ongoing one, as the system and the requirements will change over time.

4.3 Project Planning

A school/college management system project plan following the Waterfall model would typically include the following phases:

1. **Requirements Gathering:** In this phase, the project team would gather requirements from stakeholders such as teachers, administrators, and parents to understand the needs and expectations of the system. This would include meetings, interviews, and workshops to gather information on the system's functional and non-functional requirements.
2. **System Design:** In this phase, the project team would use the requirements gathered in the previous phase to design the system's architecture, user interface, and database structure. The project team would also create detailed design documents outlining the system's functionalities and technical specifications.
3. **Implementation:** In this phase, the project team would begin the implementation of the system, including the development of the software, testing, and debugging. This would include coding, testing, and fixing bugs and defects.
4. **Testing:** In this phase, the project team would conduct testing to ensure the system meets the requirements and functionalities, validate the system's performance and usability, and fix any bugs or defects.
5. **Deployment:** In this phase, the project team would roll out the system to the school or college and provide training and support to users. This would include installing the system, providing the necessary support and training for the users, and monitoring the system's performance.

6. **Maintenance:** In this phase, the project team would monitor the system's performance and make necessary updates, fix defects, and implement changes to ensure the system continues to meet the needs of the school or college.

It's important to note that the Waterfall model is a linear and sequential approach, where each phase must be completed before moving on to the next one, making it inflexible to change and it may not be the best fit for all projects, especially if the requirements change during the project.

Chapter Five: System Analysis & Design, Technology Study

5.1 System Analysis and Design

The system analysis and design process are a systematic approach for investigating a business or organizational problem, determining the requirements for a solution, and creating a detailed design for that solution. The process typically includes the following steps:

1. **Problem definition and scope:** Identify the problem to be solved and define the boundaries of the system.
2. **Feasibility study:** Determine if the proposed solution is viable and if the required resources are available.
3. **Requirements gathering:** Identify and document the specific requirements for the solution, including functional and non-functional requirements.
4. **Design:** Create a detailed design for the solution, including the overall architecture, data models, and user interface.
5. **Implementation:** Build and test the solution, including any necessary software development and data migration.
6. **Deployment:** Install and deploy the solution in the production environment.
7. **Maintenance:** Monitor and maintain the system to ensure it continues to meet the needs of the business.

The system analysis and design process are iterative, meaning that the process is repeated multiple times in order to refine the solution and ensure it meets the requirements.

5.2 System Specifications

A school/college management system typically requires the following system requirements:

For Development:

1. **Processor:** Intel/AMD/M Series with 3GHz or better
2. **RAM:** 8GB or better
3. **Storage:** 250GB or better
4. **Operating System:** 64-bit versions of Windows 8 or better, MacOS 10.14 or better, Any Linux that supports GNOME, KDE or Unity DE.
5. **Database:** SQLite compatible with OS version.

6. **Web Server:** Any latest versions of Apache or IIS.
7. **Programming Language:** Python 3 or better
8. **Monitor:** 1024 x 768 or better display
9. **Peripherals:** Standard Keyboard & Mouse
10. **Browser:** Latest version of Chrome, Edge, Opera, Firefox, etc.
11. **Network:** Minimum 5mbps

For Utilization:

1. **Processor:** Intel/AMD/M Series with 3GHz or better
2. **RAM:** 4GB or better
3. **Storage:** 250GB or better, HDD or SSD
4. **Operating System:** 32/64-bit versions of Windows 8 or better, MacOS 10.14 or better, Any Linux that supports GNOME, KDE or Unity DE.
5. **Monitor:** 1024 x 768 or better display
6. **Peripherals:** Standard Keyboard & Mouse
7. **Browser:** Latest version of Chrome, Edge, Opera, Firefox, etc.
8. **Network:** Minimum 2mbps

It's important to note that these requirements may vary depending on the specific school management system and the features it includes. It's always best to check with the vendor or developer for the specific system requirements.

5.3 Technology Study

Frontend Technologies

Frontend technologies are used to create the user interface of a web application, such as a school/college management system. These technologies include:

1. **HTML (Hypertext Markup Language):** HTML is used to create the structure and layout of web pages. It can be used to create pages for displaying student information, adding new students, and updating grades. HTML tags are used to define headings, paragraphs, lists, tables and other elements that make up the structure of the page.

2. **CSS (Cascading Style Sheets):** CSS is used to control the appearance of elements on a web page. It can be used to define the font, color, size, and other visual properties of elements. In the context of a school/college management system, CSS can be used to create a consistent look and feel across all pages, ensuring that the system is visually appealing and easy to use.
3. **JavaScript:** JavaScript is a programming language that is used to create dynamic effects on web pages. It can be used to validate user input, update the contents of a web page without refreshing the entire page, and create interactive elements such as drop-down menus and pop-ups. In the context of a school/college management system, JavaScript can be used to add interactivity to the system, making it more user-friendly and efficient.
4. **JavaScript Libraries and Frameworks:** JavaScript libraries such as React, Angular and Vue.js are widely used to build complex and interactive user interface. These libraries provide a set of pre-built components and hooks that can be reused to build the user interface faster.
5. **Responsive Design:** Responsive design techniques are used to ensure that the user interface adapts to different screen sizes and resolutions. This is important because users may access the system from a variety of devices, including desktops, laptops, tablets, and smartphones.
6. **UI/UX Design:** The user interface and user experience design of the system are important to ensure that the system is intuitive, easy to use and visually appealing.

These technologies are used together to create a responsive and interactive web interface of the school/college management system that will be used by administrators, teachers, and students.

Backend Technologies

Backend technologies are used to create the server-side functionality of a web application, such as a school/college management system. These technologies include:

1. **Django:** Django is a web framework for building web applications using the Python programming language. It provides a high-level set of tools for building web applications, including a built-in admin interface, an Object-Relational Mapper (ORM) for database management, and a built-in development server. In the context of a school/college management system, Django can be used to handle user authentication and authorization, manage the database and handle the routing of requests from the frontend to the appropriate backend functionality.
2. **Python:** Python is a general-purpose programming language that is widely used for web development. It is known for its simplicity, readability and its vast number of libraries that are available for different use cases. In the context of a school/college management system, Python can be used to implement the backend logic of the system, such as calculating grades,

generating reports, and communicating with other systems.

3. **SQLite**: SQLite is a lightweight, file-based relational database management system. It can be embedded in other applications, making it well-suited for small to medium-sized applications like a school/college management system. SQLite can be used to store and retrieve data related to students, teachers, classes, grades, and other information needed for the management of the school/college system.
4. **ORM (Object-Relational Mapping)**: An ORM is used to map the database table to the python class. Django has its own ORM which can be used to connect to the SQLite database and perform CRUD operations on the tables.

These technologies work together to create a robust and efficient backend for the school/college management system. Django and Python handle the server-side logic, while SQLite is used to store and manage the data. The ORM provide an easy way to interact with the database.

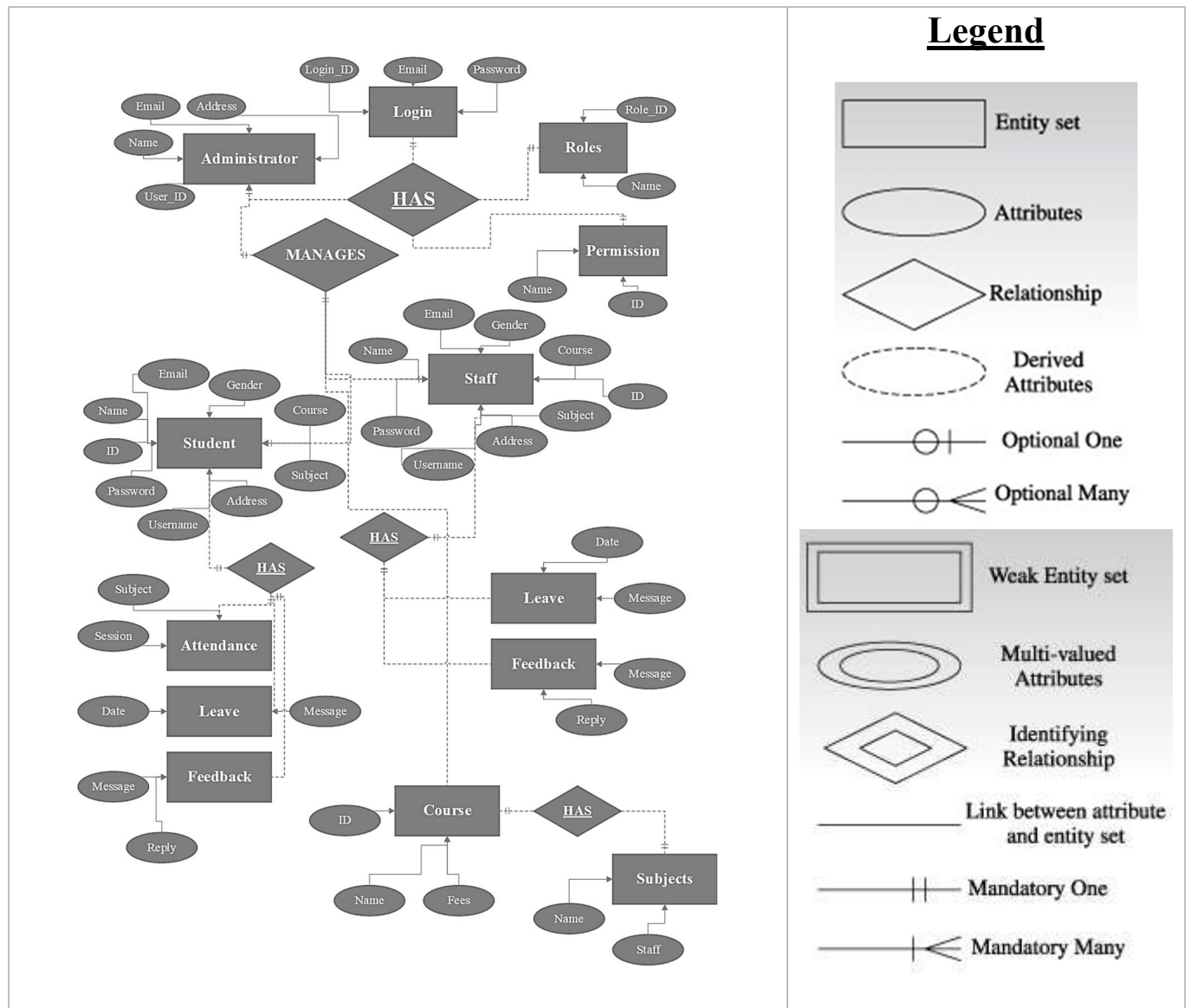
5.4 ER Diagram & DFD Diagram

An ER (entity-relationship) diagram is a graphical representation of entities and their relationships to one another. An ER diagram for a school/college management system could include the following entities:

1. **Student**: An individual who is enrolled in the school or college.
2. **Faculty**: An individual who teaches or works at the school or college.
3. **Course**: A class or subject offered at the school or college.
4. **Department**: A group of faculty and courses that share a common field of study.
5. **Enrollment**: A relationship between a student and a course, indicating that the student is enrolled in the course.
6. **Teaches**: A relationship between a faculty member and a course, indicating that the faculty member teaches the course.
7. **Belongs**: A relationship between a course and a department, indicating that the course belongs to the department.
8. **Major**: A relationship between a student and a department, indicating that the student is majoring in the department.
9. **Grade**: A relationship between a student, a course, and a grade, indicating the student's grade in the course.

These entities and relationships can be represented in an ER diagram using rectangles (for entities) and diamonds (for relationships), with lines connecting the entities and relationships. The diagram can also include attributes for each entity, such as student name, faculty salary, and course credit hours, represented as oval shapes.

It's important to note that this is a general example and the actual entities and relationships will depend on the specific requirements of the system. Following is the entity diagram for a school/college management system,







A Data Flow Diagram (DFD) is a graphical representation of the flow of data in a system, such as a school/college management system. It is used to visualize and understand the flow of data through the system, including the inputs, processes, and outputs. A DFD is composed of four main elements:

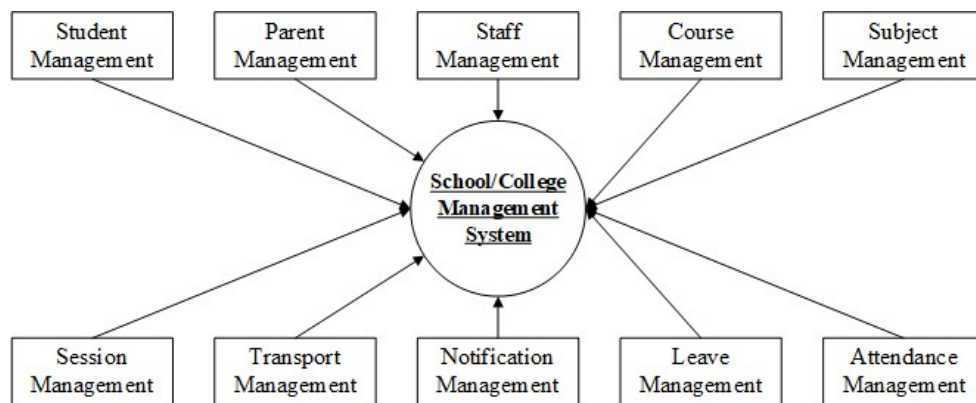
1. **External entities:** These are external systems or users that interact with the school/college management system. Examples of external entities in a school/college management system include students, teachers, parents, and administrators.
2. **Processes:** These are the activities or tasks that are performed on the data as it flows through the system. Examples of processes in a school/college management system include adding students, updating grades, generating reports, and managing class schedules.
3. **Data stores:** These are the locations where data is stored and accessed within the system. Examples of data stores in a school/college management system include student records, grade records, and class schedules.
4. **Data flows:** These are the arrows that connect the external entities, processes, and data stores. They represent the flow of data through the system. For example, a student's information may flow from an external entity to a process that adds the student to the system, and then to a data store where the student's information is stored.

In a typical school/college management system, the DFD would show how data flows from external entities, such as students and teachers, into the system where it is processed by various processes such as enrolling students, updating grades, generating reports and managing class schedule. The processed data is then stored in databases like SQLite and can be accessed by the authorized external entities.

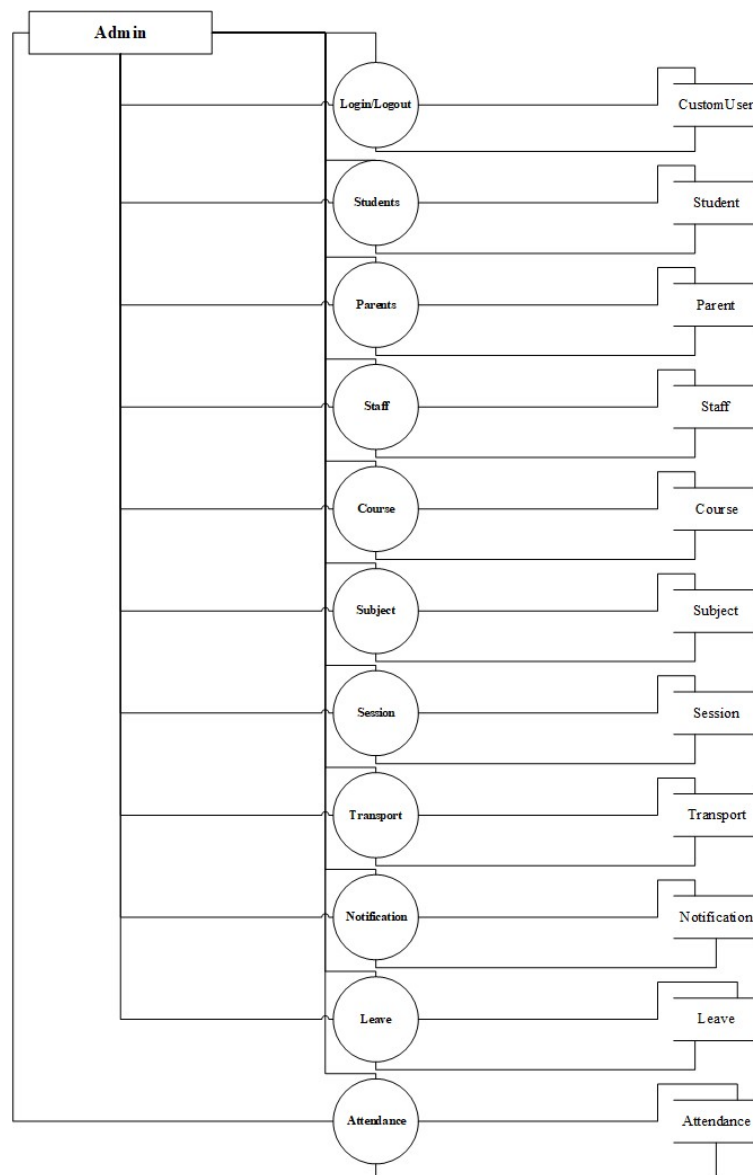
It's worth noting that DFDs come in different levels of details and different level DFDs will provide different level of information about the system. Below is the legend for Data Flow Diagrams and a system suitable Zeroth level DFD and a First Level DFD (Admin) is showcased.

	dataflow	Arrows showing direction of flow
	process	circles
	file	horizontal pair of lines
	data-source, sink	rectangular box

Zeroth Level DFD



First Level DFD (Admin)



5.5 Database Design

The database design for a school/college management system defines the structure of the data that is stored and how it is organized within the system. It includes the definition of tables, relationships between tables, and the data types of the columns within the tables.

Typically, a school/college management system requires several tables to store information about students, teachers, courses, classes, and grades. Some of the tables and their relationships are as follows:

1. Attendance

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	attendance_Date	date	Not Null
3	created At	datetime	Not Null
4	updated At	datetime	Not Null
5	session_year_id	integer	FK, Not Null
6	subject_id	integer	FK, Not Null

2. Attendance Report

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	status	bool	Not Null
3	created_at	datetime	Not Null
4	updated_at	datetime	Not Null
5	attendance_id	integer	FK, Not Null
6	student_id	integer	FK, Not Null

3. Courses

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	course_name	varchar(255)	Not Null
3	course_fees	varchar(255)	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null

4. Custom User

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	password	varchar(128)	Not Null
3	last_login	datetime	
4	is_superuser	bool	Not Null
5	username	varchar(150)	UK, Not Null
6	first_name	varchar(150)	Not Null
7	last_name	varchar(150)	Not Null
8	email	varchar(254)	Not Null
9	is_staff	bool	Not Null
10	is_active	bool	Not Null
11	date_joined	datetime	Not Null
12	user_type	varchar(10)	Not Null
13	profile_pic	varchar(100)	Not Null

5. Feedback Staffs

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	feedback	text	Not Null
3	feedback_reply	text	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	staff_id	integer	FK, Not Null

6. Feedback Students

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	feedback	text	Not Null
3	feedback_reply	text	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	student_id	integer	FK, Not Null

7. Leave Report Staff

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	leave_date	varchar(255)	Not Null
3	leave_message	text	Not Null
4	leave_status	integer	Not Null
5	created_at	datetime	Not Null
6	updated_at	datetime	Not Null
7	staff_id	integer	FK, Not Null

8. Leave Report Student

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	leave_date	varchar(255)	Not Null
3	leave_message	text	Not Null
4	leave_status	integer	Not Null
5	created_at	datetime	Not Null
6	updated_at	datetime	Not Null
7	student_id	integer	FK, Not Null

9. Notification Staffs

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	message	text	Not Null
3	created_at	datetime	Not Null
4	updated_at	datetime	Not Null
5	status	integer	
6	staff_id	integer	FK, Not Null

10. Notification Students

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	message	text	Not Null
3	created_at	datetime	Not Null
4	updated_at	datetime	Not Null
5	status	integer	
6	student_id	integer	FK, Not Null

11. Session Year

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	session_year_year	date	Not Null
3	session_end_year	date	Not Null

12. Staffs

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	gender	varchar(100)	Not Null
3	address	text	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	admin_id	bigint	FK, UK, Not Null
7	course_id	integer	FK

13. Student Fee

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	fee_amount	integer	Not Null
3	fee_type	varchar(100)	Not Null
4	fee_ref	varchar(100)	Not Null
5	fee_stat	integer	Not Null
6	created_at	datetime	Not Null
7	updated_at	datetime	Not Null
8	stud_id	integer	FK, UK Not Null

14. Student Result

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	subject_exam_marks	real	Not Null
3	subject_assignment_marks	real	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	student_id	integer	FK, Not Null
7	subject_id	integer	FK, Not Null

15. Students

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	gender	varchar(50)	Not Null
3	address	text	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	admin_id	bigint	FK, UK, Not Null
7	course_id	integer	FK, Not Null
8	session_year_id	integer	FK, Not Null

16. Subjects

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	subject_name	varchar(255)	Not Null
3	subject_code	varchar(255)	Not Null
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	course_id	integer	FK, Not Null
7	staff_id	integer	FK, Not Null

17. Transport

ID	Field	Datatype	Constraints
1	id	integer	PK, Not Null
2	address	text	Not Null
3	gender	varchar(100)	Not Null
4	cell_no	integer	Not Null
5	vehicle_no	varchar(100)	Not Null
6	vehicle_type	varchar(100)	Not Null
7	aadhar	varchar(100)	Not Null
8	dl	varchar(100)	Not Null
9	rc	varchar(100)	Not Null
10	puc	varchar(100)	Not Null
11	created_at	datetime	Not Null
12	updated_at	datetime	Not Null
13	admin_id	bigint	

Chapter Six: Implementation & Testing

6.1 Implementation of a School/College Management System

Implementation of a school/college management system involves several steps, including:

1. **Requirements gathering:** The first step is to gather requirements from the stakeholders, including administrators, teachers, students, and parents. The requirements gathering process helps to identify the key features and functionalities that are needed in the system.
2. **Design:** The next step is to design the system, including the user interface, database structure, and data flow. This stage involves creating diagrams, such as data flow diagrams (DFD), to visualize the system architecture and ensure that the design meets the requirements.
3. **Development:** In this stage, the system is developed using programming languages such as Python, HTML, CSS, and JavaScript. The system is built using a framework such as Django, which provides a high-level structure for the application and helps to reduce the development time.
4. **Testing:** Once the system is developed, it is tested to ensure that it works as expected. This involves conducting functional and non-functional testing to identify any bugs or issues in the system.
5. **Deployment:** After testing is completed, the system is deployed and made available to the stakeholders. This stage may involve setting up the server infrastructure, installing the software, and configuring the system for the specific needs of the school/college.
6. **Maintenance:** Once the system is deployed, it requires ongoing maintenance and support to ensure that it continues to function properly. This includes fixing bugs, adding new features, and ensuring that the system remains secure.

Implementing a school/college management system can be a complex process, but it can help to streamline administrative tasks and improve communication between stakeholders. With the right implementation approach, the system can be customized to meet the specific needs of the school/college and provide a robust and user-friendly solution.

6.2 Implementation of Programming Languages

Frontend Languages (HTML/CSS/JS):

HTML, CSS, and JavaScript are commonly used front-end technologies for implementing the user interface of a school/college management system.

1. HTML (Hypertext Markup Language) is used to structure the content and layout of the web pages within the system. HTML provides a set of predefined elements, such as headings, paragraphs, and links, that can be used to define the content of the pages.
2. CSS (Cascading Style Sheets) is used to style and format the HTML elements, such as changing the font, color, and layout of the pages. CSS provides a way to separate the content and presentation of the pages, making it easier to maintain and update the design of the system.
3. JavaScript is used to add interactivity and dynamic behavior to the pages. JavaScript can be used to validate user inputs, retrieve data from the server, and update the page content without having to refresh the page.

The implementation of HTML, CSS, and JavaScript in a school/college management system typically involves creating a set of templates that define the basic structure of the pages, and using CSS and JavaScript to enhance the look and behavior of the pages. The templates and styles can be reused across multiple pages, reducing the time and effort required to implement the user interface.

In addition, popular front-end frameworks, such as React, Angular, or Vue.js, can also be used to implement the user interface of a school/college management system. These frameworks provide a set of tools and components that can be used to build a responsive and dynamic user interface, reducing the development time and improving the user experience.

Backend Languages (Django, Python, SQLite)

Django, Python, and SQLite are commonly used technologies for implementing the backend of a school/college management system.

1. Django is a high-level web framework for Python that provides a structure for developing web applications. Django provides built-in features for handling user authentication, URL routing, database access, and more, making it easier to implement the backend of a school/college management system.
2. Python is a high-level programming language that is used to develop the backend logic and perform data processing. Python is known for its readability, ease of use, and large libraries for data science and web development, making it an ideal language for building the backend of a school/college management system.
3. SQLite is a lightweight, file-based database management system that is often used for small-scale applications. SQLite is easy to install and use, and provides a SQL interface for accessing and manipulating data. SQLite can be used to store information such as student records, grades, schedules, and more.

The implementation of Django, Python, and SQLite in a school/college management system typically involves defining the data models and the business logic using Django models, views, and templates. The database is created using the SQLite database management system and can be interacted with using the Django ORM (Object-Relational Mapping) which provides a Python interface for accessing the data.

The Django framework also provides features for handling HTTP requests, validating user inputs, and generating HTML pages, making it easier to implement the backend of a school/college management system. The combination of Django, Python, and SQLite provides a powerful and scalable solution for building the backend of a school/college management system.

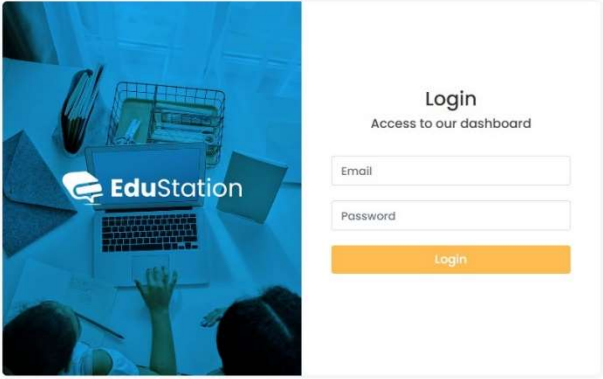
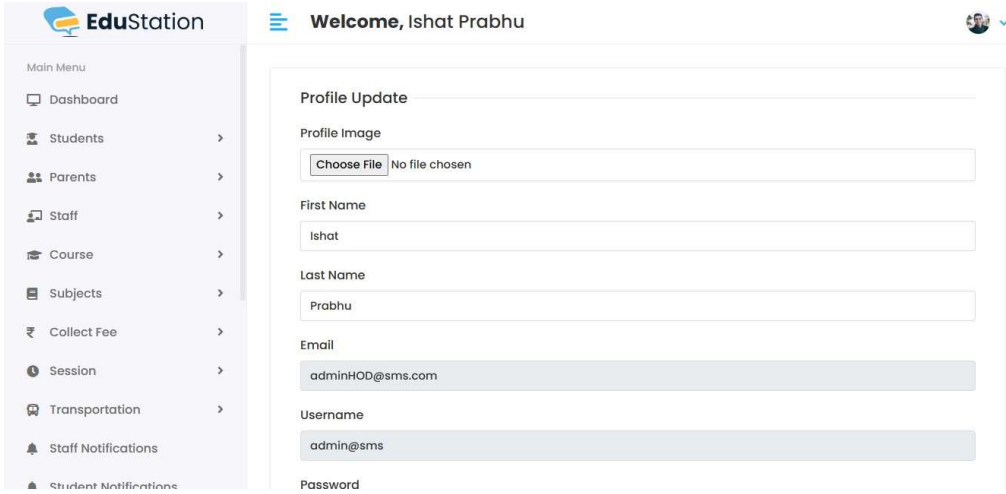
6.3 Test Cases (Admin)

#	Case Name	Case Description	Actual Output	Test Result
1	Login Screen	A login screen which authenticates user to the system.	A page with graphics and a login form for login.	PASS
2	Dashboard	Home page of the admin user which has overview of entire system.	A page with multiple graphics like chart depicting the data of the system.	PASS
3	Student	A submenu containing pages to add/view student.	Pages containing forms and table that relate to the student data.	PASS
4	Staff	A submenu containing pages to add/view staff.	Pages containing forms and table that relate to the parent data.	PASS
5	Course	A submenu containing pages to add/view courses.	Pages containing forms and table that relate to the course data.	PASS
6	Subject	A submenu containing pages to add/view subjects	Pages containing forms and table that relate to the subject data.	PASS
7	Collect Fee	A submenu containing pages to add/view student fees.	Pages containing forms and table that relate to the fees data.	PASS
8	Session	A submenu containing pages to add/view session.	Pages containing forms and table that relate to the session data.	PASS

9	Transport	A submenu containing pages to add/view drivers.	Pages containing forms and table that relate to the driver data.	PASS
10	Staff Notification	A page that shows sent notifications and list of staff for sending notifications.	A table and a modal box to display valid data	PASS
11	Student Notification	A page that shows sent notifications and list of students for sending notifications.	A table and a modal box to display valid data	PASS
12	View Attendance	A page for admin to view attendance of a student.	A form to choose and filter the valid data.	PASS
13	Staff Feedback	A page that shows sent feedback replies and list of staff for sending and reading feedbacks.	A table and a modal box to display valid data	PASS
14	Student Feedback	A page that shows sent feedback replies and list of students for sending and reading feedbacks.	A table and a modal box to display valid data	PASS
15	Staff Leave	A page that shows the list of staff who have sent a leave request.	A table and buttons to approve or reject leaves.	PASS
16	Student Leave	A page that shows the list of staff who have sent a leave request.	A table and buttons to approve or reject leaves.	PASS
17	Logout	A link to logout from the system.	HREF Redirect to login page.	PASS

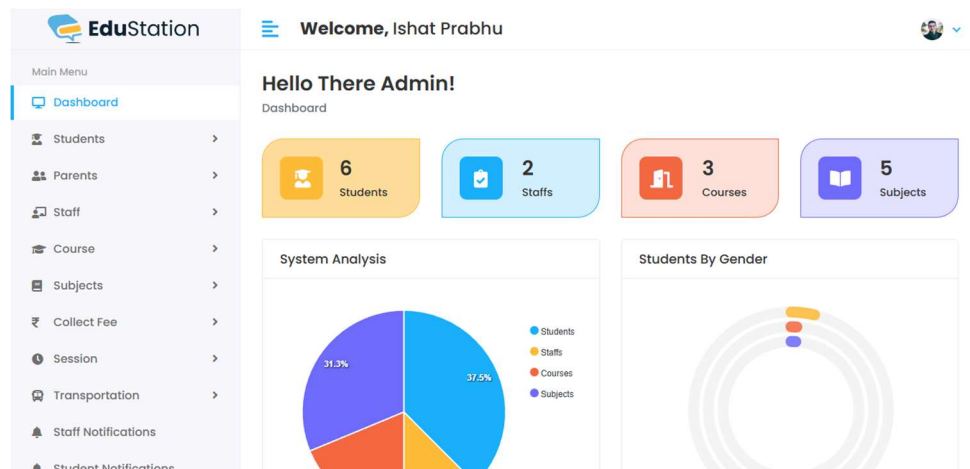
Chapter Seven: System Results

7.1 General

Login Page	
Update Profile	

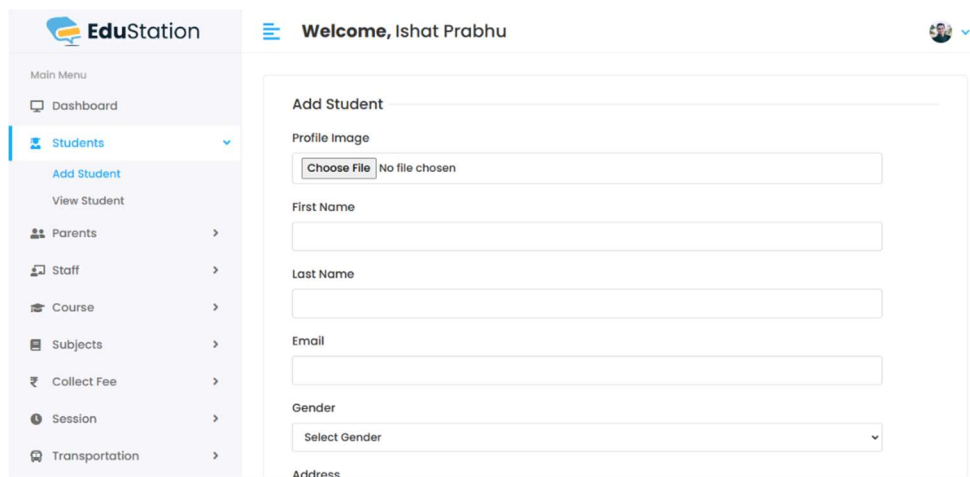
7.2 Administrator Module

Login Page



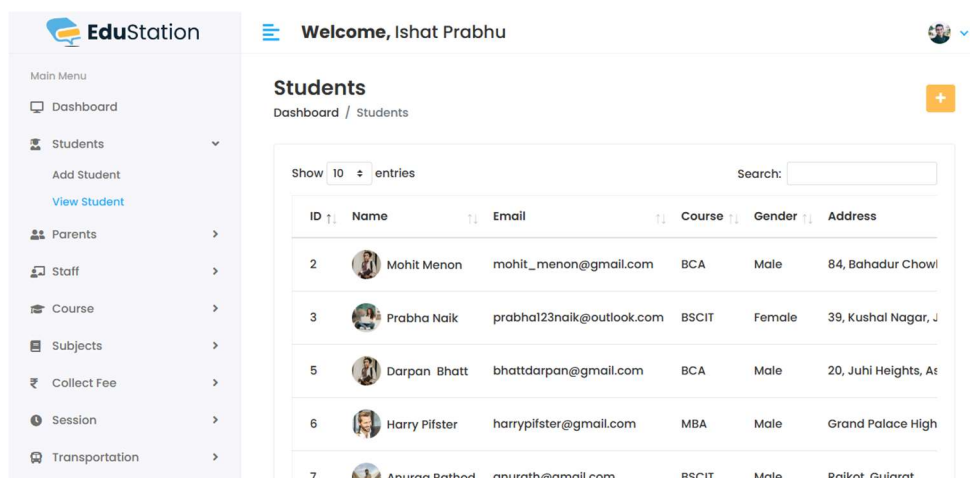
The dashboard shows a sidebar menu with options: Dashboard, Students, Parents, Staff, Course, Subjects, Collect Fee, Session, Transportation, Staff Notifications, and Student Notifications. The main content area displays a welcome message, a 'Hello There Admin!' greeting, and a summary of system statistics: 6 Students, 2 Staffs, 3 Courses, and 5 Subjects. Below this, there are two charts: 'System Analysis' (a pie chart showing 31.3% for Students, 37.5% for Staffs, and 31.2% for Courses) and 'Students By Gender' (a donut chart showing a distribution of genders).

Student > Add Student



The 'Add Student' form includes fields for Profile Image (with a 'Choose File' button), First Name, Last Name, Email, Gender (a dropdown menu), and Address. The sidebar menu is updated to show 'Add Student' and 'View Student' under the 'Students' category.

Student > View Student



The 'Students' page displays a table of student records. The table has columns for ID, Name, Email, Course, Gender, and Address. The sidebar menu is updated to show 'View Student' under the 'Students' category.

ID	Name	Email	Course	Gender	Address
2	Mohit Menon	mohit_menon@gmail.com	BCA	Male	84, Bahadur Chowk
3	Prabha Naik	prabha123naik@outlook.com	BSCIT	Female	39, Kushal Nagar, J
5	Darpan Bhatt	bhattdarpan@gmail.com	BCA	Male	20, Juhl Heights, As
6	Harry Pifster	harrypifster@gmail.com	MBA	Male	Grand Palace High
7	Anurag Rathod	anurath@gmail.com	BSCIT	Male	Rajkot, Gujarat

Staff > Add Staff

Main Menu

- Dashboard
- Students
- Parents
- Staff**
 - Add Staff
 - View Staff
- Course
- Subjects
- Collect Fee
- Session
- Transportation

Welcome, Ishat Prabhu

Add Staff

Profile Image

No file chosen

First Name

Last Name

Email

Gender

Select Gender

Address

Staff > View Staff

Main Menu

- Dashboard
- Students
- Parents
- Staff**
 - Add Staff
 - View Staff
- Course
- Subjects
- Collect Fee
- Session
- Transportation

Welcome, Ishat Prabhu

Staffs

Dashboard / Staff

Show 10 entries Search:

ID	Name	Email	Gender	Address
3	Kate McReary	katemcreary@gmail.com	Female	Thomson Street, Central Par
4	Praveen Mehotra	mehotra.praveen@gmail.com	Male	45, Mohit-Gunj, Surat

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Course > Add Course

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course**
- Subjects
- Collect Fee
- Session
- Transportation
- Staff Notifications
- Student Notifications

Welcome, Ishat Prabhu

Add Course

Course Name

Course Fees

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Course > View Course

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course**
 - Add Course
 - View Course**
- Subjects
- Collect Fee
- Session
- Transportation

Welcome, Ishat Prabhu

Course

Dashboard / Course

Show 10 entries
Search:

ID	Name	Fees	Created On	Updated On	Action
1	BCA	650000	Jan. 7, 2023, 4:35 p.m.	Jan. 7, 2023, 4:35 p.m.	Edit Delete
2	BSCIT	700000	Jan. 7, 2023, 4:36 p.m.	Jan. 7, 2023, 4:36 p.m.	Edit Delete
4	MBA	1200000	Jan. 11, 2023, 7:33 a.m.	Jan. 11, 2023, 7:33 a.m.	Edit Delete

Showing 1 to 3 of 3 entries

Previous 1 Next

Subject > Add Subject

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects**
 - Add Subject**
 - View Subject
- Collect Fee
- Session
- Transportation

Welcome, Ishat Prabhu

Add Subject

Subject / Add Subject

Subject Information

Subject Name
Subject Code

Course
Staff

Select Course
Assign Staff

Add Subject

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Subject > View Subject

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects**
 - Add Subject
 - View Subject**
- Collect Fee
- Session
- Transportation

Welcome, Ishat Prabhu

View Subjects

Dashboard / Subject

Show 10 entries
Search:

ID	Name	Code	Assigned Course	Assigned Staff	Created O
2	Environmental Engineering	BCA-001	BCA	Kate McReary	Jan. 10, 2023
3	Java Programming	BSCIT-001	BSCIT	Kate McReary	Jan. 11, 2023
5	Programming in ASP.NET	BCA-002	BCA	Kate McReary	Jan. 11, 2023
6	Financial Management	MBA-001	MBA	Praveen Mehotra	Jan. 11, 2023
7	Human Resource Management	MBA-002	MBA	Praveen Mehotra	Jan. 11, 2023

Fees > Add Fees

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects
- Collect Fee**
 - Add Fees
 - View Paid
- Session
- Transportation

Welcome, Ishat Prabhu

Collect Fees

Student / Collect Fees

Collect Fees

Student

Select Student

Fee Amount

Payment Method

Select Type

Reference Number

Collect Fees

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Fees > View Paid

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects
- Collect Fee**
 - Add Fees
 - View Paid**
- Session
- Transportation

Welcome, Ishat Prabhu

Students

Dashboard / Paid Fees

Show 10 entries

Search:

ID	Name	Email	Course	Course Fee	Session
1	Anurag Rathod	anurath@gmail.com	BSCIT	700000	Jan. 1, 2023 - March 15, 20

Showing 1 to 1 of 1 entries

Previous

1

Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Session > Add Session

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects
- Collect Fee
- Session**
 - Add Session**
 - View Session
- Transportation

Welcome, Ishat Prabhu

Add Session

Session / Add Session

Session Information

Session Start Date

dd-mm-yyyy

Session End Date

dd-mm-yyyy

Add Session

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Session > View Session

EduStation

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects
- Collect Fee
- Session**
 - Add Session
 - View Session**
- Transportation

Welcome, Ishat Prabhu

View Session

Dashboard / Session

Show 10 entries
Search:

ID	Start Date	End Date	Action
1	Jan. 1, 2023	March 15, 2026	Edit Delete
2	Jan. 1, 2023	April 15, 2026	Edit Delete

Showing 1 to 2 of 2 entries
[Previous](#) [Next](#)

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Transport > Add Driver

EduStation

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects
- Collect Fee
- Session
- Transportation**
 - Add Driver**
 - View Drivers

Welcome, Ishat Prabhu

Add Driver Information

Profile Image

[Choose File](#) No file chosen

First Name

Last Name

Email

Phone Number

Gender

Transport > View Drivers

EduStation

Main Menu

- Dashboard
- Students
- Parents
- Staff
- Course
- Subjects
- Collect Fee
- Session
- Transportation**
 - Add Driver
 - View Drivers**

Welcome, Ishat Prabhu

Transport

Dashboard / Transport

Show 10 entries
Search:

ID	Name	Email	Contact	Gender	Address
1	Mahesh Soni	maheshsoni@gmail.com	9510258753	Male	Parvati Nagar, Ambuja C

Showing 1 to 1 of 1 entries
[Previous](#) [Next](#)

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Staff Notifications

[Dashboard](#)
[Students](#)
[Parents](#)
[Staff](#)
[Course](#)
[Subjects](#)
[Collect Fee](#)
[Session](#)
[Transportation](#)
[Staff Notifications](#)
[Student Notifications](#)

Welcome, Ishat Prabhu

Staffs

Dashboard / Notification

View Sent Notifications

Show 10 entries

ID	Name	Email	Action
3	Kate McReary	katemcreary@gmail.com	Send Notification
4	Praveen Mehotra	mehotra.praveen@gmail.com	Send Notification

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Student Notifications

[Transportation](#)
[Staff Notifications](#)
[Student Notifications](#)
[Parent Notifications](#)
[View Attendance](#)
[Staff Feedback](#)
[Student Feedback](#)
[Parent Feedback](#)
[Staff Leave](#)
[Student Leave](#)
[Logout](#)

Welcome, Ishat Prabhu

Student

Dashboard / Notification

View Sent Notifications

Show 10 entries

ID	Name	Email	Action
2	Mohit Menon	mohit_menon@gmail.com	Send Notification
3	Prabha Naik	prabha23naik@outlook.com	Send Notification
5	Darpan Bhatt	bhattdarpan@gmail.com	Send Notification
6	Harry Pilster	harrypilster@gmail.com	Send Notification
7	Anurath Rathod	anurath@gmail.com	Send Notification

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Student Attendances

[Transportation](#)
[Staff Notifications](#)
[Student Notifications](#)
[Parent Notifications](#)
[View Attendance](#)
[Staff Feedback](#)
[Student Feedback](#)
[Parent Feedback](#)
[Staff Leave](#)
[Student Leave](#)
[Logout](#)

Welcome, Ishat Prabhu

Attendance

Admin / View Attendance

View Attendance

Subject Select Subject Session Year Assign Staff

Attendance Date dd-mm-yyyy

Show Attendance

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Staff Feedback

- Transportation
- Staff Notifications
- Student Notifications
- Parent Notifications
- View Attendance
- Staff Feedback**
- Student Feedback
- Parent Feedback
- Staff Leave
- Student Leave
- Logout

Welcome, Ishat Prabhu

Feedback

Dashboard / Staff

Feedback Replies

Show 10 entries

Search:

ID	Name	Feedback	Date	Action
1	Kate McReary	A better system than previous one!	Jan. 9, 2023, 1 p.m.	Replied

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Student Feedback

- Transportation
- Staff Notifications
- Student Notifications
- Parent Notifications
- View Attendance
- Staff Feedback
- Student Feedback**
- Parent Feedback
- Staff Leave
- Student Leave
- Logout

Welcome, Ishat Prabhu

Feedback

Dashboard / Student

Feedback Replies

Show 10 entries

Search:

ID	Name	Feedback	Date	Action
1	Mohit Menon	This System is good	Jan. 9, 2023, 4:04 p.m.	Replied

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Staff Leave

- Transportation
- Staff Notifications
- Student Notifications
- Parent Notifications
- View Attendance
- Staff Feedback
- Student Feedback
- Parent Feedback
- Staff Leave**
- Student Leave
- Logout

Welcome, Ishat Prabhu

Staff Leave

Dashboard / Staff Leave

Show 10 entries

Search:

ID	Name	Date	Reason	Action
1	Kate	2023-01-10	Leaving for a teacher's seminar	Approved

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Student Leave

EduStation Welcome, Ishat Prabhu

Student Leave
Dashboard / Student Leave

Show 10 entries Search:

ID	Name	Date	Reason
1	Mohit	2023-01-13	Going to cousin's wedding
2	Mohit	2023-01-24	Visiting Regional Science Center with family, kindly approve my request.

Showing 1 to 2 of 2 entries Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

7.3 Staff Module

Dashboard

EduStation Welcome, Kate McReary

Hello There Staff!
Dashboard

5 Students Under Me 8 Attendance Taken 1 Leaves Taken 3 Total Subjects

Total Attendance Data

Attendance by Subjects

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Take Attendance

EduStation Welcome, Kate McReary

Attendance
Staff / Take Attendance

Take Attendance


Subject Session Year

Select Subject Assign Staff

Fetch Student(s)

Copyright © 2023 EduStation. Developed by Anshu Joshi.

View Attendance



Main Menu

- Dashboard
- Take Attendance
- View Attendance**
- Add Result
- Apply Leave
- Notifications
- Feedback
- Logout

Welcome, Kate McReary

Attendance

Staff / View Attendance

View Attendance

Subject

Select Subject

Session Year

Assign Staff


Attendance Date

dd-mm-yyyy

Show Attendance

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Add Result



Main Menu

- Dashboard
- Take Attendance
- View Attendance
- Add Result**
- Apply Leave
- Notifications
- Feedback
- Logout

Welcome, Kate McReary

Add Result

Dashboard / Add Result

Add Result

Subject

-- Select Subject --


Session Year

-- Select Session Year --

Fetch Student

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Apply Leave



Main Menu

- Dashboard
- Take Attendance
- View Attendance
- Add Result
- Apply Leave**
- Notifications
- Feedback
- Logout

Welcome, Kate McReary

Leave Application

Staff / Apply Leave

View Applications

Apply For Leave


Leave Date

dd-mm-yyyy

Leave Reason

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Notification

EduStation

Main Menu

- Dashboard
- Take Attendance
- View Attendance
- Add Result
- Apply Leave
- Notifications**
- Feedback
- Logout

Welcome, Kate McReary

Notification

Dashboard / Notifications


Show 10 entriesSearch:

Message	Action
Welcome to SMS	Already Read

Showing 1 to 1 of 1 entriesPreviousNext

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Feedback

EduStation

Main Menu

- Dashboard
- Take Attendance
- View Attendance
- Add Result
- Apply Leave
- Notifications
- Feedback**
- Logout

Welcome, Kate McReary

Feedback

Staff / Feedback

Feedback History

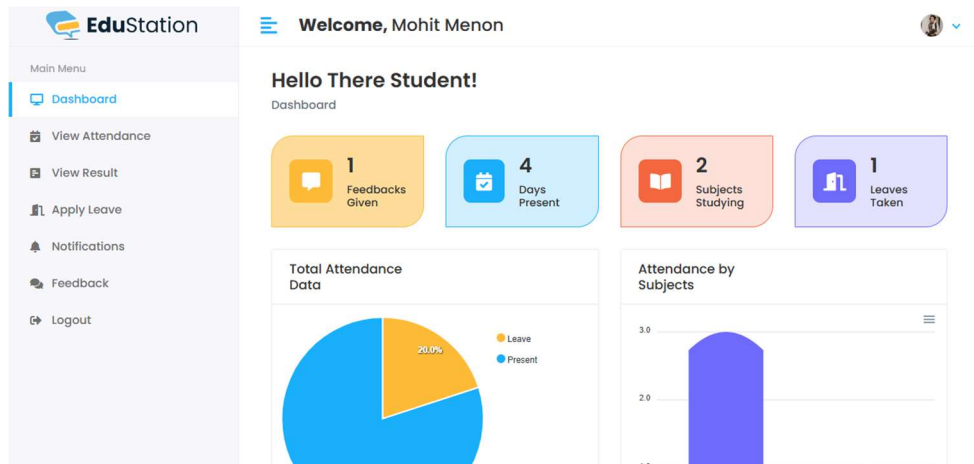
Write your feedback

Feedback

Send Feedback

7.4 Student Module

Dashboard



View Attendance


The 'View Attendance' form in the EduStation interface. It includes a sidebar menu with options: Dashboard, View Attendance, View Result, Apply Leave, Notifications, Feedback, and Logout. The main content area has a 'View Attendance' heading, a 'Subject' dropdown menu with 'Select Subject' as the placeholder, and a 'Show Attendance' button. Below the form is a copyright notice: 'Copyright © 2023 EduStation. Developed by Anshu Joshi.'

View Result

The 'View Result' page in the EduStation interface. It includes a sidebar menu with options: Dashboard, View Attendance, View Result, Apply Leave, Notifications, Feedback, and Logout. The main content area has a 'View Result' heading and a 'Student / Result' breadcrumb. Below this is a table with columns: ID, Subject, Assignment Marks, Exam Mark, and Result. The table shows one entry for 'Environmental Engineering' with an 'Assignment Marks' of 23.0 and an 'Exam Mark' of 69.0, resulting in a 'Pass' status. The table is paginated with 'Showing 1 to 1 of 1 entries' and navigation buttons for 'Previous', '1', and 'Next'. Below the table is a copyright notice: 'Copyright © 2023 EduStation. Developed by Anshu Joshi.'

ID	Subject	Assignment Marks	Exam Mark	Result
1	Environmental Engineering	23.0	69.0	Pass

Leave



Main Menu

- Dashboard
- View Attendance
- View Result
- Apply Leave**
- Notifications
- Feedback
- Logout

Welcome, Mohit Menon

Leave Application

Student / Apply Leave

View Applications


Apply For Leave

Leave Date

dd-mm-yyyy

Leave Reason

Notification



Main Menu

- Dashboard
- View Attendance
- View Result
- Apply Leave
- Notifications**
- Feedback
- Logout

Welcome, Mohit Menon

Notification

Student / Notifications

Show 10 entries

Search:


Message	Action
Welcome to the system	Already Read

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © 2023 EduStation. Developed by Anshu Joshi.

Feedback



Main Menu

- Dashboard
- View Attendance
- View Result
- Apply Leave
- Notifications
- Feedback**
- Logout

Welcome, Mohit Menon

Feedback

Student / Feedback

Feedback History

Write your feedback

Feedback

Send Feedback

Chapter Eight: Conclusion & Future Works

8.1 Summary of the project

A school/college management system project is a web application developed using the Django framework, SQLite database, and Python programming language. It allows administrators to manage various aspects of the institution, including student information, course offerings, teacher assignments, and grades. The system provides a user-friendly interface for students, teachers, and parents to access information and perform tasks such as viewing schedules, submitting assignments, and tracking progress. The project leverages the power of Python, Django, and SQLite to create a robust and efficient solution for school/college management.

8.2 Future Improvements

Here are some potential improvements that can be made in a school/college management system:

1. Integration with virtual learning platforms
2. Improved data security and privacy features
3. Enhanced reporting and analytics capabilities
4. Mobile app access for students, teachers, and parents
5. Automated fee payment and bookkeeping
6. Predictive analytics to improve student outcomes
7. Gamification elements to enhance student engagement
8. Personalized learning and progress tracking
9. Integration with social media and communication tools
10. Cloud-based architecture for better scalability and accessibility.

Chapter Nine: References

9.1 List of references

The following references and web resources were useful in the development of a school/college management system:

#	Name	URL
1	HTML & CSS	https://www.w3schools.com/html/html_css.asp
2	JavaScript	https://www.w3schools.com/js/
3	Django official documentation	https://docs.djangoproject.com/en/3.2/
4	Python official documentation	https://docs.python.org/3/
5	SQLite official documentation	https://sqlite.org/docs.html
6	Django Girls tutorial	https://tutorial.djangogirls.org/en/
7	Full Stack Python Django Guide	https://django.fullstackpython.com/
8	Real Python's Django web development with Python tutorial	https://realpython.com/django-web-development-with-python/
9	SQLite Tutorial	https://www.sqlitetutorial.net/
10	Django and SQLite3 Tutorial	https://simpleisbetterthancomplex.com/series/2017/09/11/a-complete-beginners-guide-to-django-part-2.html
11	Django REST framework tutorial	https://www.django-rest-framework.org/
12	Bootstrap for creating responsive and visually appealing user interfaces.	https://getbootstrap.com/
13	JavaScript frameworks such as jQuery	JavaScript frameworks such as jQuery (https://jquery.com/) and React (https://reactjs.org/).

Note: It is important to keep in mind the latest version of these technologies and references before starting development.

Chapter Ten: Appendices

10.1 List of abbreviations and relatable terms

ORM - Object-Relational Mapping

URL - Uniform Resource Locator

MVT - Model-View-Template

CBV - Class-Based Views

FBV - Function-Based Views

DRF - Django Rest Framework

CSRF - Cross-Site Request Forgery

DTL - Django Template Language

MIDDLEWARE - Middleware layer in Django

HTTP - Hypertext Transfer Protocol

DB - Database

AJAX - Asynchronous JavaScript and XML

SQL - Structured Query Language

API - Application Programming Interface

JS - JavaScript

HTML - Hypertext Markup Language

CSS - Cascading Style Sheets

WSGI - Web Server Gateway Interface

FTP - File Transfer Protocol

SSL - Secure Sockets Layer

HTTPS - Hypertext Transfer Protocol Secure

REST - Representational State Transfer

JSON - JavaScript Object Notation

XML - eXtensible Markup Language

DOM - Document Object Model

MVC - Model-View-Controller

VM - Virtual Machine

CRUD - Create, Read, Update, Delete

CLI - Command Line Interface

GUI - Graphical User Interface

OOP - Object-Oriented Programming

UML - Unified Modeling Language

SSH - Secure Shell

SMTP - Simple Mail Transfer Protocol

IMAP - Internet Message Access Protocol

POP - Post Office Protocol

LDAP - Lightweight Directory Access Protocol

DNS - Domain Name System

IP - Internet Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

IP address - Internet Protocol Address

VPN - Virtual Private Network

WAN - Wide Area Network

LAN - Local Area Network

MODEL - Django Model class

VIEW - Django View function

TEMPLATE - Django Template for HTML

APP - Django Application

PYTHON - Programming language

DJANGO - High-level Python Web framework

WSGI - Web Server Gateway Interface

Flask - Micro web framework for Python

Pyramid - Web framework for Python

Tkinter - GUI toolkit for Python

Pygame - Gaming framework for Python

NumPy - Numerical computing library for Python

SciPy - Scientific computing library for Python

Matplotlib - Plotting library for Python

Pandas - Data analysis and manipulation library for Python

OpenCV - Computer vision library for Python

TensorFlow - Machine learning library for Python

PyTorch - Machine learning library for Python

Keras - High-level neural networks API for Python

scikit-learn - Machine learning library for Python

NLP - Natural Language Processing

ML - Machine Learning

AI - Artificial Intelligence

DL - Deep Learning

CNN - Convolutional Neural Network

RNN - Recurrent Neural Network

LSTM - Long Short-Term Memory

GRU - Gated Recurrent Unit

Dense - Fully connected layer in neural network

Dropout - Regularization technique in neural network

Optimizer - Algorithm

ADMIN - Django Admin interface

ADMIN SITE - Django administrative interface for managing models

FORMS - Django forms for user input validation

MODELFORM - Django form for model class

URL CONF - Django URL configuration

MIGRATION - Django database migration tool

MANAGER - Django model manager

Q - Django query objects

F - Django query expressions

GIS - Django geographic information system

DJANGO CORE - Django core framework

DJANGO REST FRAMEWORK - Django toolkit for building APIs

DJANGO CMS - Django content management system

DJANGO MODEL ADMIN - Django administrative interface for managing models

PYTHON SHELL - Python REPL interface

PYTHON PACKAGE INDEX (PyPI) - Python package repository

IPYTHON - Interactive Python shell

JUPYTER - Interactive notebook for Python and other languages

IDLE - Python integrated development environment

ANACONDA - Python distribution for scientific computing and data science

VENV - Python virtual environment

ENVIRONMENT VARIABLES - System variables for Python

SETUPTOOLS - Python library for building and distributing packages

DISCOVERY - Python package for discovering plugins.