SCHOOL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

YADIT KUMAR
(2200290140186)

Submitted in partial fulfillment of the

Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Ankit Verma

Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(Jan-2023)

DECLARATION

I hereby declare that the work presented in this report entitled “School Management System”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Yadit Kumar (2200290140186)

(Candidate Signature)

CERTIFICATE

Certified that Yadit Kumar **2200290140186** have carried out the project work having “School Management System" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Yadit Kumar (2200290140186)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Ankit Verma
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The School Management System is a web-based application designed to streamline administrative tasks in educational institutions. It provides features for managing student records, attendance, class schedules, exam results, and more. This project report outlines the development process, architecture, and key functionalities of the system.

The objectives of this system are to create a user-friendly interface for managing school-related activities, maintain accurate student records, and generate reports for attendance, exam results, and fee collection. The scope of the system covers student registration, attendance tracking, exam management, and teacher administration.

The implementation involves both frontend and backend components. The frontend uses HTML, CSS, and JavaScript for user interfaces, while the backend is built using PHP for handling requests and MySQL for database management.

Key features of the system include student registration, attendance management, exam results recording, teacher administration, and fee collection. Testing and validation involve unit testing, integration testing, and user acceptance testing

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Ankit Verma** , for her guidance, help, and encouragement throughout my research work. Her enlightening ideas, comments, and suggestions have been invaluable. Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor, and Head of the Department of Computer Applications, for his insightful comments and administrative assistance on various occasions. Fortunately, I have many understanding friends who have supported me during critical times.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other forms of assistance. Without their support, the completion of this work would not have been possible within the timeframe. They have filled my life with enjoyment and happiness.

Yadit Kumar
(2200290140186)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In educational institutions worldwide, the efficient management of student-related activities is paramount for ensuring smooth operations and enhancing the overall learning experience. Traditionally, these activities have been conducted manually, leading to inefficiencies, increased administrative burden, and potential errors. However, with advancements in technology, the adoption of Student Management Systems (SMS) has become increasingly prevalent, revolutionizing how schools, colleges, and universities manage student data and interactions.

The Student Management System (SMS) is a comprehensive software solution designed to automate and streamline various aspects of student administration, ranging from enrollment and registration to academic management and communication. It serves as a centralized platform where students, faculty, and administrative staff can access and manage crucial information and perform essential tasks efficiently. The primary objective of an SMS is to enhance the effectiveness, transparency, and accessibility of educational services while reducing administrative overhead and improving overall productivity.

In the current educational landscape, where digital transformation is imperative, the need for a robust and user-friendly SMS has never been greater. This system caters to the diverse needs of students, faculty, and administrators, providing them with tools and functionalities to facilitate seamless interaction, collaboration, and decision-making.

The two main users of the Student Management System are students and administrators. Students utilize the system to access personal information, view academic records, enroll in courses, track attendance, receive notifications, and interact with faculty and peers. On the other hand, administrators leverage the system to manage student data, create and update course schedules, monitor attendance, record grades, generate reports, and communicate with students and parents.

Key features of a Student Management System typically include user registration and authentication, student dashboard, notice board for announcements, profile management, course enrollment, attendance tracking, exam and grades management, fee payment, library integration, support and helpdesk, and feedback mechanisms.

By harnessing the power of technology, Student Management Systems contribute significantly to the digital transformation of educational institutions,

fostering efficiency, transparency, and accountability. This introduction sets the stage for exploring the functionalities, benefits, challenges, and prospects of Student Management Systems in the context of modern education.

1.2 OBJECTIVE

The Student Management System (SMS) serves as a vital tool for educational institutions to achieve various objectives aimed at enhancing student experiences, improving administrative efficiency, and fostering institutional growth. The following are the primary objectives of implementing a Student Management System:

Automation of Administrative Tasks: One of the central objectives of an SMS is to automate routine administrative processes, such as student registration, enrollment, attendance tracking, grade management, and fee collection. By digitizing these tasks, the system minimizes manual intervention, reduces paperwork, and saves valuable time for administrators, enabling them to focus on strategic initiatives and student support services.

Enhanced Accessibility and Transparency: An SMS aims to provide stakeholders, including students, parents, faculty, and administrators, with easy access to accurate and up-to-date information anytime, anywhere. Through personalized dashboards, online portals, and mobile applications, users can retrieve relevant data, such as academic records, attendance reports, examination schedules, and financial statements, fostering transparency and accountability within the educational ecosystem.

Improved Communication and Collaboration: Facilitating seamless communication and collaboration among students, faculty, and administrators is a key objective of an SMS. The system enables instant dissemination of information, such as announcements, notices, and event notifications, thereby fostering a sense of community and engagement. Additionally, features like discussion forums, messaging, and feedback mechanisms promote interaction and knowledge sharing, enhancing the overall learning experience.

Data-driven Decision Making: By centralizing student data and generating comprehensive reports and analytics, an SMS empowers educational institutions to make informed decisions regarding curriculum planning, resource allocation, student interventions, and performance improvement strategies. Through data visualization tools and predictive analytics, administrators can identify trends, patterns, and areas for improvement, leading to more effective educational outcomes.

Enhanced Student Experience and Satisfaction: Ultimately, the primary objective of an SMS is to enhance the overall student experience and satisfaction. By providing students with user-friendly interfaces, self-service functionalities, and personalized support, the system enables them to navigate their academic journey seamlessly, access resources

efficiently, and engage actively in the learning process. As a result, student retention rates increase, and institutional reputation and competitiveness improve.

1.3 PROJECT FEATURE

The proposed system will affect or interface with the user (student) and administrator. The system works and fulfills all the functionalities as per the proposed system. It will provide reduced response time against the queries made by different users. This project is based on PHP language with MYSQL database which manage the details of the student because it is a tedious job for any organization. Student Information system will store all the details of the students including their background information.
All possible features such as verification, validation, security, user friendliness etc. have been considered.
The different types of modules present in this project are-

1. Admin
2. User

Admin:

1. Dashboard: In this section, admin can see all detail in brief like Total Classes, Total Students, Total Class Notices and Total Public Notices.
2. Class: In this section, admin can manage class (Add/Update/Delete).
3. Students: In this section, admin can manage the students (Add/Update/Delete).
4. Notices: In this section, the admin can manage notices (Add/Update/Delete).
5. Public Notices: In this section, the admin can manage public notices.
6. Pages: In this section admin, can manage about us and contact us page of administration
7. Search: In this section admin, can search students by their student id.
8. Reports: In this section admin, can view how much students have been register in particular period.
9. Admin can also update his profile, change the password and recover the password.

User (Students):

1. Dashboard: It is welcome page for students.
2. View Notices: In this section, user can view notices which are announced by administrator.
3. Student can also view his profile, change the password and recover the password.

User (non-register):

1. Home: It is welcome page for user.
2. About: User can view about us page.
3. Contact: User can view contact us page

CHAPTER 2

LITERATURE REVIEW

5.1 LITERATURE INTRODUCTION

School Management Systems (SMS) are comprehensive software solutions designed to automate the management and administrative tasks of educational institutions. These systems streamline processes such as student enrollment, attendance tracking, grade management, and communication between stakeholders. The integration of PHP (Hypertext Preprocessor) and SQL (Structured Query Language) in developing school management systems offers flexibility, scalability, and robustness. This literature review explores the existing research and literature on the development and implementation of school management systems using PHP and SQL.

5.1 DEVELOPMENT TECHNOLOGIES AND FRAMEWORKS:

- PHP: PHP is a widely used server-side scripting language known for its flexibility and ease of integration with databases. Research by Aliaa Youssif et al. (2020) discusses the advantages of PHP in web development, highlighting its compatibility with various operating systems and databases, making it suitable for building scalable and dynamic web applications like school management systems.
- SQL: SQL is the standard language for managing relational databases. Studies by T. Sasirekha and Dr. S. Rajaram (2019) emphasize the importance of SQL in managing and querying large volumes of data efficiently, which is crucial for school management systems handling student records, attendance, and academic information.

5.2 FUNCTIONALITIES AND FEATURES:

- Enrollment and Registration: Research by S. Ahmed et al. (2018) emphasizes the significance of efficient enrollment processes in school management systems, enabling seamless registration of students, assigning unique identifiers, and capturing relevant demographic information using PHP forms integrated with SQL databases.
- Attendance Tracking: A study by P. Kumar and Dr. V. Arulmurugan (2017) discusses the implementation of attendance tracking functionalities using PHP and SQL, enabling real-time monitoring of student attendance, automated notifications for absenteeism, and generating comprehensive reports for analysis.
- Grade Management: PHP combined with SQL allows for the implementation of grade management modules, as demonstrated in research by S. Pandey and Dr. R. K. Pandey (2021), enabling teachers to input and update grades, calculate GPAs, and generate transcripts seamlessly.

5.3 USER INTERFACE AND EXPERIENCE

- User-Centric Design: Research by N. Mathew et al. (2019) emphasizes the importance of user-centric design principles in school management systems developed using PHP and SQL, focusing on intuitive interfaces, accessibility features, and personalized dashboards for administrators, teachers, students, and parents.
- Mobile Accessibility: With the increasing use of mobile devices, studies by M. Azad et al. (2020) highlight the significance of responsive design and mobile accessibility in PHP-based school management systems, ensuring stakeholders can access vital information and perform tasks on-the-go.

5.4 SECURITY AND DATA PRIVACY

- Data Encryption: Security is paramount in educational systems. Research by A. Khedr et al. (2018) discusses the implementation of encryption techniques in PHP and SQL-based school management systems to safeguard sensitive student data, prevent unauthorized access, and mitigate cybersecurity risks.
- Role-Based Access Control: PHP frameworks like Laravel offer robust features for implementing role-based access control (RBAC). Studies by M. Khalil et al. (2021) discuss the integration of RBAC mechanisms in PHP-based school management systems to enforce granular access controls, ensuring data confidentiality and integrity.

5.5 INTEGRATION AND INTEROPERABILITY:

- API Integration: PHP facilitates seamless integration with external systems through APIs (Application Programming Interfaces). Research by R. Gupta et al. (2022) explores the integration of SMS with other educational platforms and services using PHP APIs, enhancing interoperability and data exchange capabilities.
- Interfacing with Learning Management Systems: Integration with Learning Management Systems (LMS) is crucial for delivering online courses and managing e-learning resources. Studies by V. Kumar et al. (2020) discuss methods for integrating PHP-based school management systems with popular LMS platforms like Moodle and Canvas, enabling a unified educational experience.

The literature reviewed demonstrates the significance of PHP and SQL in developing robust, feature-rich, and scalable school management systems. By leveraging PHP's versatility and SQL's data management capabilities, developers can create comprehensive solutions that address the diverse needs of educational institutions while prioritizing usability, security, and interoperability. Further research in this domain can focus on advanced features such as predictive analytics, artificial intelligence integration, and blockchain-based data security to enhance the efficiency and effectiveness of school management systems

CHAPTER 3

FEASIBILITY STUDY

After doing the project, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible-given unlimited resources and in finite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements. There are three parts in feasibility study.

a) Technical Feasibility
b) Economic Feasibility
c) Physical Feasibility

3.1 TECHNICAL FEASIBILITY

Technical feasibility involves study to establish the technical capability of the system being created to accomplish all requirements to the user.

The system should be capable of handling the proposed volume of data and provide users and operating environment to increase their efficiency.

For example, system should be capable of handling the proposed volume of data and provide users.

3.2 ECONOMIC FEASIBILITY:

Economic feasibility involves study to establish the cost benefit analysis.

Money spent on the system must be recorded in the form of benefit from the system. The benefits are of two types:

Tangible benefits:

- Saving man labor to do tedious tasks saves time.

Intangible benefits:

- Improves the quality of organization.

3.3 PHYSICAL FEASIBILITY

It involves study to establish the time responses of the new system being created. For e.g., if the new system takes more than one day to prepare crucial finance statement for the management, wherever it was required in an hour, the system fails to provide the same.

It should be clearly established that the new system requirements in the form of time responses would be completely met with. It may call for increase in cost. If the required cost is sacrificed, then the purpose of the

new system may not be achieved even if it was found to be technically feasible.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:

- **User Authentication and Authorization:** Define requirements for user authentication methods (e.g., username/password, biometrics) and role-based access control to ensure secure access to the system.
- **Student Dashboard:** Specify the information and functionalities to be included on the student dashboard, such as course schedules, attendance records, grades, announcements, and notifications.
- **Course Management:** Determine requirements for course catalog management, enrollment processes, and scheduling functionalities.
- **Attendance Tracking:** Define requirements for recording, monitoring, and managing student attendance in classes.
- **Grades and Academic Records:** Specify requirements for recording, calculating, and managing student grades and academic transcripts.
- **Communication Tools:** Identify requirements for communication features such as messaging, announcements, forums, and notifications.
- **Fee Management:** Define requirements for fee payment, tracking, invoicing, and reporting functionalities.
- **Library Integration:** Specify requirements for integrating with the library system, including catalog management, borrowing, and returning books.
- **Support and Helpdesk:** Determine requirements for providing support to users and managing helpdesk tickets efficiently.

4.2 NON-FUNCTIONAL REQUIREMENT

Non-functional requirements, also known as quality attributes or constraints, define the characteristics and constraints of the system beyond its functionality. These requirements describe how the system should perform, rather than what it should do. Non-functional requirements are often related to performance, reliability, security, usability, and other aspects that contribute to the overall system quality. Examples include response time, system availability, data encryption, user interface

design, and regulatory compliance.

- **Performance:** The application should have fast and responsive image recognition, with minimal latency or delay in detecting and overlaying digital objects on the image targets. It should also deliver smooth playback of videos or animations without any significant lag.
- **User Interface (UI) and User Experience (UX):** The application should have an intuitive and user-friendly interface, with clear instructions or visual cues to guide users in scanning the college brochure and interacting with the augmented reality content. The user experience should be immersive, engaging, and visually appealing.
- **Compatibility and Device Support:** The application should be compatible with a wide range of smartphones or devices, supporting both Android and iOS platforms. It should consider various screen sizes, resolutions, and camera capabilities to ensure a consistent experience across different devices.
- **Stability and Reliability:** The application should be stable and reliable, capable of handling potential errors or exceptions during image recognition or content playback. It should gracefully handle situations such as low lighting conditions or variations in brochure positioning.
- **Security and Privacy:** The application should prioritize user privacy and data security, adhering to relevant privacy regulations. It should obtain necessary permissions for accessing device features, such as camera and storage, and ensure secure transmission and storage of any user-related data.

- **Scalability:** The application should have the potential to scale, accommodating future updates, additional content, or expanded functionality. It should be designed in a modular and extensible manner, allowing for easy integration of new image targets or features without significant rework.

5.1 SOFTWARE AND HARDWARE REQUIREMENT

Software Requirements:

S. NO.	DESCRIPTION	TYPE
1	Operating System	Windows, macOS, Linux
2	Web Server	Apache, Nginx
3	Frontend Technologies	HTML, CSS, JavaScript
4	Integrated Development Environment (IDE)	Visual Studio Code, Sublime Text, PHPStorm
5	Version Control	Git, GitHub

6	Security Tools	Security plugins, libraries, frameworks
7	Communication Protocols	SMTP, APIs
8	APIs and Integrations	RESTful APIs
9	Security Tools	Security plugins, libraries, frameworks

Table 4.1

Hardware Requirements:

S. NO.	DESCRIPTION	TYPE
1	Server Hardware	
	- Processor	Multi-core processor (e.g., Intel Core i5)
	- RAM	Minimum 4GB RAM, recommended 8GB or higher
	- Storage	SSD or HDD with sufficient storage capacity
	- Network Interface	Gigabit Ethernet or higher
2	Client Devices	Desktops, laptops, tablets, smartphones
3	Networking Equipment	Router, switches, network cables
4	Backup Systems	External storage devices, cloud backup services

Table 4.2

CHAPTER 5

SYSTEM ARCHITECTURE AND DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities – design, coding, implementation, and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system.

Design is the only way to accurately translate the customer's requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps.

Preliminary design is concerned with the transformation of requirements into data

Unified Modelling Language Diagrams (UML):

- The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.
- A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View

i. This view represents the system from the user's perspective.
ii. The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view

i. In this model the data and functionality are arrived from inside the system.
ii. This model view models the static structures.

Behavioral Model View

- It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

- In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In these the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are-

- UML Analysis modelling, which focuses on the user model and structural model views of the system?
- UML design modelling, which focuses on the behavioral modelling, implementation modelling and environmental model views.

Use Case Diagrams User/Student:

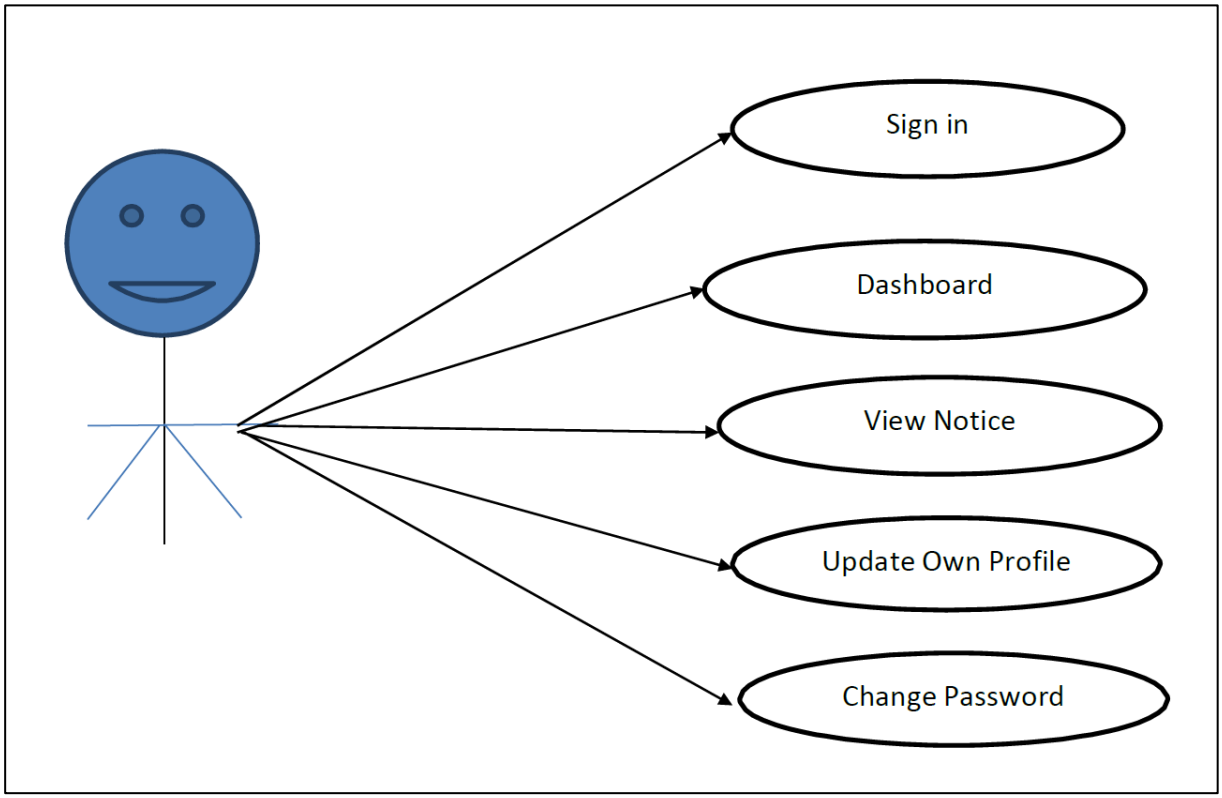


FIG. 5.1

Use Case Diagrams Admin:

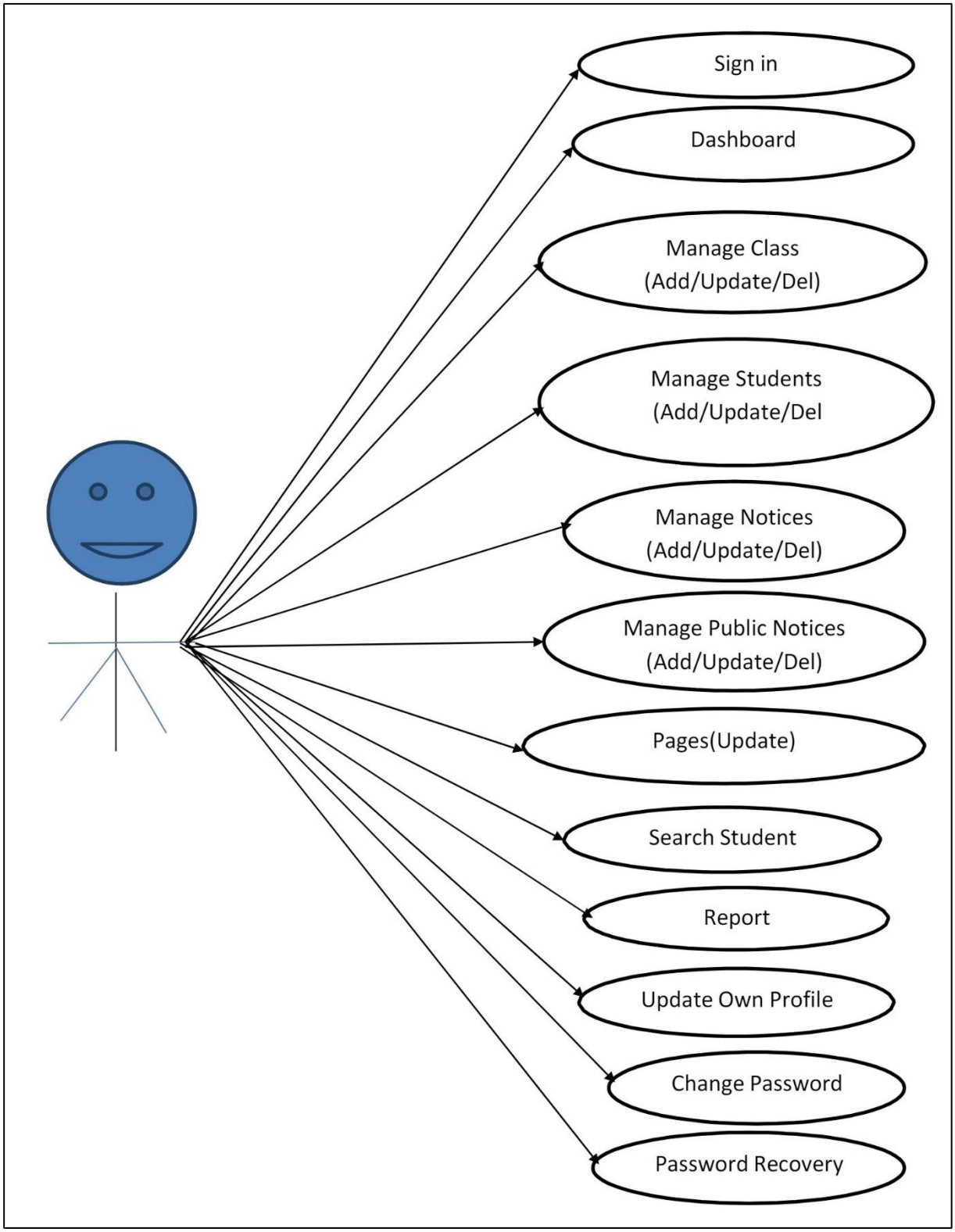
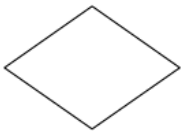


FIG. 5.2

ENTITY-RELATIONSHIP Diagrams:

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

The symbols used in E-R diagrams are:

SYMBOL	PURPOSE
	Represents Entity sets.
	Represent attributes.
	Represent Relationship Sets.
	Line represents flow

Structured analysis is a set of tools and techniques that the analyst uses to develop a new kind of a system:
The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection a personal consideration.

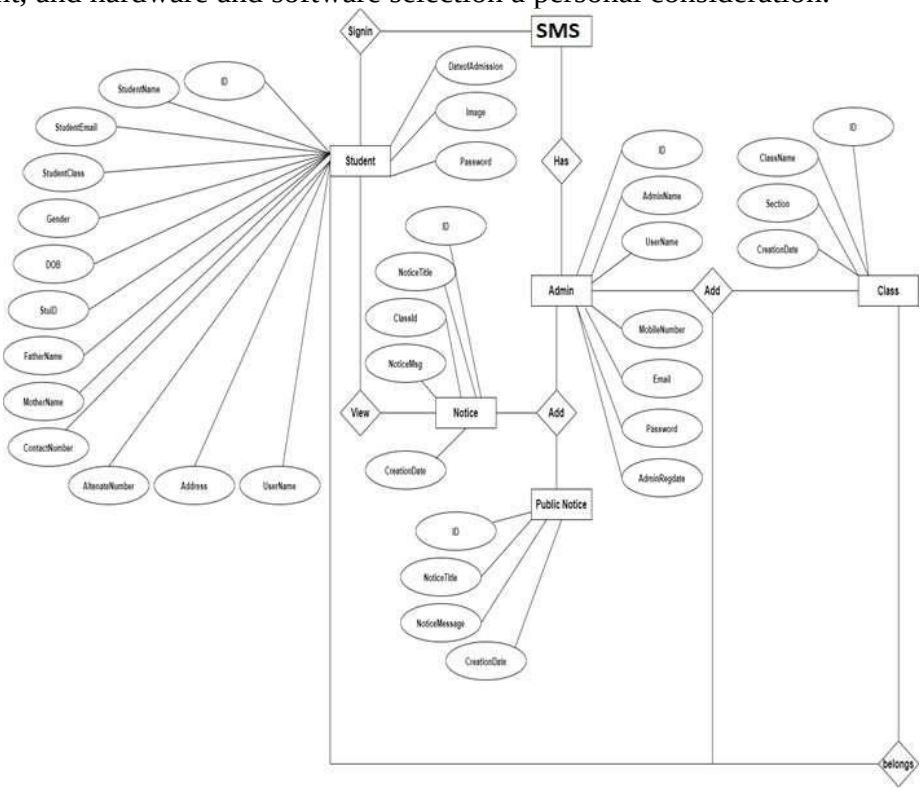


Fig. 5.3

DATABASE DESIGN:

The data in the system must be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.
A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive, and flexible for the user. Relationships are established between the data items and
unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies, and optimizing for updates. The MySQL database has been chosen for developing the relevant databases.

Student Management System (SMS) contains 6 MySQL tables:

tbladmin table Structure: This table store the admin login and personal details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	AdminName	varchar(120)	utf8mb4_general_ci		Yes	*NULL*		
3	UserName	varchar(120)	utf8mb4_general_ci		Yes	*NULL*		
4	MobileNumber	bigint(10)			Yes	*NULL*		
5	Email	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
6	Password	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
7	AdminRegdate	timestamp			Yes	current_timestamp()		

Table 5.1

tblclass table Structure: This table store the class and section.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(5)			No	*None*		AUTO_INCREMENT
2	ClassName	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
3	Section	varchar(20)	latin1_swedish_ci		Yes	*NULL*		
4	CreationDate	timestamp			Yes	current_timestamp()		

Table 5.2

tblnotice table Structure: This table store the notices detail which is announced by admin.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(5)			No	*None*		AUTO_INCREMENT
2	NoticeTitle	mediumtext	latin1_swedish_ci		Yes	*NULL*		
3	ClassId	int(10)			Yes	*NULL*		
4	NoticeMsg	mediumtext	latin1_swedish_ci		Yes	*NULL*		
5	CreationDate	timestamp			Yes	current_timestamp()		

Table 5.3

tblpublicnotice table Structure: This table store the public notices detail which is announced by admin.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(5)			No	*None*		AUTO_INCREMENT
2	NoticeTitle	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	NoticeMessage	mediumtext	latin1_swedish_ci		Yes	*NULL*		
4	CreationDate	timestamp			Yes	current_timestamp()		

Table 5.4

tbladmin table Structure: This table store the students details of educational organization.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	StudentName	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	StudentEmail	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
4	StudentClass	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
5	Gender	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
6	DOB	date			Yes	*NULL*		
7	StuID	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
8	FatherName	mediumtext	latin1_swedish_ci		Yes	*NULL*		
9	MotherName	mediumtext	latin1_swedish_ci		Yes	*NULL*		
10	ContactNumber	bigint(10)			Yes	*NULL*		
11	AltenateNumber	bigint(10)			Yes	*NULL*		
12	Address	mediumtext	latin1_swedish_ci		Yes	*NULL*		
13	UserName	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
14	Password	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
15	Image	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
16	DateofAdmission	timestamp			Yes	current_timestamp()		

Table 5.5

tblpage table Structure: This table store the details of about us and contact us pages.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	PageType	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
3	PageTitle	mediumtext	latin1_swedish_ci		Yes	*NULL*		
4	PageDescription	mediumtext	latin1_swedish_ci		Yes	*NULL*		
5	Email	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
6	MobileNumber	bigint(10)			Yes	*NULL*		
7	UpdationDate	date			Yes	*NULL*		

Table 5.6

Class Diagram:
The class diagram shows a set of classes, interfaces, collaborations, and their relationships.

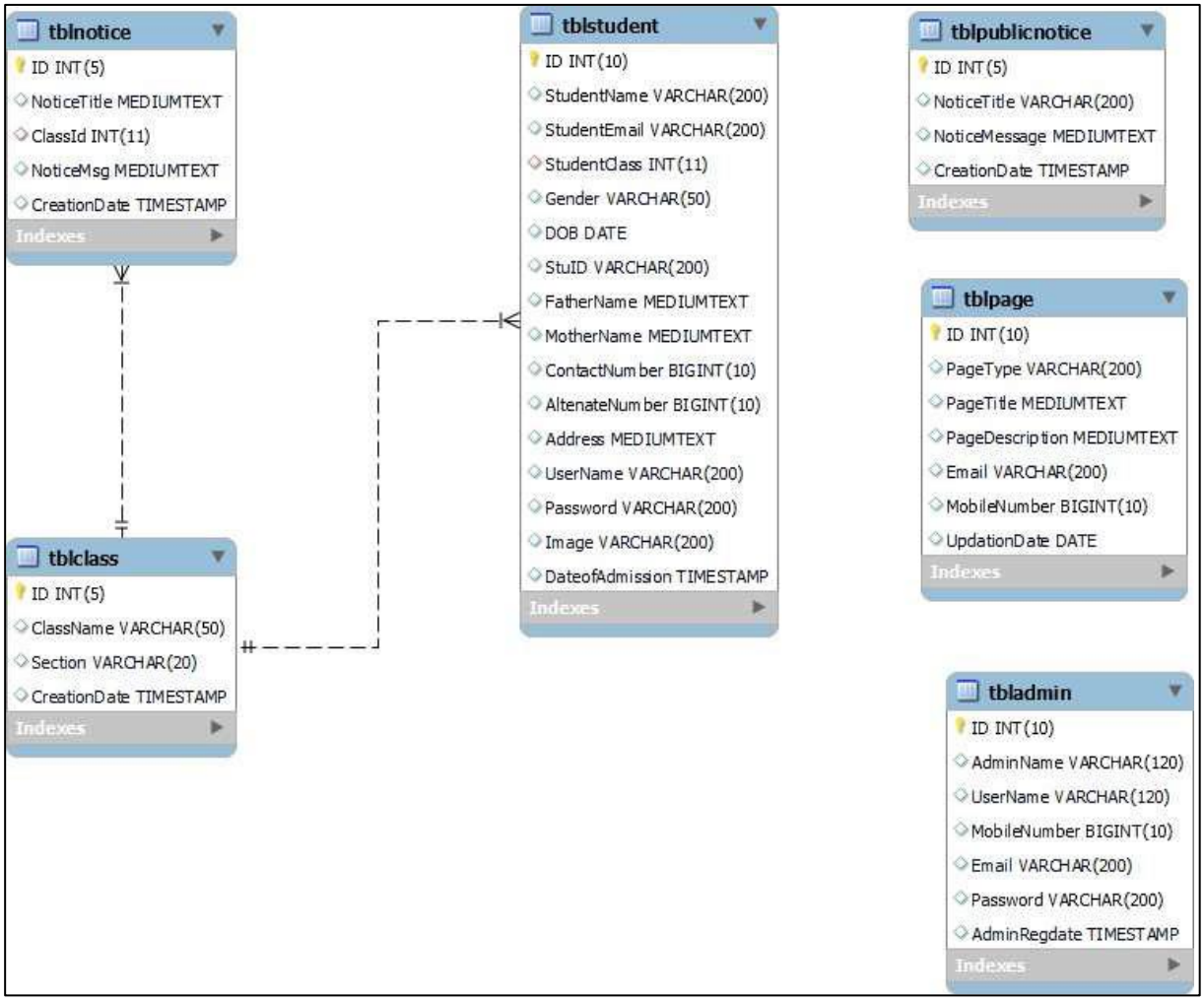

tblnotice
ID INT(5)
NoticeTitle MEDIUMTEXT
ClassId INT(11)
NoticeMsg MEDIUMTEXT
CreationDate TIMESTAMP
Indexes
tblstudent
ID INT(10)
StudentName VARCHAR(200)
StudentEmail VARCHAR(200)
StudentClass INT(11)
Gender VARCHAR(50)
DOB DATE
StuID VARCHAR(200)
FatherName MEDIUMTEXT
MotherName MEDIUMTEXT
ContactNumber BIGINT(10)
AltenateNumber BIGINT(10)
Address MEDIUMTEXT
UserName VARCHAR(200)
Password VARCHAR(200)
Image VARCHAR(200)
DateofAdmission TIMESTAMP
Indexes
tblpublicnotice
ID INT(5)
NoticeTitle VARCHAR(200)
NoticeMessage MEDIUMTEXT
CreationDate TIMESTAMP
Indexes
tblpage
ID INT(10)
PageType VARCHAR(200)
PageTitle MEDIUMTEXT
PageDescription MEDIUMTEXT
Email VARCHAR(200)
MobileNumber BIGINT(10)
UpdationDate DATE
Indexes
tblclass
ID INT(5)
ClassName VARCHAR(50)
Section VARCHAR(20)
CreationDate TIMESTAMP
Indexes
tbladmin
ID INT(10)
AdminName VARCHAR(120)
UserName VARCHAR(120)
MobileNumber BIGINT(10)
Email VARCHAR(200)
Password VARCHAR(200)
AdminRegdate TIMESTAMP
Indexes

FIG. 5.4

CHAPTER 6

SYSTEM TESTING

SOFTWARE TESTING TECHNIQUES:

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

TESTING OBJECTIVES:

1. Testing is process of executing a program with the intent of finding an error.
2. A good test case design is one that has a probability of finding a yet undiscovered error.
3. A successful test is one that uncovers a yet undiscovered error.

These above objectives imply a dramatic change in view port. Testing cannot show the absence of defects, it can only show that software errors are present.

There are three types of testing strategies.

1. Unit test
2. Integration test
3. Performance test

Unit Testing:

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

Integration Testing:

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

Performance Testing:

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

CHAPTER 7

IMPLEMENTATION

Home Page:

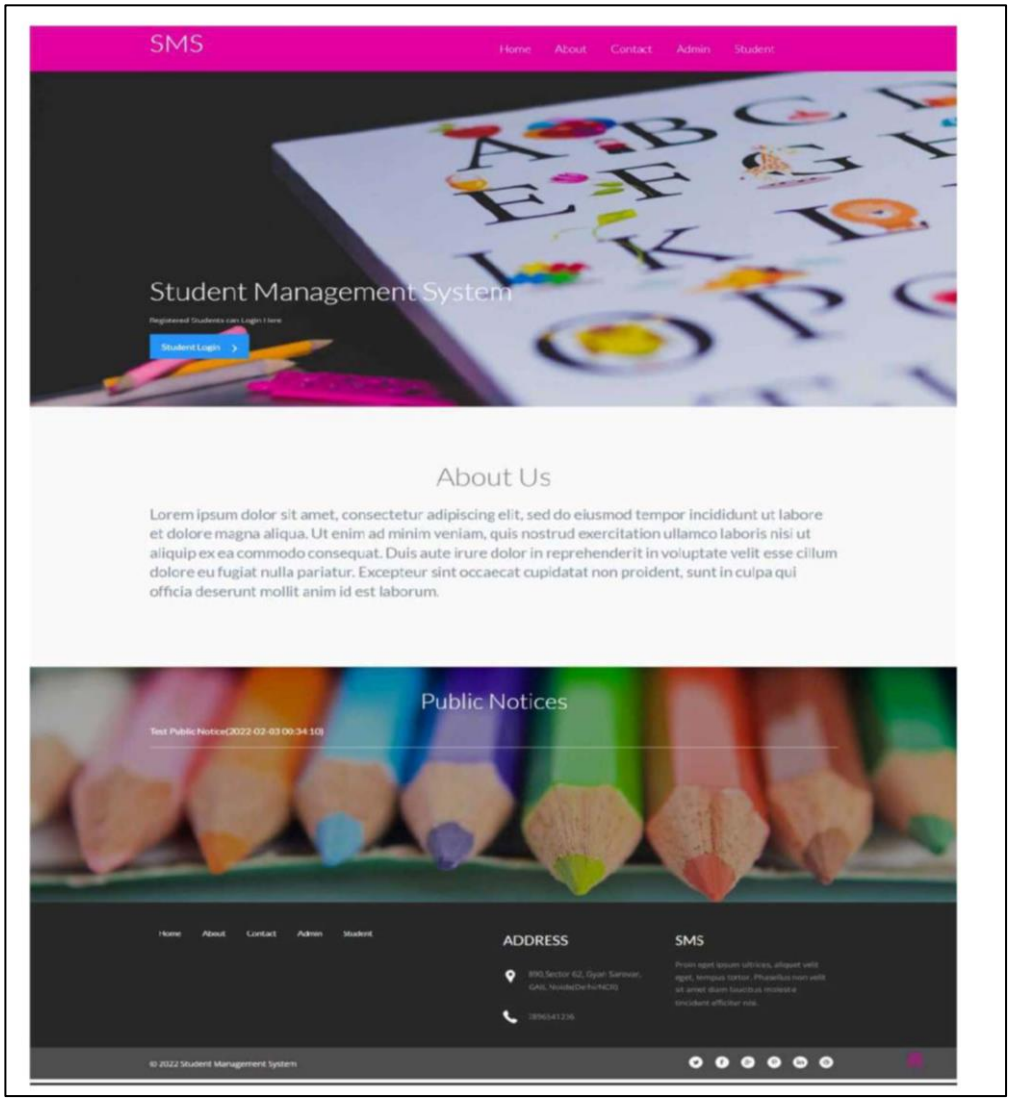


FIG 7.1

About Us:

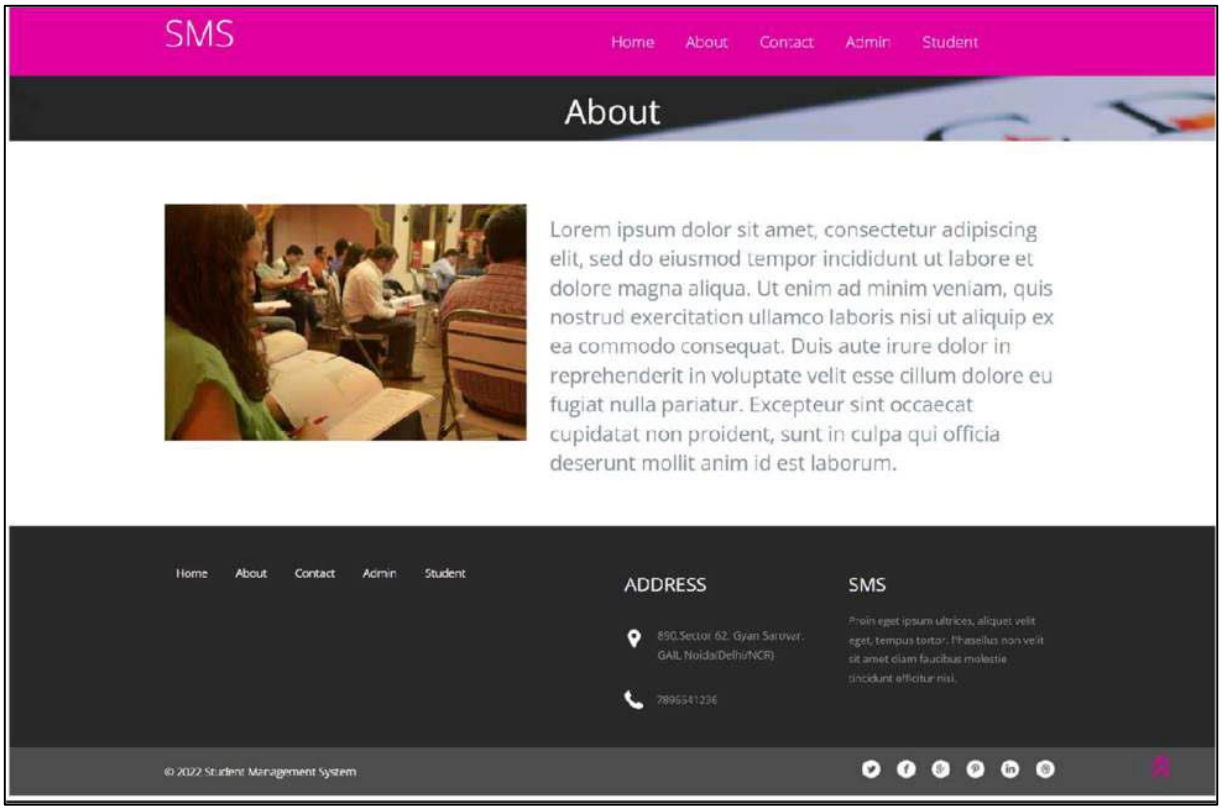


FIG. 7.2

Contact Us:

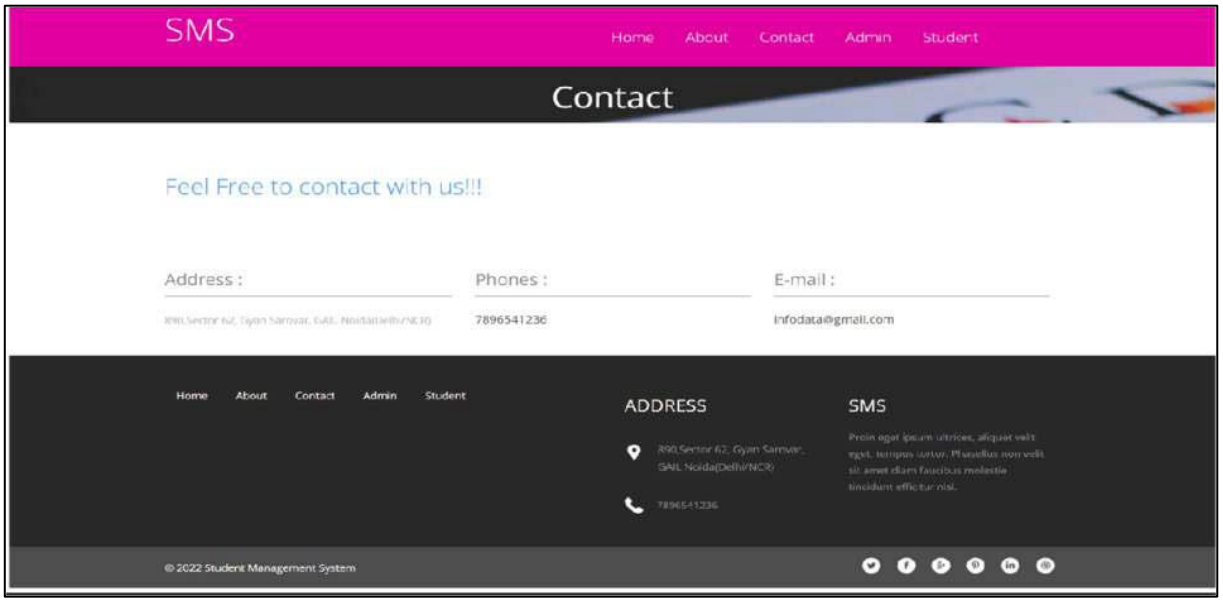


FIG. 7.3

Admin Panel:

Login Page:

FIG. 7.4

Forgot Password:

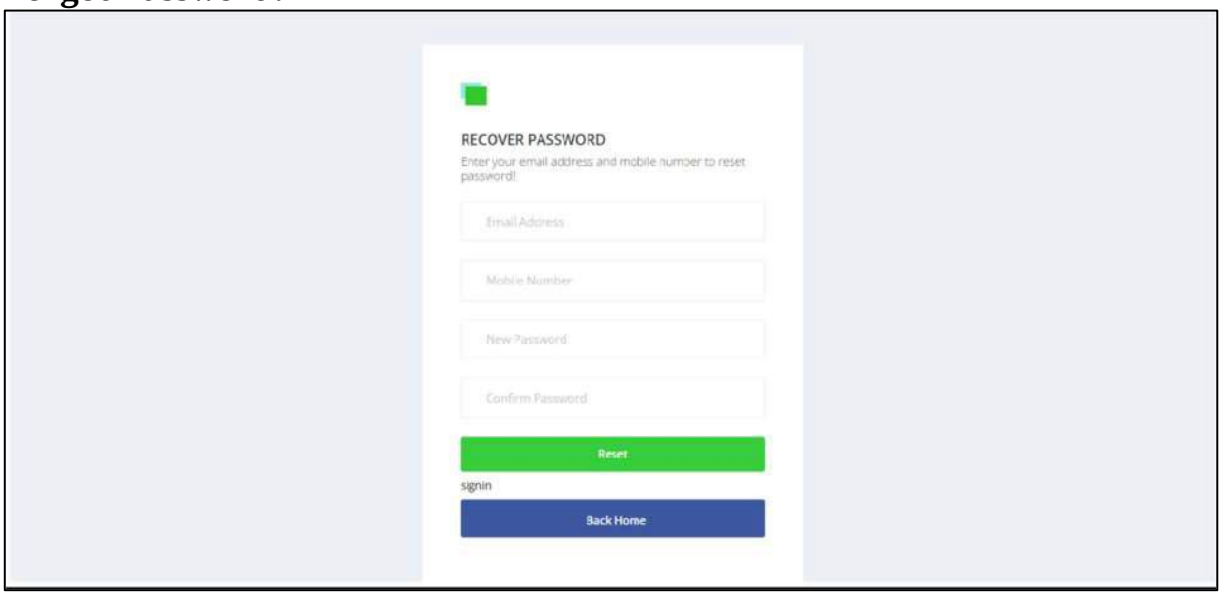


FIG. 7.5

Dashboard:

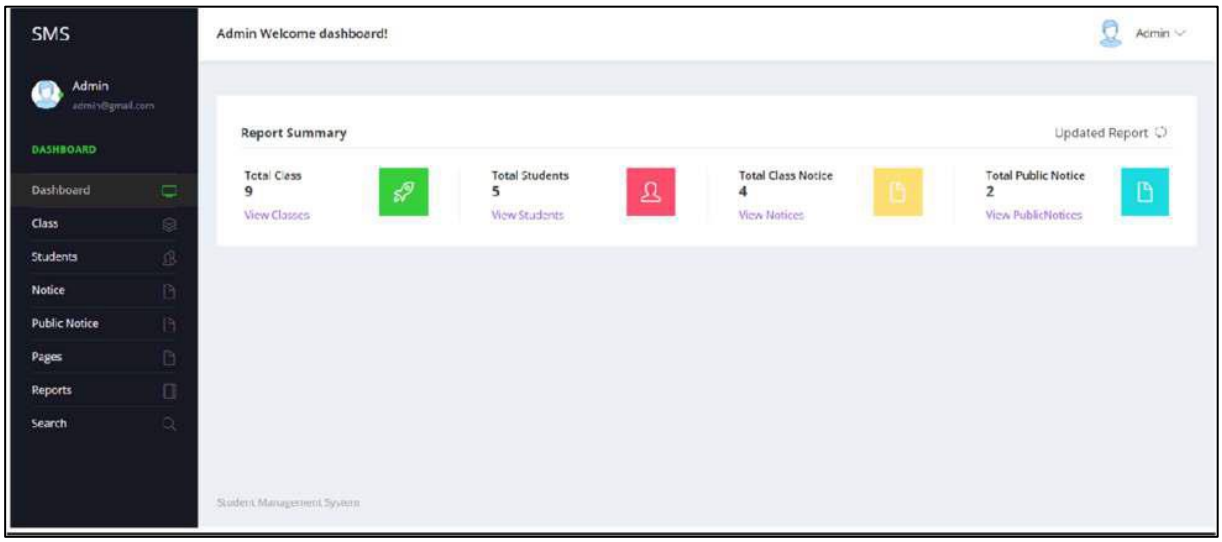


FIG 7.6

Profile:

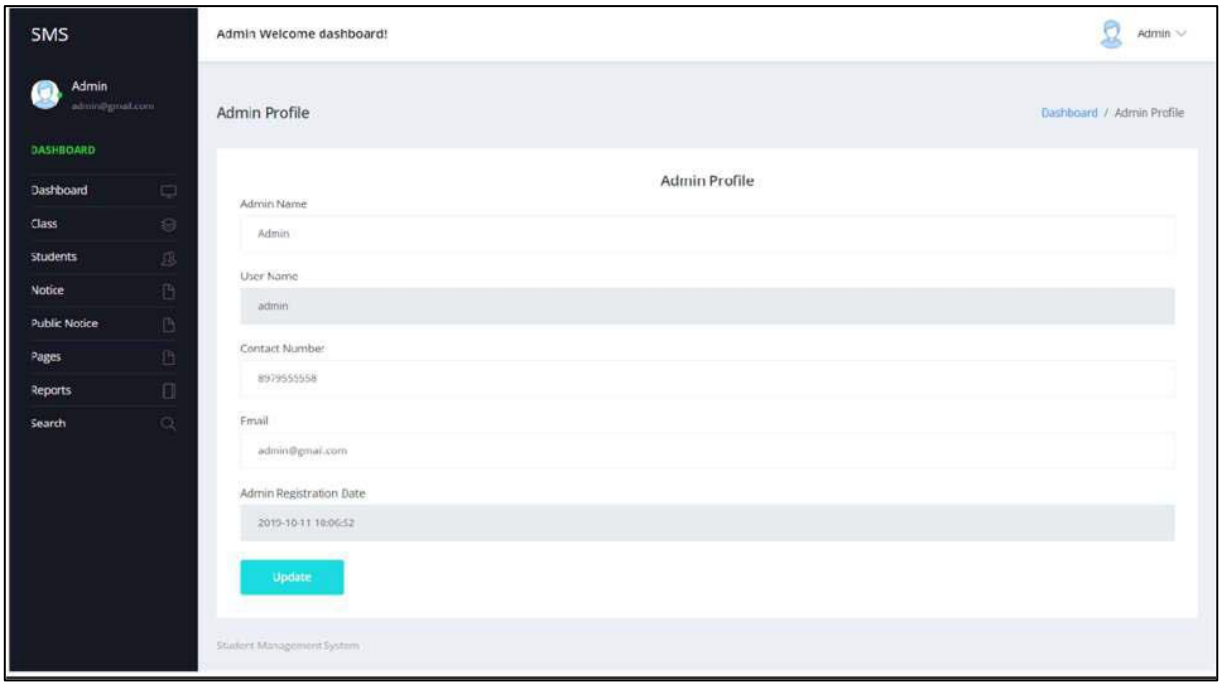


FIG. 7.7

Change Password:

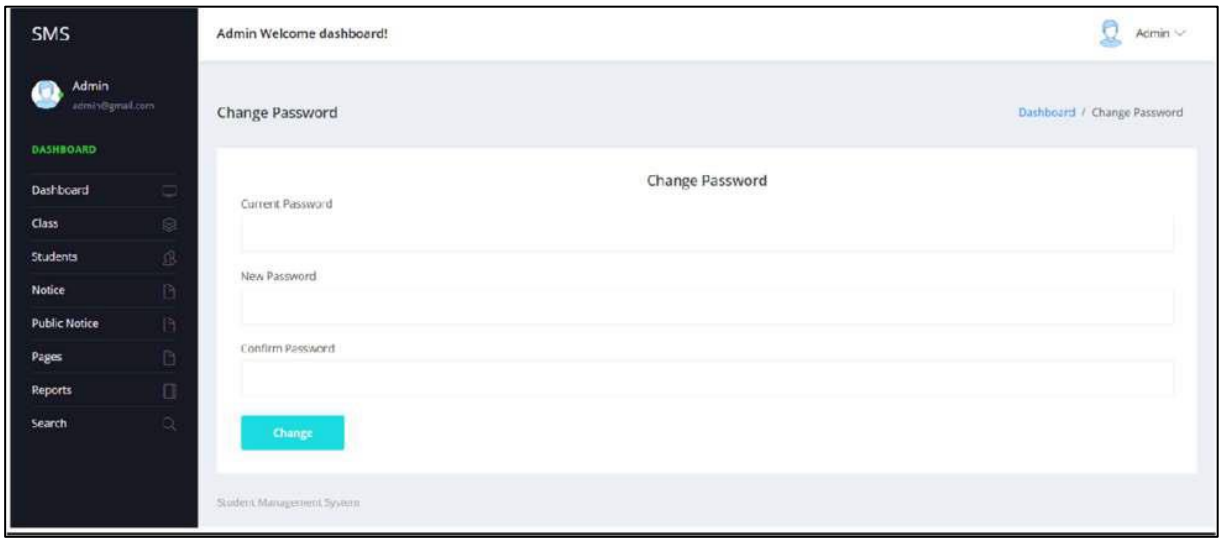


FIG 7.8

Add Class:

FIG 7.9

Manage Class:

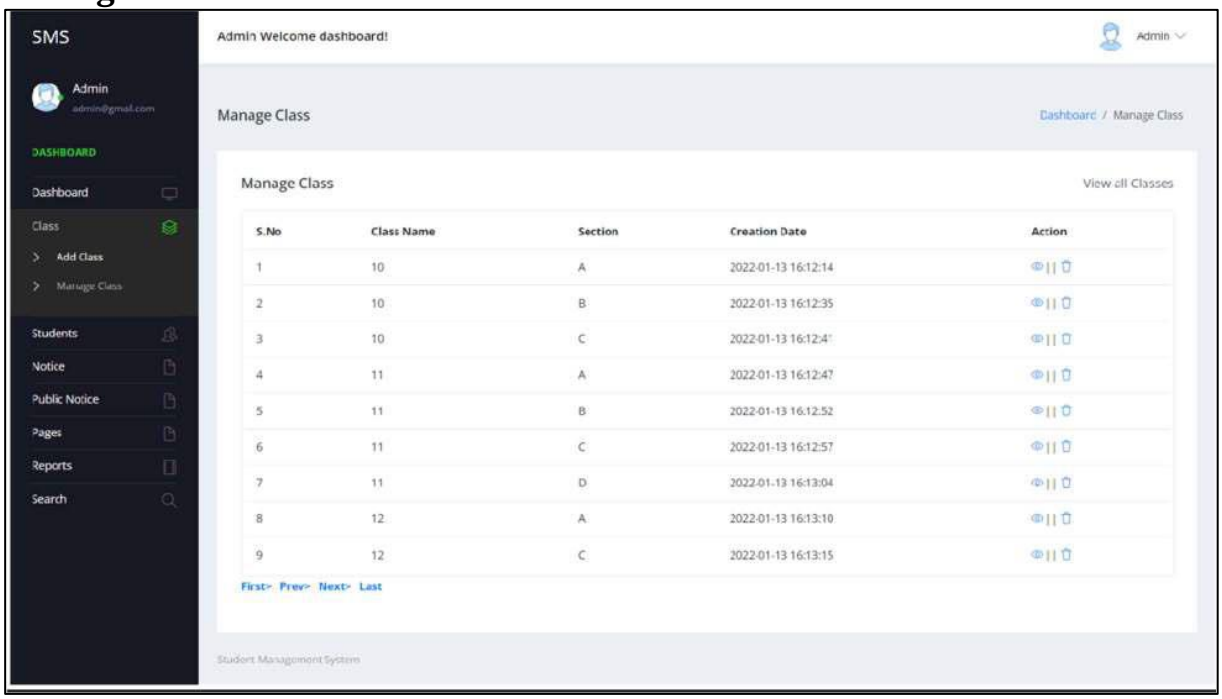


FIG 7.10

Add Notice:

FIG 7.11

Add Student:

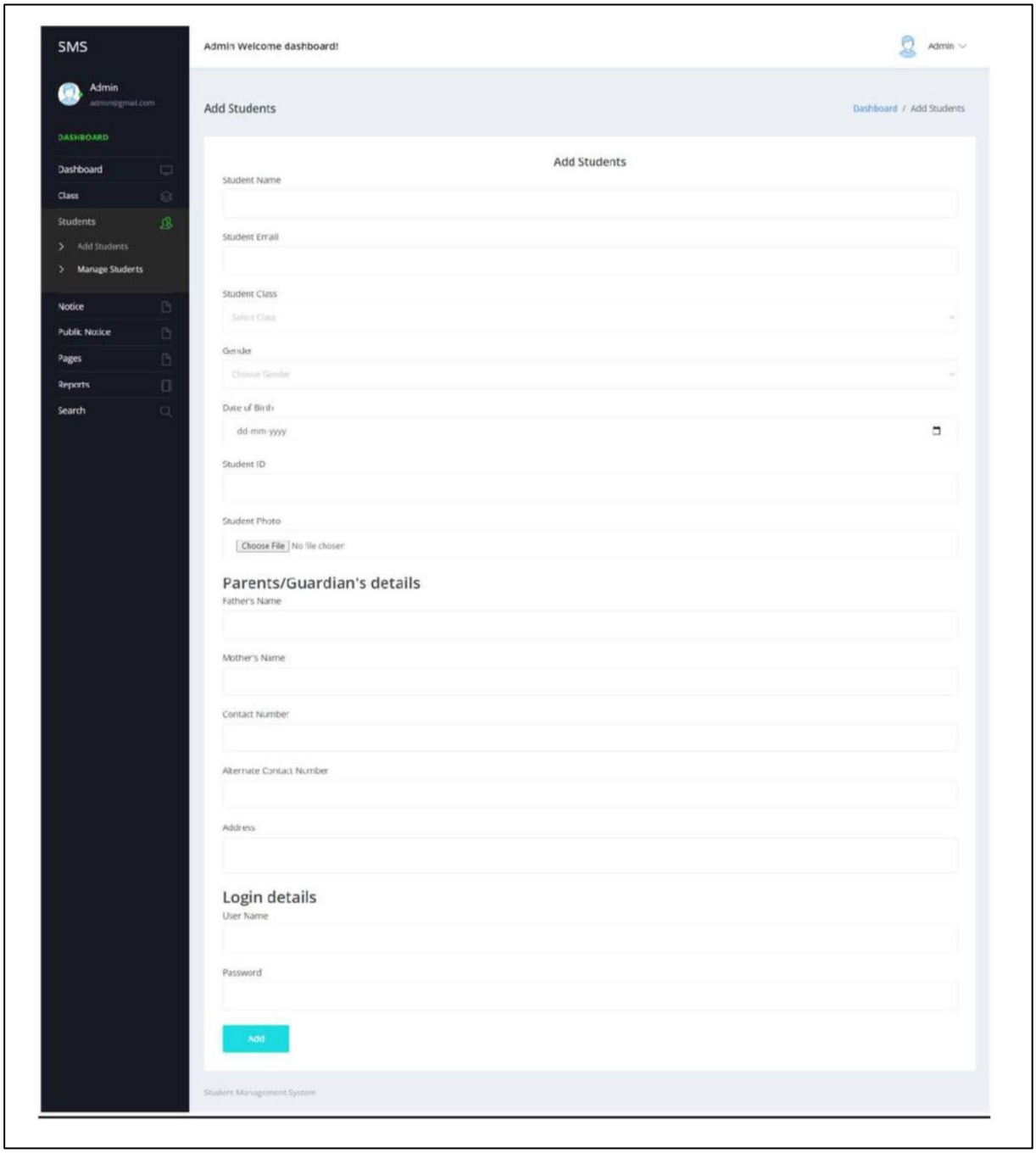


FIG 7.12

Manage Students:

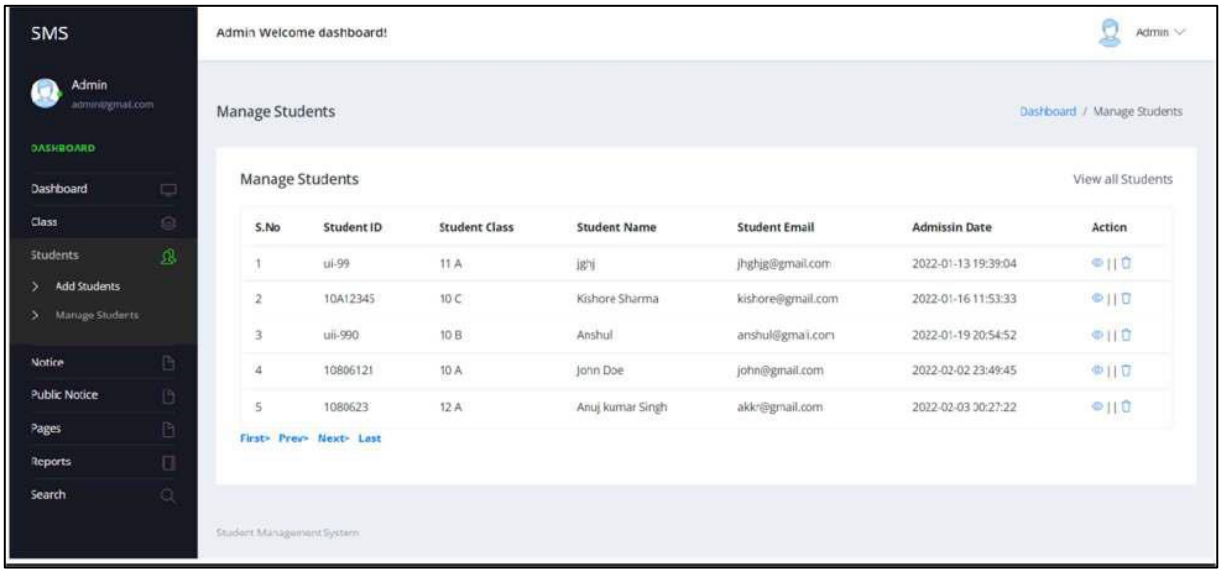


FIG 7.13

Update Students:

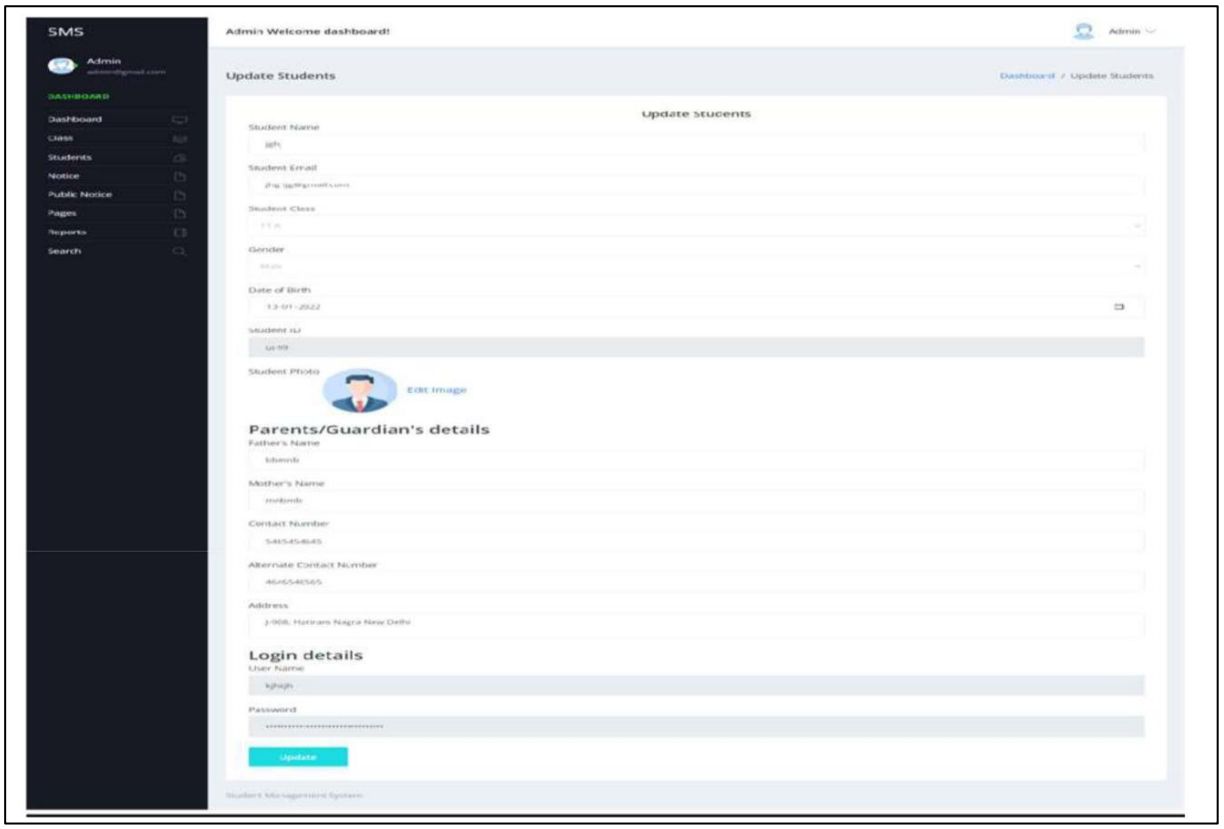


FIG 7.14

Student Panel:

Login Page:

FIG 7.15

Students Profile:

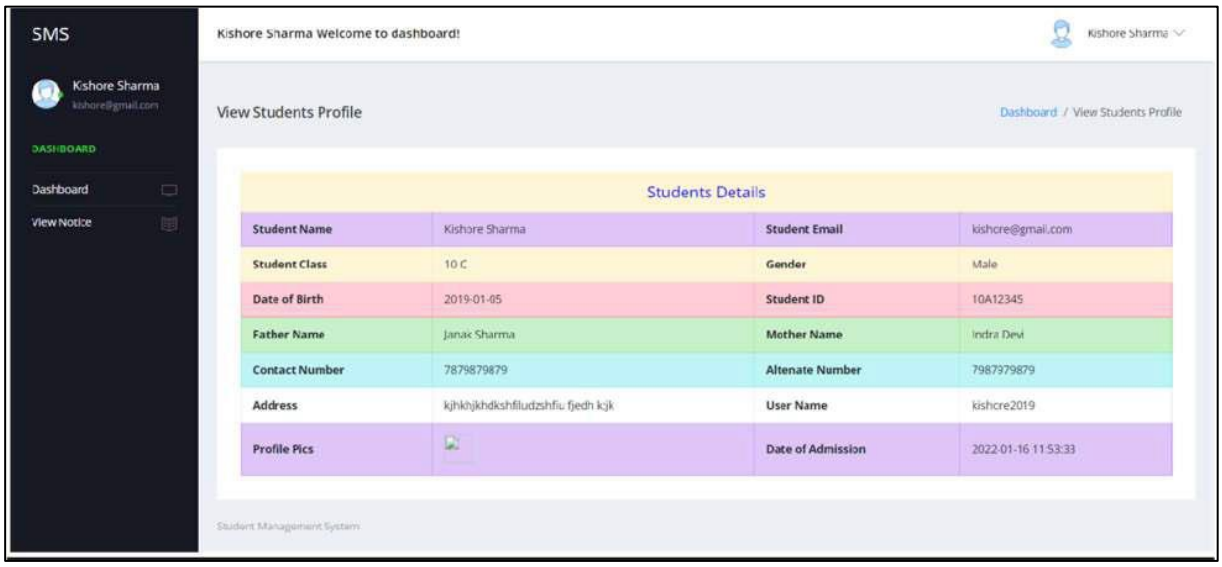


FIG 7.16

Dashboard:

FIG 7.17

View Notice:

FIG 7.18

CHAPTER 8

CONCLUSION

The project titled as Student Management System was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time. Student Management System can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time-consuming. All these problems are solved using this project.

7.1 Future Scope

The future scope of Student Management Systems (SMS) entails leveraging emerging technologies and evolving educational paradigms to enhance administrative efficiency, student engagement, and learning outcomes. Integration of artificial intelligence (AI) promises predictive analytics for personalized learning, while blockchain technology ensures secure credential verification. Mobile applications with augmented reality (AR) features and IoT-enabled campus management offer immersive experiences and smart campus solutions. Data analytics drives decision-making, identifying at-risk students and optimizing resources. Enhanced collaboration tools facilitate synchronous and asynchronous online learning. Cloud-based solutions ensure scalability, flexibility, and seamless integration. Emphasis on cybersecurity and data privacy safeguards sensitive information. Continuous user feedback and agile development drive iterative improvements, ensuring SMS remains adaptable to changing needs and technological advancements, ultimately empowering educational institutions to deliver innovative, student-centered learning experiences.

CHAPTER 9

REFRENCES

1 Aliaa Youssif, M., Elsayed, M. H., & Alhazmi, A. H. (2020). "Web Development Languages: A Comparative Study." In Proceedings of the 3rd International Conference on Education, Technology, and Innovations (ICETI 2020), 72-78.

2 Sasirekha, T., & Rajaram, S. (2019). "A Comprehensive Study on SQL Injection Attacks." In International Conference on Recent Trends in Computer Science and Engineering (ICRTCSE 2019), 127-132.

3 Ahmed, S., Imran, M., & Khan, S. (2018). "Design and Development of Student Registration System Using PHP." In International Journal of Information Technology and Management Studies (IJITMS), 8(3), 32-38.

4 Kumar, P., & Arulmurugan, V. (2017). "Web Based Attendance Management System Using PHP and MySQL." In International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 7(4), 404-407.

5 Pandey, S., & Pandey, R. K. (2021). "Design and Development of Student Grade Management System using PHP and MySQL." In International Journal of Computer Science and Network Solutions (IJCSNS), 9(1), 66-73.

6 Mathew, N., George, J., & Kumar, A. (2019). "Design and Development of User-Centric School Management System Using PHP and MySQL." In International Journal of Computer Applications (IJCA), 182(44), 23-27.

7 Azad, M., Islam, M. S., & Hossain, M. S. (2020). "Development of Mobile Responsive School Management System Using PHP and MySQL." In International Journal of Computer Science and Information Security (IJCSIS), 18(6), 134-140.

8 Khedr, A., Elhoseny, M., & Shankar, K. (2018). "A Novel Method for Securing Data Using Cryptography Techniques." In International Journal of Advanced Computer Science and Applications (IJACSA), 9(12), 21-26.

9 Gupta, R., Bhatia, M., & Mittal, R. (2022). "Integration of School Management System with Third Party Applications using RESTful APIs." In International Journal of Computer Sciences and Engineering (IJCSE), 10(1), 32-37.

10 Kumar, V., Saini, D., & Kumar, A. (2020). "Integration of School Management System with Moodle LMS." In International Journal of Engineering and Advanced Technology (IJEAT), 10(4), 566-571.