

PROJECT AUTHOR: VISHAL SIDDHA

TABLE OF CONTENTS

	Page No.
1. Introduction to AMS.....	5
2. Problem Statement .....	6
3. Proposed Solution .....	6
4. Requirement specification .....	7
5. Tech Stack.....	8
6. Design.....	9
6.1 UML Diagram.....	10
6.2 Class Diagram.....	15
6.3 ER Diagram.....	16
6.4 Data Tables .....	19
7. Implementation and System Testing.....	22
8. User Interface .....	25
9. Conclusion .....	43
10. References .....	44

## Introduction to AMS

### Introduction:-

In present academic system, regular class attendance of students' plays a significant role in performance assessment and quality monitoring. The conventional methods practiced in most of the institutions are by calling names or signing on papers, which is highly time-consuming and insecure. This project presents the Attendance Monitoring System for convenience or data reliability.

Attendance Monitoring System is a web application developed for daily student attendance in schools, colleges and institutes. It facilitates to access the attendance information of a particular student in a particular class and shift. The information is sorted by the operators, which will be provided by the staff for a particular class. This system will also help in evaluating attendance criteria of a student.

### Purpose:-

The purpose of developing attendance monitoring system is to computerized the tradition way of taking attendance. Another purpose for developing this application is to generate the report automatically.

### Scope:-

Attendance Monitoring System project is developed as a web application and it will work over web.

## Problem Statement

In present all attendance work done on the paper. The whole year attendance is stored in the registers. We can't generate reports as per our requirements because its take more time to calculate the attendance.

- Manual Control: All report calculation is done manually so there is a chance of error.
- Lots of paper work: Attendance is maintained in the register so lots of paper required.
- Not user friendly: The present system is not user friendly because data is not stored in structured and proper format.
- Time consuming: It takes a lot of time to calculate attendance manually.

## Proposed Solution

Attendance monitoring system deals with the monitoring of the student's attendance details. It will generate the attendance of the student on basis of presence in class. It is maintained on the daily and shift basis of their attendance. The staffs will be provided with the separate username & password to mark the student's attendance status.

The staffs handling the particular shift responsible to make the attendance for all students. Only if the student present on that particular shift, the attendance will be calculated. Reports of students will be generated by particular class as well as by their name.

## Requirement Specification

### Hardware Configuration :

Client Side:

RAM	512 MB
Hard disk	10 GB
Processor	1.0 GHz

Server side:

RAM	1 GB
Hard disk	20 GB
Processor	2.0 GHz

### Software Requirement:

Client Side:

Web Browser	Google Chrome or any compatible browser
Operating System	Window or any equivalent OS

Server Side:

Web Server	APACHE
Server side Language	PHP5.6 or above version
Database Server	MYSQL
Web Browser	Google Chrome or any compatible browser
Operating System	Window or any equivalent OS

### Tech Stack

Front-End: HTML, CSS, JavaScript, Bootstrap, jQuery

Back-End: PHP

Database: MySQL

Tools: Visual Studio Code, Git and GitHub, XAMPP

Client: Web Browsers like Chrome, Firefox etc

## Design

### Design Introduction:

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.

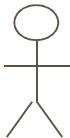
Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps.

Preliminary design is concerned with the transformation of requirements into data.

### UML Diagrams:

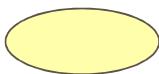
#### Actor:

A coherent set of roles that users of use cases play when interacting with the use 'cases.



#### Use case:

A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.



UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

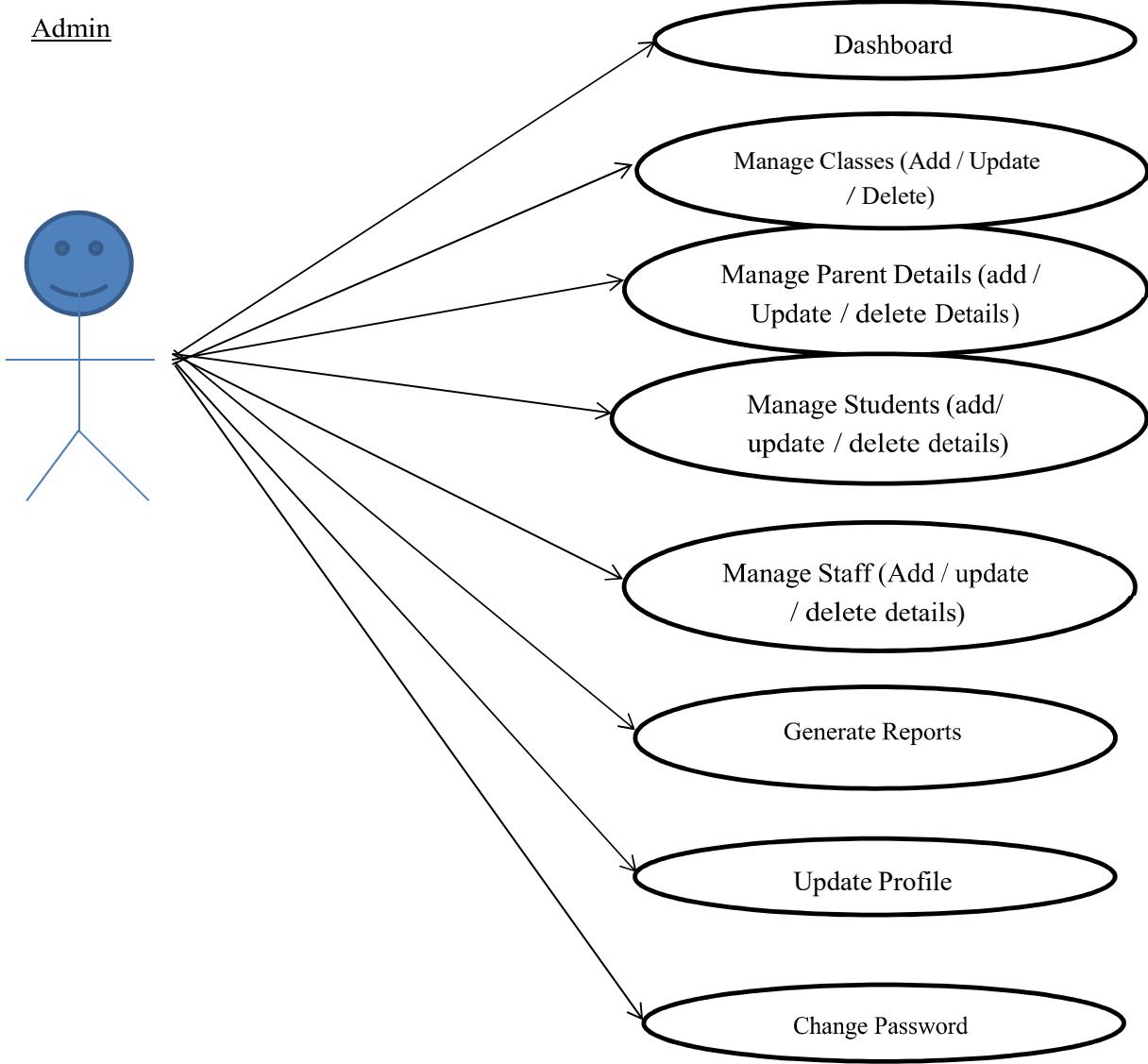


Fig : Admin diagram.

Staff:

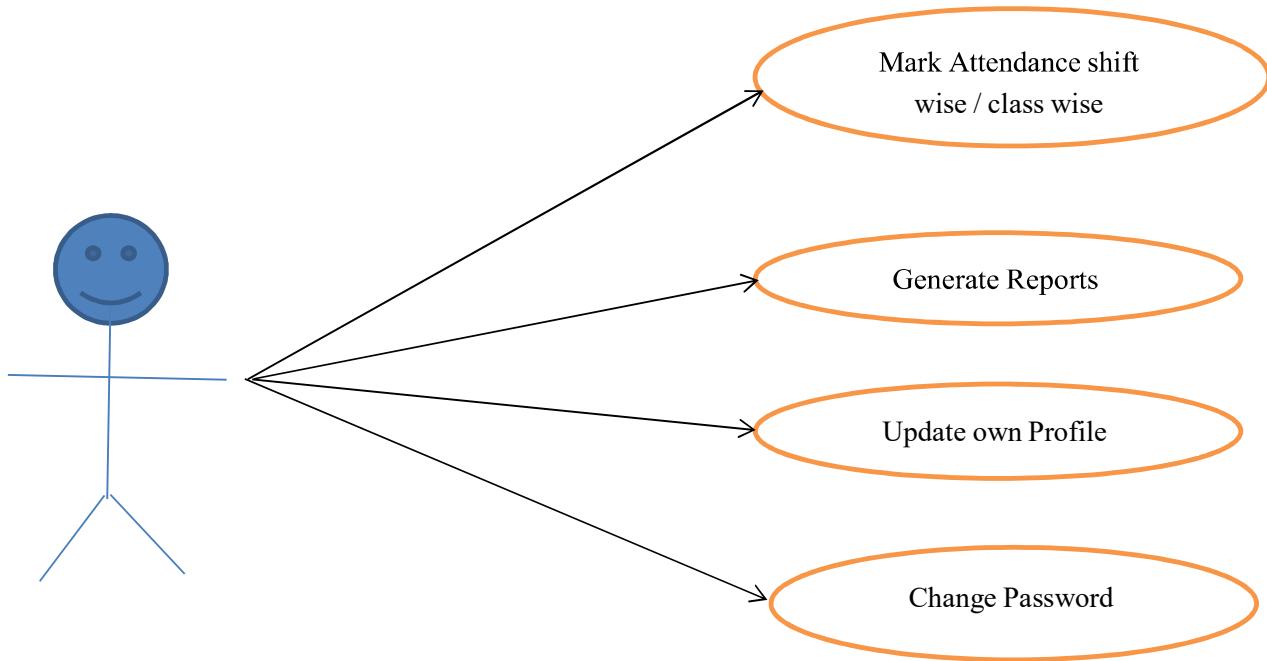


Fig : Staff diagram.

Parent:

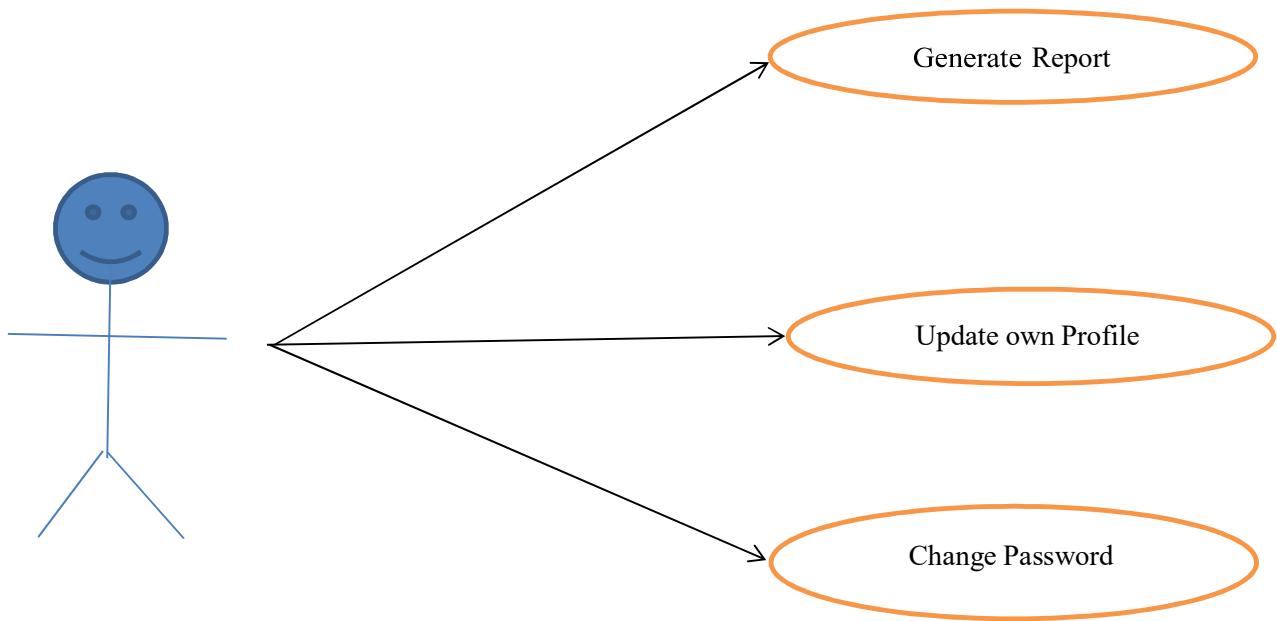


Fig : Parent diagram.

Student:

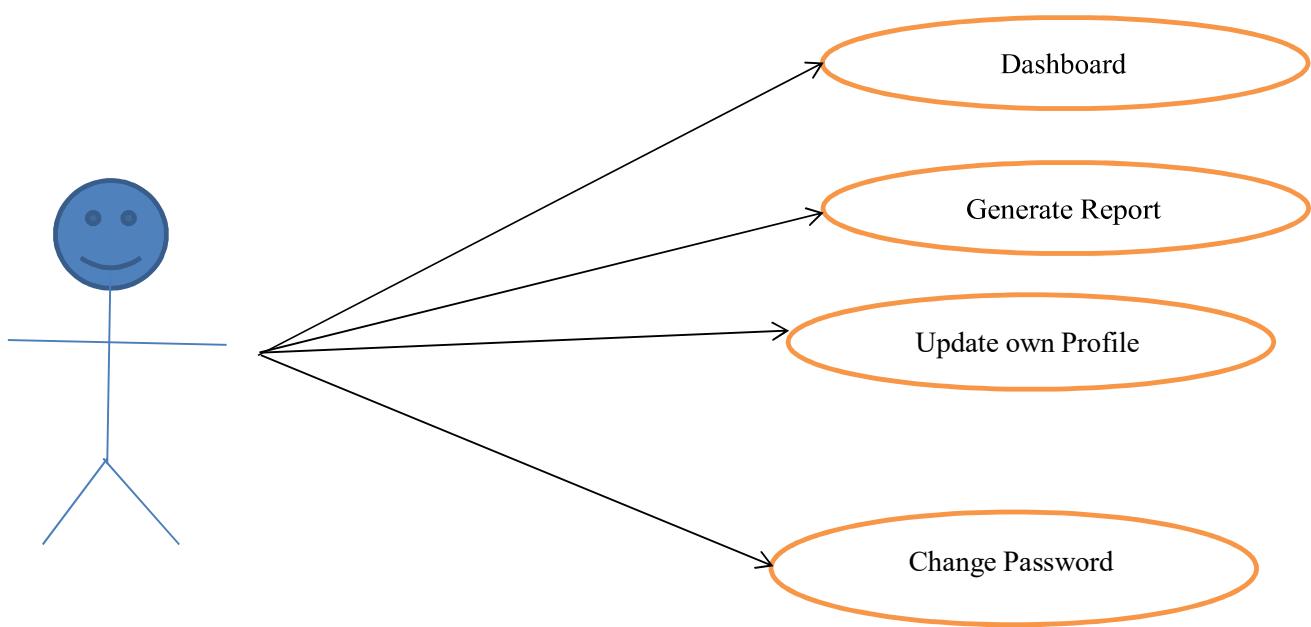
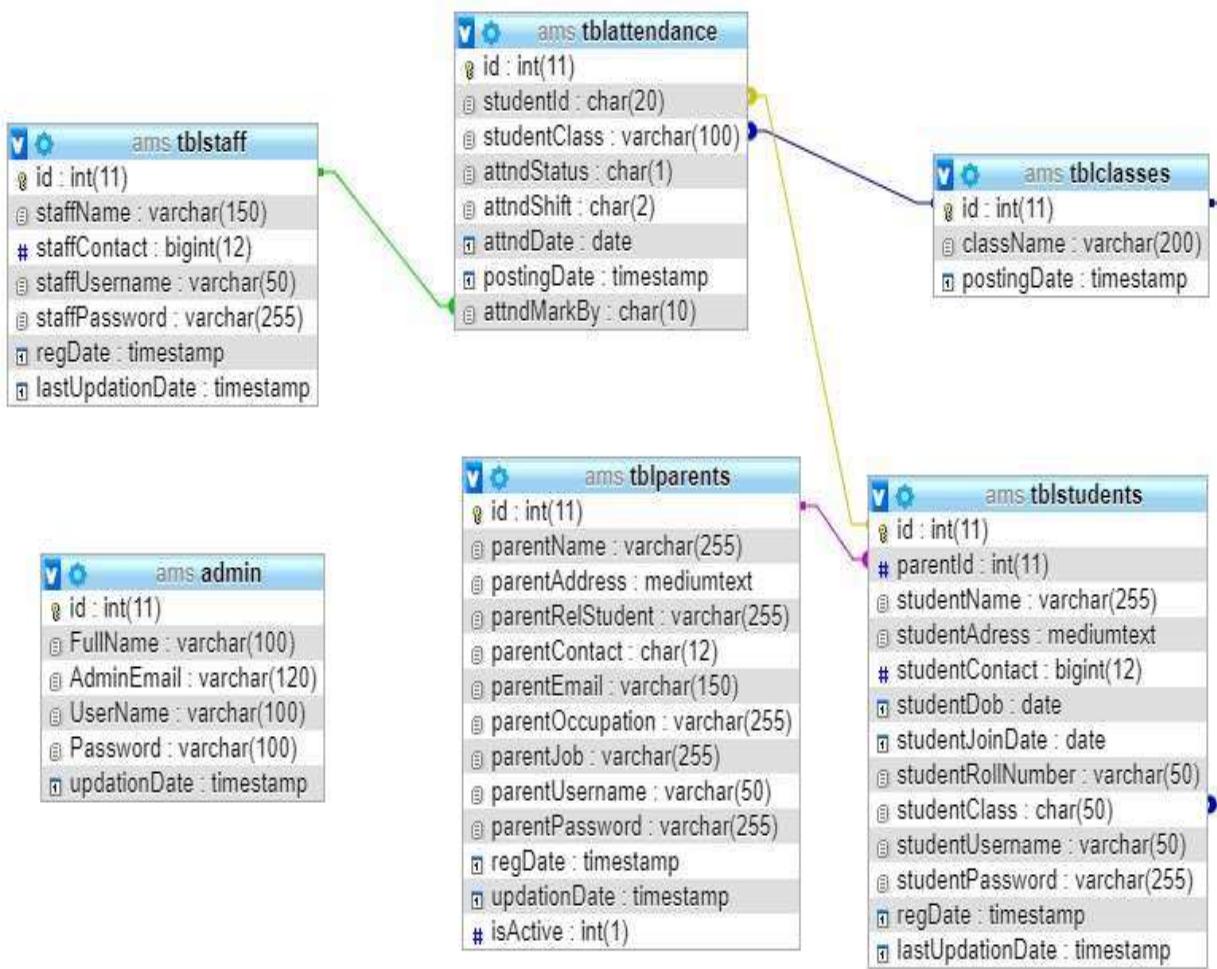


Fig : Student diagram.

Class Diagram:

A description of set of objects that share the same attributes operations, relationships, and semantics



ER Diagram:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen

wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

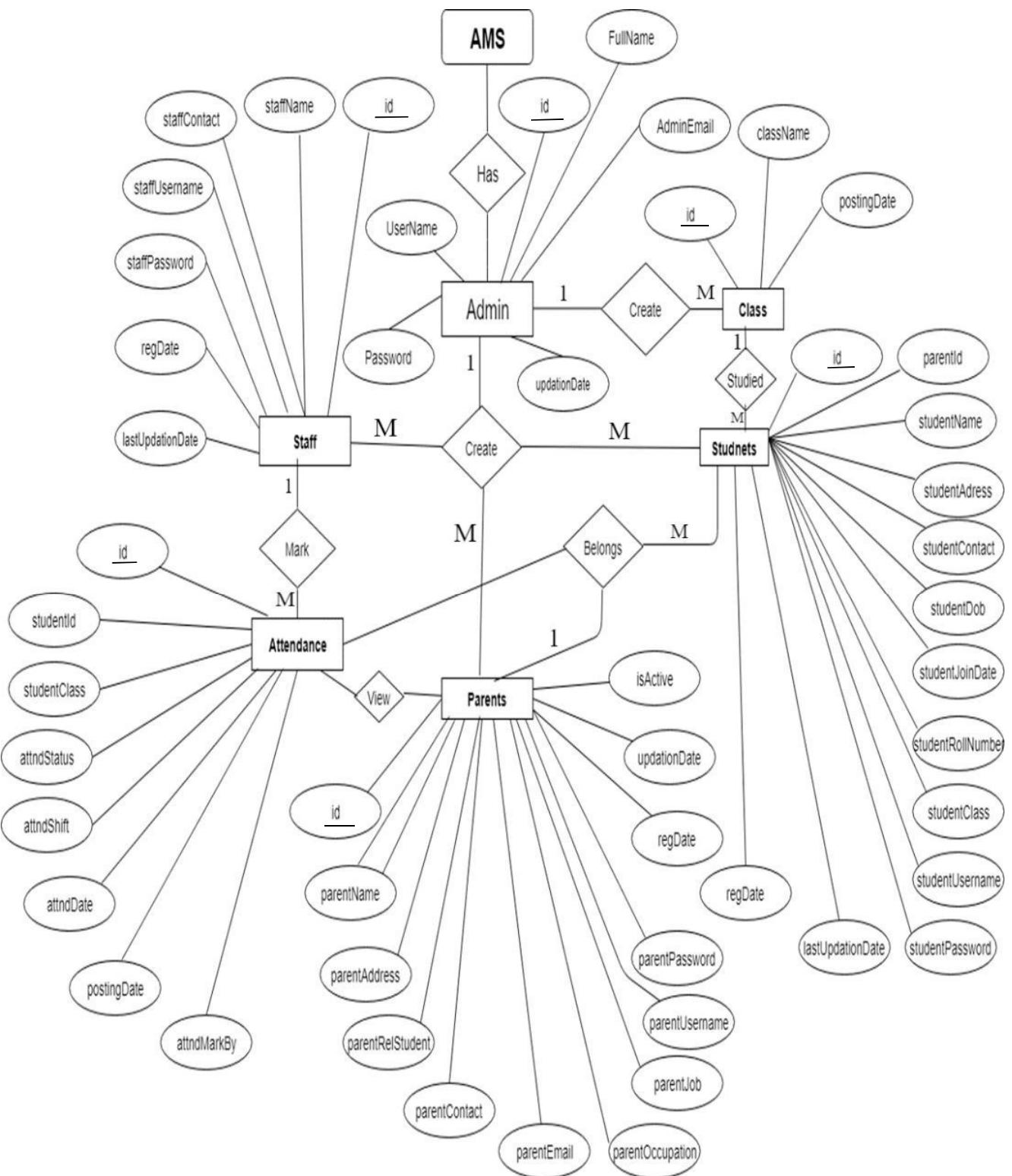
#### ER Notation:

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- Entities are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
- Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required.  
Optional existence is shown by placing a circle next to the entity that is optional.



## MySQL Data Tables

Admin Table:

The screenshot shows the MySQL Workbench interface for the 'admin' table. The table has six columns: id, FullName, AdminEmail, UserName, Password, and updationDate. The 'id' column is defined as int(11) with AUTO\_INCREMENT, 'FullName' as varchar(100), 'AdminEmail' as varchar(120), 'UserName' as varchar(100), 'Password' as varchar(100), and 'updationDate' as timestamp with a default value of 0000-00-00 00:00:00.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	FullName	varchar(100)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
3	AdminEmail	varchar(120)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
4	UserName	varchar(100)	latin1_swedish_ci		No	None			Change  Drop  More
5	Password	varchar(100)	latin1_swedish_ci		No	None			Change  Drop  More
6	updationDate	timestamp			No	0000-00-00 00:00:00		ON UPDATE CURRENT_TIMESTAMP()	Change  Drop  More

Buttons at the bottom include: Check all, With selected:, Browse, Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Add to central columns, Remove from central columns.

Table 1: admin

Class Table:

The screenshot shows the MySQL Workbench interface for the 'tblclasses' table. The table has three columns: id, className, and postingDate. The 'id' column is defined as int(11) with AUTO\_INCREMENT, 'className' as varchar(200), and 'postingDate' as timestamp with a default value of current\_timestamp().

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	className	varchar(200)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
3	postingDate	timestamp			Yes	current_timestamp()			Change  Drop  More

Buttons at the bottom include: Check all, With selected:, Browse, Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Add to central columns, Remove from central columns.

Table 2: tblclasses

## Staff Table:

The screenshot shows the MySQL Workbench interface for the 'tblstaff' table. The table has 7 columns: id, staffName, staffContact, staffUsername, staffPassword, regDate, and lastUpdateDate. The 'id' column is defined as int(11) with AUTO\_INCREMENT, 'staffName' as varchar(150), 'staffContact' as bigint(12), 'staffUsername' as varchar(50), 'staffPassword' as varchar(255), 'regDate' as timestamp with a default value of current\_timestamp(), and 'lastUpdateDate' as timestamp with a default value of ON UPDATE CURRENT\_TIMESTAMP(). Primary key constraints are applied to the 'id' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	staffName	varchar(150)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
3	staffContact	bigint(12)			Yes	NULL			Change  Drop  More
4	staffUsername	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
5	staffPassword	varchar(255)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
6	regDate	timestamp			Yes	current_timestamp()			Change  Drop  More
7	lastUpdateDate	timestamp			Yes	NULL		ON UPDATE CURRENT_TIMESTAMP()	Change  Drop  More

With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns  
 Remove from central columns

Table 3: tbstaff

## Students Table:

The screenshot shows the MySQL Workbench interface for the 'tbstudents' table. The table has 13 columns: id, parentId, studentName, studentAddress, studentContact, studentDob, studentJoinDate, studentRollNumber, studentClass, studentUsername, studentPassword, regDate, and lastUpdateDate. The 'id' column is defined as int(11) with AUTO\_INCREMENT, 'parentId' as int(11), 'studentName' as varchar(255), 'studentAddress' as mediumtext, 'studentContact' as bigint(12), 'studentDob' as date, 'studentJoinDate' as date, 'studentRollNumber' as varchar(50), 'studentClass' as char(50), 'studentUsername' as varchar(50), 'studentPassword' as varchar(255), 'regDate' as timestamp with a default value of current\_timestamp(), and 'lastUpdateDate' as timestamp with a default value of ON UPDATE CURRENT\_TIMESTAMP(). Primary key constraints are applied to the 'id' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	parentId	int(11)			Yes	NULL			Change  Drop  More
3	studentName	varchar(255)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
4	studentAddress	mediumtext	latin1_swedish_ci		Yes	NULL			Change  Drop  More
5	studentContact	bigint(12)			Yes	NULL			Change  Drop  More
6	studentDob	date			Yes	NULL			Change  Drop  More
7	studentJoinDate	date			Yes	NULL			Change  Drop  More
8	studentRollNumber	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
9	studentClass	char(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
10	studentUsername	varchar(50)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
11	studentPassword	varchar(255)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
12	regDate	timestamp			No	current_timestamp()			Change  Drop  More
13	lastUpdateDate	timestamp			Yes	NULL		ON UPDATE CURRENT_TIMESTAMP()	Change  Drop  More

With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns  
 Remove from central columns

Table 4: tblstudents

Parents Table:

The screenshot shows the 'tblparents' table structure in MySQL Workbench. The table has 13 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<b>id</b>	int(11)	latin1_swedish_ci	No	None		AUTO_INCREMENT		Change  Drop  More
2	<b>parentName</b>	varchar(255)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
3	<b>parentAddress</b>	mediumtext	latin1_swedish_ci	Yes	NULL				Change  Drop  More
4	<b>parentRelStudent</b>	varchar(255)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
5	<b>parentContact</b>	char(12)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
6	<b>parentEmail</b>	varchar(150)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
7	<b>parentOccupation</b>	varchar(255)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
8	<b>parentJob</b>	varchar(255)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
9	<b>parentUsername</b>	varchar(50)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
10	<b>parentPassword</b>	varchar(255)	latin1_swedish_ci	Yes	NULL				Change  Drop  More
11	<b>regDate</b>	timestamp		Yes	current_timestamp()				Change  Drop  More
12	<b>updationDate</b>	timestamp		Yes	NULL		ON UPDATE CURRENT_TIMESTAMP()		Change  Drop  More
13	<b>isActive</b>	int(1)		Yes	NULL				Change  Drop  More

Below the table, there are several buttons for managing the schema: Check all, With selected:, Browse, Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Add to central columns, Remove from central columns, and a scroll bar.

Table 5: tblparents

Attendance Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT		Change  Drop  More
2	studentId	char(20)	latin1_swedish_ci		No	None			Change  Drop  More
3	studentClass	varchar(100)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
4	attndStatus	char(1)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
5	attndShift	char(2)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
6	attndDate	date			Yes	NULL			Change  Drop  More
7	postingDate	timestamp			Yes	current_timestamp()			Change  Drop  More
8	attndMarkBy	char(10)	latin1_swedish_ci		Yes	NULL			Change  Drop  More

Table 6: tblattendance

## Implementation and System Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

### System Testing

The goal of the system testing process was to determine all faults in our project. The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing:

1. Unit testing
2. Integration testing

### Unit Testing

Unit testing is commenced when a unit has been created and effectively reviewed. In order to test a single module we need to provide a complete environment i.e. besides the section we would require :

- The procedures belonging to other units that the unit under test calls
- Non local data structures that module accesses
- A procedure to call the functions of the unit under test with appropriate parameters

#### 1. Test for the admin module

- Testing admin login form-This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.
- Student account addition- In this section the admin can verify student details from student academic info and then only add student details to main database it contains add and update buttons if user click add button data will be added to student database and if he clicks update button the student data will be updated
- Parent account addition- In this section the admin can add parent details to main database it contains add and update buttons if user click add button data will be added to parents database and if he clicks update button the parents data will be updated.

- Staff account addition- In this section the admin can add staff details to main database it contains add and update buttons if user click add button data will be added to staff database and if he clicks update button the staff data will be updated.
- Report Generation: admin can Generate report from the main database.

## 2. Test for Student login module

- Test for Student login Form- This form is used for log in of Student .In this student enter, username and password if details are correct student login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for these details

## 3. Test for Parents login module

- Test for Parents login Form- This form is used for log in of Parents. In this Parent enter, username and password if details are correct Parent login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for these details

## 4. Test for Staff login module-

- Test for Parents login Form- This form is used for log in of Staff. In this Staff enter, username and password if details are correct Staff login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for these details.

## Integration Testing

In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

## <User Interface >

Landing Page <http://localhost/ams>

### Login Screen:

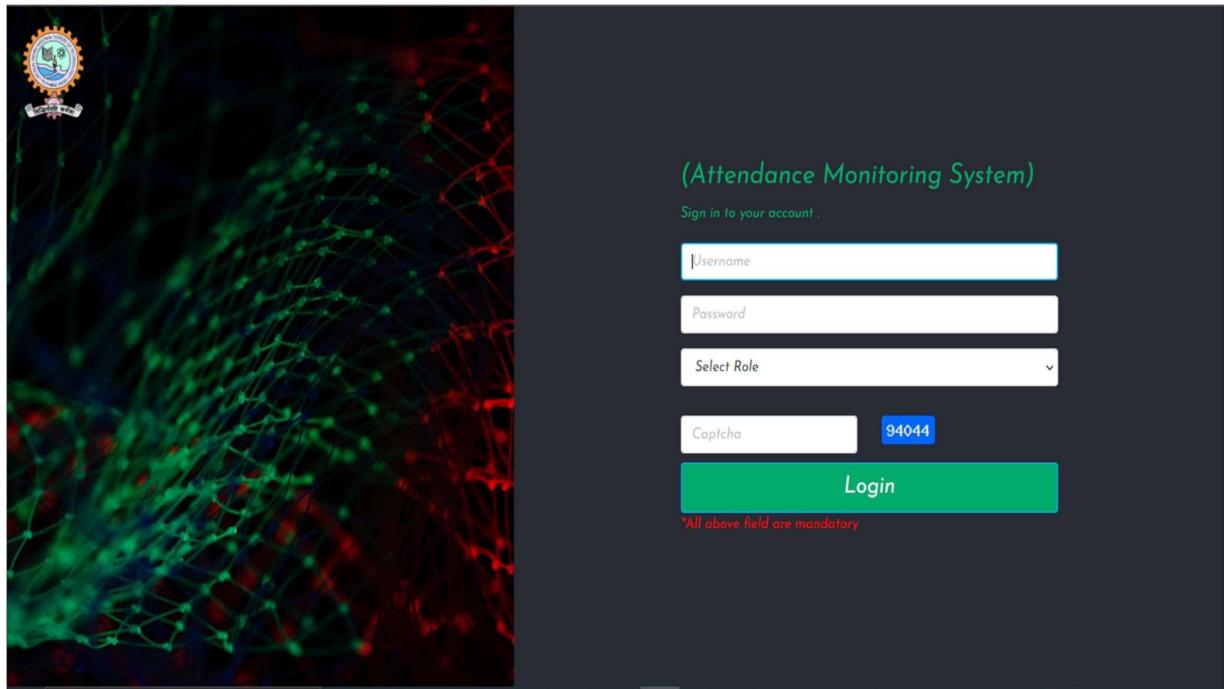


Fig. 1: Login page

## ADMIN MODULE

[Localhost/ams/admin](http://localhost/ams/admin)

Add Parent:

≡ (Attendance Monitoring System)

Admin

USER INTERFACE

- Dashboard
- Parents
- Students
- Staff
- Class
- Reports

## ② Add a Parent

Fill the info

Parent Name :	Parent Name
Parent Address :	Enter parent address
Relation with Student :	Parent relation with student
Parent Contact :	Parent Contact
Parent Email :	Parent Email
Parent Occupation :	Parent Occupation
Job :	job
Username :	username
Password :	Password

**Submit**

The screenshot shows a dark-themed web application interface. On the left is a sidebar with navigation links: Dashboard, Parents (selected), Students, Staff, Class, and Reports. The main content area has a title '② Add a Parent'. Below it is a form titled 'Fill the info' with various input fields for parent information. The fields include 'Parent Name', 'Parent Address', 'Relation with Student', 'Parent Contact', 'Parent Email', 'Parent Occupation', 'Job', 'Username', and 'Password'. A 'Submit' button is located at the bottom of the form.

Fig. 2.1: Admin can add a parent by filling the above details.

Manage Parents:

#	Parent Name	Parent Contact	Username	Reg Date	Action
1	Vikas Kumar Sharma	7979732664	vikas123	2022-07-12 01:25:46	
2	Mudit Jain	6386506592	mudit123	2022-07-12 01:32:30	
3	Radhika Karode	9111401605	radhika12	2022-07-12 01:40:54	

Fig. 2.2 :Admin can edit or delete the parent details.

#### Edit Parent:

Fill the info

Parent Name :	Vikas Kumar Sharma
Parent Address :	Bihar
Relation with Student :	Father
Parent Contact :	7979732664
Parent Emailid :	vikas21638@gmail.com
Parent Occupation :	Service
Job :	Debugger
Username :	vikas123
<b>Update</b>	

Fig. 2.3: Except username, any other detail can be edited.

#### Admin>student Add a

#### student:

**Add a Student**

**Fill the info**

Parent :	Select
Student Name :	Student Name
Student Address :	Enter parent address
Student Contact :	Student Contact
Date of Birth(DOB) :	yyyy-mm-dd
Join Date :	yyyy-mm-dd
Class :	Select
Roll Number :	Roll Number
Username :	username
Password :	Password
<b>Submit</b>	

Show desktc

Fig. 2.4: Admin can add a student by filling the below details.

### Manage Students:

**Manage Students Data**

**Students Details**

#	Student Name	Class	Roll No	Student Contact	Action
1	Jai Kishan	B.Tech	1	7979732664	
2	Rakesh Mandal	B.Tech	2	7979732664	
3	Deepak Singh Kushwaha	M.Sc	3	7979732664	
4	Tarun Verma	M.Sc	4	7979732664	
5	Suraj Prasad	M.Tech	5	7979732664	
6	Saloshi Trivedi	M.Tech	6	7979732664	
7	Sumit Verma	MBA	7	7979732664	
8	Ojasvi	MBA	8	7979732664	
9	Umanag Agrawal	MCA	9	7979732664	
10	Neha Joshi	MCA	10	7979732664	
11	Priya	Ph.D	11	7979732664	
12	Aman	Ph.D	12	7979732664	

Fig. 2.5: Admin can edit or delete the student details.

Edit Student:

**Fill the info**

Reg Date :	2022-07-12 01:59:31
Parent :	Mudit Jain
Student Name :	Jai Kishan
Student Address :	Allahabad
Student Contact :	7979732664
Date of Birth(DOB) :	2000-01-01 yyyy-mm-dd
Join Date:	2020-01-01 yyyy-mm-dd
Class :	B.Tech
Roll Number :	1
Username :	jai123

**Update**

Fig. 2.6: Except username, any other details can be edited.

### Admin>staff

#### Add a staff:

**Fill the info**

Staff Name :	Staff Name
Staff Contact :	Staff Contact
Username :	username
Password :	Password

**Submit**

Fig. 2.7: Admin can add a staff by filling the below details.

### Manage Staff:

The screenshot shows the 'Manage Staff Data' page of the Attendance Monitoring System. The left sidebar has a dark theme with white text and includes a 'USER INTERFACE' section with 'Dashboard', 'Parents', 'Students', 'Staff' (which is expanded), 'Class', and 'Reports'. The main content area has a light background and displays a table titled 'Staff Details' with the following data:

#	Staff Name	Staff Contact	Username	Reg Date	Action
1	RSY	9973376091	rsy123	2022-07-12 01:46:08	
2	AKS	9973376090	aks123	2022-07-12 01:47:07	
3	VKV	9973376090	vkv123	2022-07-12 01:48:18	
4	SSD	9973376090	ssd123	2022-07-12 01:48:56	
5	AG	9973376090	ag123	2022-07-12 01:49:25	

Fig. 2.8: Admin can edit or delete the staff details.

### Edit Staff:

The screenshot shows the 'Edit a Staff' page of the Attendance Monitoring System. The left sidebar has a dark theme with categories: Dashboard, Parents, Students, Staff, Class, and Reports. The main area shows the title 'Edit a Staff' and a form titled 'Fill the info'. It contains fields for 'Reg date' (2022-07-12 01:46:08), 'Last Updation date' (2022-07-13 21:07:05), 'Staff Name' (RSY), 'Staff Contact' (9973376091), and 'Username' (rsy123). A blue 'Update' button is at the bottom.

Fig. 2.9: Except username, any other detail can be edited.

### Admin>class

#### Add class:

The screenshot shows the 'Add a Class' page of the Attendance Monitoring System. The left sidebar has a dark theme with categories: Dashboard, Parents, Students, Staff, Class, and Reports. The main area shows the title 'Add a Class' and a form titled 'Fill the info'. It contains a field for 'Class Name' and a green 'Submit' button.

Fig. 2.10: Admin can add a new class by filling class name.

#### Manage Class:

The screenshot shows the 'Attendance Monitoring System' interface. On the left, a dark sidebar lists 'USER INTERFACE' categories: Dashboard, Parents, Students, Staff, Class (selected), and Reports. The main area is titled 'Class : Manage' and contains a sub-section titled 'Manage Class Data'. A table titled 'Class Details' lists six entries:

#	Class	Posting Date	Action
1	B.Tech	2022-07-12 01:41:52	
2	M.Tech	2022-07-12 01:42:29	
3	MCA	2022-07-12 01:43:02	
4	MBA	2022-07-12 01:43:32	
5	M.Sc	2022-07-12 01:43:48	
6	Ph.D	2022-07-12 01:44:13	

Fig. 2.11: Admin can edit a class name or can delete a class.

Edit Class:

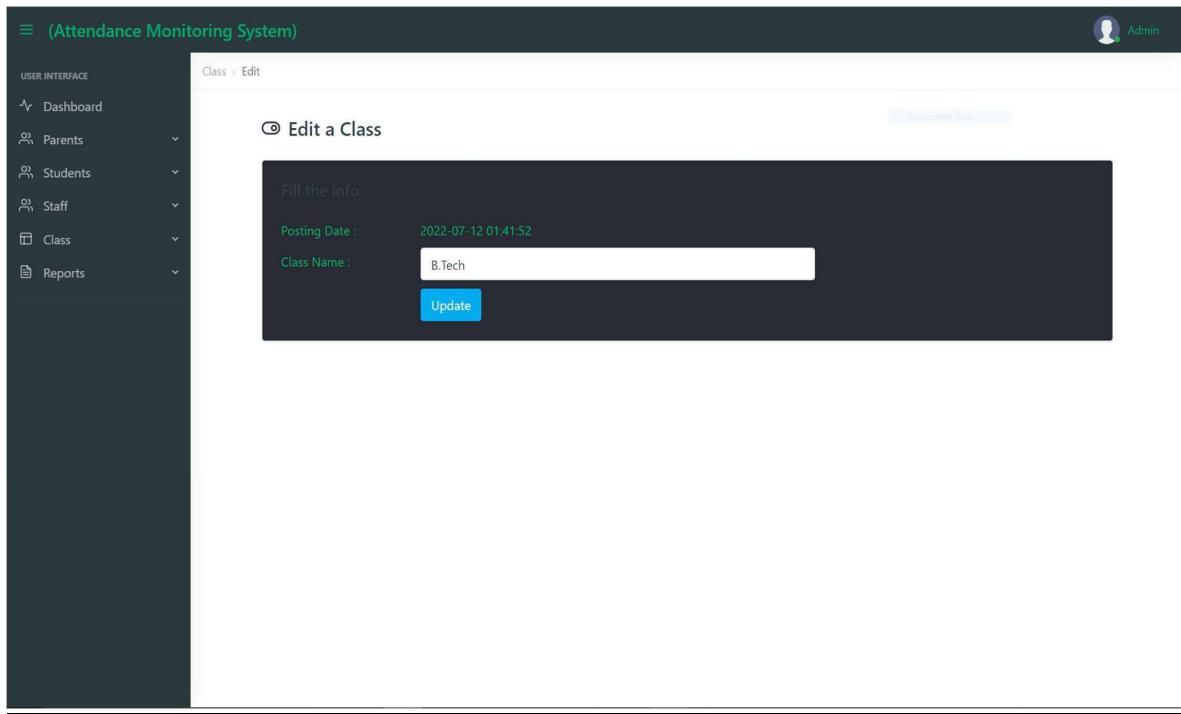


Fig. 2.12: Class Name can be edited.

## Reports

### Class Wise Report:

The screenshot shows the 'Attendance Monitoring System' interface. On the left, a dark sidebar menu titled 'USER INTERFACE' includes 'Dashboard', 'Parents', 'Students', 'Staff', 'Class', and 'Reports'. Under 'Reports', there are three options: 'Report by Class', 'Report by Name', and 'Graph Report'. The main content area is titled 'Report > Classwise Report Date selection'. It features a form titled 'Classwise Attendance Report' with three input fields: 'Class' (dropdown menu), 'From Date' (text input), and 'To Date' (text input). A 'Submit' button is at the bottom right of the form.

Fig. 2.13: Class wise report can be generated from a particular date to a particular date.

### Attendance report of a particular class:

The screenshot shows the 'Attendance Monitoring System' interface. The sidebar menu is identical to the previous screenshot. The main content area is titled 'Report > Classwise Report'. It displays a report titled 'Attendance Report of B.Tech class From 2022-07-10 To 2022-07-13'. Below the title is a table with the following data:

#	Date	Shift 1			Shift 2		
		Present	Late	Absent	Present	Late	Absent
1	2022-07-10	1	1	0	0	1	1
2	2022-07-11	1	0	1	1	1	0
3	2022-07-12	2	0	0	1	1	0
4	2022-07-13	2	0	0	0	1	1
<b>Total</b>		<b>6</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>4</b>	<b>2</b>

Fig. 2.14: Attendance report of a class.

Name Wise Report:

The screenshot shows a dark-themed web application interface. On the left is a sidebar with a tree menu under 'USER INTERFACE' labeled 'Dashboard', 'Parents', 'Students', 'Staff', 'Class', and 'Reports'. The main content area has a header 'Report > Studentwise Report Date selection'. Below this is a section titled 'Studentwise Attendance Report' with the following fields:

- Class :
- Student :
- From Date :
- To Date :

A green 'Submit' button is at the bottom right of the form.

Fig. 2.15: Name wise report can be generated from a particular date to a particular date.

Student Report:

The screenshot shows the same dark-themed web application interface as Fig. 2.14. The main content area has a header 'Report > Classwise Report'. Below this is a section titled 'Attendance Report of Jai Kishan B.Tech class student From 2022-07-10 To 2022-07-13' with a table:

#	Date	Shift 1	Shift 2
1	2022-07-10	Late	Late
2	2022-07-11	Present	Present
3	2022-07-12	Present	Late
4	2022-07-13	Present	Absent

Fig. 2.16: Attendance report of a particular student.

### Graph Report:

The screenshot shows a web-based application interface for an 'Attendance Monitoring System'. The top navigation bar includes a logo, the title 'Attendance Monitoring System', and a user icon labeled 'Admin'. A sidebar on the left, titled 'USER INTERFACE', contains a tree-view menu with categories: 'Dashboard', 'Parents', 'Students', 'Staff', 'Class', and 'Reports'. The main content area is titled 'Report > Classwise Graph Report Date selection' and features a sub-section titled 'Classwise Graph Report'. This section contains four input fields: 'Class' (dropdown menu with 'Select' option), 'Attendance Status' (dropdown menu with 'Present' option), 'From Date' (text input field with placeholder 'yyyy-mm-dd'), and 'To Date' (text input field with placeholder 'yyyy-mm-dd'). A green 'Submit' button is located at the bottom of these fields.

Fig. 2.17: Graphical report can be generated from a particular date to a particular date.

### Bar graph report:

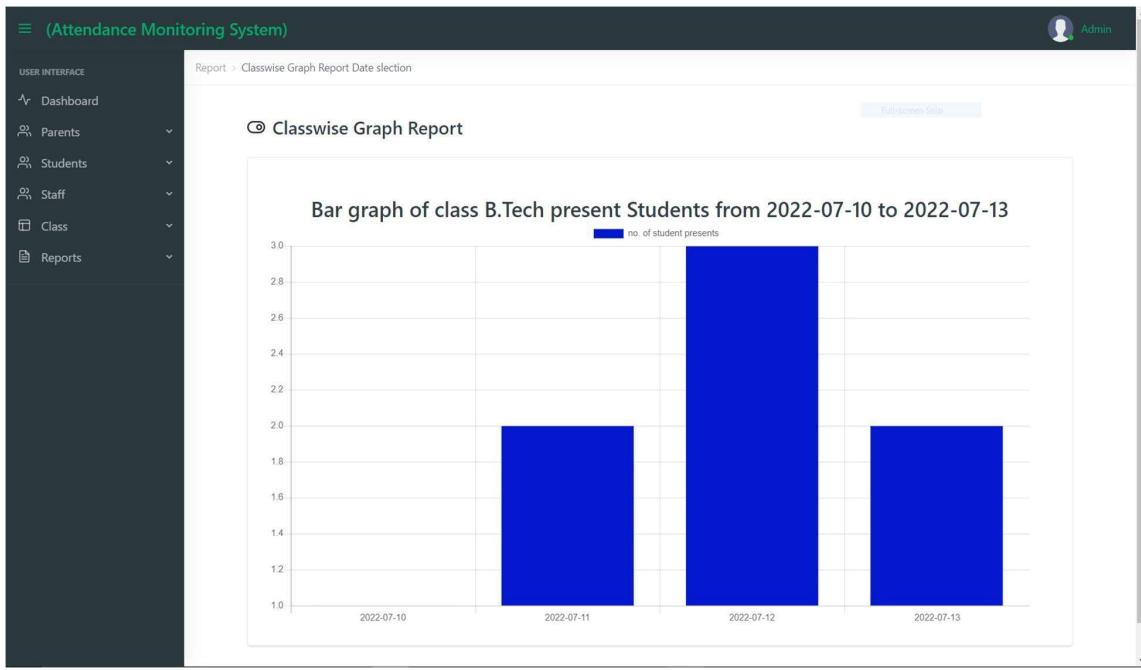


Fig. 2.18: Bar graph report

#### Admin Profile:

The screenshot shows the 'Update Profile' page for the admin. The sidebar menu on the left is identical to Fig. 2.18. The main content area is titled 'Update Profile' and contains a form titled 'Update the info'. It includes fields for 'Last Updation date' (2022-07-12 12:47:59), 'Name' (Mudit Jain), 'Admin Email' (mudit.2020ca044@mnnit.ac.in), and 'Username' (admin). A blue 'Update' button is at the bottom of the form. The top right corner shows the admin user icon and the word 'Admin'.

Fig. 2.19: Information about the admin

### Admin Change Password:

The screenshot shows a dark-themed web application interface for an 'Attendance Monitoring System'. At the top right, there is a user icon labeled 'Admin'. On the left, a vertical sidebar titled 'USER INTERFACE' contains navigation links: 'Dashboard', 'Parents', 'Students', 'Staff', 'Class', and 'Reports'. The main content area is titled 'Change Password' and contains three input fields: 'Current Password:', 'New Password:', and 'Confirm Password:'. Below these fields is a blue 'Change' button.

Fig. 2.20: Admin can change the password

## PARENT MODULE

[Localhost/ams/parent](http://localhost/ams/parent)

### Parent Profile:

The screenshot shows a dark-themed web application interface. On the left, a sidebar menu lists 'USER INTERFACE' with options: Report, My Profile, and Logout. The main content area has a header 'Parent > Add' and a sub-header 'Add a Parent'. A form titled 'Fill the info' contains the following fields with their values:

Parent Name :	Vikas Kumar Sharma
Parent Address :	Bihar
Relation with Student :	Father
Parent Contact :	7979732664
Parent Occupation :	Service
Job :	Debugger
Username :	vikas123

A blue 'Update' button is located at the bottom right of the form.

Fig. 3.1: Details of the parent

### Change Password:

The screenshot shows a dark-themed web application interface. On the left, a sidebar menu lists 'USER INTERFACE' with options: Report, My Profile, and Logout. The main content area has a header 'Parent > Change Password' and a sub-header 'Change Password'. A form contains three fields:

Current Password :	[Redacted]
New Password :	[Redacted]
Confirm Password :	[Redacted]

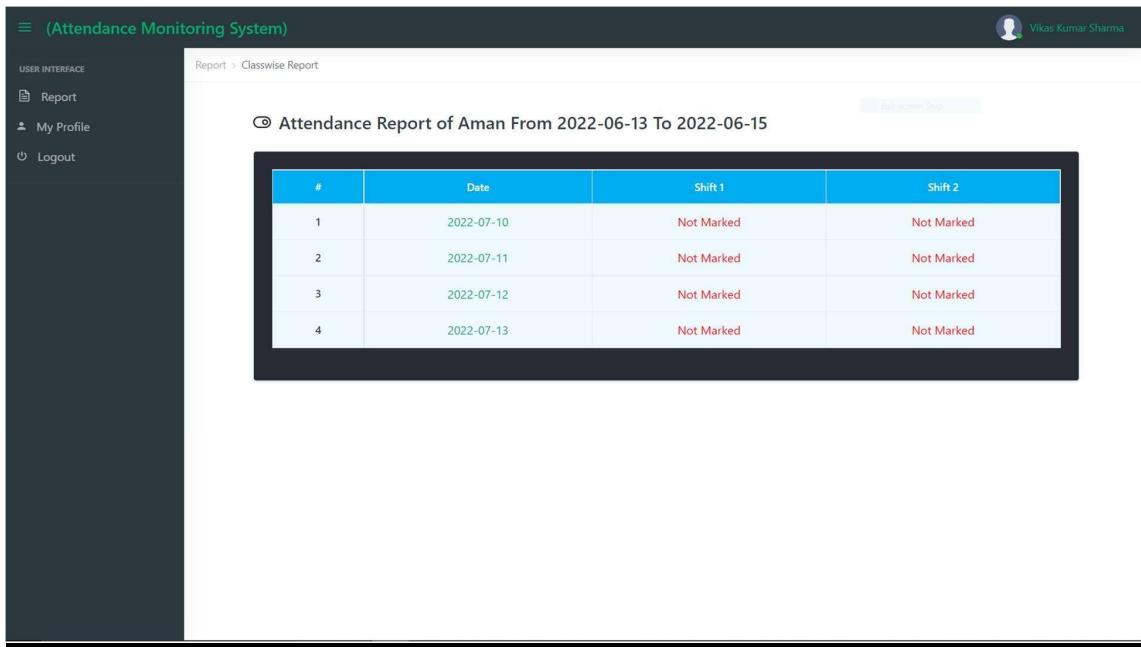
A blue 'Change' button is located at the bottom right of the form.

Fig. 3.2: Parent can change their password [Attendance Report:](#)

The screenshot shows a dark-themed web application interface. At the top left is the title '(Attendance Monitoring System)'. On the top right is a user profile icon labeled 'Mudit Jain'. A navigation bar on the left includes 'USER INTERFACE' and links for 'Report', 'My Profile', and 'Logout'. The main content area has a header 'Report > Studentwise Report Date selection'. Below this is a section titled 'Attendance Report' with a subtitle '(Attendance Report)'. It contains three input fields: 'Student:' with a dropdown menu showing 'Deepak Singh Kushwaha', 'From Date:' with the value '2022-07-11', and 'To Date:' with the value '2022-06-15'. A blue 'Submit' button is at the bottom of the form.

Fig. 3.3: Parent can check their child's attendance record by filling the above details

Attendance Report:



The screenshot shows the Attendance Monitoring System interface. The top navigation bar includes a logo, the title '(Attendance Monitoring System)', and a user profile for 'Vikas Kumar Sharma'. The left sidebar has a 'USER INTERFACE' section with 'Report' selected, along with links for 'My Profile' and 'Logout'. The main content area is titled 'Report > Classwise Report' and displays a report for student 'Aman' from June 13, 2022, to June 15, 2022. The report table has columns for '#', 'Date', 'Shift 1', and 'Shift 2'. The data shows four entries, all marked as 'Not Marked'.

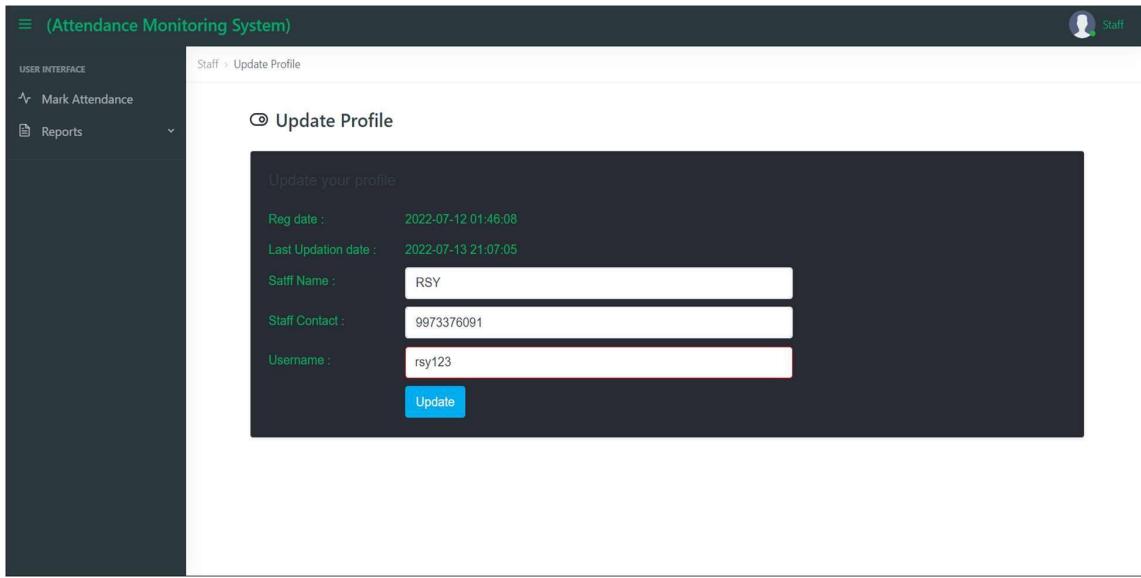
#	Date	Shift 1	Shift 2
1	2022-07-10	Not Marked	Not Marked
2	2022-07-11	Not Marked	Not Marked
3	2022-07-12	Not Marked	Not Marked
4	2022-07-13	Not Marked	Not Marked

Fig. 3.4: Results of the attendance record of that student

### STAFF MOUDLE

[localhost/ams/staff](http://localhost/ams/staff)

Staff Profile:



The screenshot shows the Staff Profile update form. The top navigation bar includes a logo, the title '(Attendance Monitoring System)', and a user profile for 'Staff'. The left sidebar has a 'USER INTERFACE' section with 'Mark Attendance' and 'Reports' selected. The main content area is titled 'Update Profile' and contains fields for 'Reg date', 'Last Updation date', 'Staff Name', 'Staff Contact', and 'Username'. The 'Staff Name' field is populated with 'RSY', 'Staff Contact' with '9973376091', and 'Username' with 'rsy123'. A blue 'Update' button is at the bottom.

Update your profile

Reg date : 2022-07-12 01:46:08  
Last Updation date : 2022-07-13 21:07:05

Staff Name : RSY

Staff Contact : 9973376091

Username : rsy123

Update

Fig. 4.1: Details of the particular staff after login

### Select Shift:

The screenshot shows the 'Attendance Monitoring System' interface. The top navigation bar includes a logo, the system name '(Attendance Monitoring System)', and a 'Staff' user icon. The left sidebar has a 'USER INTERFACE' section with 'Mark Attendance' and 'Reports' options. The main content area is titled 'Mark Attendance' and contains two dropdown menus: 'Class' (set to 'Select') and 'Shift' (set to 'Shift 1'). A green 'Submit' button is at the bottom.

Fig. 4.2: Staff will select the class and shift for attendance

### Mark Attendance:

This screenshot continues from Fig. 4.2. It shows the 'Mark Attendance' page with the same dropdown menus for 'Class' and 'Shift'. Below the dropdowns is a table listing student attendance data. The table has columns for #, Student Name, Class, Roll Number, and Status. Two rows are shown: one for Deepak Singh Kushwaha (M.Sc, Roll 3) with status OP, OL, OA, and another for Tarun Verma (M.Sc, Roll 4) with status OP, OL, OA. At the bottom of the table is a green 'Submit' button.

#	Student Name	Class	Roll Number	Status
1	Deepak Singh Kushwaha	M.Sc	3	OP    OL    OA
2	Tarun Verma	M.Sc	4	OP    OL    OA

Fig. 4.3: Staff can mark the student as either present, late or absent

## STUDENT MOUDLE

[Localhost/am/student](#)

Dashboard:

The screenshot shows the Student Dashboard of the Attendance Monitoring System. At the top, there is a header bar with the title '(Attendance Monitoring System)' and a user profile icon for 'Jai Kishan'. On the left, a sidebar titled 'USER INTERFACE' contains links for 'Dashboard', 'Report', 'My Profile', and 'Logout'. The main content area displays attendance statistics for the last 30 days across six categories: Shift 1 (3 Day Present), Shift 1 (1 Day Late), Shift 1 (0 Day Absent), Shift 2 (1 Day Present), Shift 2 (2 Day Late), and Shift 2 (1 Day Absent). Below this, a table titled 'Your Last 10 attendance' shows four entries with columns for '#', 'Date', 'Shift 1', and 'Shift 2'.

#	Date	Shift 1	Shift 2
1	2022-07-10	Late	Late
2	2022-07-11	Present	Present
3	2022-07-12	Present	Late
4	2022-07-13	Present	Absent

Fig. 5.1: Dashboard shows attendance record of last 30 days

Edit Profile:

≡ (Attendance Monitoring System)

User Interface

- Dashboard
- Report
- My Profile
- Logout

>Edit profile

Update the info

Reg Date :	2022-07-12 01:59:31
Parent :	Mudit Jain
Student Name :	Jai Kishan
Student Address :	Allahabad
Student Contact :	7979732664
Date of Birth(DOB) :	2000-01-01 yyyy-mm-dd
Join Date:	2020-01-01 yyyy-mm-dd
Class :	B.Tech
Roll Number :	1
Username :	jai123

**Update**

Fig. 5.2: Student can edit their profile after login

### Student Report:

≡ (Attendance Monitoring System)

User Interface

- Dashboard
- Report
- My Profile
- Logout

Report > Classwise Report

Attendance Report of Jai Kishan class student From 2022-07-10 To 2022-07-14

#	Date	Shift 1	Shift 2
1	2022-07-10	Late	Late
2	2022-07-11	Present	Present
3	2022-07-12	Present	Late
4	2022-07-13	Present	Absent

Fig. 5.3: Attendance report from a particular date to a particular date

### Change Password:

The screenshot shows a dark-themed web interface for changing a password. On the left, a sidebar lists 'USER INTERFACE' with options like 'Dashboard', 'Report', 'My Profile', and 'Logout'. The main content area is titled 'Change Password'. It features three input fields: 'Current Password', 'New Password', and 'Confirm Password', each with a corresponding label above it. Below these fields is a blue rectangular button labeled 'Change'.

Fig. 5.4: Student can change his/her password after login

### CONCLUSION

This Application provides a computerized version of Attendance Monitoring System which will benefit the students as well as the staff. It makes entire process online where student and parents can view attendance and can generate reports. It also has a facility for students and parents login where student and parent can login and can see attendance on the daily basis. It has a facility of staff's login where staff can mark the of particular and shift. The Application was designed in such a way that future changes can be done easily. The following conclusions can be deduced from the development of the project:

- Automation of the entire system improves the productivity.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.

- The System has adequate scope for modification in future if it is necessary.
- Currently the entire project is running on XAMPP sever Apache. Later the project will be shifted to the php web hosting website like DigitalOcean.
- We are planning to add more features using already collected data.

## References

For PHP

1. <https://www.w3schools.com/php/default.asp>
2. <https://www.sitepoint.com/php/>
3. <https://www.php.net/>

For MySQL

1. <https://www.mysql.com/>
2. <http://www.mysqltutorial.org>

For XAMPP

1. <https://www.apachefriends.org/download.html>