PROJECT REPORT

ON

E-SCHOOL MANAGEMENT SYSTEM

FOR

REOTRIX Pvt.Ltd.

IN PARTIAL FULFILMENT OF

MASTER IN COMPUTER APPLICATION

BY

ISHWAR DHOTE

SUBMITTED TO

UNIVERSITY OF PUNE

MASTER IN COMPUTER APPLICATION

MAHARASHTRA EDUCATION SOCIETY's

INSTITUTE OF MANAGEMENT AND CAREER COURSES

(IMCC), PUNE-411029

Acknowledgement

I wish to record my heartfelt gratitude and sincere to **Dr. V. H. Inamdar, Director IMCC**. I am sincerely thankful to **Dr. S. D. Deshpande, H.O.D.** for their timely support and guidance. I also express sincere thanks to **Dr.Manasi Bhate**, **Head T.P.O** for her encouragement, support and valuable guidance.

I would specially like to thank **Mrs. Minakshi More** our internal guide and **Mr. Krishna Maheshwari** who motivated, inspired and supported me in my project development and academic year.

Last but not the least my sincere thanks to my parents, friends and family members for their continuous support, inspiration and encouragement without which this project would not have been success.

Thanking you,

ISHWAR DHOTE

Index

Sr.No	Title	Page No
1	**Introduction**	
	1.1 Company profile	1
	1.2 Existing system and need for system	3
	1.3 Scope of work	5
	1.4 Operating Environment – Hardware and Software	6
	1.5 Detail Description of Technology used	7
2	**Proposed system**	
	2.1 Proposed system	12
	2.2 Objectives of system	13
	2.3 User requirements	14
3	**Analysis & Design**	
	3.1 Object diagram	15
	3.2 Class diagram	16
	3.3 Use Case diagram	17
	3.4 Activity diagram	21
	3.5 Sequence diagram	26
	3.6 Entity Relationship diagram	28
	3.7 Module Hierarchy diagram	29
	3.8 Component diagram	30
	3.9 Deployment diagram	31
	3.10 Module Specification	32
	3.11 Web Site Map diagram	35
	3.12 User Interface design	36
	3.13 Data Dictionary	48
	3.14 Table specifications	51
	3.15 Test procedures and implementation	61
4.	**User manual**	
	4.1 User manual	67
	4.2 Operations manual / Menu Explanation	73
	4.3 Program Specification	75

5	**Drawbacks and Limitations**	79
6	**Proposed Enhancement**	80
7	**Conclusions**	81
8	**Bibliography**	
	ANNEXURES:	
	ANNEXURE 1:User Interface Screens	
	ANNEXURE 2:Data Output Reports	
	ANNEXURE 3:Sample Program Code	

CHAPTER 1: INTRODUCTION

1.1 Company profile:-

Technologies pvt.ltd

Reotrix technologies started in 2014, in Pune, India. We are an emerging organization which is providing IT solutions to many businesses. We have expertise on various solutions from Web designing to Digital Marketing. We are leading provider of computer software to consumers and small businesses across a wide range of computing and communications devices and software. We offer services of entire software, web development, Content Management Solutions and Creative Design from the very beginning until the end. Our overall process includes concept, design, development, incorporation, and implementation. We have a squad of experts, who are working on modern technologies and tools with vast experience to carry out web-based projects, e-commerce, mobile applications

and programming scripting languages. Our development team strive hard to meet the client requirements through consistent efforts.

We believe in transparent business based on ethical protocols practice. Customer satisfaction has always been the key to our success. We started as a beginner in India, completed many projects with meeting the timelines, customer satisfaction, and passion towards it. The credibility has been well established with the list of satisfied customers ranging from corporate companies to E-commerce establishments. We make your presence on The Internet with a corporate identity and strong online marketing in major search engines. Crossing over the geographical boundaries, we can work on the website design, website development, web marketing, e-commerce solutions and software development for organizations based in any location in global destinations of USA, Canada, UK, Australia and so on. We have the expertise and technology to serve anyone based anywhere in the world.

1.2 Existing System:-

Under the existing system there was no any medium to make direct communication between parents and administration. To inform about status and study reports to particular parents of a particular student, administration have to write letter to make them inform. Students have to check their notice board to get information about details about their routine, exam time tables, about their results and marks scored in particular exam subject wise.

There are manual interactions between parents and teachers which are causing problems in establishing a good environment for transparency. In order to solve this problem this e school management software will be helpful. School management will update every students details that are taken place in school with marks, attendance etc to database and provide user name and password for every student.

1.2 Need for System:-

Existing system is manual system. This makes it slow. And record of information are available, but on paper. If anyone wants to some old information, then it again consumes large amount time.

And if parents of students wanted to know something about child's progress, then they have to come to the school.

So as considering above situations, a web based "E School Management System" is needed. This web application will store all information and records online. And parents can see and check their child's progress online i.e. on schools web site.

1.3 Scope of Work

E-School Management System is a web-based School Management Application. It will design for better interaction between students, teachers, parents & management. The main purpose of building this application is, the parents of Students are very busy nowadays, so they can't be able to monitor their children and their Activities properly and regularly. This E School management system will help the parents to monitor their children from anywhere. They can be able to check their children's academic performance from a remote location.

Some the modules are as given below:-

- Manage Student
- Manage Parent
- Manage Teacher
- Manage Exam
- Manage Daily Attendance
- Manage Subject

1.4 Operating Environment:-

Hardware Requirements:

- ✓ Processor type: Minimum: Intel P4
- ✓ RAM: Minimum: 1GB
- ✓ Processor speed: 1 GHz or higher
- ✓ Hard disk: 160 GB or more

Software Requirements:

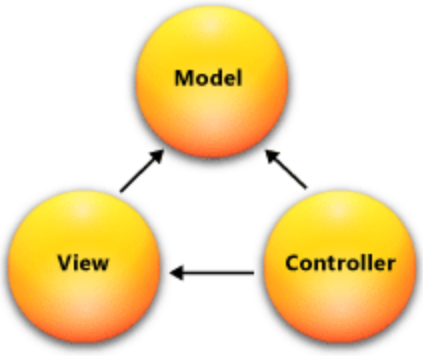
- ✓ Front End : - Visual studio 2010 (ASP.Net with C#.NET)
- ✓ Back End : - Microsoft SQL Server 2008

1.5 Detail Description of Technology Used:-

MVC- 3.0:-

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentication.

MVC design pattern



MVC is a standard design pattern that many developers are familiar with.

The MVC pattern helps you create applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. The pattern specifies where each kind of logic should be located in the application.

- The UI logic belongs in the view.
- Input logic belongs in the controller.
- Business logic belongs in the model.

This separation helps you manage complexity when you build an application, because it enables you to focus on one aspect of the

implementation at a time. For example, you can focus on the view without depending on the business logic.

Three tier Architecture:-

Three-tier architecture is a client–server architecture in which the user interface (presentation), functional process logic ("business rules"),computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. The three-tier model is a software architecture pattern.

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the *presentation tier* would only affect the user interface code.

Typically, the user interface runs on a

desktop PC or workstation and uses a standard graphical user interface, functional process logic that may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe that contains the computer data storage logic.

Business logic Layer:-

Business logic could be anywhere in a program. Business logic often changes. Thus it is often seen as desirable to make the code that implements the business logic relatively isolated, or loosely coupled. This makes it more likely that changes to business logic will require a small set of code changes, in only one part of the code.

Data Access Layer:-

A **data access layer** (DAL) is a layer of a computer program which provides simplified access to data stored in persistent storage of some kind, such as an entity-relational database. This acronym is prevalently used in Microsoft ASP.NET environments.

Service layer:-

A s**ervice layer** is a conceptual layer within network service provider architecture. It aims at providing middleware that serves third-party value-added services and applications at a higher application layer. The concept of service layer is used in contexts such as intelligent networks .

CHAPTER 2: PROPOSED SYSTEM

2.1 Proposed System

- Proposed system is E school Management System that manage school activity over the internet.
- This proposal will be developed for school. It is web based application.
- The administrator of the system is operations dashboard which is accessible to operations user.
- Operations dashboard where the administrator can add different other users like student, teacher, parents etc.
- Parents can monitor their children's activity.
- Parent checks their children academic performance from a remote location
- Administrator will add attendance of students, marks, assignment into system and update this into database.
- Administrator responsible for tracking student activity and updating.

2.2 Objectives of System-

- Track the progress activity of the student.

- Parent can easily look at their children activity.

- Parents of students are very busy now days so they can monitor their children activity properly and regularly from any remote location.
- To make better interaction between student, teacher and parents.
- Parents can check their children's academic performance from remote location.

- Administrator will make all updating in system.

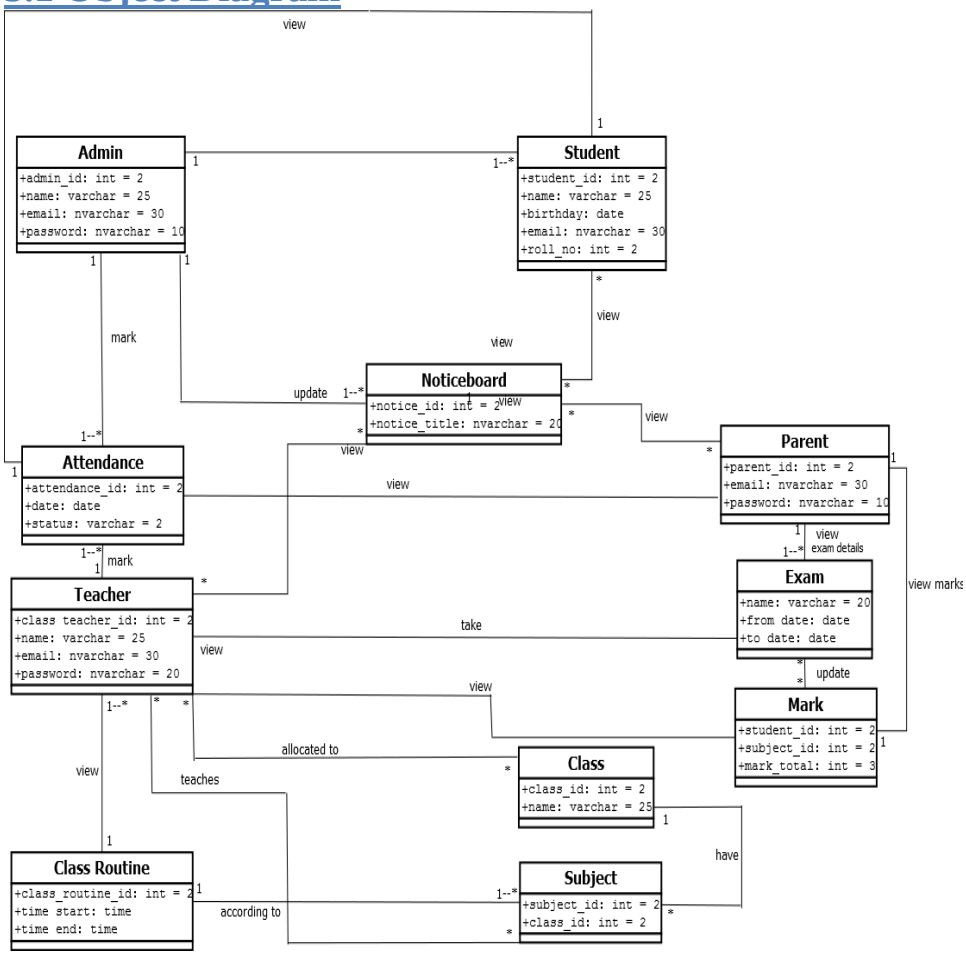
- Parents can track their children activity over internet.

2.3 User Requirements-

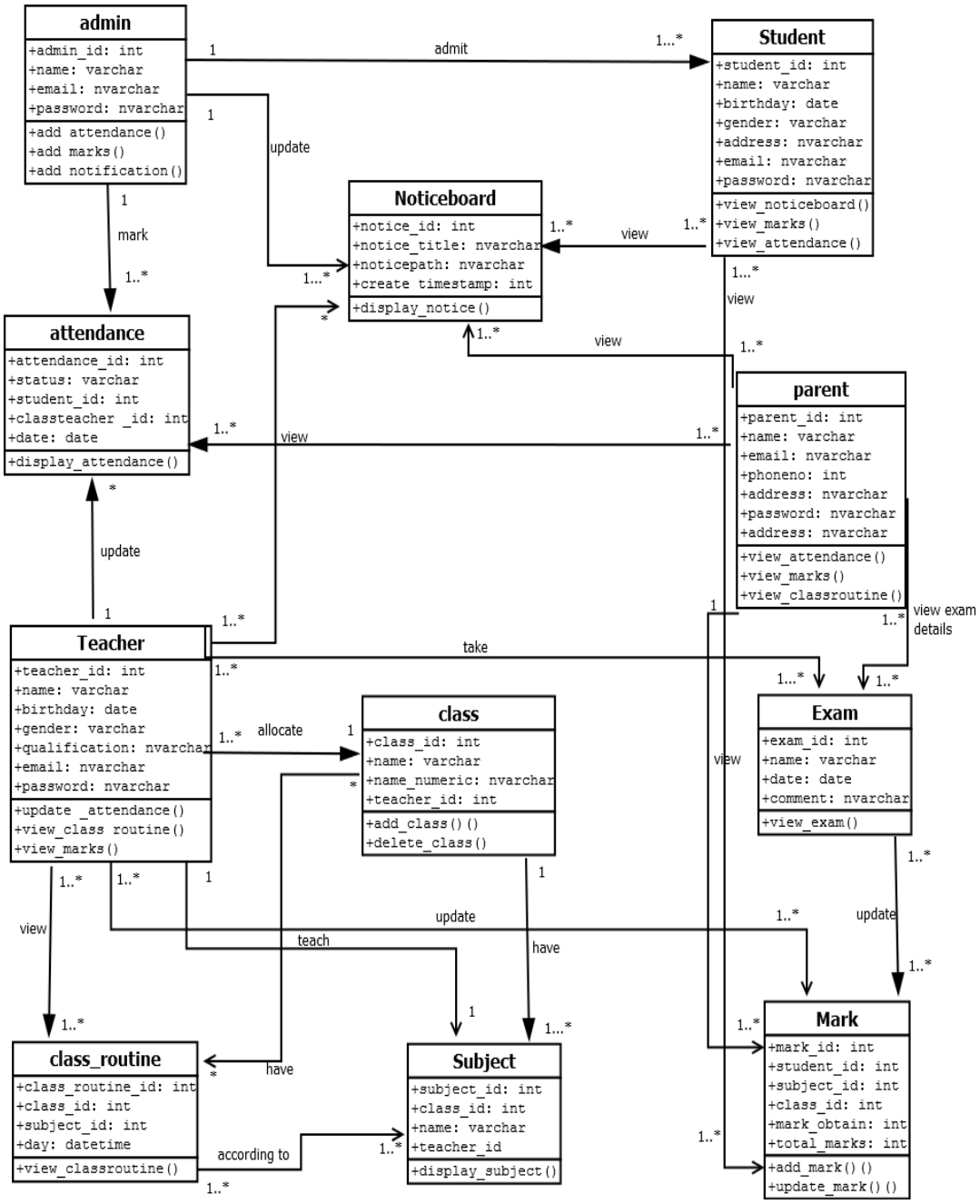
- To make job of parents easier. So that the parents can keep eye on their child's progress.
- To keep all the record and information online rather than keeping it on paper.
- To have communication with each other (i.e. parents and teacher) whenever required.
- Organization need to able to handle all updating activity by administrator.
- Parents need to track the attendance, marks, and assignments of their child.
- Organization need to track exam time table of the student.
- Organization need to display holiday list.
- A platform which can be accessible all over the world.

CHAPTER 3: ANALYSIS AND DESIGN

3.1 Object Diagram

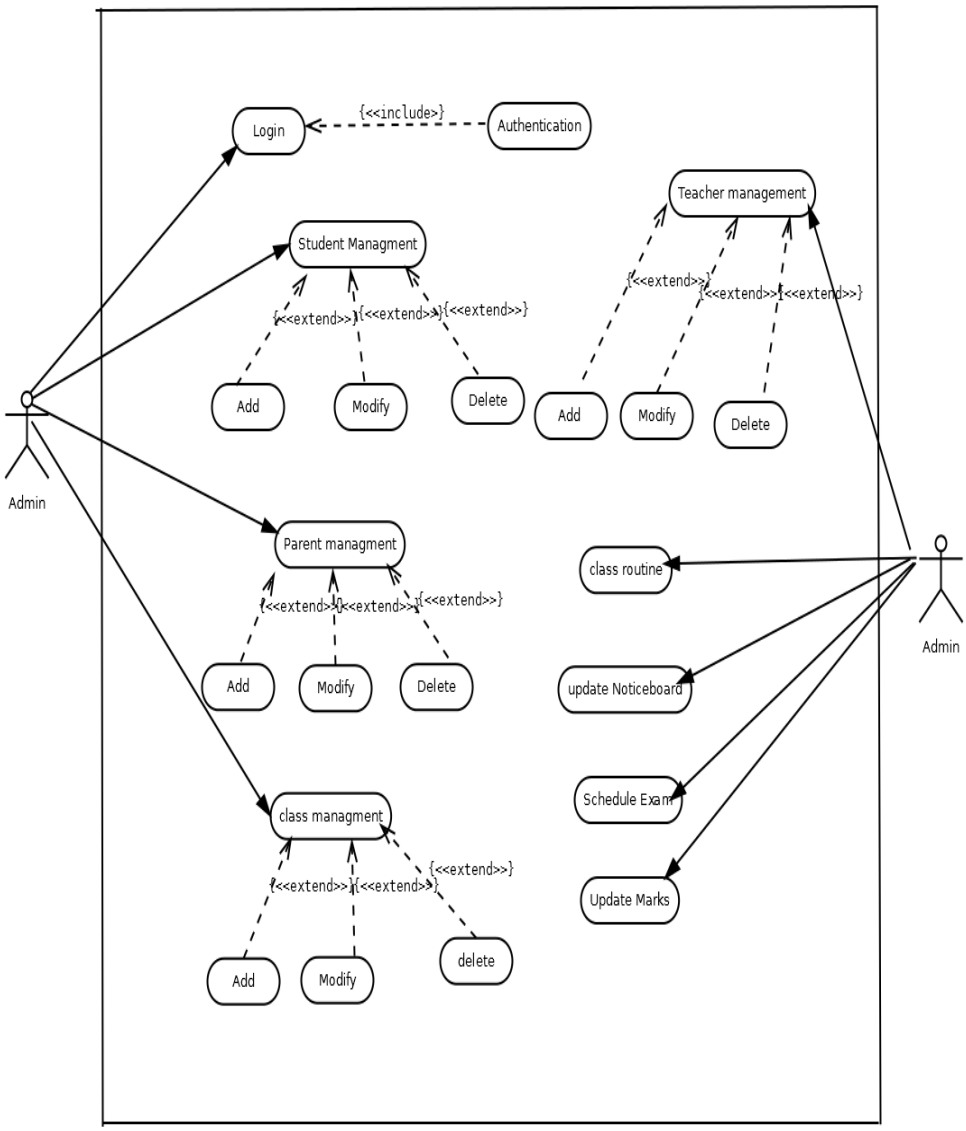


3.2 Class Diagram

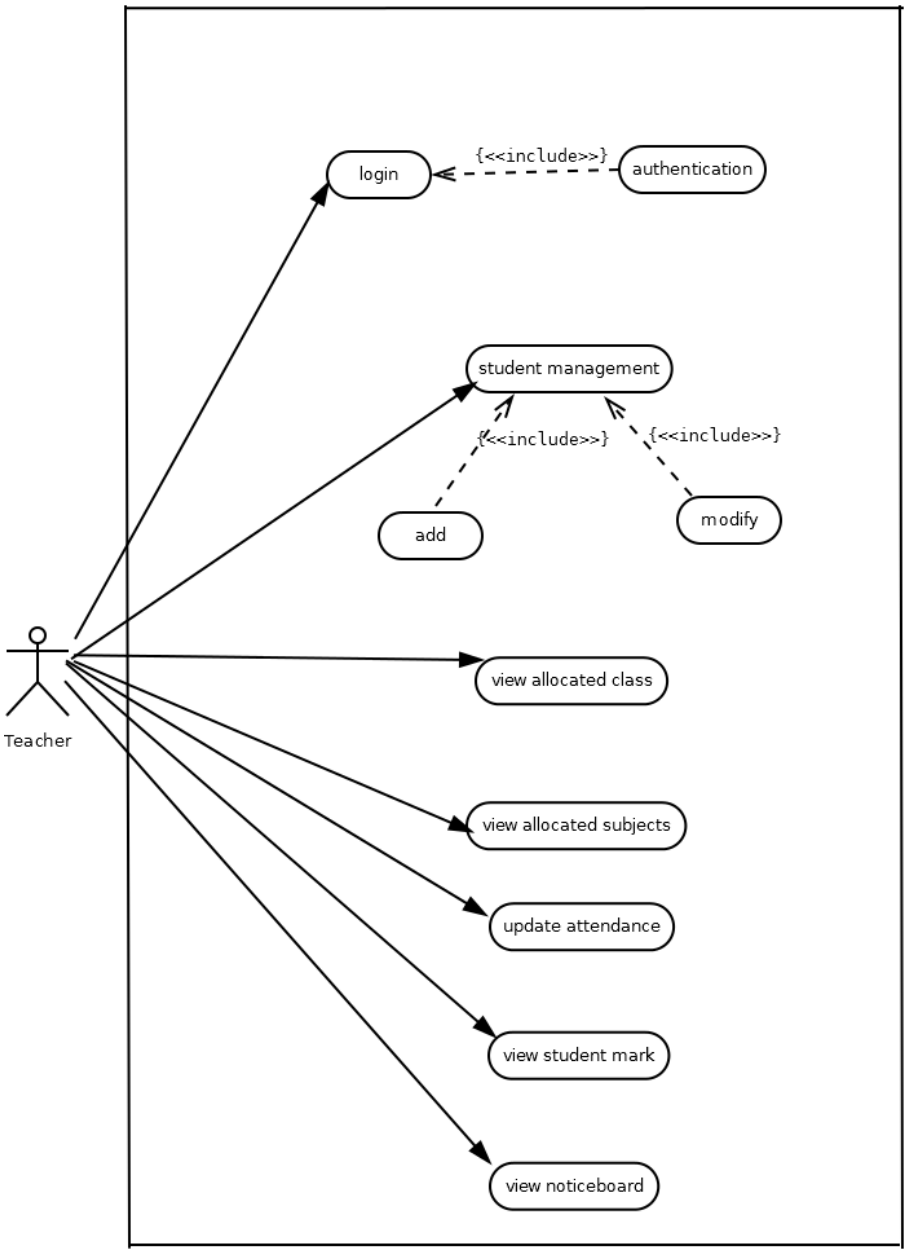


3.3 Use case Diagram

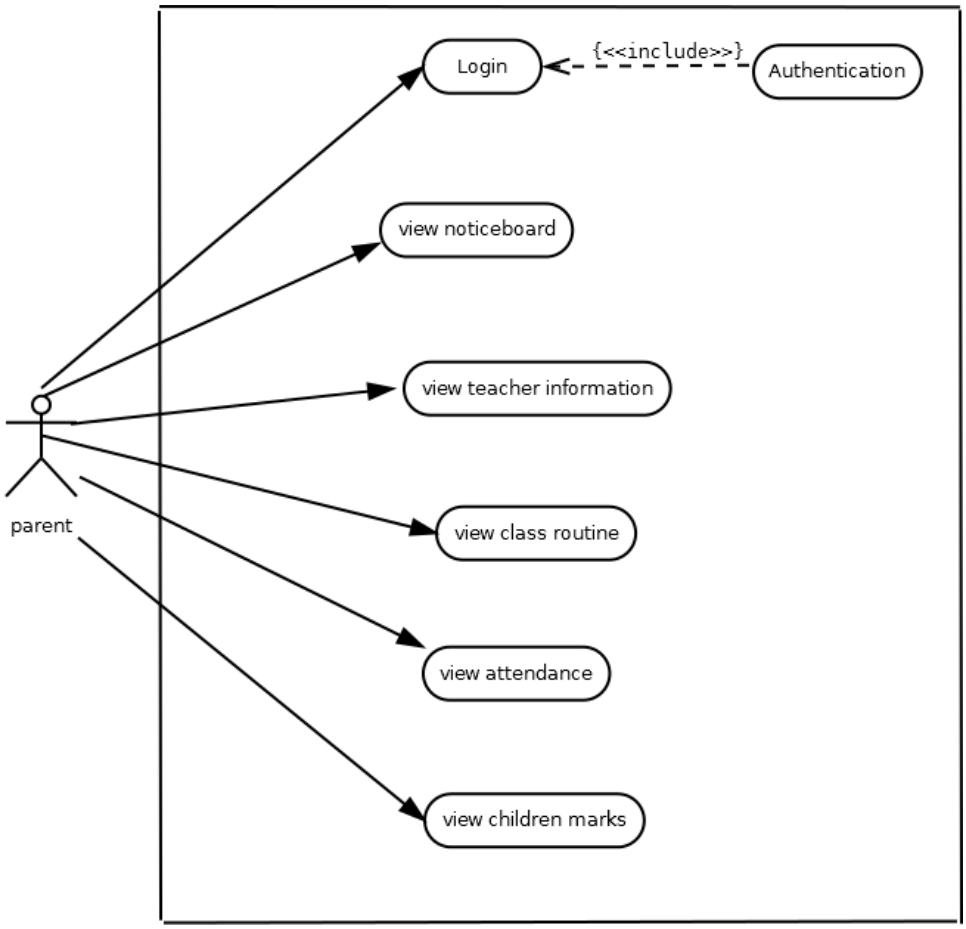
Admin:



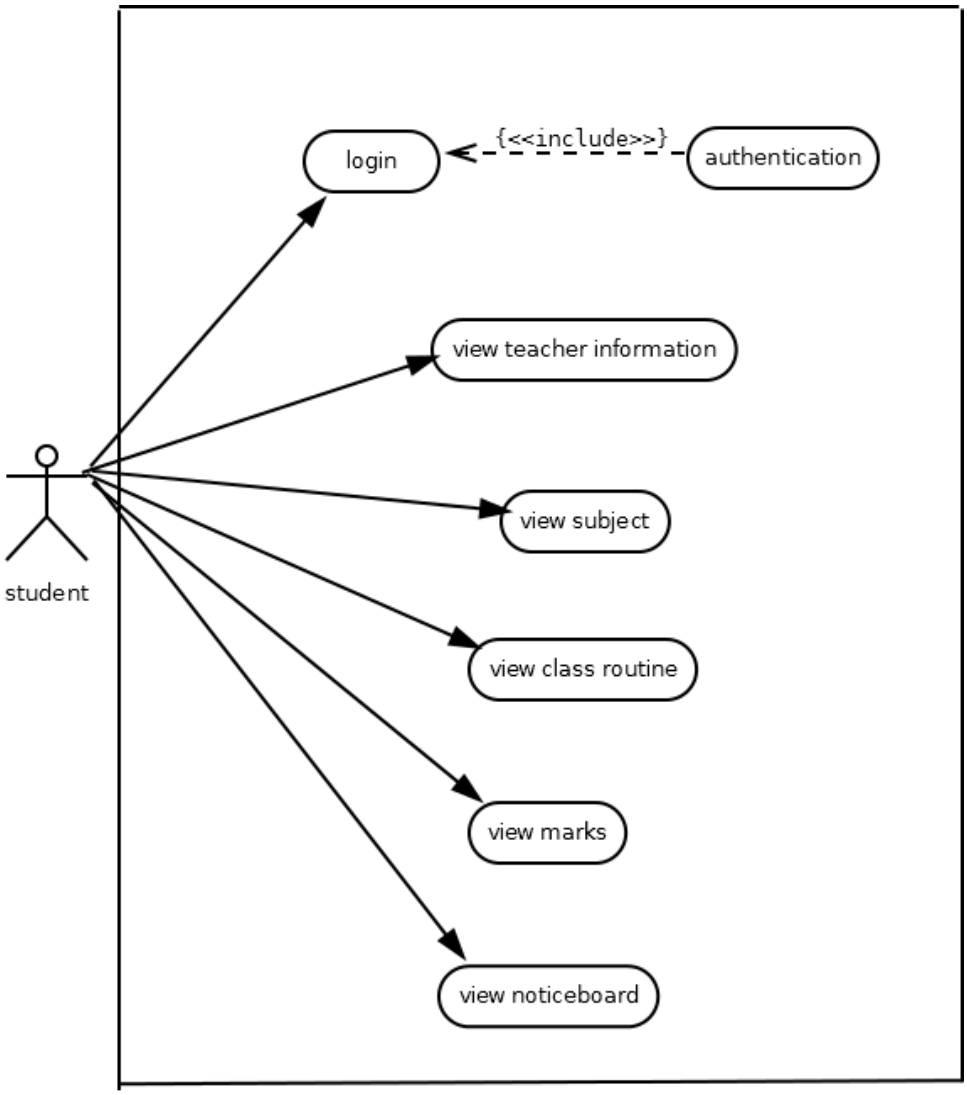
Teacher:


login
{<<include>>}
authentication
student management
{<<include>>}
{<<include>>}
add
modify
Teacher
view allocated class
view allocated subjects
update attendance
view student mark
view noticeboard

Parent:

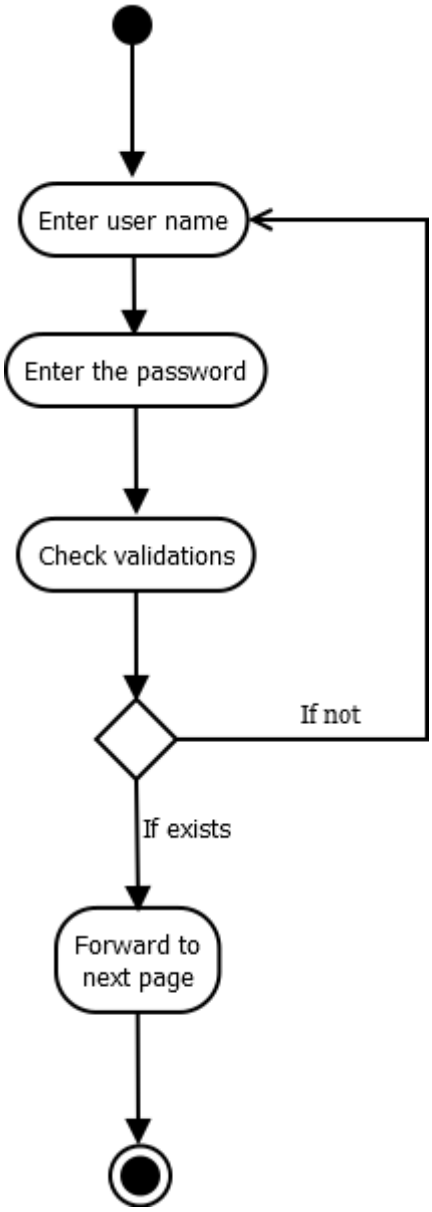

Login
{<<include>>}
Authentication
view noticeboard
view teacher information
parent
view class routine
view attendance
view children marks

Student:

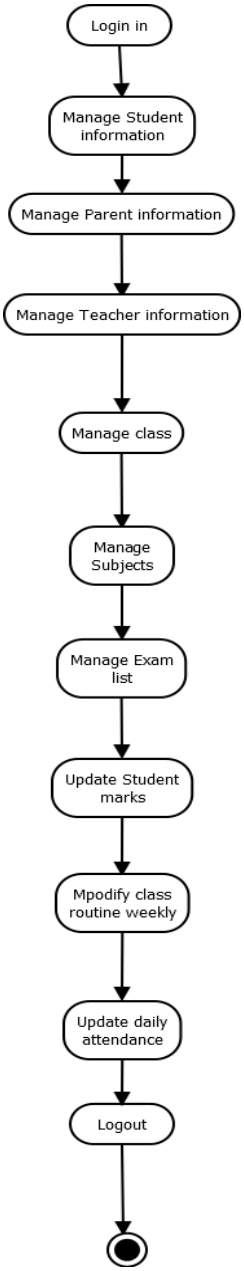

login
{<<include>>}
authentication
view teacher information
view subject
view class routine
view marks
view noticeboard
student

3.4 Activity Diagram:

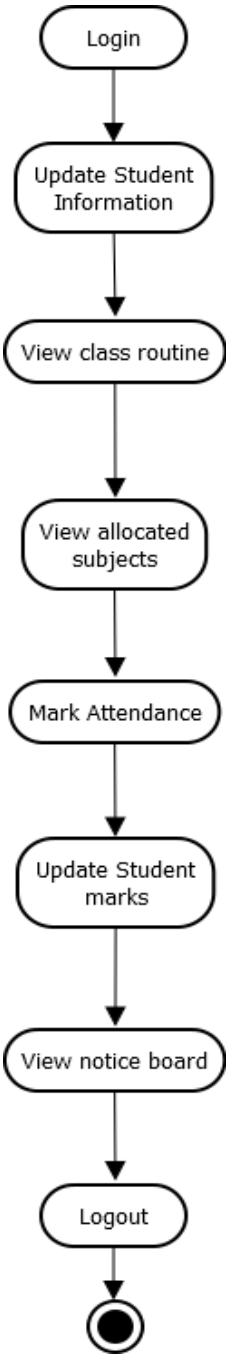
Login Form



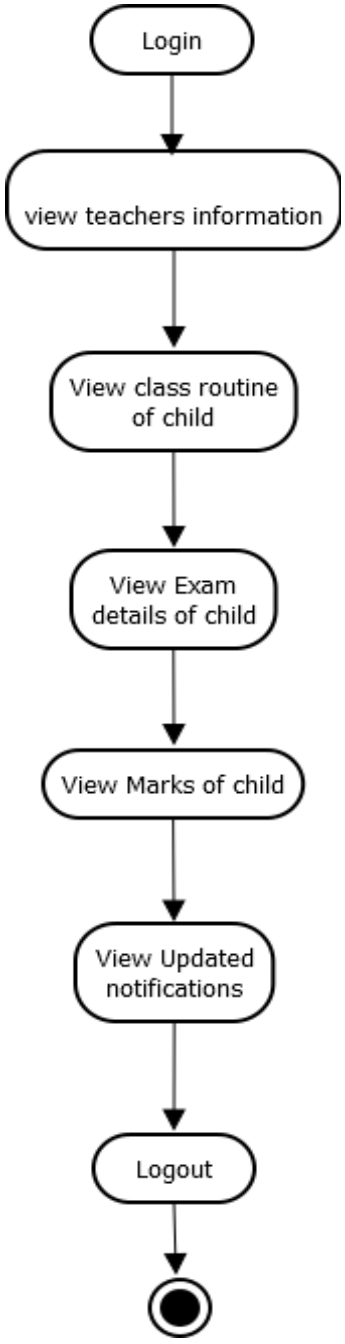
Admin:


Login in
Manage Student information
Manage Parent information
Manage Teacher information
Manage class
Manage Subjects
Manage Exam list
Update Student marks
Mpodify class routine weekly
Update daily attendance
Logout

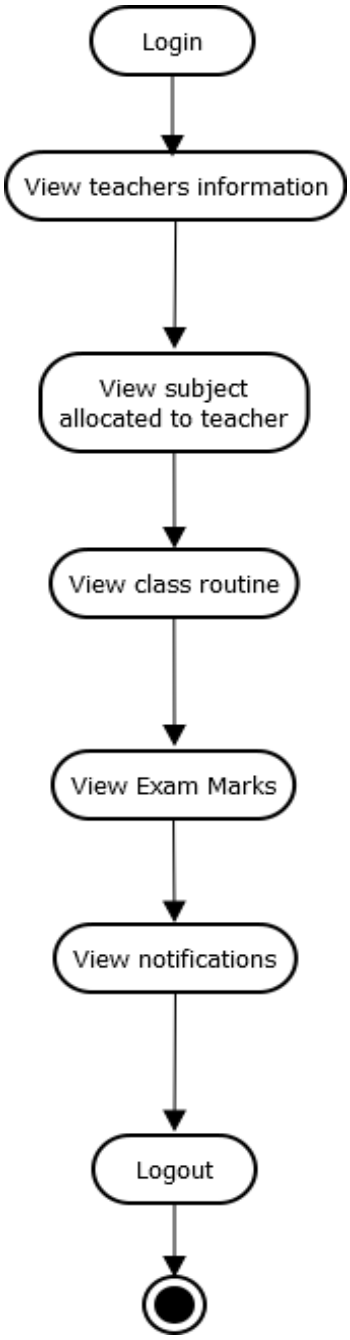
Teacher:


Login
Update Student Information
View class routine
View allocated subjects
Mark Attendance
Update Student marks
View notice board
Logout

Parent:

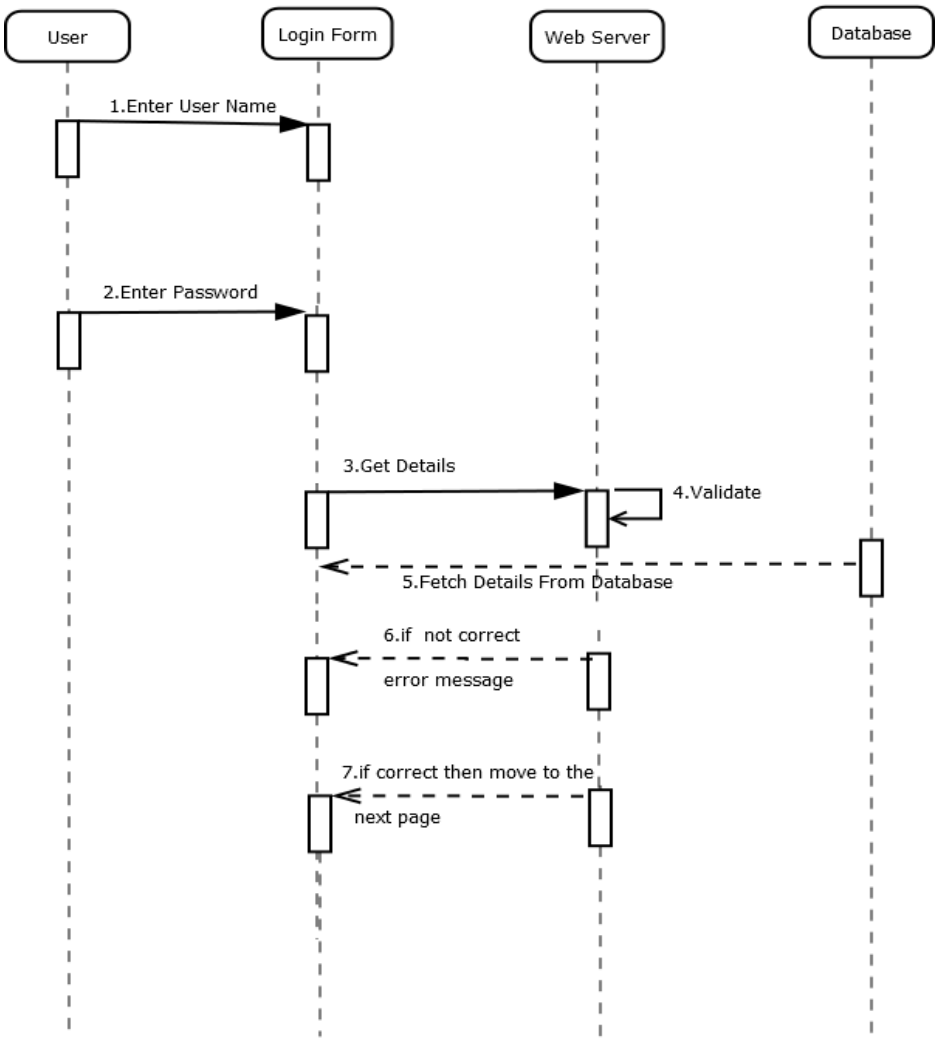

Login
view teachers information
View class routine of child
View Exam details of child
View Marks of child
View Updated notifications
Logout

Student:

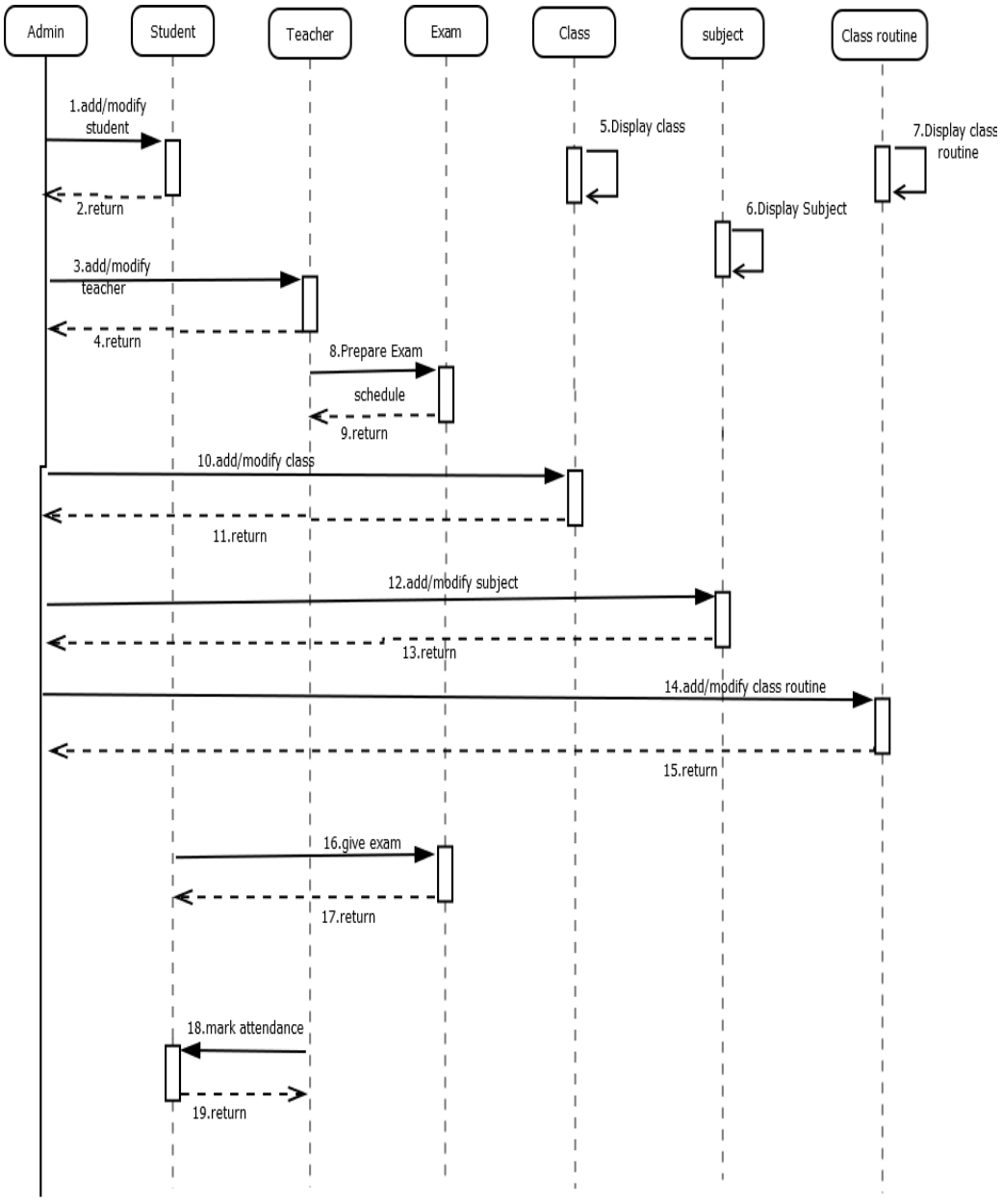

Login
View teachers information
View subject allocated to teacher
View class routine
View Exam Marks
View notifications
Logout

3.5 Sequence Diagram

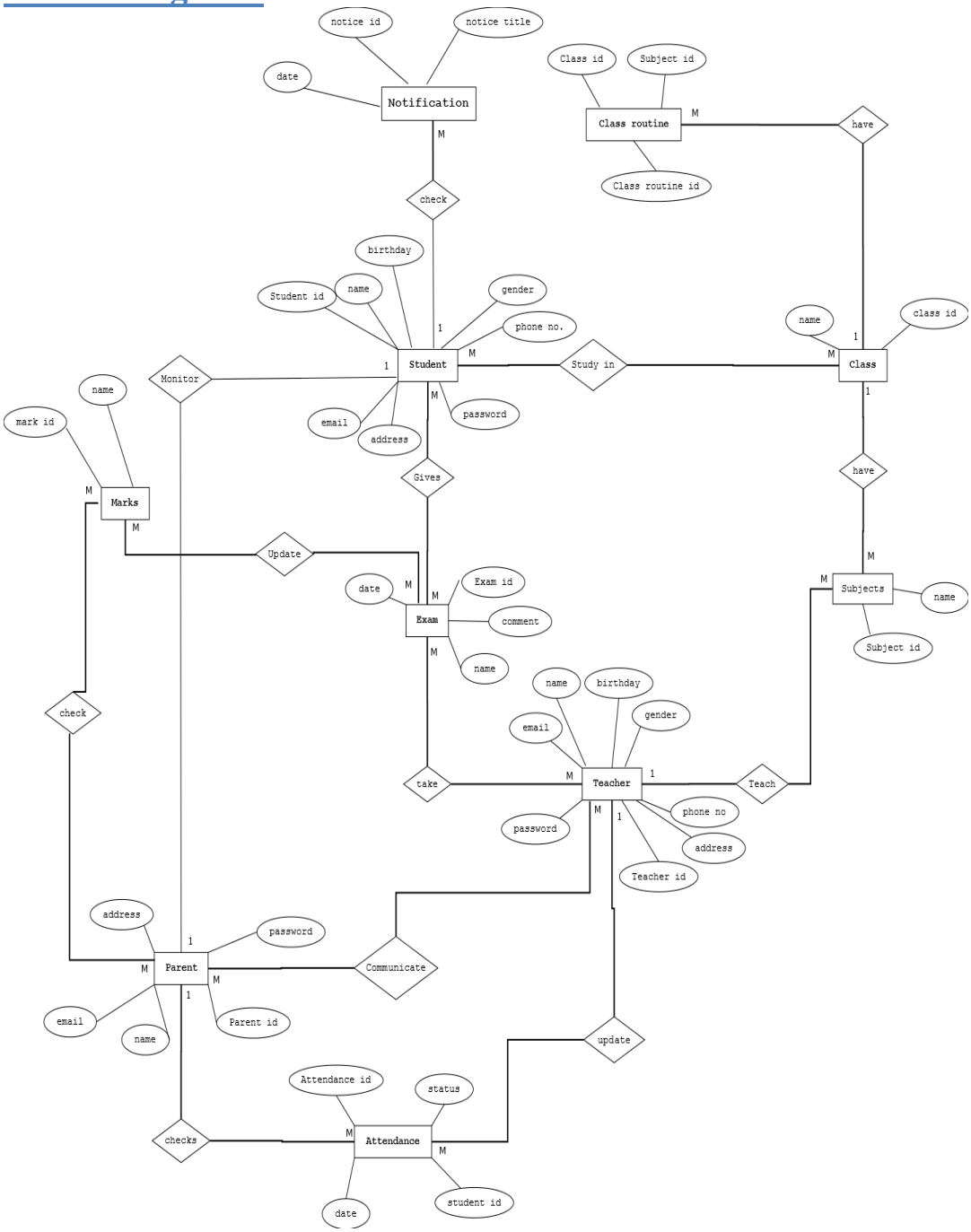
Login form:



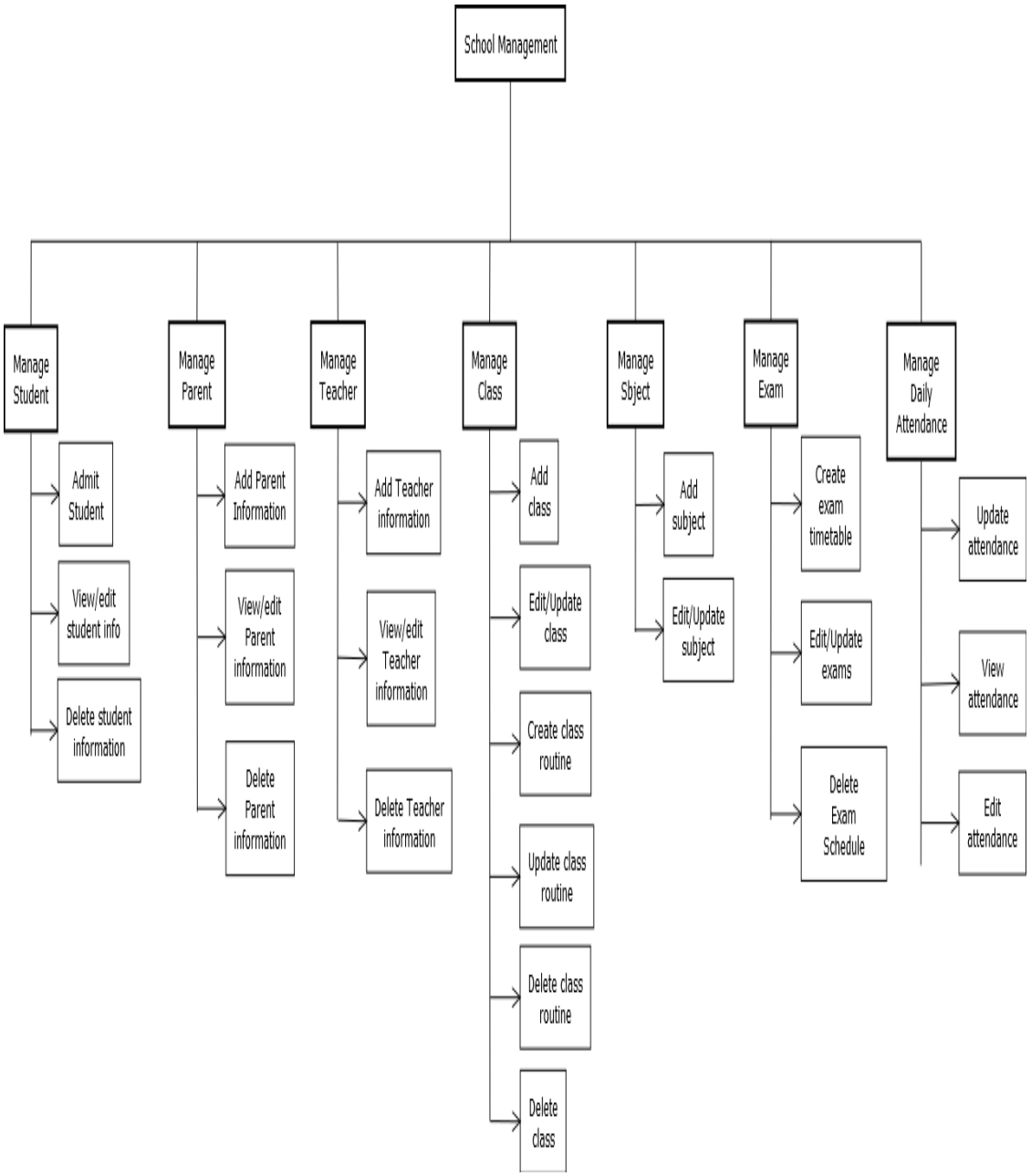
Admin:


Admin
Student
Teacher
Exam
Class
subject
Class routine
1.add/modify student
2.return
5.Display class
7.Display class routine
6.Display Subject
3.add/modify teacher
4.return
8.Prepare Exam schedule
9.return
10.add/modify class
11.return
12.add/modify subject
13.return
14.add/modify class routine
15.return
16.give exam
17.return
18.mark attendance
19.return

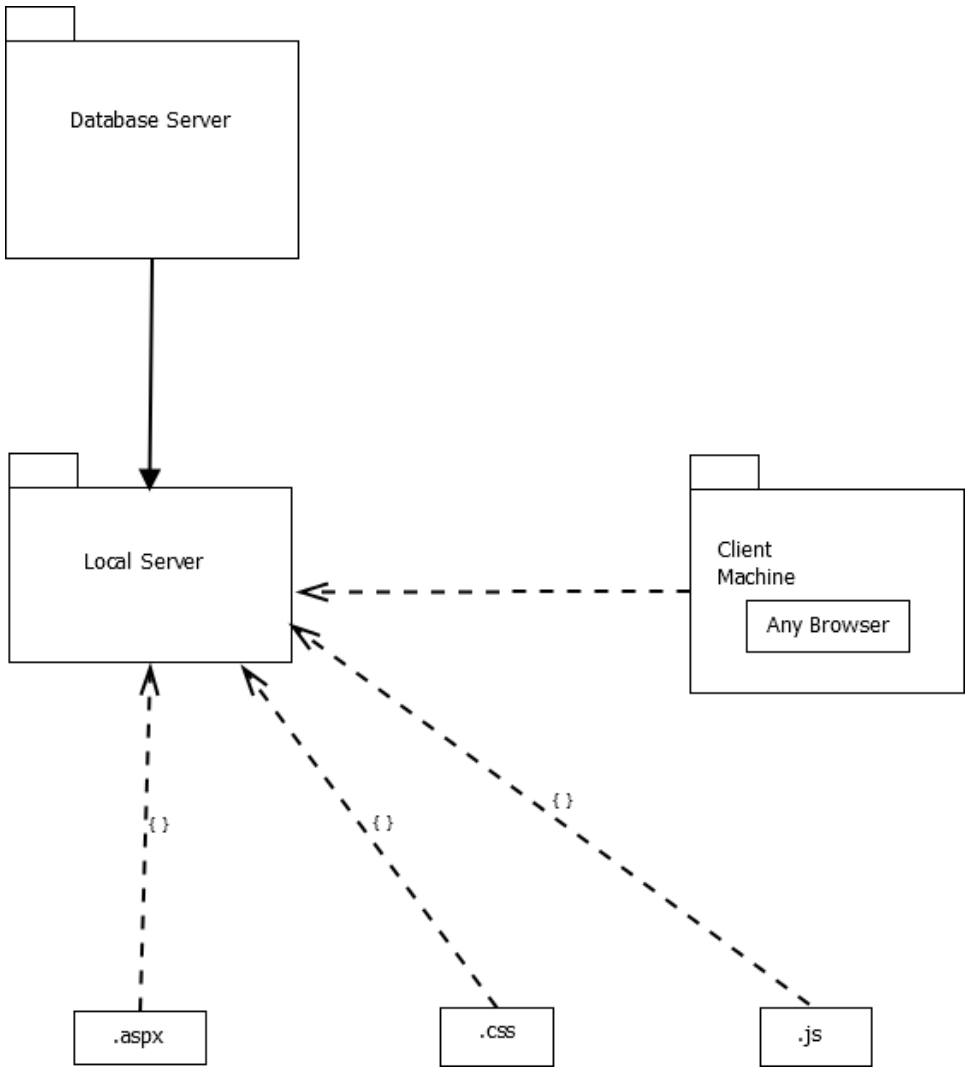
3.6 ER Diagram:



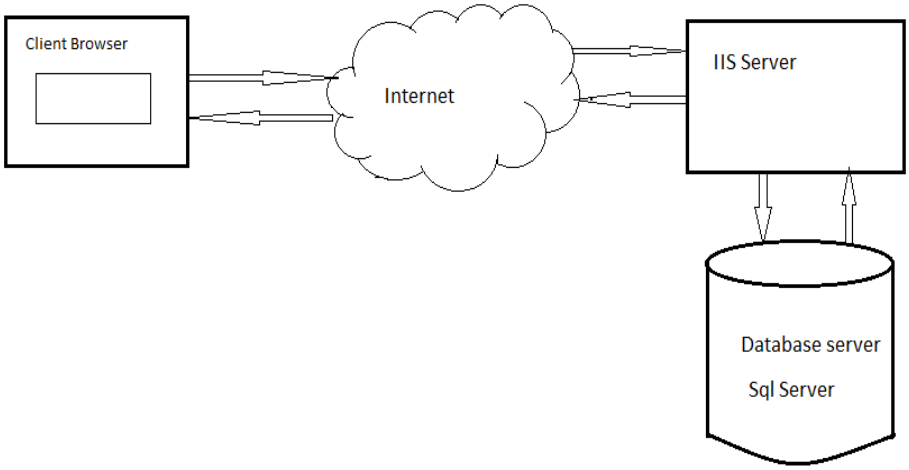
3.7 Module Hierarchy Diagram:



3.8 Component Diagram:



3.9 Deployment Diagram



3.10 Module Specification:

The Modules are as follows:

1) **Manage Student**
2) **Manage Parent**
3) **Manage Teacher**
4) **Manage Exam**
5) **Manage Daily Attendance**
6) **Manage Subject**
7) **Manage Class**
8) **Manage Invoice/Payment**

1. **Manage Student**

   Function student has the ability to add, edit or delete students from the system.

2. **Manage Parent**

   Function parent has the ability to add, edit or delete parents form the system. Once added by the admin, then parent can be able to see their child's progress.

3. **Manage Teacher**

   Function teacher has the ability to add, edit or delete teachers form the system.

4. **Manage Exam**

   Function exam has the ability to add exam timetable, edit or update exam schedule and can delete the older exam schedules.

5. **Manage Daily Attendance**

   The attendance module holds the commands for entering attendance status for a particular student of a particular class for a particular day

6. **Manage Subject**

   The subject module adds subjects with respect to the class. And can be able to edit or update if required.

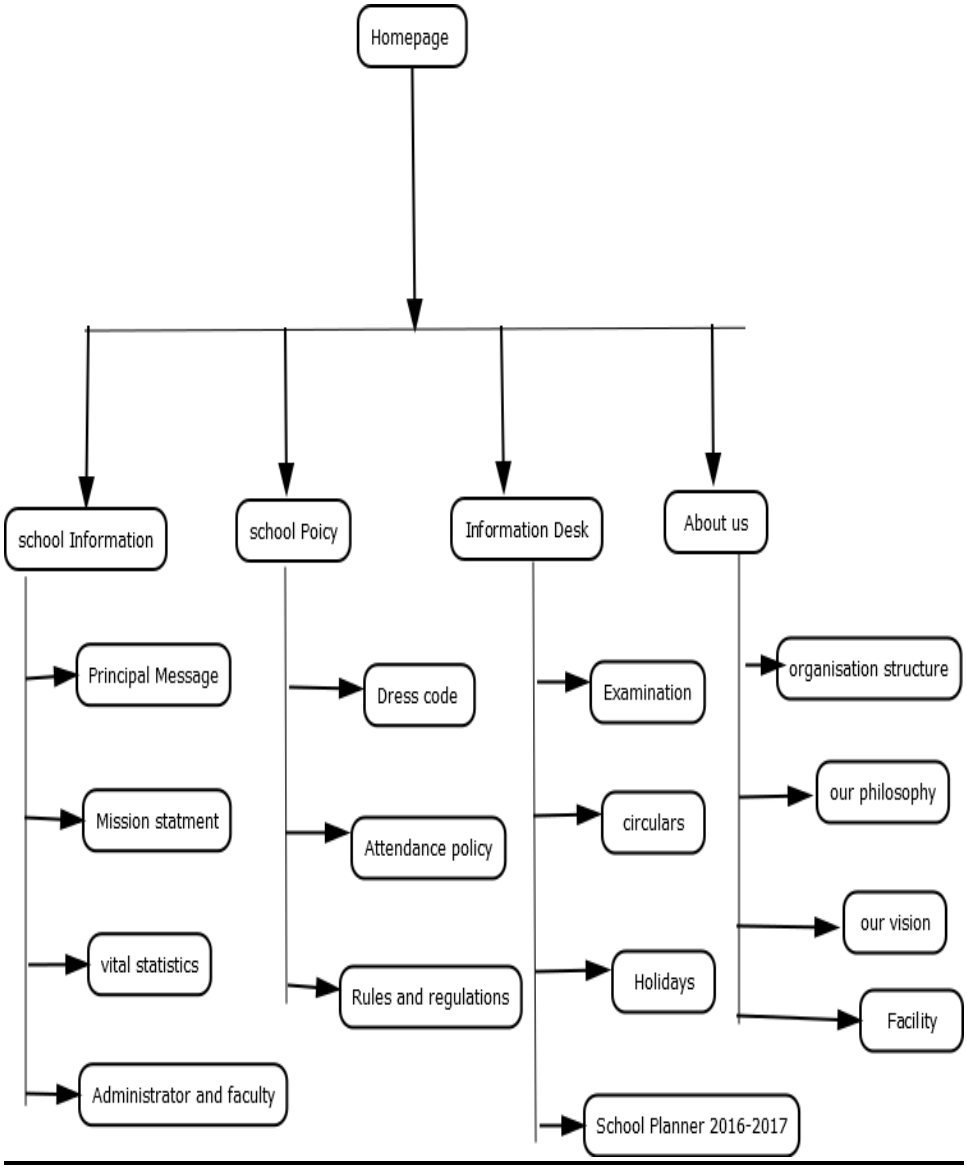
7. **Manage Class**

   The class module has the ability to add class, edit or update class. It also can create class routine and can update it. It also has power to delete class and class routine.

8. **Manage Invoice/Payment**

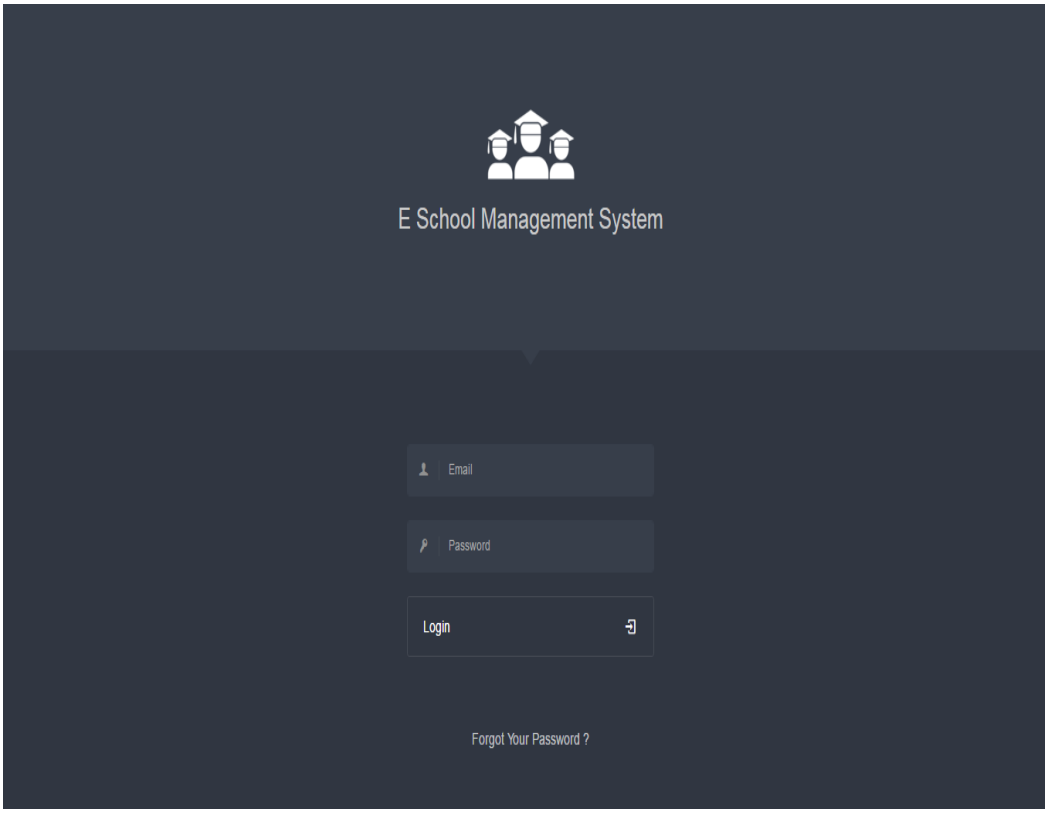
   The manage invoice module deals with school fees paid by student. It has the ability to generate report/invoice in printable format.

3.12 Web Site Map Diagram



3.13 User Interface Design

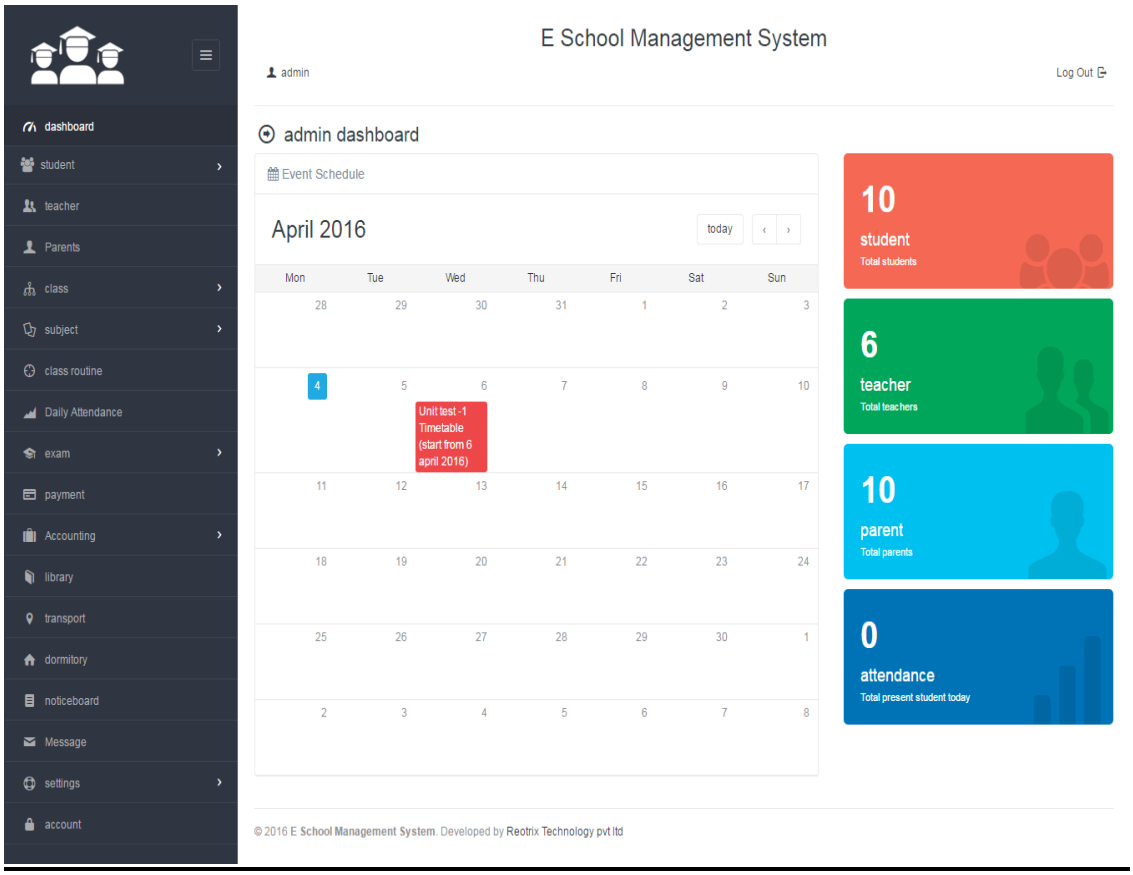
Login Page



DESCRIPTION

Login page, the first screen appears after entering the url. This allows user to login into system.

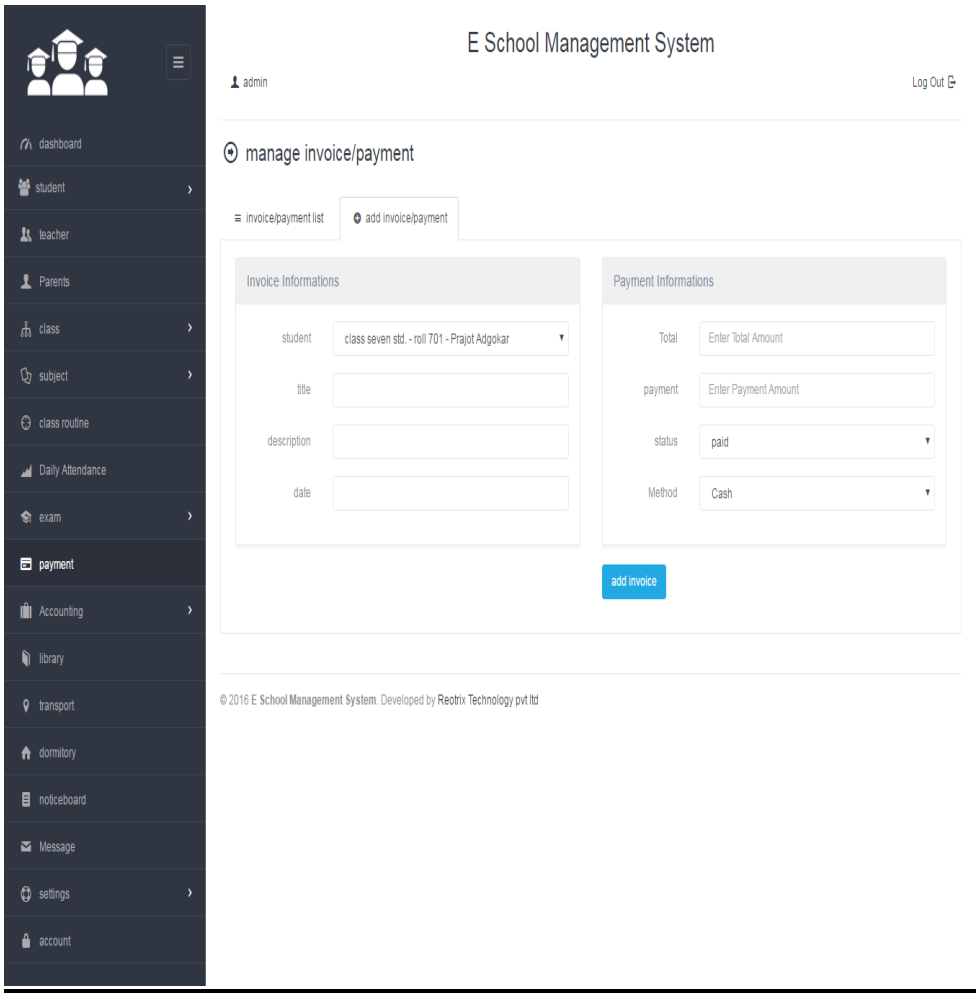
Admin Dashboard



DESCRIPTION

Admin can control all information and functionalities from admin dashboard.

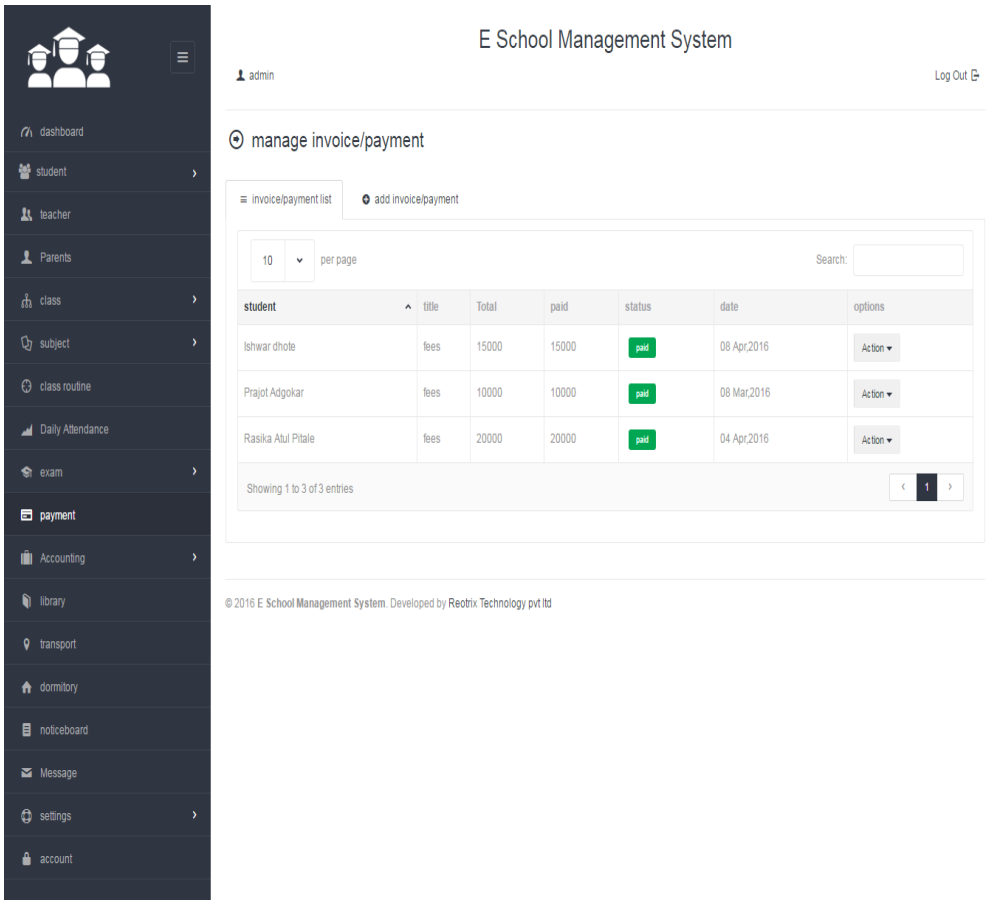
Manage Invoice



DESCRIPTION

Admin can be able to manage invoice record here.

Invoice/Payment list

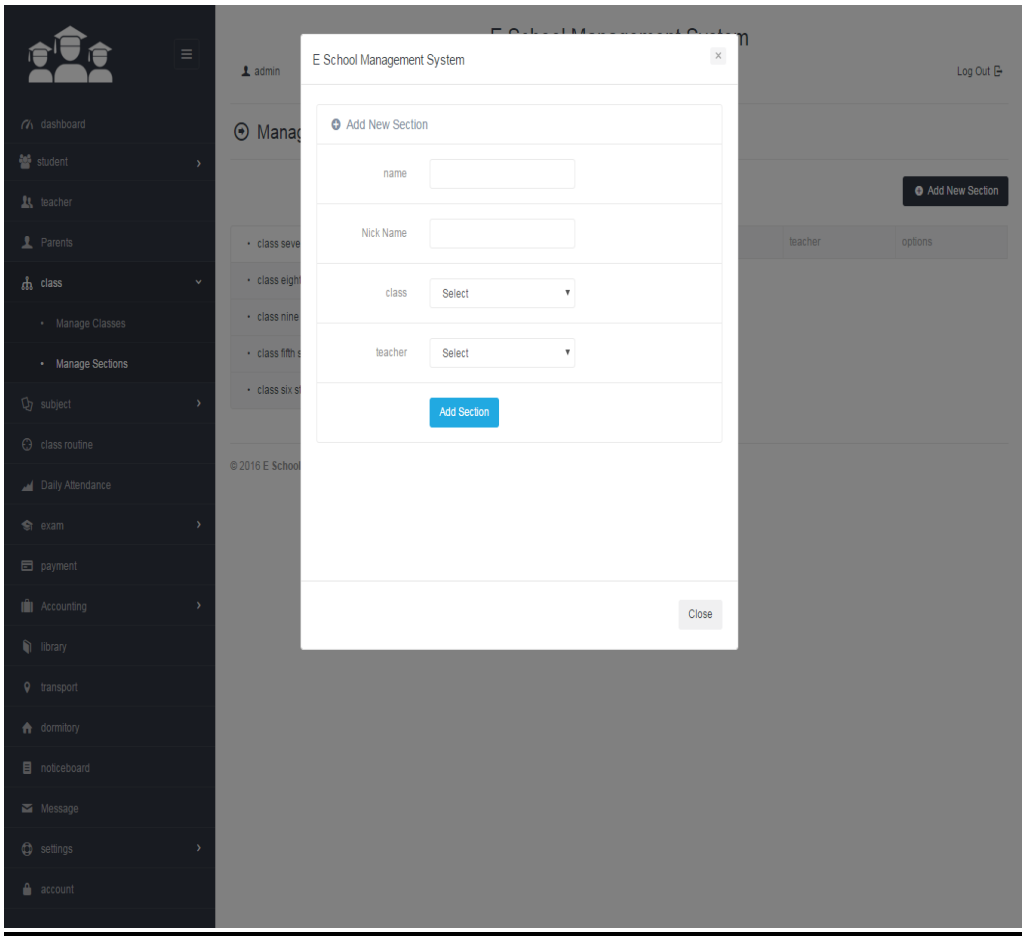


.

DESCRIPTION

This is the screen which shows list of payments made by different students to the admin.

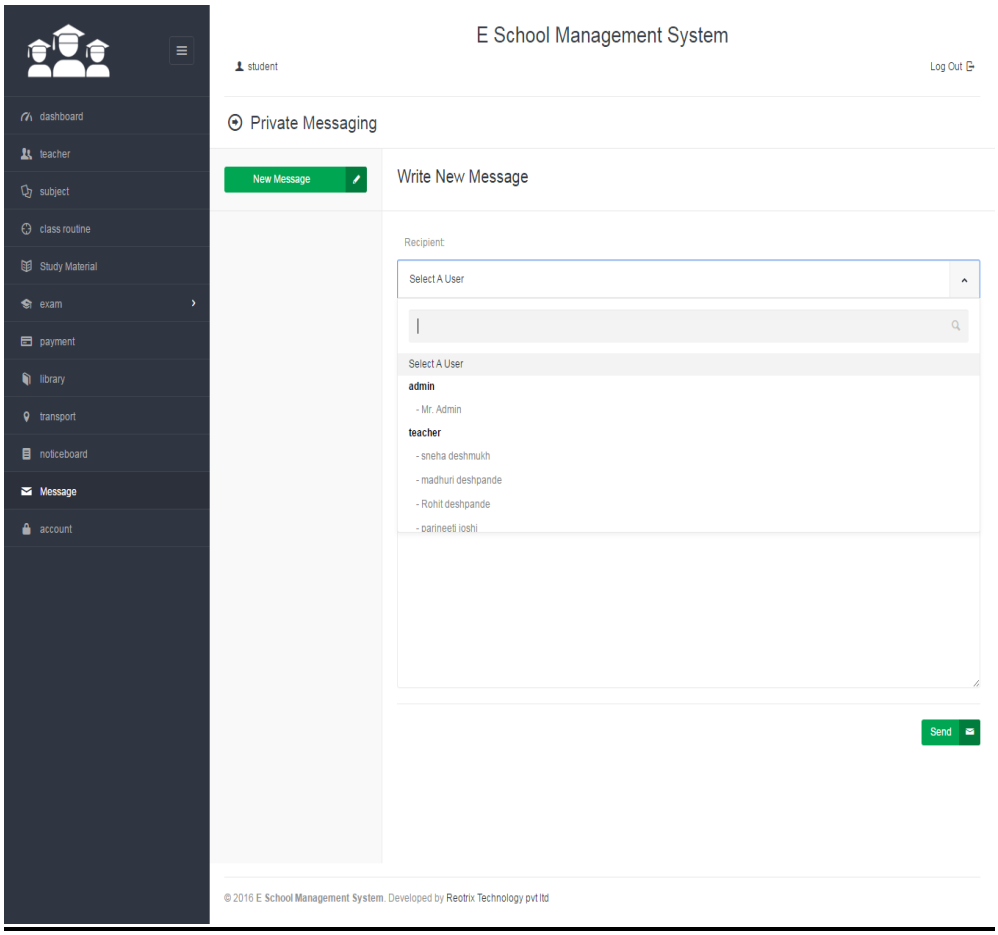
Add New Section



DESCRIPTION

New class section (i.e. new section of same class) can be added by admin.

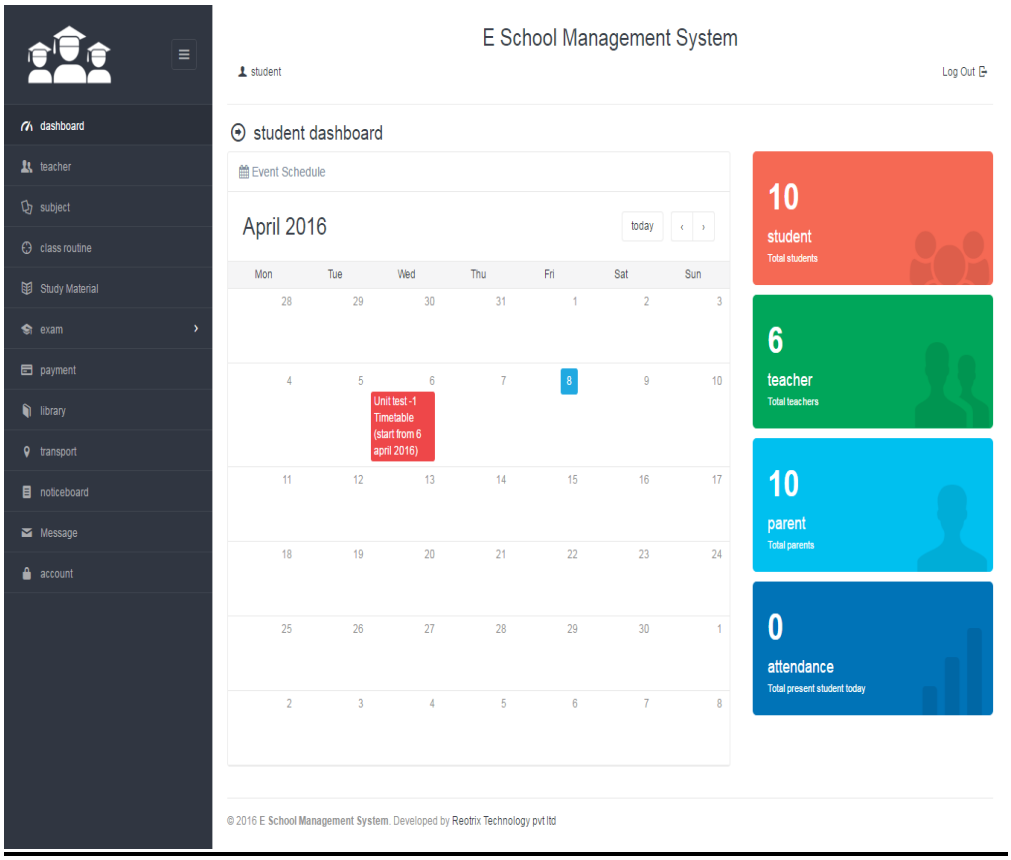
Private Massage



DESCRIPTION

Student can message teacher or admin. The message is totally private. Only respective student and teacher or admin can be able to see the messages.

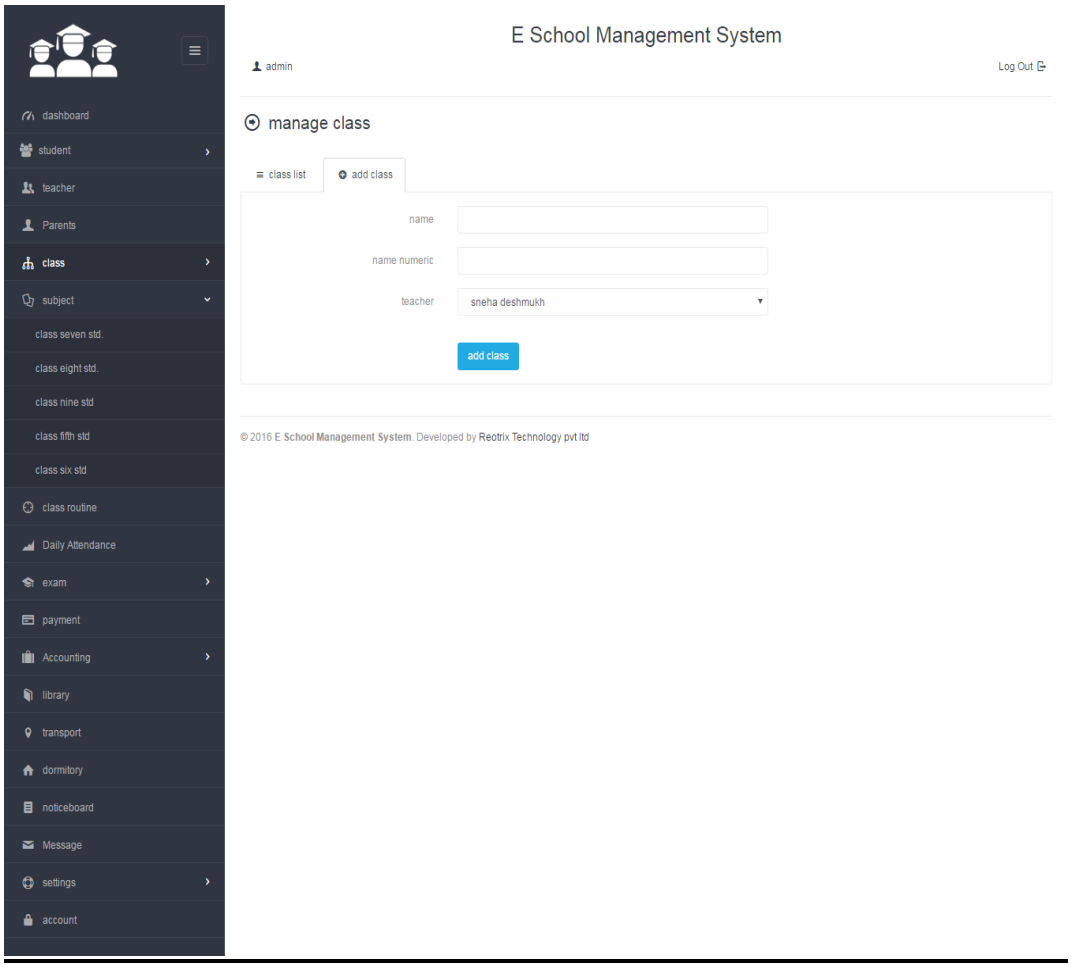
Student Dashboard



DESCRIPTION

This is the place where student can interact with the system. Student is not allowed to make any changes to the system. She/he can only check the notifications, daily schedule etc.

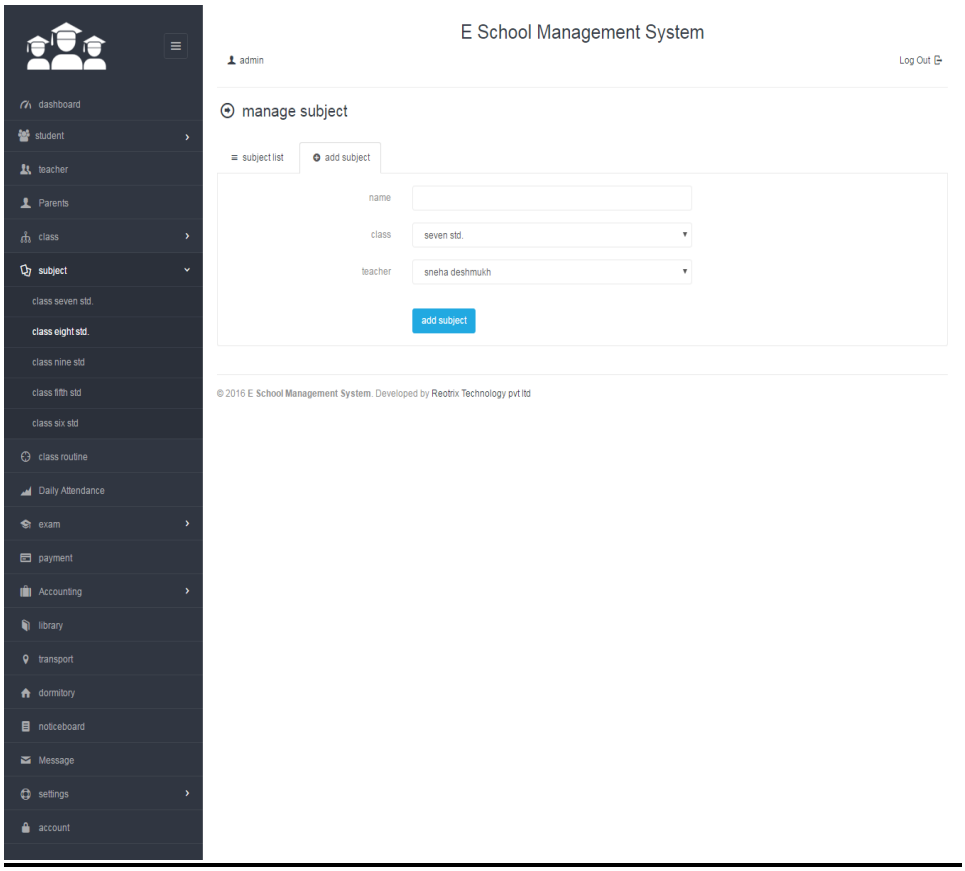
Manage Class



DESCRIPTION

All available class information shown and manipulated on manage class screen.

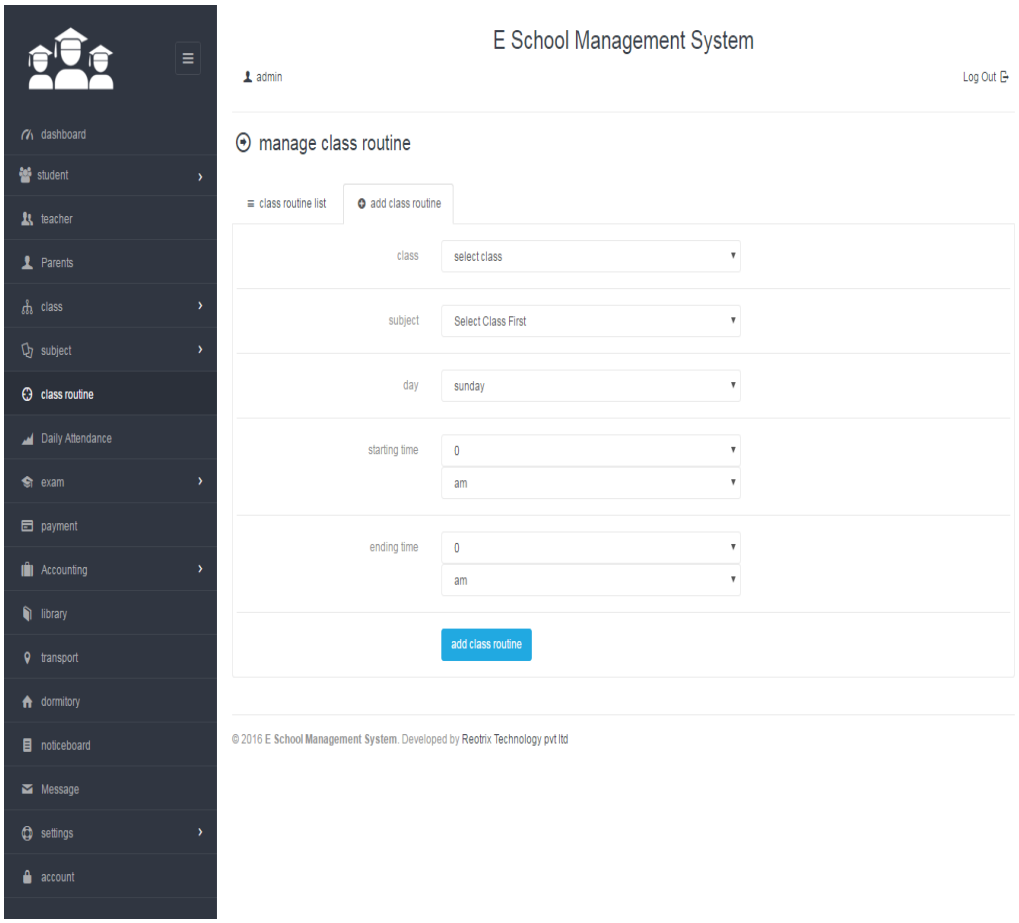
Add Subject



DESCRIPTION

New subject can be added.

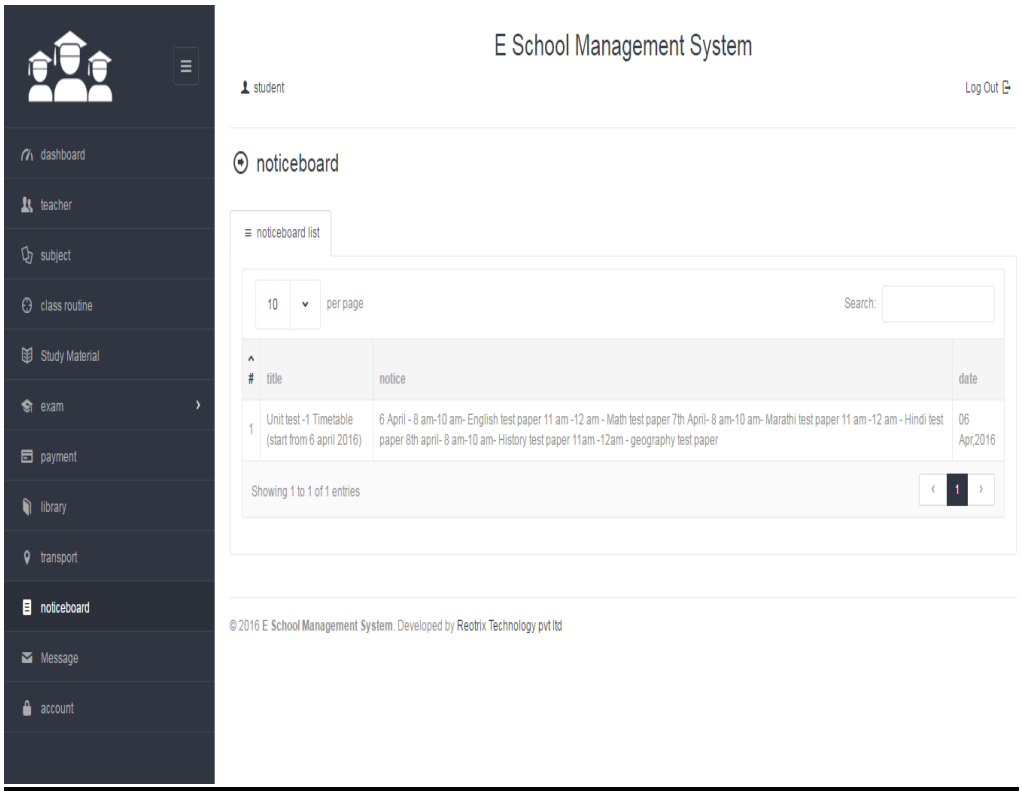
Add Class Routine



DESCRIPTION

Admin will add new class routine.

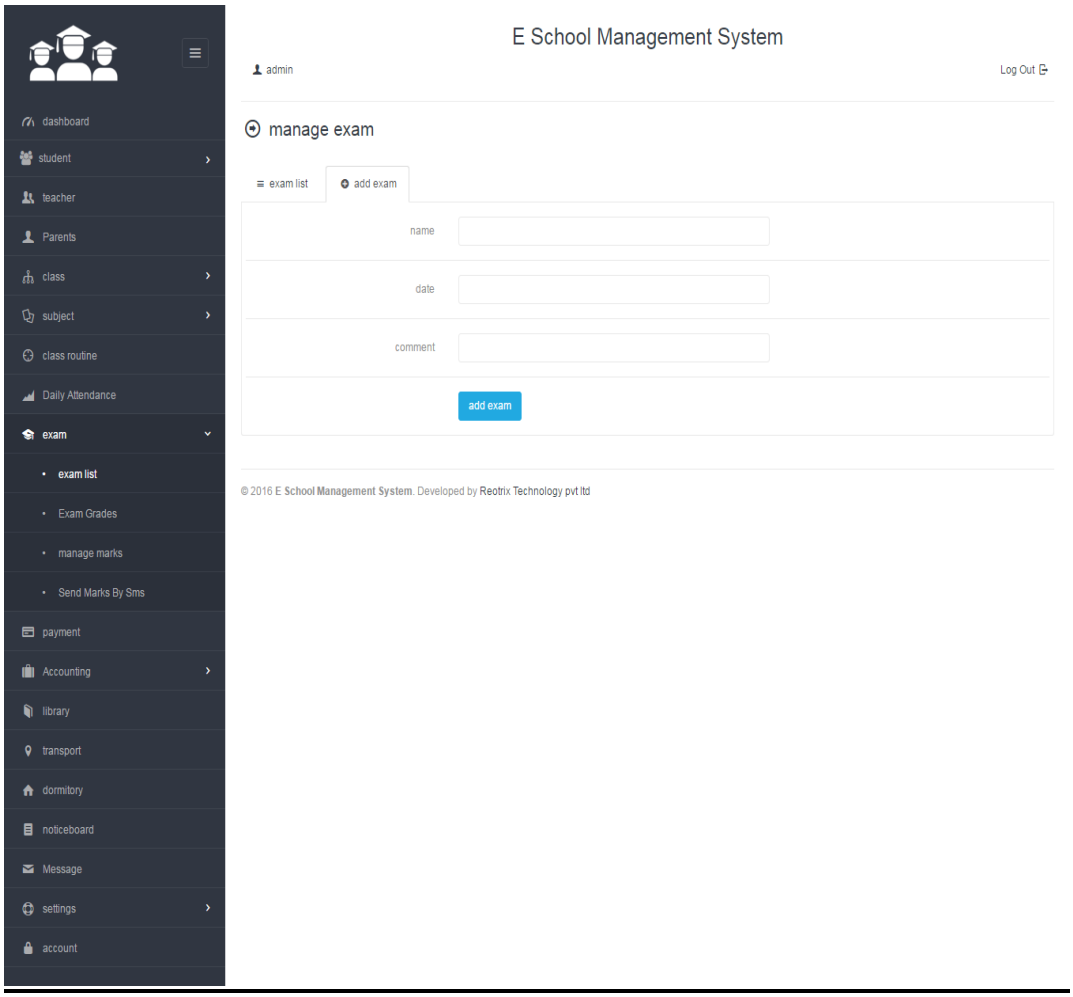
Notice board



DESCRIPTION

All the important notices by teacher or admin are displayed. And student can see all the notices and any kind of updates here.

Add Exam



DESCRIPTION

Add new exam.

3.14 Data Dictionary

Sr.No	*Field Name*	*Data Type*	*Description*
1	address	nvarchar(30)	Student home address
2	address	nvarchar(30)	Parents address
3	address	nvarchar(30)	Teacher address
4	admin_id	int (2)	Primary key of Admin table
5	attendance_id	int(2)	Primary key of Attendance table
6	birthday	date	Student date of birth
7	birthday	date	Teacher date of birth
8	class strength	int(2)	Number of students in class
9	class teacher id	int(2)	Primary key[teacher],foreign key[attendance, class, subject]
10	class_id	int (2)	Primary key [Class],foreign key [subject, class_routine,student, mark]
11	class_routine_id	int(2)	Primary key [Class routine]
12	comment	varchar(50)	Description about student performance
13	create timestamp	int(2)	timestamp
14	date	date(-)	Attendance date
15	date	date(-)	Exam date
16	day	date time(-)	Class day

17	Email	nvarchar(30)	Admin email address
18	email	nvarchar(30)	Student email
19	email	nvarchar(30)	Parents email address
20	email	nvarchar(30)	Teacher email address
21	exam schedule id	int(2)	Primary key [Exam schedule] foreign key[Mark]
22	exam_id	int(2)	Primary key [Exam] ,foreign key[Exam schedule]
23	from date	date(-)	Exam start date
24	gender	varchar(8)	Student gender
25	gender	varchar(8)	Teacher gender
26	mark _obtained	int(3)	Subject marks
27	mark _total	int(3)	Total marks
28	Name	varchar(25)	Admin name
29	Name	varchar(25)	Class name
30	Name	varchar(25)	Student name
31	name	varchar(25)	Parent name
32	name	varchar(25)	Teacher name
33	name	varchar(25)	Subject name
34	name	varchar(20)	Exam name
35	Name Numeric	nvarchar(5)	Numeric value of class
36	notice path	nvarchar(20)	Notice path
37	notice title	nvarchar(20)	Notice title
38	notice-id	int(2)	Primary key of Notice board table
39	parent_id	int(2)	Primary key [parent],foreign key[student]
40	password	nvarchar(10)	Unique password admin
41	password	nvarchar(15)	Student’s unique password
42	password	nvarchar	Parents unique password

43	password	nvarchar(20)	Teachers unique password
44	phone	int(10)	Student Home contact no
45	phone	int(10)	Parent mobile number
46	phone	int(10)	Teacher contact no
47	profession	varchar(20)	Parent profession
48	qualification	varchar(20)	Teacher qualification
49	Roll no	int(2)	Student roll no
50	status	varchar(2)	Attendance status of student
51	student_id	int(2)	Primary key [Student], Foreign key[attendance, class routine, mark]
52	subject	varchar(5)	Exam Subject
53	subject_id	int(2)	Primary key [Subject], Foreign key[class routine, mark]
54	time	time(-)	Exam time
55	time end	time(-)	Class end time
56	time start	time(-)	Class start time
57	To date	date(-)	Exam end date
58	total mark	int(3)	Total marks of exam

3.15 Table Specification

1) Admin

Column Name	Data type	Length	Constraints	Description
admin_id	int	2	Not Null	Primary key
Name	varchar	25	Not Null	Admin name
Email	nvarchar	30	Not Null	Email address
password	nvarchar	10	Not Null	Unique password

2) Class

Column Name	Data type	Length	Constraints	Description
class_id	int	2	Not Null	Primary key
Name	varchar	25	Not Null	Class name
Name Numeric	nvarchar	5	Not Null	Class numeric name
class_teacher_id	int	2	Not Null	Foreign key [Teacher]
class strength	int	2	Not Null	No. of students

3) Student

Column Name	Data type	Length	Constraints	Description
student_id	int	2	Not Null	Primary key
Name	varchar	25	Not Null	Student name
birthday	date	_	Not Null	Student date of birth
gender	varchar	8	Not Null	Student gender
address	nvarchar	30	Not Null	Home address
phone	int	10	Not Null	Home contact no.
email	nvarchar	30	Not Null	Email address
password	nvarchar	15	Not Null	unique password
class_id	int	2	Not Null	
parent_id	int	2	Not Null	Foreign _key [parent]
Roll no	int	2	Not Null	School roll no.

4) **Parent**

Column Name	**Data type**	**Length**	**Constraints**	**Description**
parent_id	int	2	Not Null	Primary key
name	varchar	25	Not Null	Parent name
email	nvarchar	30	Not Null	parent email address
password	nvarchar	10	Not Null	unique password
phone	int	10	Not Null	Contact number
address	nvarchar	30	Not Null	Permanent address
profession	varchar	20	Not Null	Parent Profession

5) **Teacher**

Column Name	**Data type**	**Length**	**Constraints**	**Description**
class teacher_id	int	2	Not Null	Primary key
name	varchar	25	Not Null	Teacher name
birthday	date	_	Not Null	Teacher date of birth

gender	varchar	8	Not Null	Teacher gender
address	nvarchar	30	Not Null	Teacher home address
phone	int	10	Not Null	Contact no.
email	nvarchar	30	Not Null	Email address
password	nvarchar	20	Not Null	unique password
qualification	varchar	20	Not Null	Teacher qualification

6) **Subject**

Column Name	**Data type**	**Length**	**Constraints**	**Description**
subject_id	int	2	Not Null	Primary key
name	varchar	25	Not Null	Subject name
class_id	int	2	Not Null	Foreign key [class]
class teacher_id	int	2	Not Null	Foreign key [teacher]

7) Attendance

Column Name	Data type	Length	Constraints	Description
attendance_id	int	2	Not Null	Primary key
status	varchar	2	Not Null	Attendance status
student_id	int	2	Not Null	Foreign key [student]
date	date	_	Not Null	Attendance date
class teacher id	int	2	Not Null	Foreign key[teacher]

8) Class routine

Column Name	Data type	Length	Constraints	Description
class_routine _id	int	2	Not Null	Primary key
class_id	int	2	Not Null	Foreign key [class]
subject_id	int	2	Not Null	Foreign key [subject]
time start	time	_	Not Null	Class start time
time end	time	_	Not Null	Class end time

day	date time	_	Not Null	Class day

9) Exam

Column Name	Data type	Length	Constraints	Description
exam_id	int	2	Not Null	Primary key
name	varchar	20	Not Null	Exam name
from date	date	_	Not Null	Exam start date
To date	date	_	Not Null	Exam end date
class_id	int	2	Not Null	Foreign key [class]

10) Mark

Column Name	Data type	Length	Constraints	Description
student_id	int	2	Not Null	Foreign key [student]
subject_id	int	2	Not Null	Foreign key [subject]
exam-schedule_id	int	2	Not Null	Foreign key [exam schedule]

mark _obtained	int	3	Not Null	Marks in a subject
mark _total	int	3	Not Null	Total of all subjects
comment	varchar	50	Not Null	Comment on marks

11) Notice board

Column Name	**Data type**	**Length**	**Constraints**	**Description**
notice-id	int	2	Not null	Primary key
notice title	nvarchar	20	Not null	Title of notice
notice path	nvarchar	20	Not null	Notice path
create timestamp	int	2	Not null	Notice time

12) Exam schedule

Column Name	**Data type**	**Length**	**Constraints**	**Description**
exam schedule id	int	2	Not null	Primary key
exam_id	int	2	Not null	Foreign key [exam]

date	date	_	Not null	Exam date
subject	varchar	5	Not null	Subject of exam
time	time	_	Not null	Exam time
total mark	int	3	Not null	Marks of exam

3.16 Test Procedure and implementations

Definition:

A test case is simply a test with formal steps and instructions; test cases are valuable because they are repeatable, reproducible under the same environments, and easy to improve upon with feedback. A test case is a detailed procedure that fully tests a feature or an aspect of a feature. Whereas the test plan describes what to test, a test case describes how to perform a particular test.

Unit Testing

This testing focuses verification on the module. Using procedural design description as a guide; important control paths are tested to uncover errors within the boundaries of the module. The relative complexity of tests and uncovered errors is limited by the constraints scope established for the unit testing. The unit testing can be conducted in parallel for the several modules.

Integration Testing

Generally a combined approach known as sandwich testing using features of top down testing strategy for the upper levels of the program structure, coupled with the bottom up strategy for subordinate modules.

Functional Testing

This is done for each module and sub module of the system. Functional testing serves as a means of validation weather the functionality of the system confirms the original user requirements i.e. done the module do what is supposed to do. The separate schedules were made for functional testing.

System Testing

System testing is done when the entire system has been fully integrated. The purpose of the system testing is to test how the different modules interact with each other and whether the system provides the functionality that was expected.

Test Procedure:

1. Create a test plan.

2. Plan the test lab.

3. Design the test lab.

4. Develop the test lab.

5. Design the test cases.

6. Conduct the tests.

Test Case: 1

Test Case Name: User Login Page

Test no	**Step to be executed**	**Test values**	**Expected result**	**Actual result**	**status**
1	User id and password	User id=”test” and password =”test”	Home screen will appear	Home page appear	pass
2	User id and password	User id=”” and password =”test”	Error message	Error message	pass
3	User id and password	User id=”test” and password =””	Error message	Error message	pass
4	User id and password	User id=”” and password =””	Error message	Error message	pass

Test Case: 2

Test Case Name: Administrator Home Page

Test no	Step to be executed	Test values	Expected result	Actual result	status
1	After logging in as administrator	Check whether admin home page open or not	Make sure that administrator home page should open	The administrator home page is opening	pass
2	Click on student admission form	Check whether student admission form open or not	Make sure student admission form should open	Student admission form is opened	pass
3	Click on teacher form	Check whether teacher information form open or not	Make sure teacher information form should open	Teacher information form is opened	pass
4	Click on logout link	Check whether logout successful or not	Make sure you are logged out	Administrator logged out	pass

Test Case: 3

Test Case Name: Teacher Home Page

Test no	**Step to be executed**	**Test values**	**Expected result**	**Actual result**	**status**
1	After logging in as teacher	Check whether teacher home page open or not	Make sure that the teacher home page should open	teacher home page should opening	pass
2	Click on notice board link	Check whether notice board page open or not	Make sure that the notice board page should open	Notice board page should opening	pass
3	click on update student attendance	Check whether update student attendance or not	Make sure that the update student attendance	Updated student attendance successfully	pass
4	Click on allocated subject	Check whether allocated subject display or not	make sure that the allocated subjects are display or not	Allocated subject are display	pass

Test Case: 4

Test Case Name: Parent Home Page

Test no	**Step to be executed**	**Test values**	**Expected result**	**Actual result**	**status**
1	After logging in as parent	Check whether parent home page open or not	Make sure that the parent home page should open	parent home page should opening	pass
2	Click on notice board link	Check whether notice board page open or not	Make sure that the notice board page should open	Notice board page should opening	pass
3	click on attendance	Check whether children attendance display or not	Make sure that the children attendance display	Display children attendance	pass
4	Click on mark	Check whether children mark are display or not	make sure that the children mark are display or not	Display children marks	pass

Test Case: 5

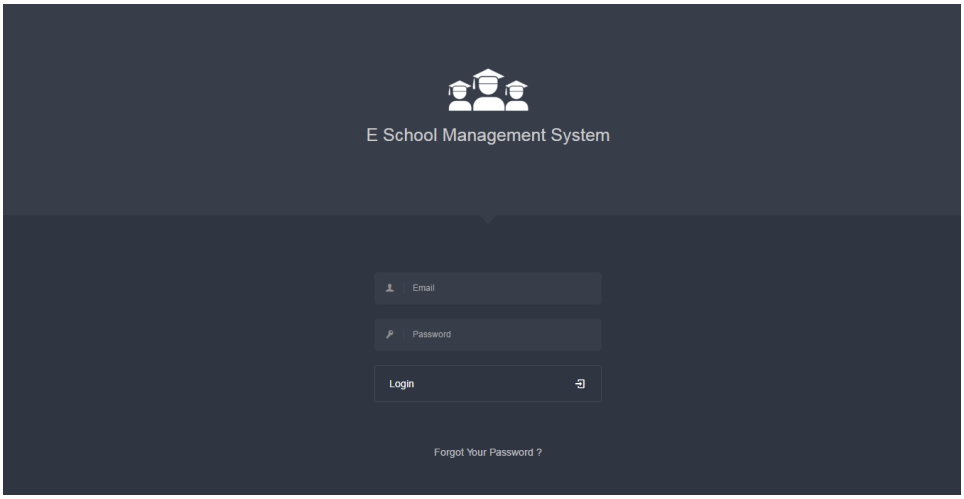
Test Case Name: Student Home Page

Test no	Step to be executed	Test values	Expected result	Actual result	status
1	After logging in as student	Check whether student home page open or not	Make sure that the student home page should open	student home page should opening	pass
2	Click on notice board link	Check whether notice board page open or not	Make sure that the notice board page should open	Notice board page should opening	pass
3	click on view teacher information	Check whether teacher information display or not	Make sure that the teacher information should display	Display teacher information	pass
4	Click on view subject	Check whether subject display or not	make sure that the subject are display or not	Display subject	pass
5	Click on logout link	Check whether student is log out or not	Student should be logged out	Student is logged out	pass

CHAPTER 4: USER MANUAL

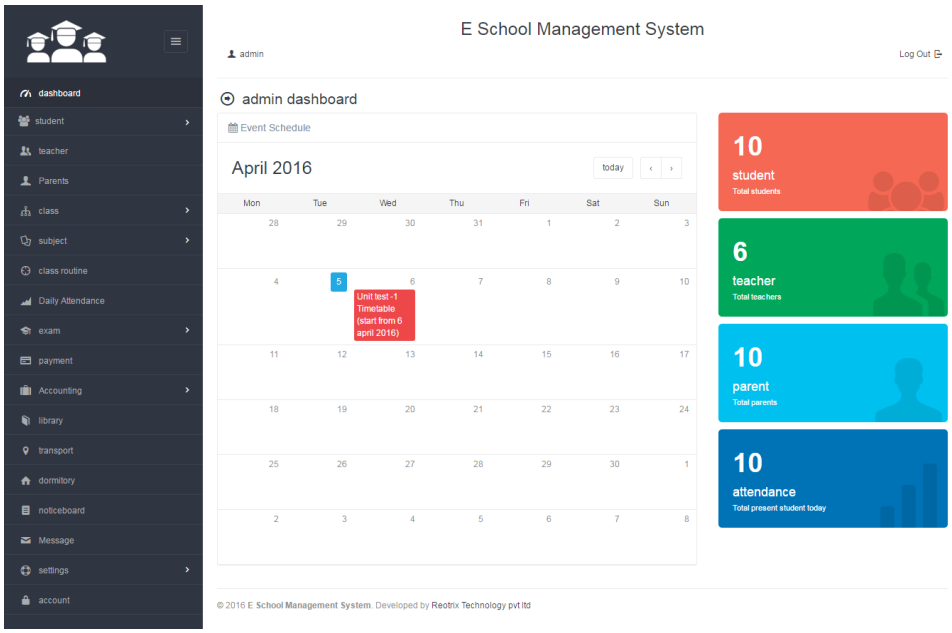
4.1 User Manual

E school Management System is a web-based application, which is designed mainly for parents and students. It is way easy to handle. When user opens the website, he/she will be provided login screen, which is shown below



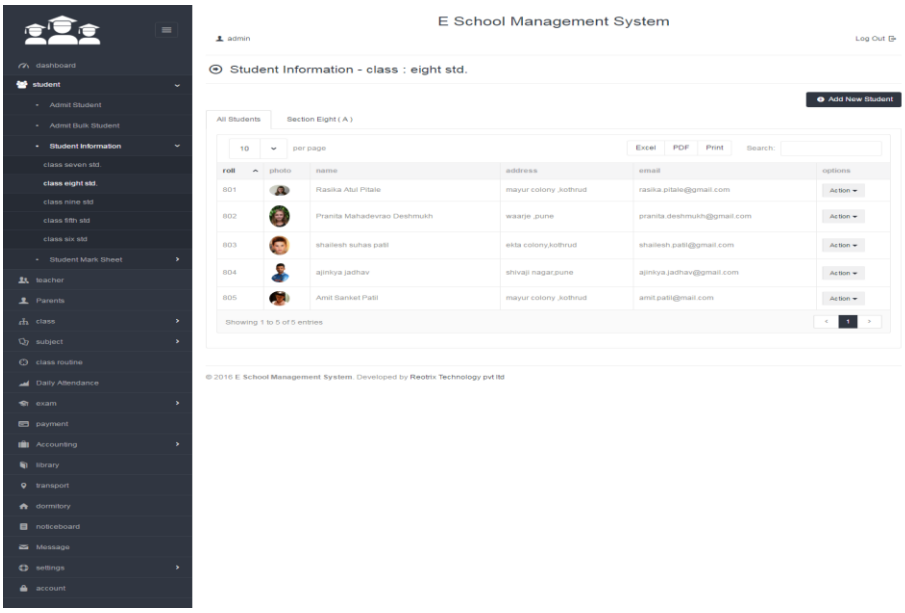
According to the username and password, a user can log in into the system as a student, as a teacher, or as a parent or as an admin. Each of this personality will be provide separate and unique username and passwords.

Admin is the one who can make changes in the system. After logged in into system as an admin, he/she can see an admin dashboard



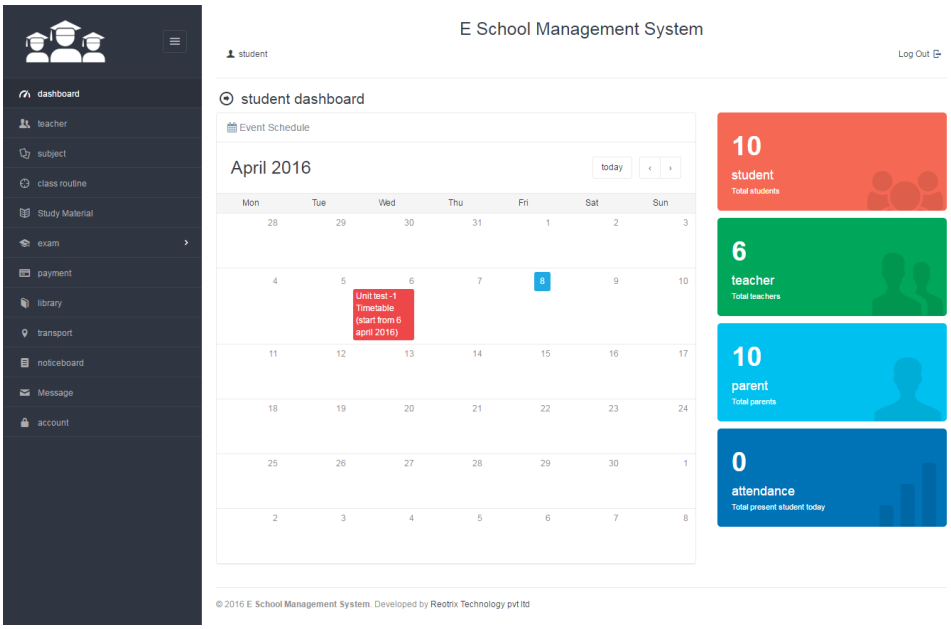
This is the workspace of admin. Admin have control to all over the information and functionalities.

For example, admin can see information entered by the students and can add or delete them from the system.



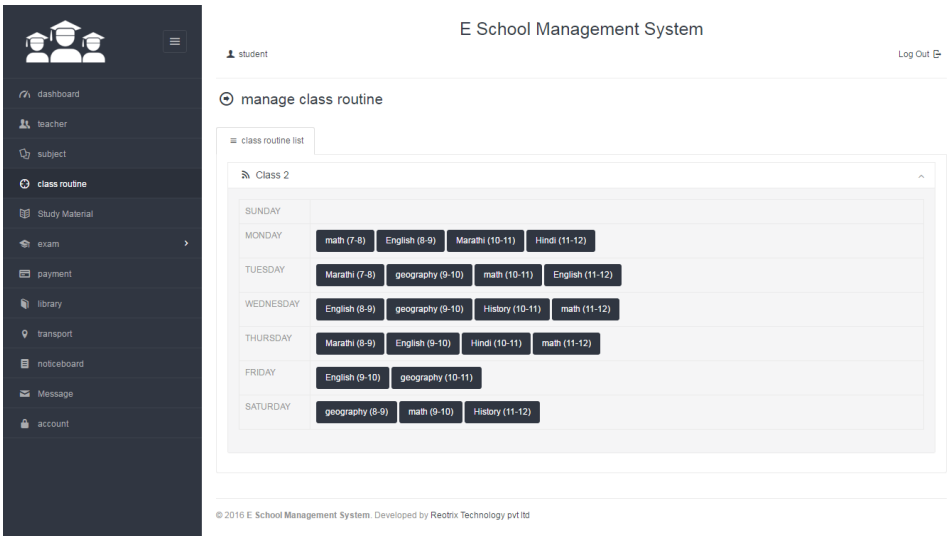
This screen shows the students of particular class. Admin can update or delete particular information from here.

Like admin, students also have their work space available in the system. And that is student dashboard. It is shown in next screen.



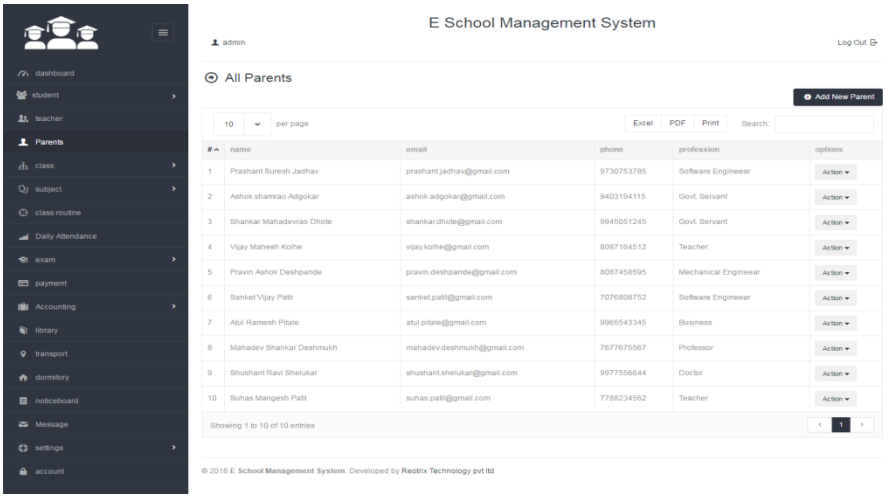
It is kind of same as admin in terms of looks, but the functionalities are way different.

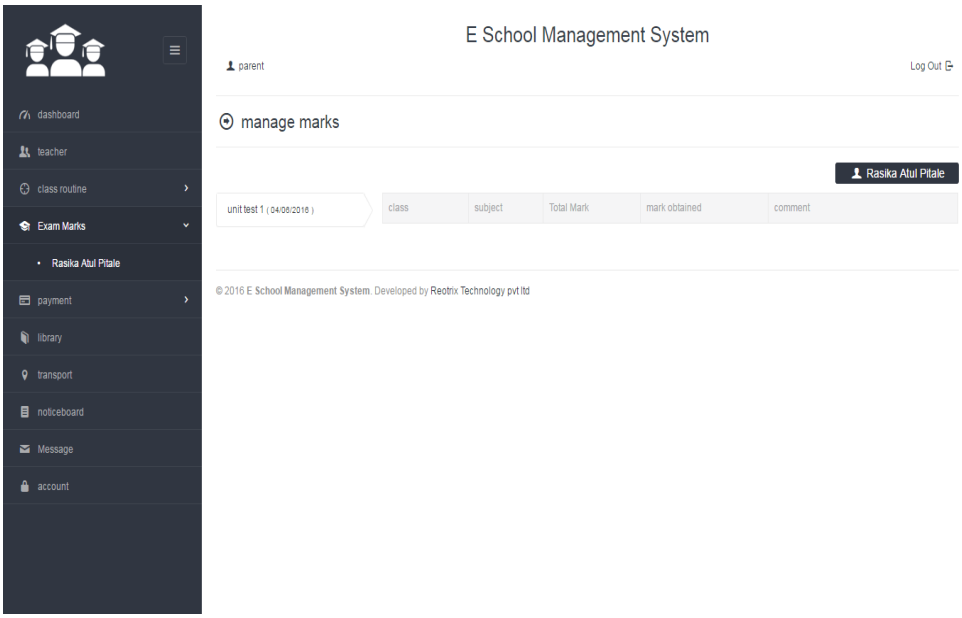
For example, students can look after their progress and school routines by themselves.



The student can see their regular class routine as shown in the above screen.

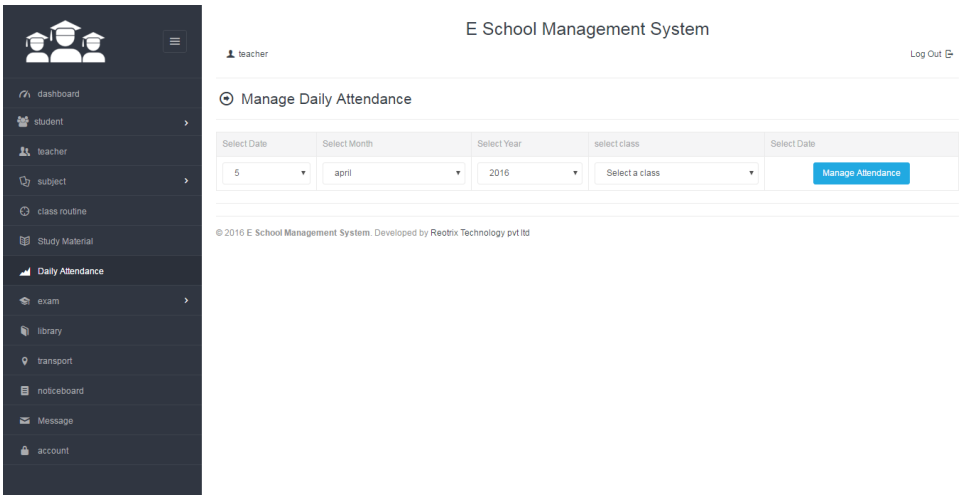
The basic concept was to include parents into the system. So down here parents added by the admin into the system.





And then, parents can see their child's progress like marks.

On the final node, teacher can mark and update daily attendance.



4.2 Operations Manual

Operations are specified below:-

1. Admin
    - Admin can add, update and delete student.
    - Admin is the one who can update all the notifications and reports.
    - Admin can add and update teacher into the system.
    - Can add, delete and update parents.
    - Can upload assignments for student.
    - Updates the class routine and exam schedule.
2. Student
    - Student can see the class routine.
    - Student can be able to see notifications and exam schedules.
    - Student can check respective subject teacher.

- They can see their own marks.

3. Parent
    - Parent can check progress of their child.
    - They can see child's marks (subject wise and total).
    - They can check notifications and messages by admin.
    - Parents can see class routine and exam schedule of their child.
4. Teacher
    - Teacher can mark the daily attendance of students.
    - Teacher can upload assignments.
    - Teacher can see all students' marks.
    - Teacher can see class routine.

4.3 Program Specification

Manage Student

Module	Manage student
Program Name	Add student
Purpose	Add new Student
Input Details	The required field should not be null and details should get stored.
Output	Student details are stored in student table. Student will get username and password

Manage Parent

Module	Manage parent
Program Name	Add Parent
Purpose	Add New Parent
Input Details	The required field should not be null and Parent details should get stored.
Output	Parent details are stored in parent table. The parent

	will get an username and password

Manage Teacher

Module	Manage Teacher
Program Name	Add Teacher
Purpose	Add new Teacher
Input Details	The required field should not be null and Teacher details should get stored.
Output	Teacher details are stored in Teacher table. The teacher will get an username and password

Manage class

Module	Manage class
Program Name	Add Class
Purpose	Add new Class
Input Details	The required field should not be null and class details should get stored.

Output	Class details are stored in class table.

Manage Exam

Module	Manage Exam
Program Name	Add Exam details
Purpose	Add new Exam details
Input Details	The required field should not be null and exam details should get stored.
Output	Exam details are stored in Exam table.

Manage notice board

Module	Manage notice board
Program Name	Add notification on notice board
Purpose	Add new notification
Input Details	The required field should not be null and notification details should get stored and notice board should be updated.

Output	Notice board details are stored in notice board table. The user will see updated notification message .

DRAWBACKS AND LIMITATIONS

Drawbacks and Limitations of the System:

1. All rights are reserved with the administration.
2. The user cannot edit anything at the system.
3. Only the registered user can perform login.
4. The user can only update the reports, student's information at the system.
5. The user needs anything then, he/she has to request to admin.

PROPOSED ENHANCEMENT

Proposed Enhancements

This module has some limitations. These can be overcome and also make some enhancement in this project.

- The generation of SMS service can be carried out.
- The user can update their requirements and information via mobile.
- An updating mail can be send to parents for notice and for results.
- Centralized admission process for the student admission.

CONCLUSION

Conclusion

The E school management system becomes the way of connection between parents and teachers. It breaks the tradition where parents had to meet teachers for student report and progress.

Now days, school students are more interested in upcoming technologies and can understand it very quickly. So school management system provides them platform to have some little but way interesting and useful interaction while studying in school.

This system is basically designed to include parents. And parents can look after their child's progress and can interact with teachers from home or from workplace.

Teachers can add online assignments; keep the record of attendance and marks.

BIBLIOGRAPHY

Bibliography

Books

- .NET PROGRAMMING (BLACK BOOK)
- HTML, DHTML, Ivan Bayross.
- ASP.NET in 21 days.
- Microsoft .NET Framework 2.0 window based client development.

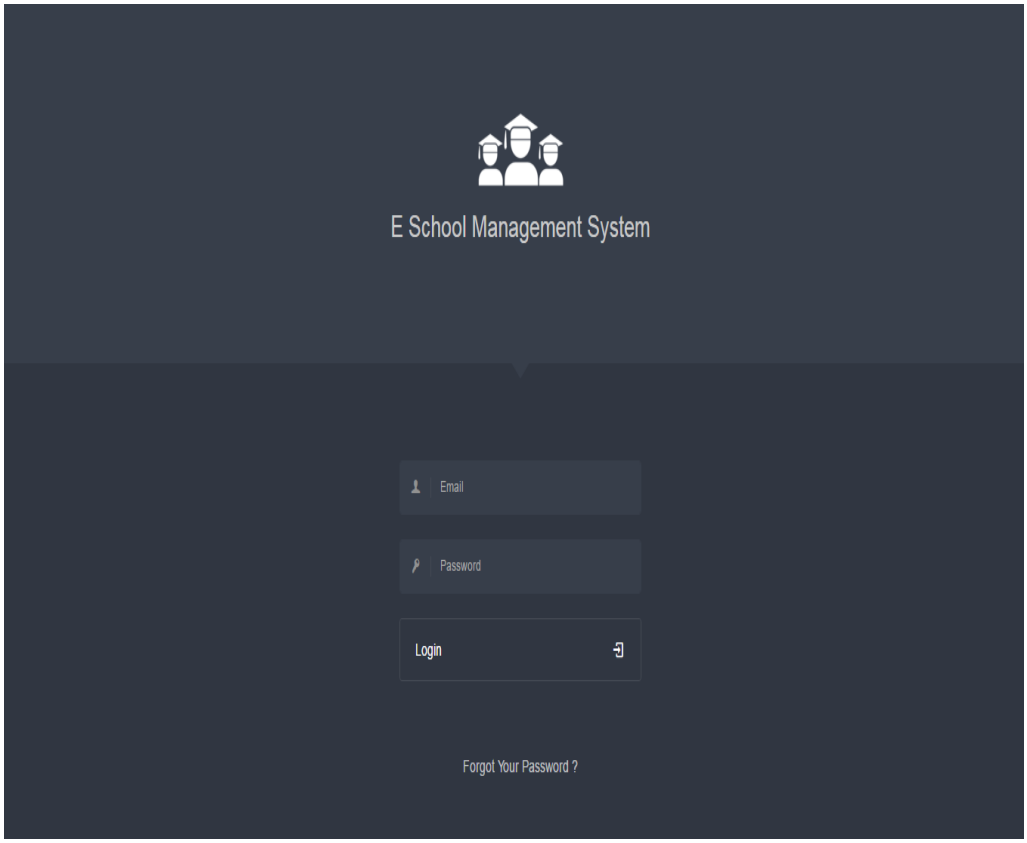
Web references

- http://docs.embarcadero.com/
- http://asp.net-tutorials.com/
- http://www.asp.net/aspnet/overview/making-websites-with-aspnet/making-websites-with-aspnet
- https://msdn.microsoft.com/en-us/library/k4cbh4dh.aspx
- http://www.stackoverflow.com/

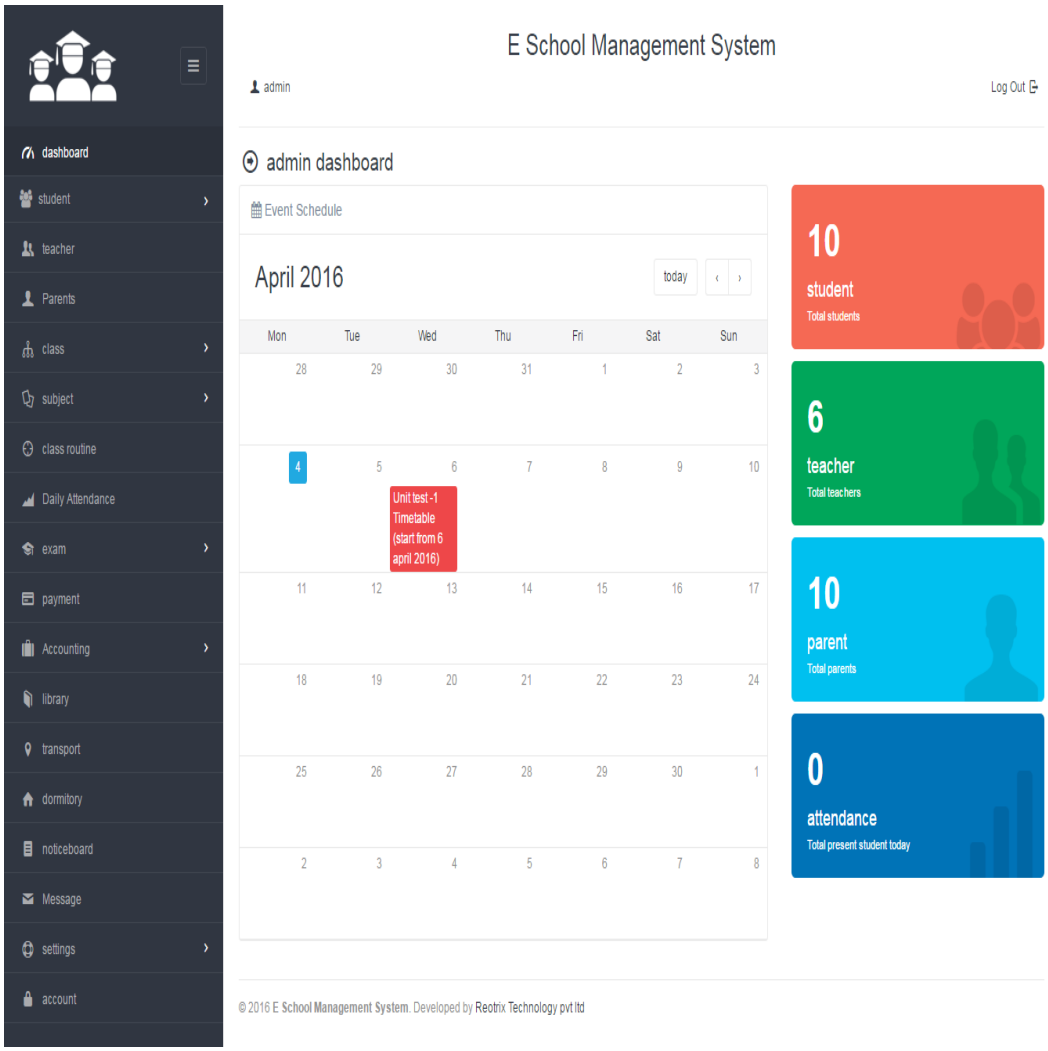
ANNEXURES

ANNEXURES 1: USER INTERFACE SCREENS

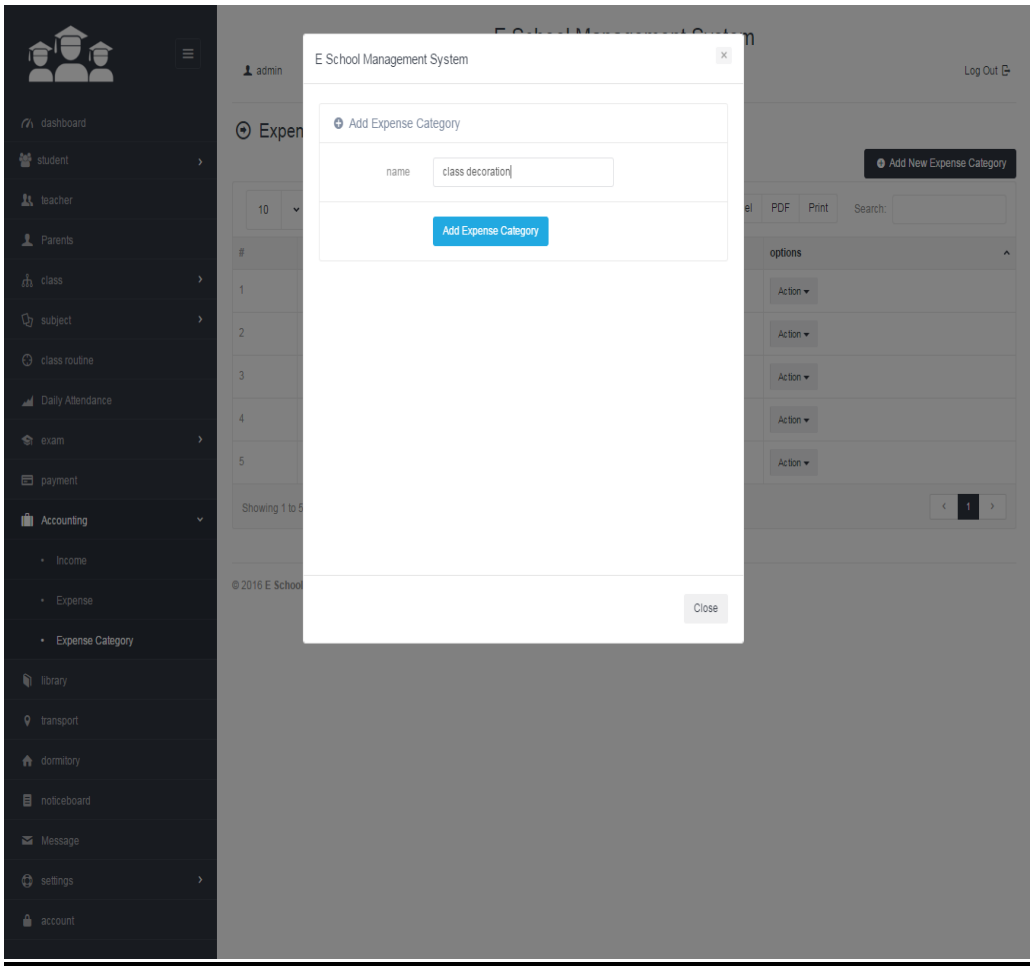
Login Screen



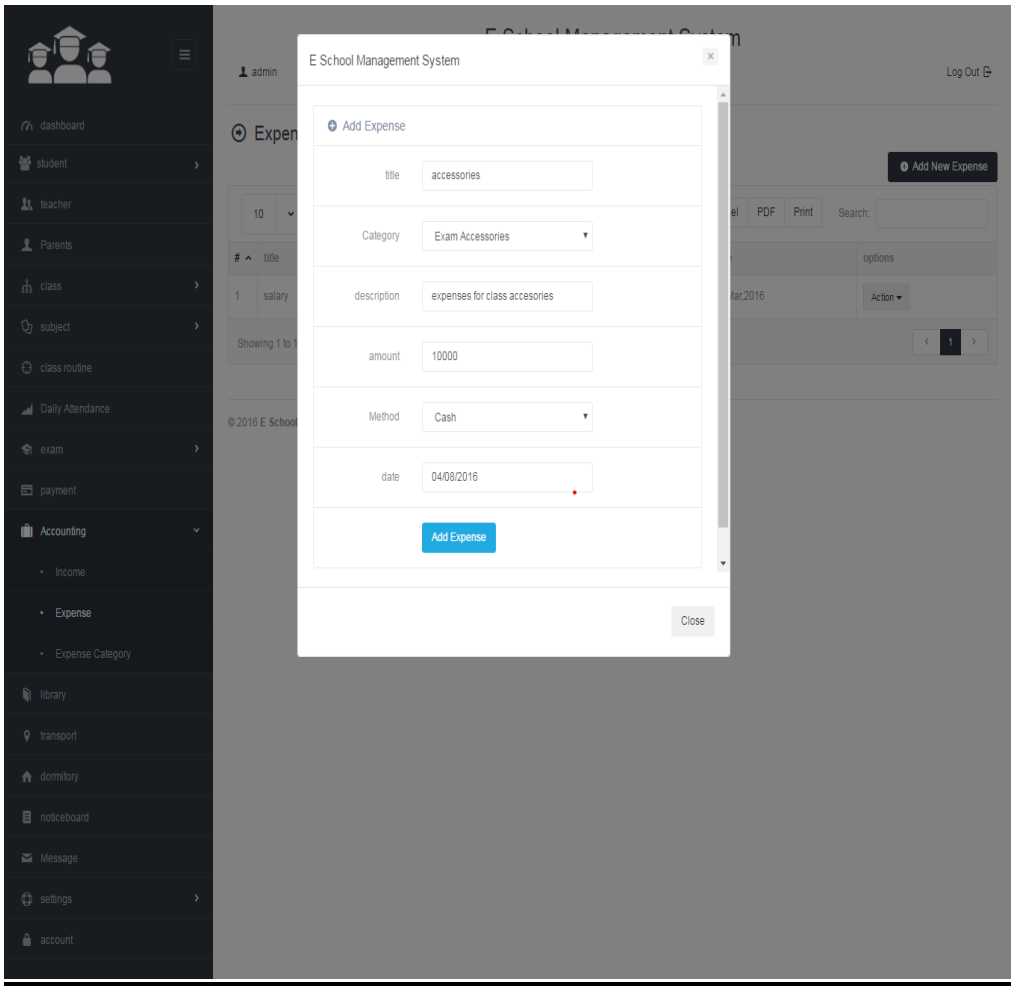
Admin Dashboard



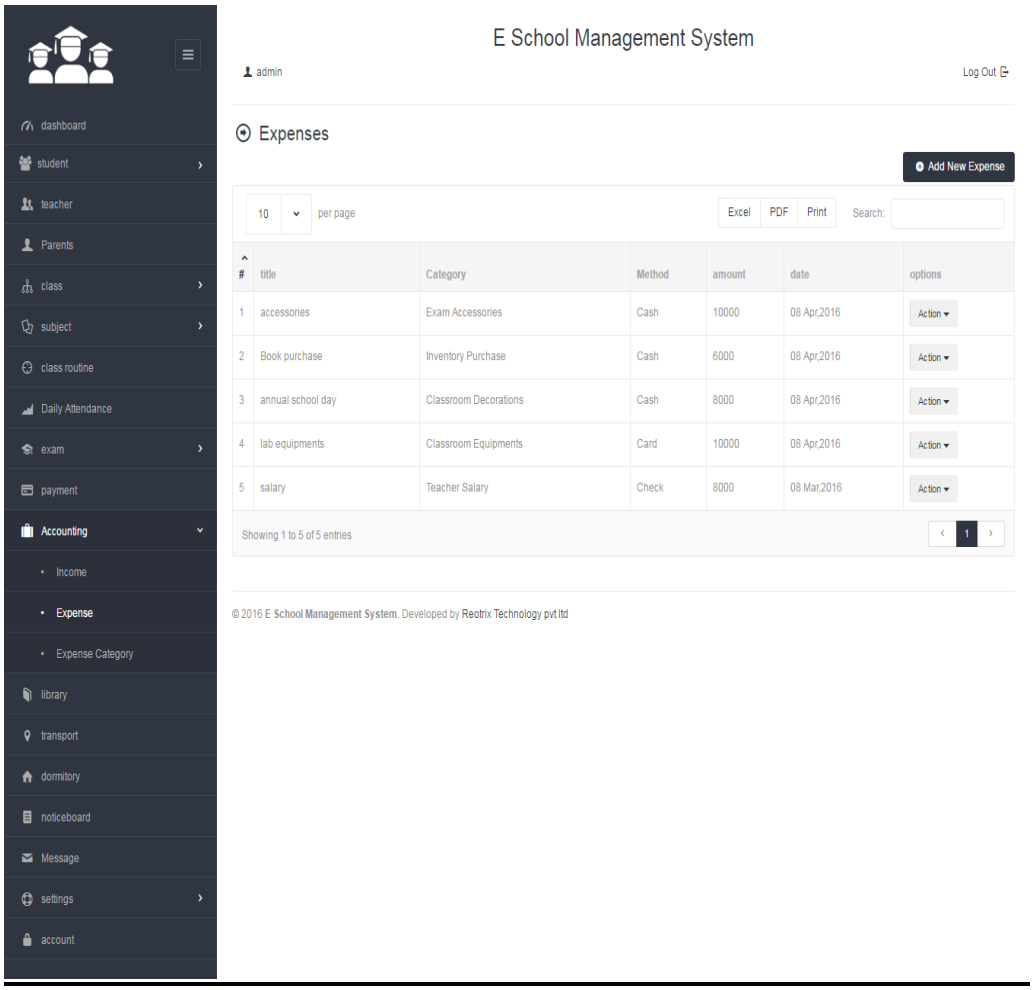
Add Expenses Category



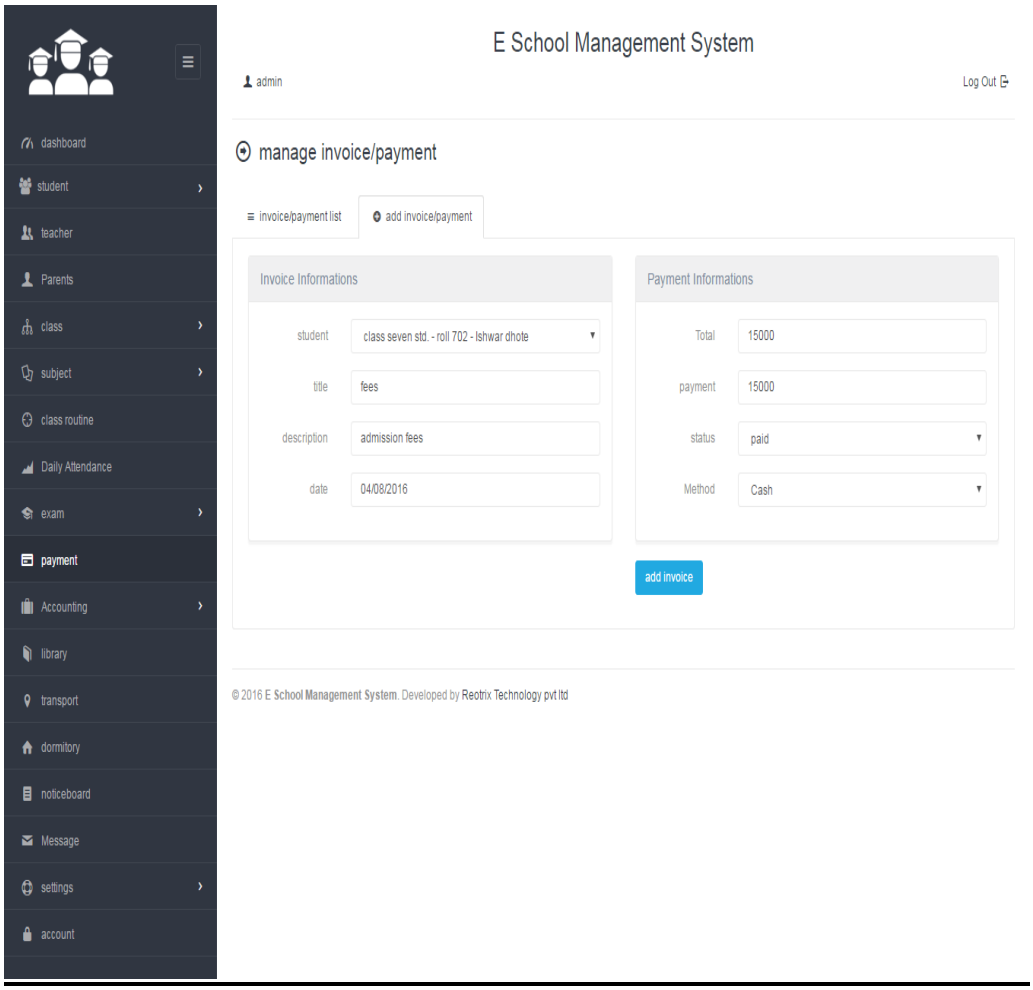
Add Expenses



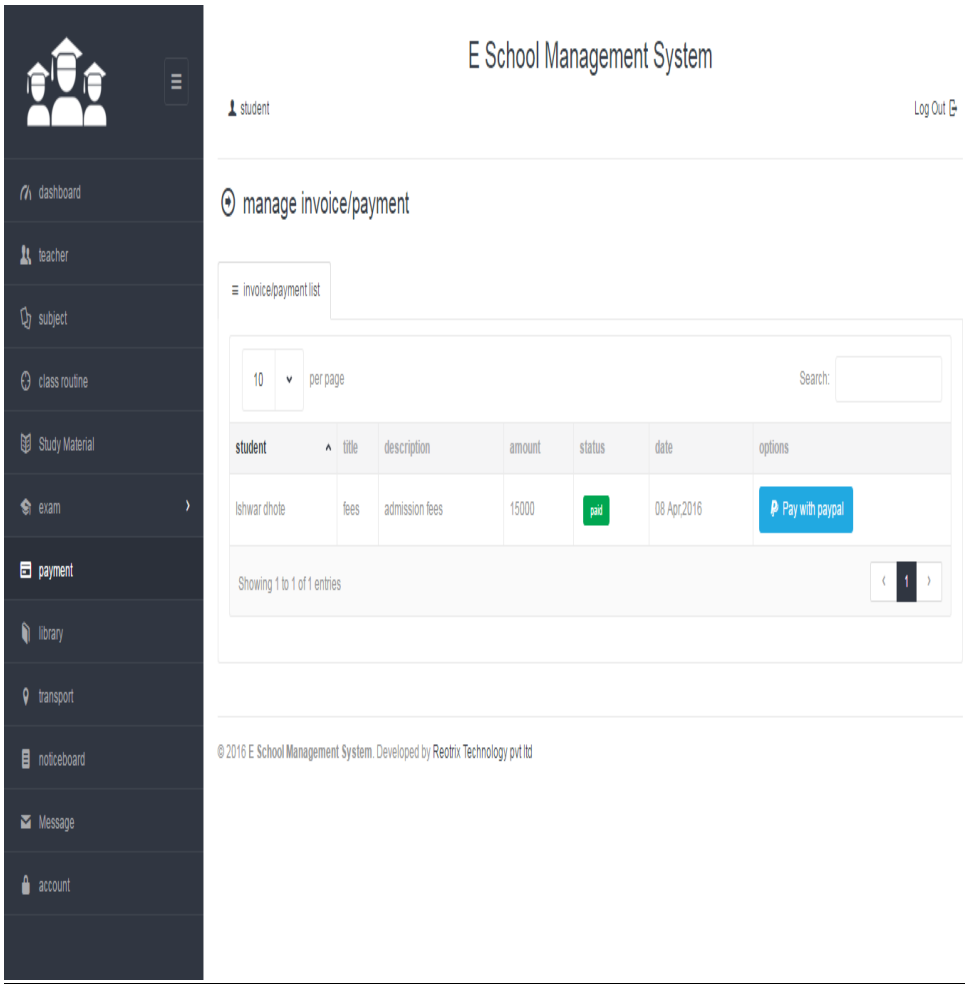
List of Expenses



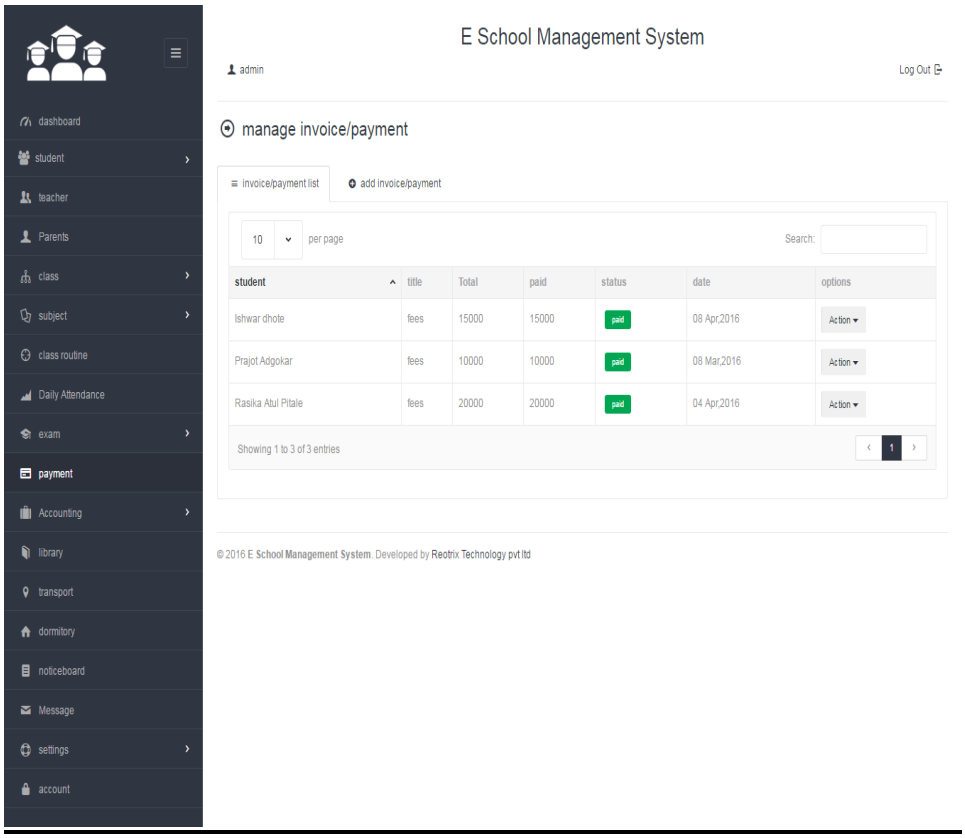
Manage invoice/Payment



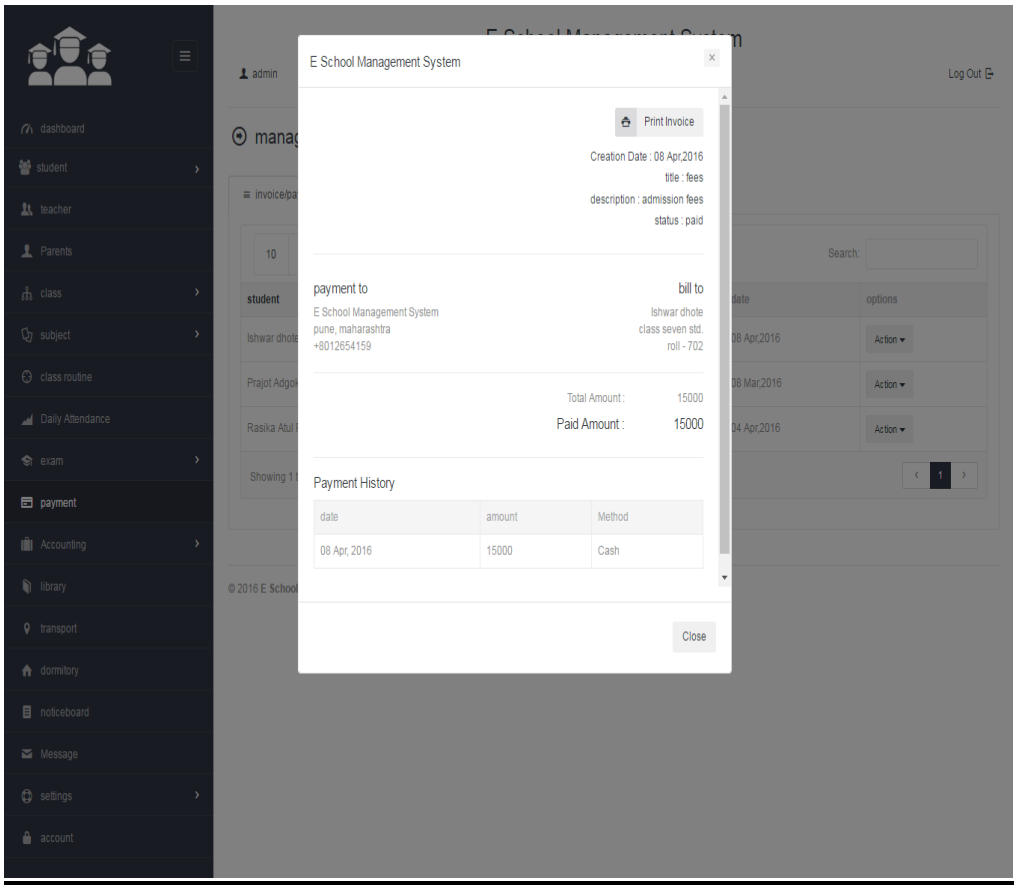
List of Invoice



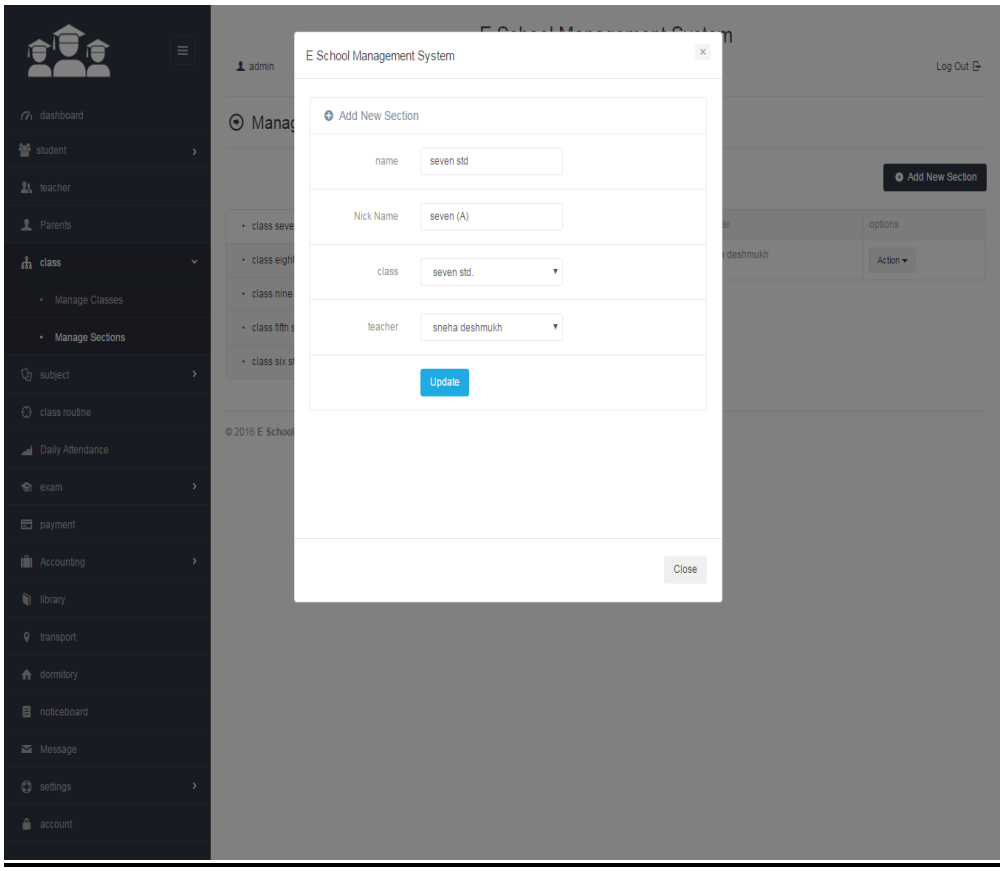
Manage invoice/payment



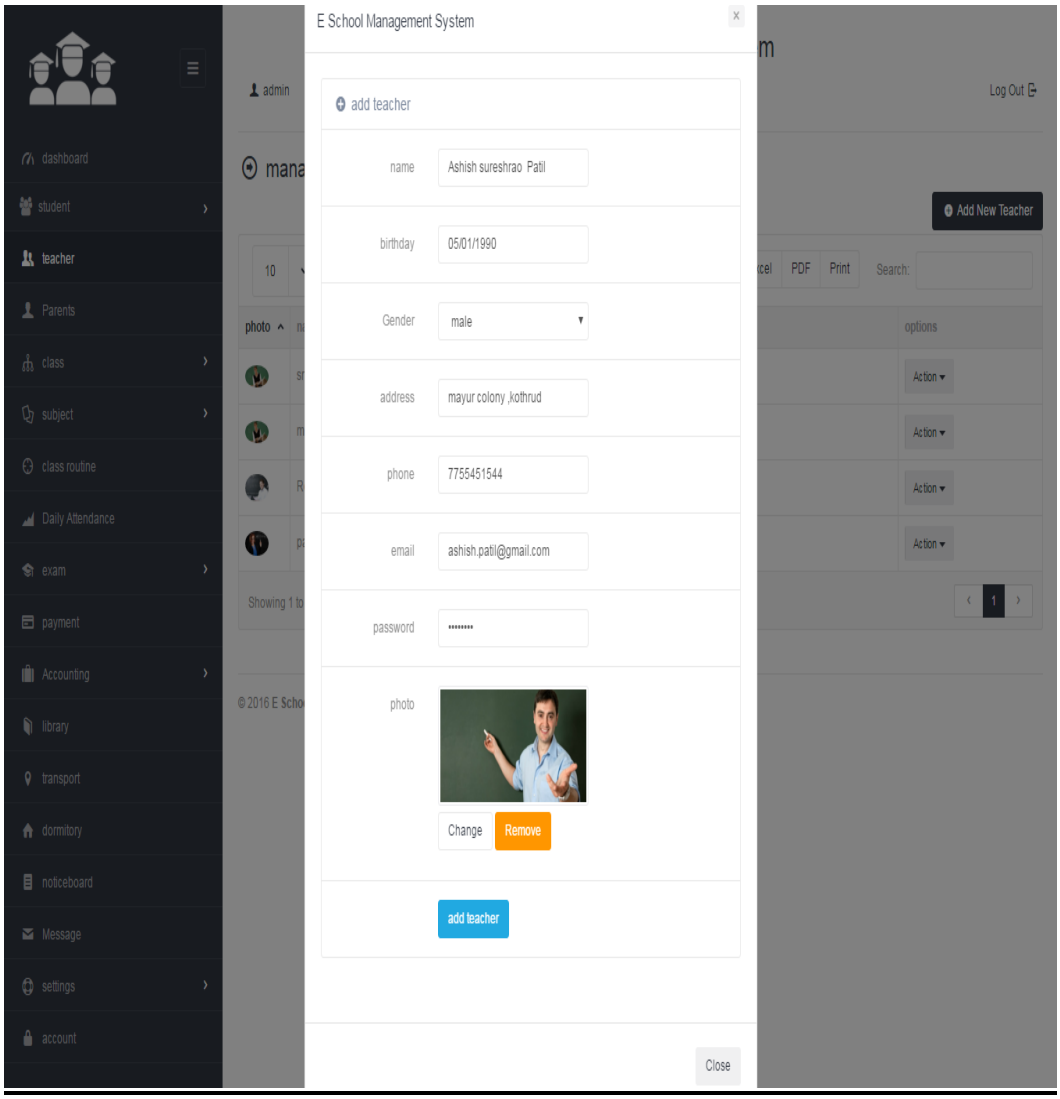
Invoice


E School Management System
Print Invoice
Creation Date : 08 Apr,2016
title : fees
description : admission fees
status : paid
payment to
E School Management System
pune, maharashtra
+8012654159
bill to
Ishwar dhote
class seven std.
roll - 702
Total Amount : 15000
Paid Amount : 15000
Payment History
date
amount
Method
08 Apr, 2016
15000
Cash
Close
admin
Log Out
dashboard
student
teacher
Parents
class
subject
class routine
Daily Attendance
exam
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account

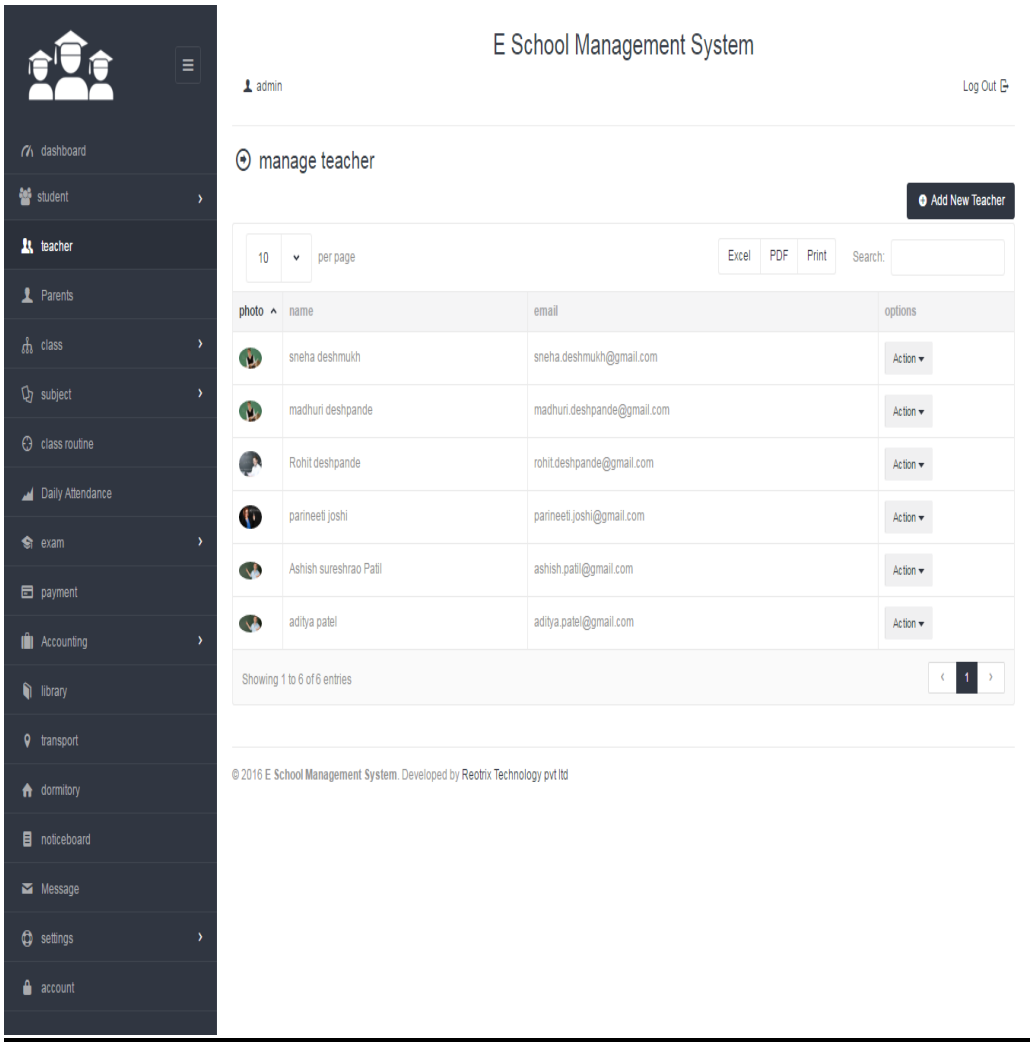
Add Section


E School Management System
admin
Log Out
dashboard
student
teacher
Parents
class
Manage Classes
Manage Sections
subject
class routine
Daily Attendance
exam
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
Add New Section
Add New Section
name
seven std
Nick Name
seven (A)
class
seven std.
teacher
sneha deshmukh
Update
Close
options
Action

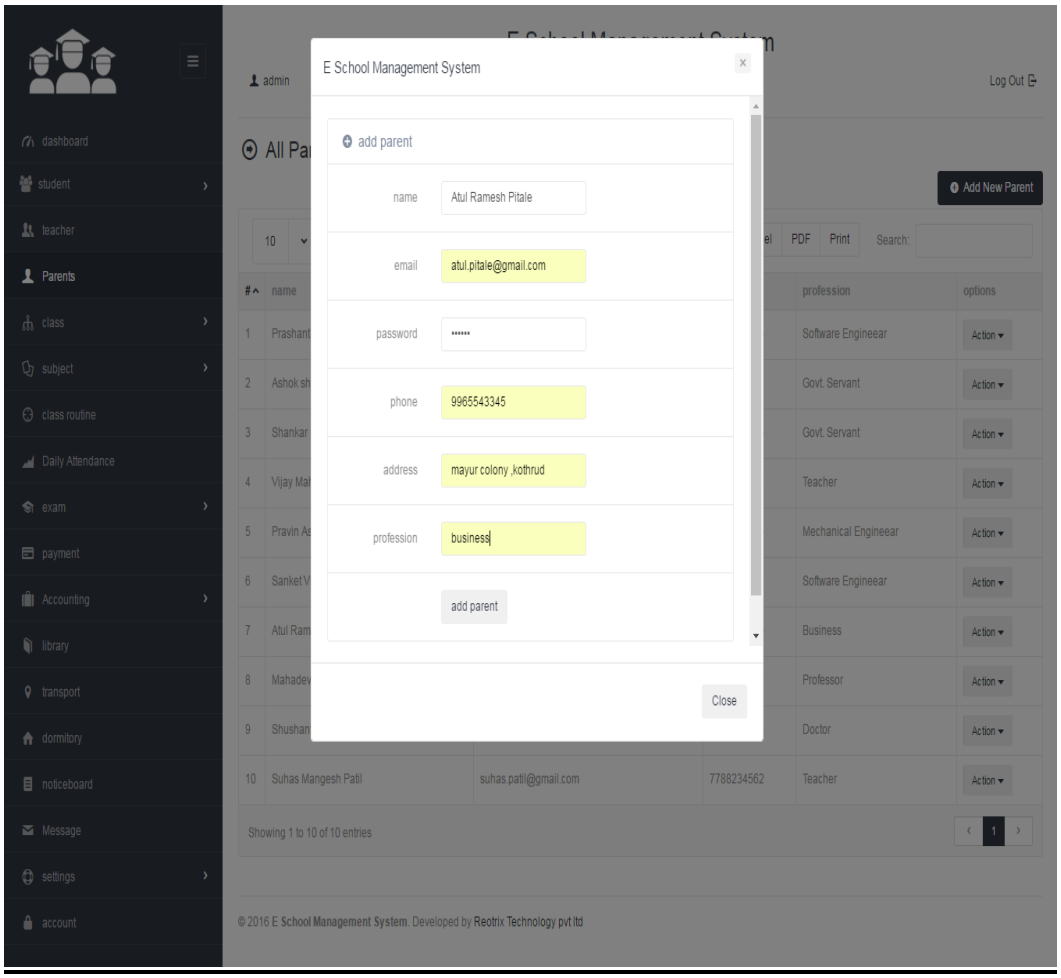
Add Teacher


E School Management System
add teacher
name
Ashish sureshrao Patil
birthday
05/01/1990
Gender
male
address
mayur colony ,kothrud
phone
7755451544
email
ashish.patil@gmail.com
password
photo
Change
Remove
add teacher
Close
dashboard
student
teacher
Parents
class
subject
class routine
Daily Attendance
exam
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
admin
Log Out
Add New Teacher
PDF
Print
Search:
options
Action

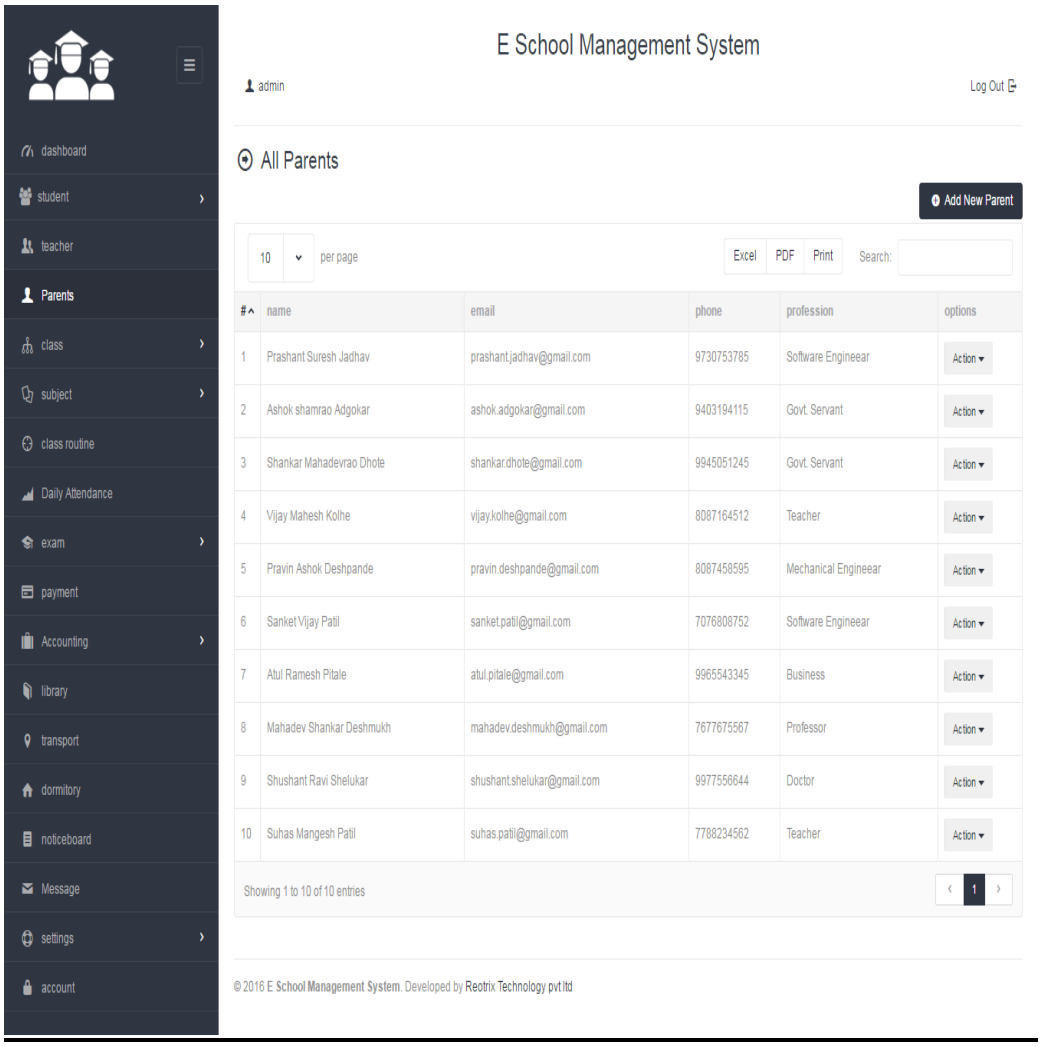
Manage Teacher



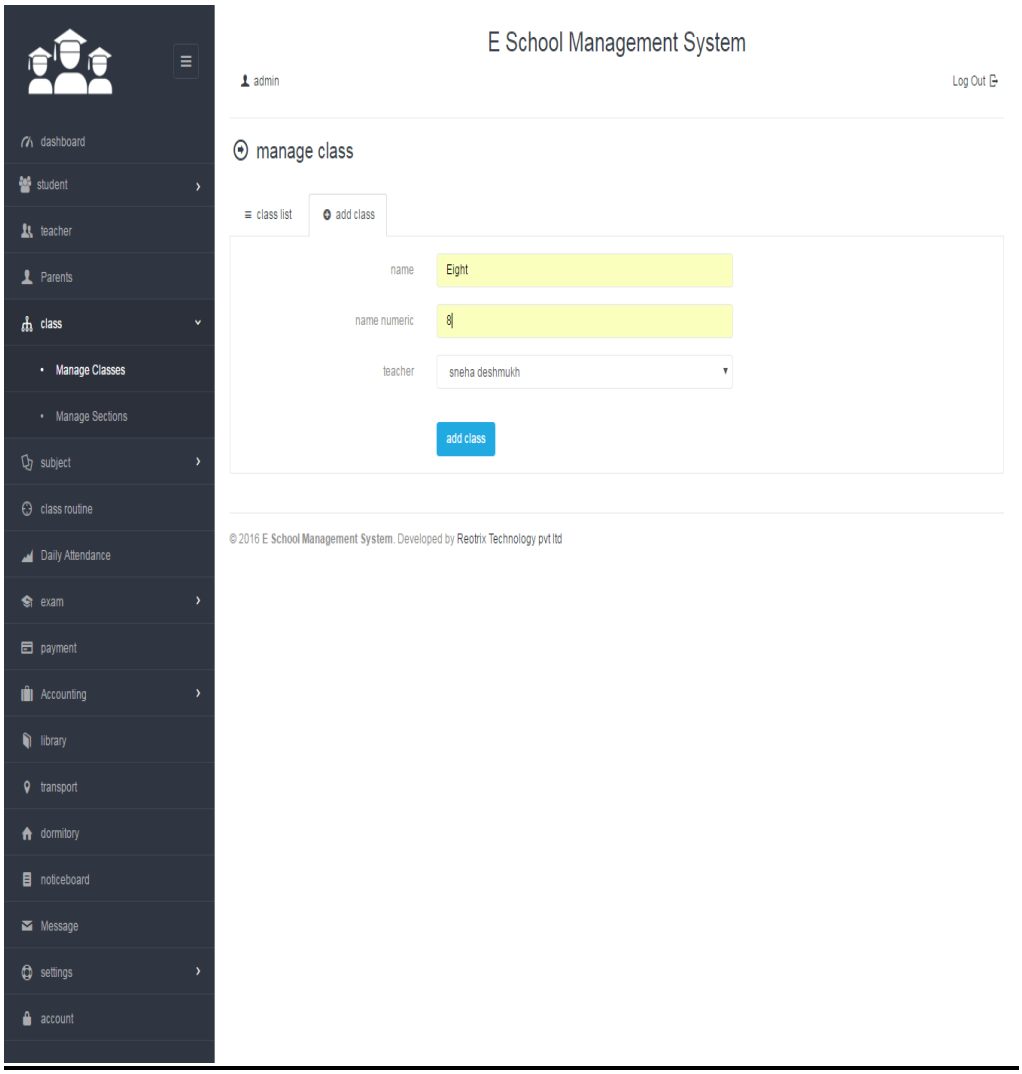
Add Parent


E School Management System
add parent
name
Atul Ramesh Pitale
email
atul.pitale@gmail.com
password
phone
9965543345
address
mayur colony ,kothrud
profession
business
add parent
Close
admin
Log Out
Add New Parent
dashboard
student
teacher
Parents
class
subject
class routine
Daily Attendance
exam
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
PDF
Print
Search:
profession
options
Software Engineear
Govt. Servant
Govt. Servant
Teacher
Mechanical Engineear
Software Engineear
Business
Professor
Doctor
Teacher
Action
10 Suhas Mangesh Patil suhas.patil@gmail.com 7788234562
Showing 1 to 10 of 10 entries
© 2016 E School Management System. Developed by Reotrix Technology pvt ltd

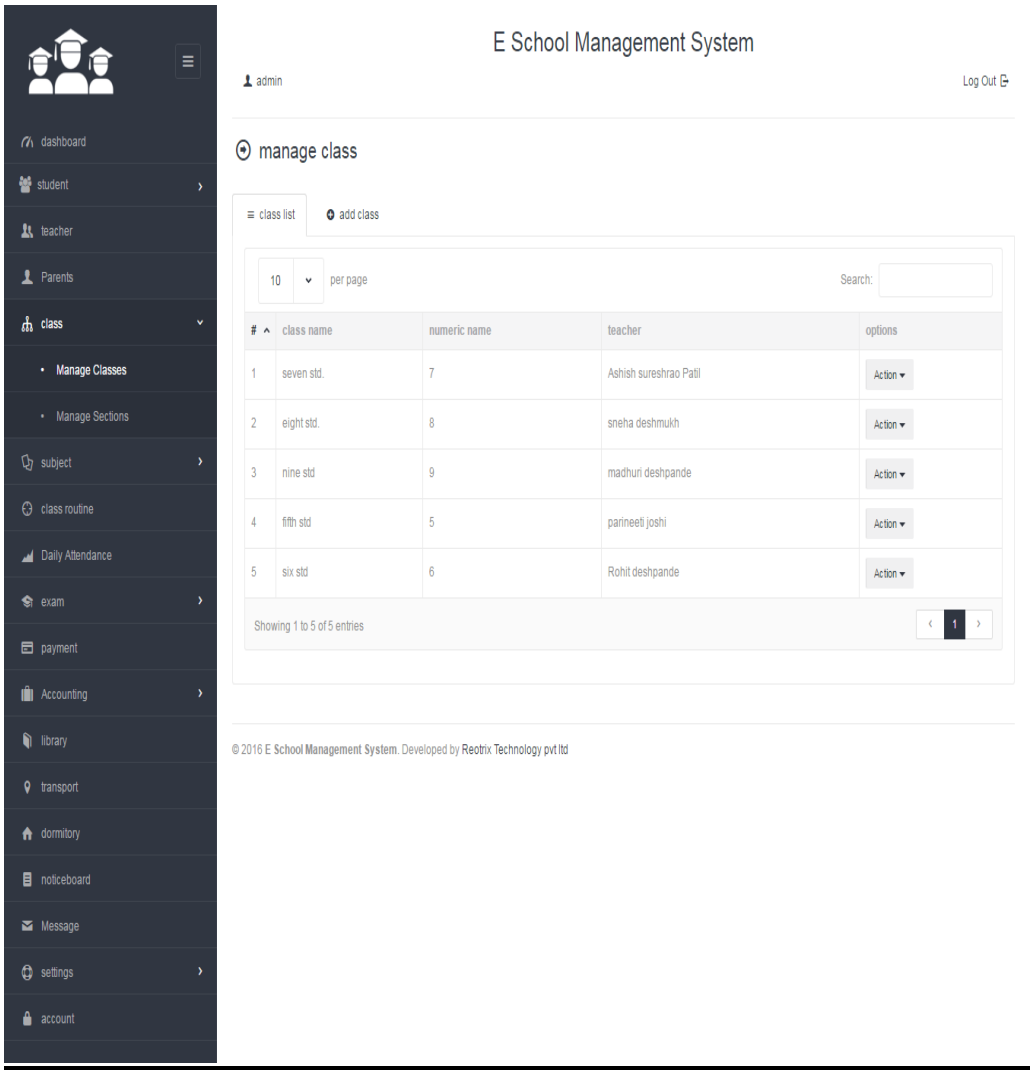
All Parents List



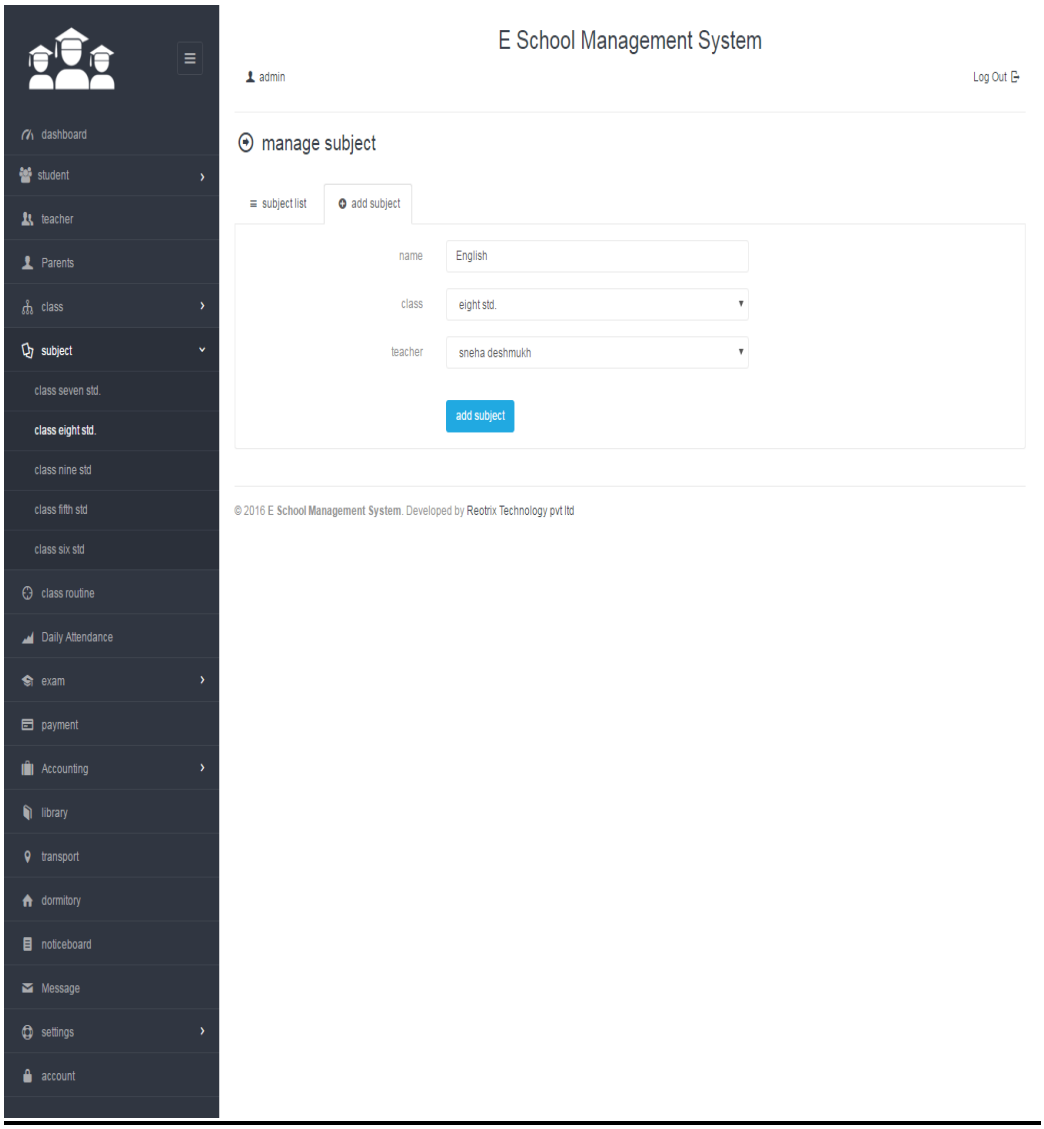
Manage Class



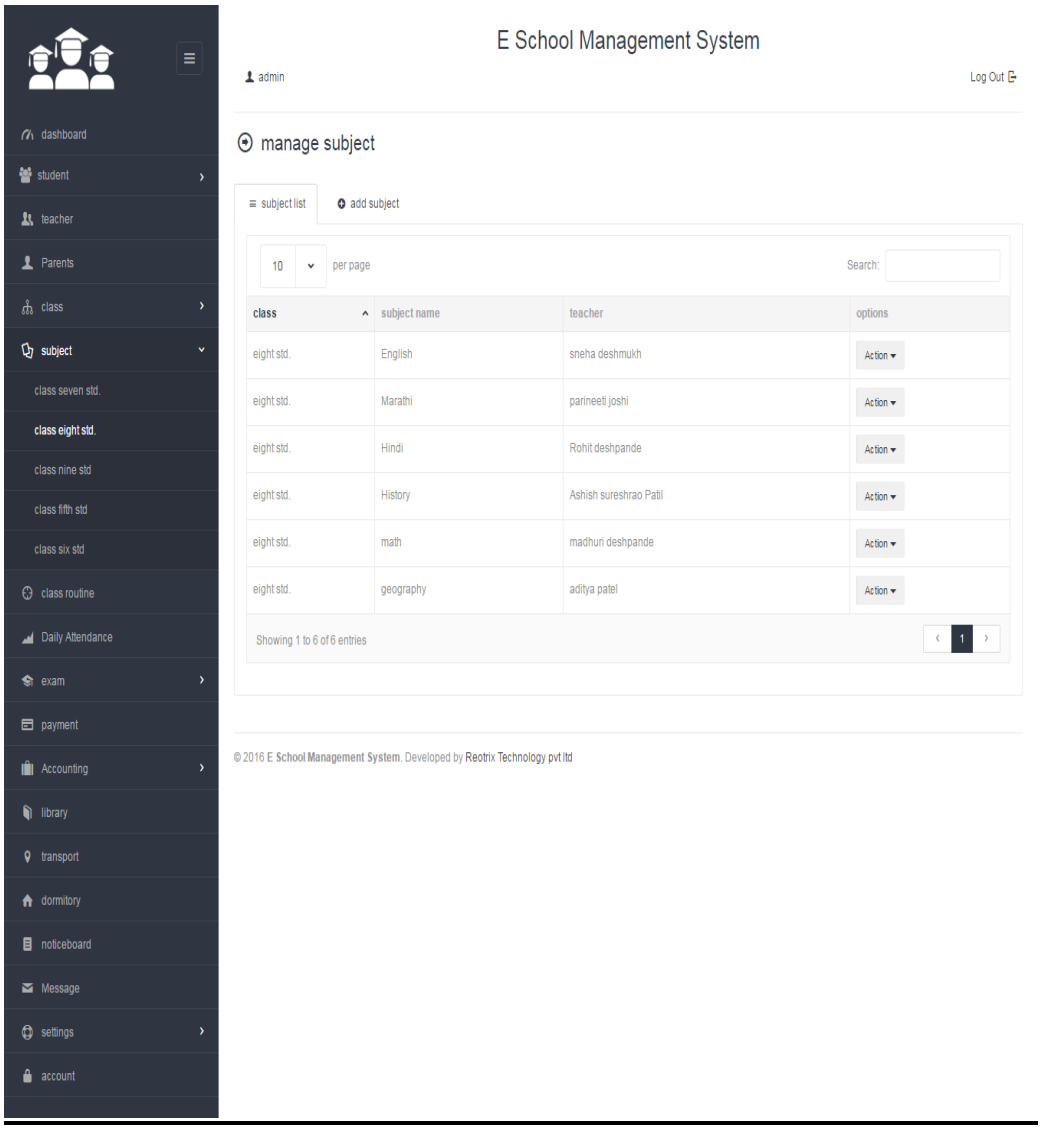
Class List


E School Management System
admin
Log Out
dashboard
student
teacher
Parents
class
Manage Classes
Manage Sections
subject
class routine
Daily Attendance
exam
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
manage class
class list
add class
10
per page
Search:
class name numeric name teacher options
1 seven std. 7 Ashish sureshrao Patil Action
2 eight std. 8 sneha deshmukh Action
3 nine std 9 madhuri deshpande Action
4 fifth std 5 parineeti joshi Action
5 six std 6 Rohit deshpande Action
Showing 1 to 5 of 5 entries
1
© 2016 E School Management System. Developed by Reotrix Technology pvt ltd

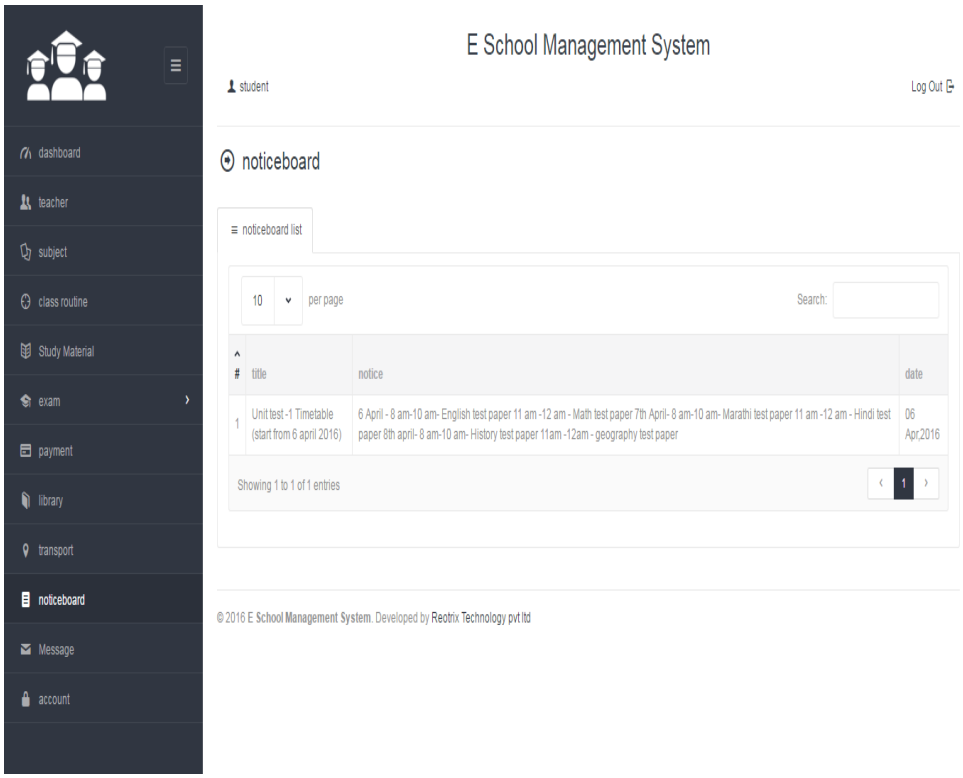
Add Subject


E School Management System
admin
Log Out
dashboard
student
teacher
Parents
class
subject
class seven std.
class eight std.
class nine std
class fifth std
class six std
class routine
Daily Attendance
exam
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
manage subject
subject list
add subject
name
English
class
eight std.
teacher
sneha deshmukh
add subject
© 2016 E School Management System. Developed by Reotrix Technology pvt ltd

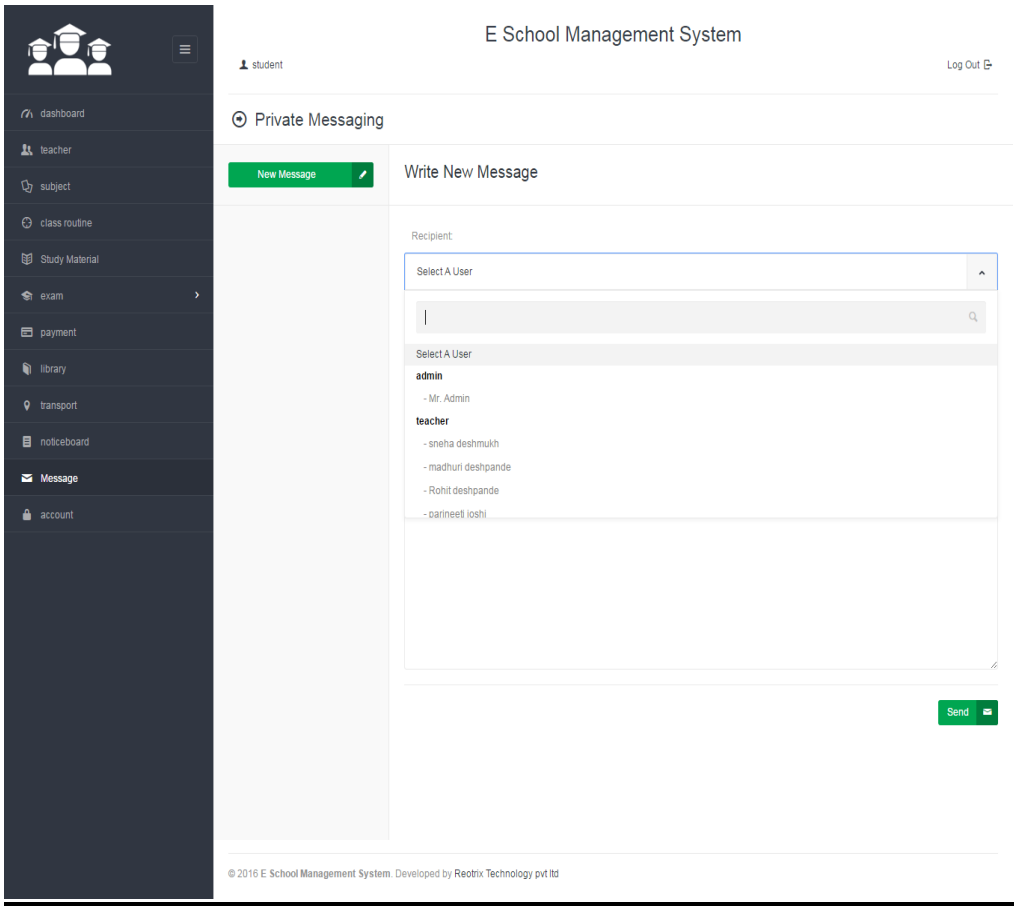
Subjects List



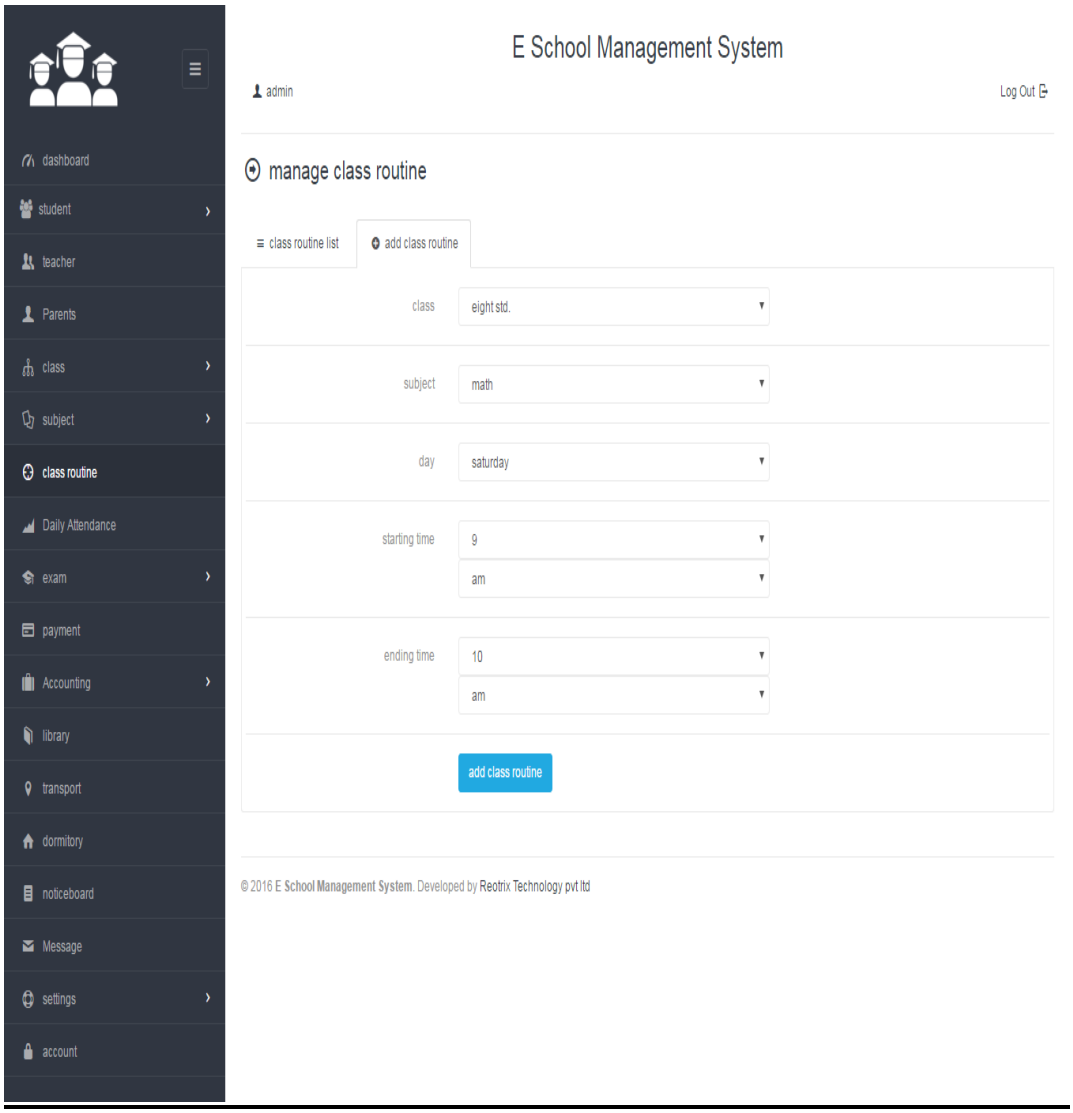
Notice Board



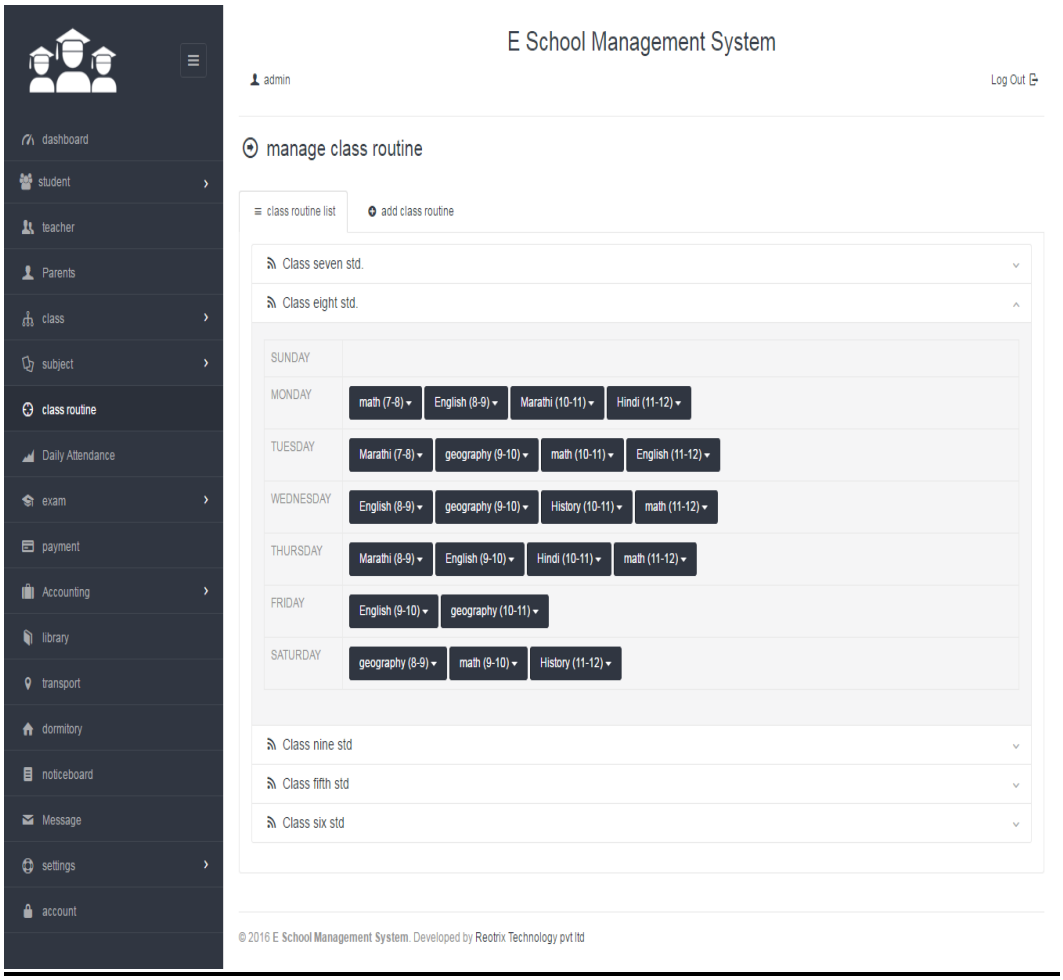
Private Message



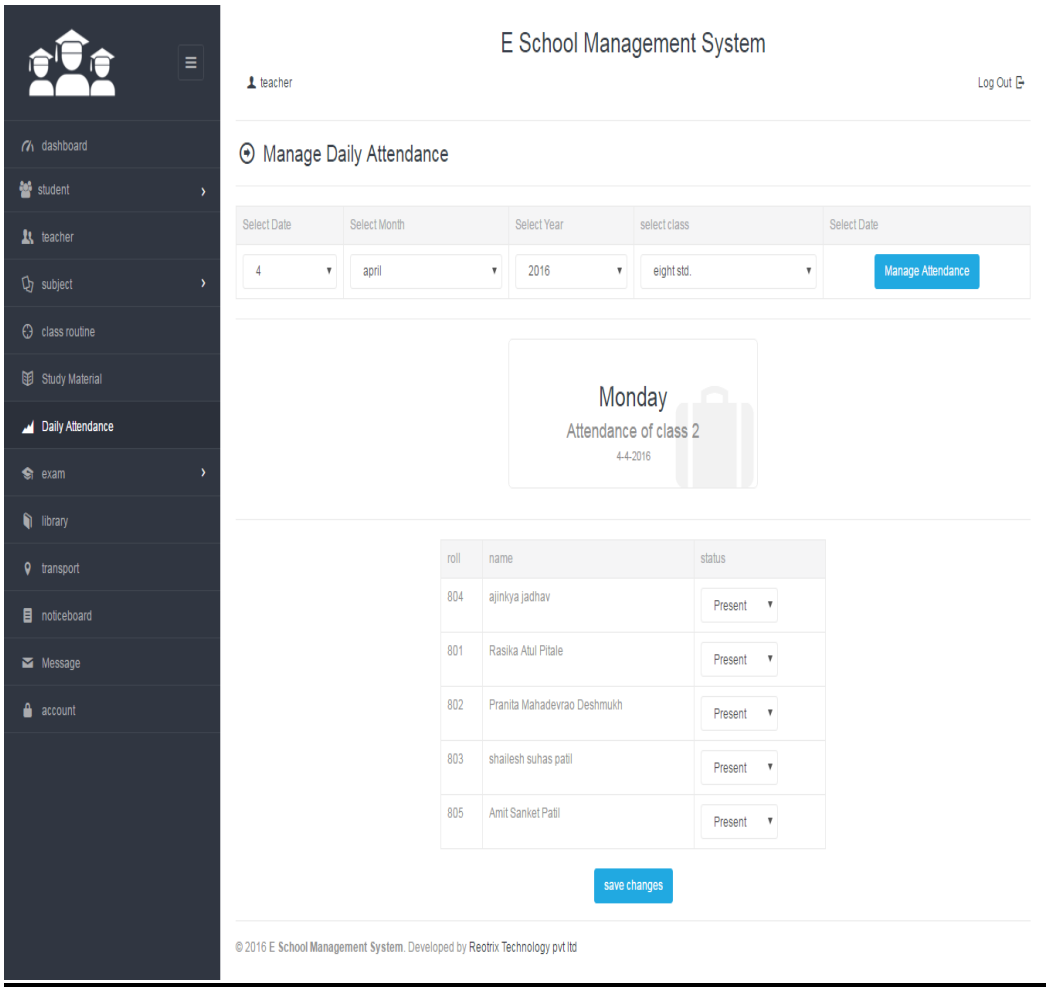
Update Class Routine



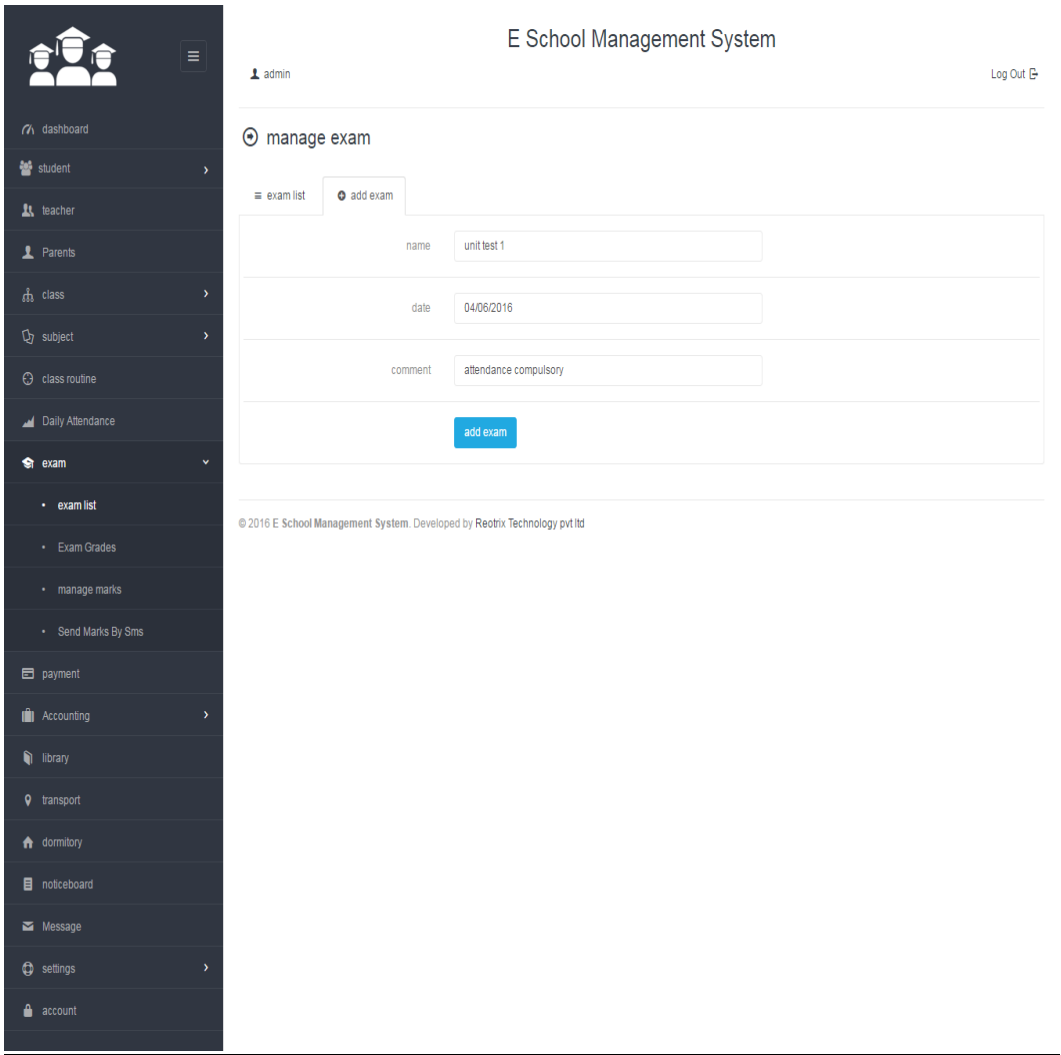
Class Routine



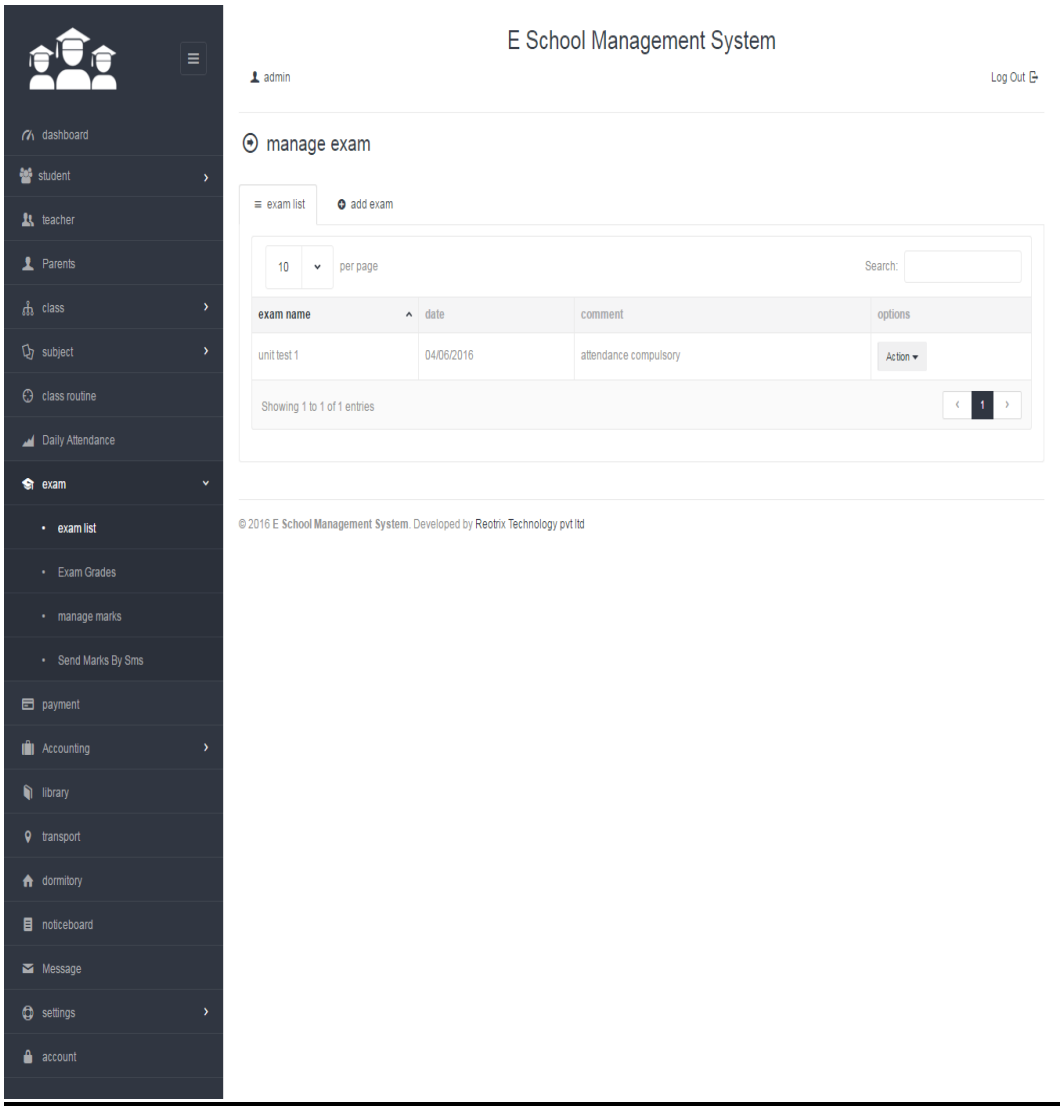
Daily Attendance



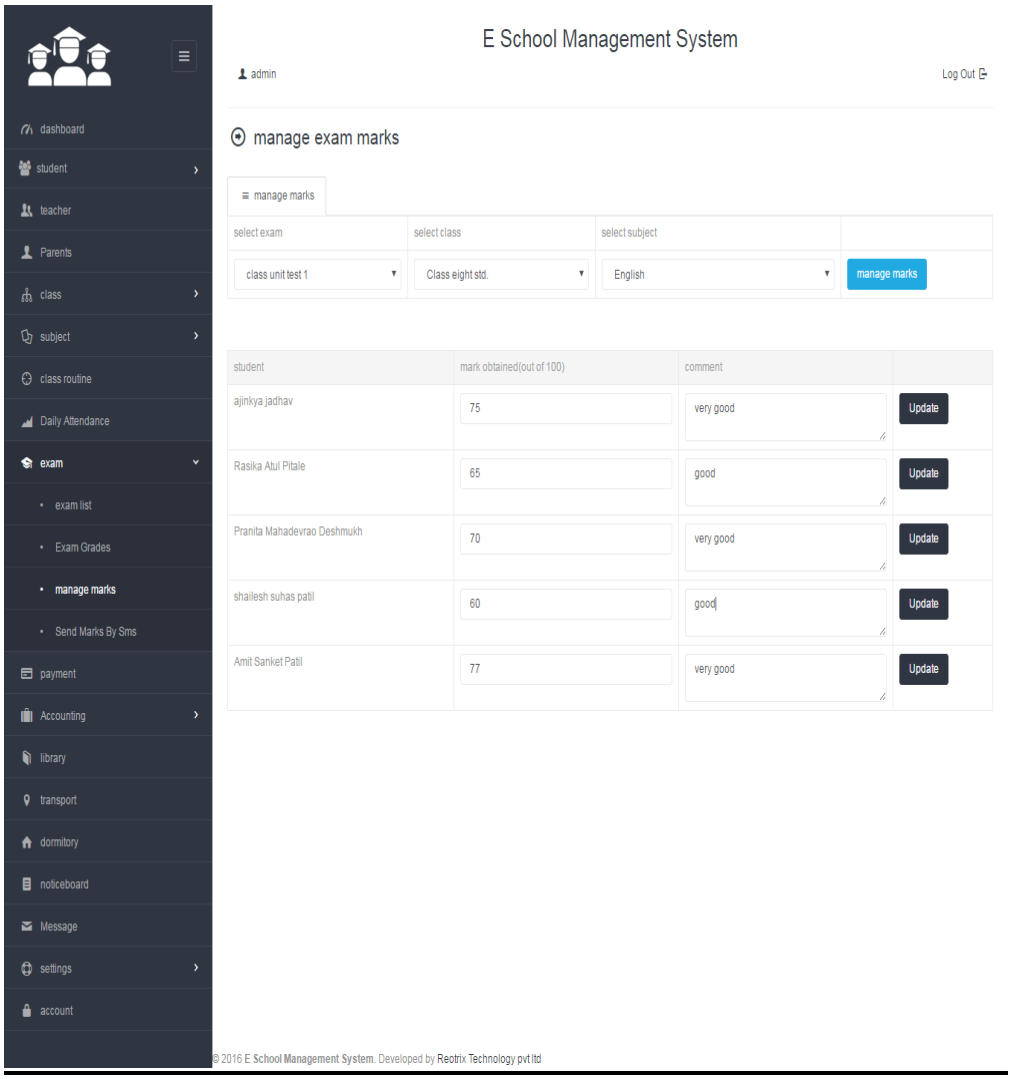
Add Exam


E School Management System
admin
Log Out
dashboard
student
teacher
Parents
class
subject
class routine
Daily Attendance
exam
exam list
Exam Grades
manage marks
Send Marks By Sms
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
manage exam
exam list
add exam
name
unit test 1
date
04/06/2016
comment
attendance compulsory
add exam
© 2016 E School Management System. Developed by Reotrix Technology pvt ltd

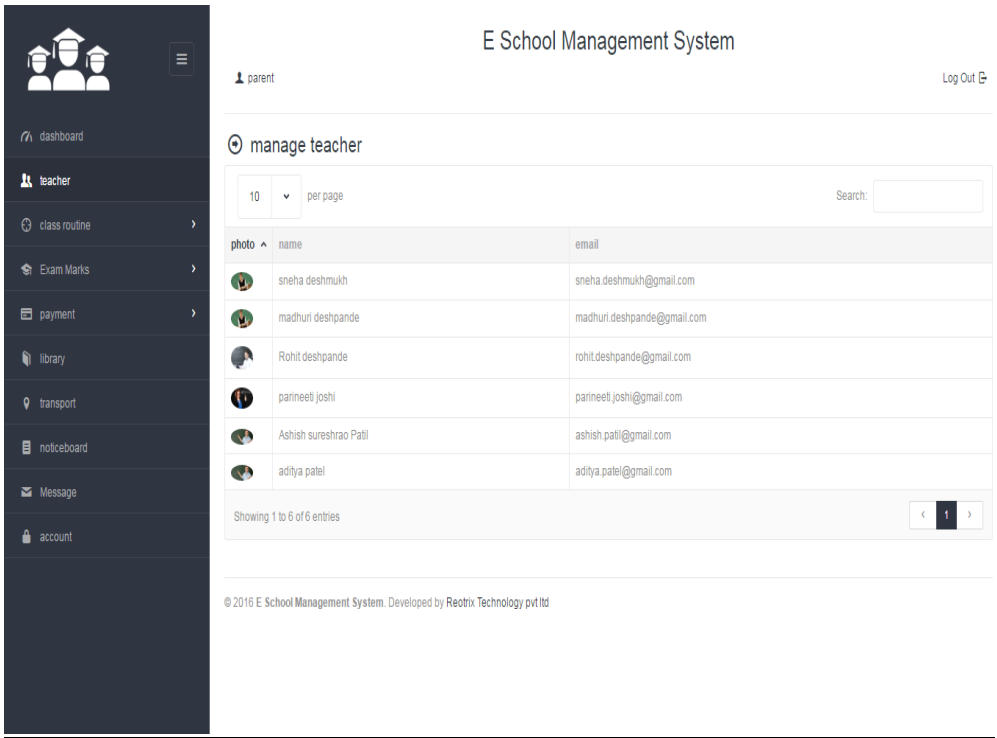
Manage Exam


E School Management System
admin
Log Out
dashboard
student
teacher
Parents
class
subject
class routine
Daily Attendance
exam
exam list
Exam Grades
manage marks
Send Marks By Sms
payment
Accounting
library
transport
dormitory
noticeboard
Message
settings
account
manage exam
exam list
add exam
10
per page
Search:
exam name
date
comment
options
unit test 1
04/06/2016
attendance compulsory
Action
Showing 1 to 1 of 1 entries
1
© 2016 E School Management System. Developed by Reotrix Technology pvt ltd

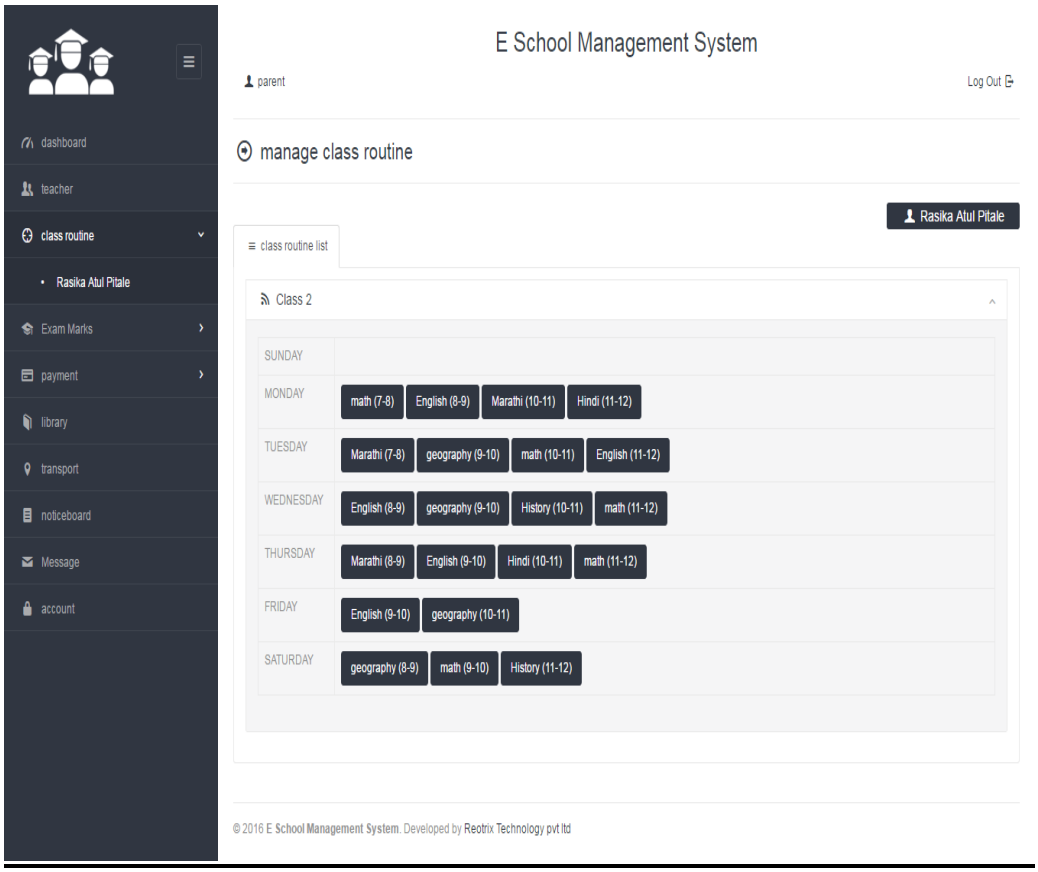
Manage Exam Marks



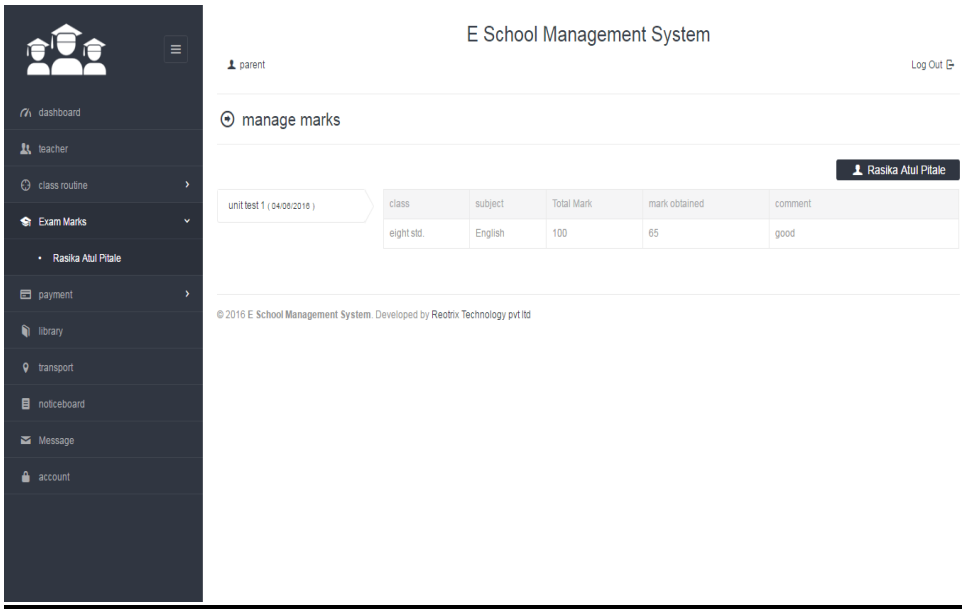
Manage Teacher Screen



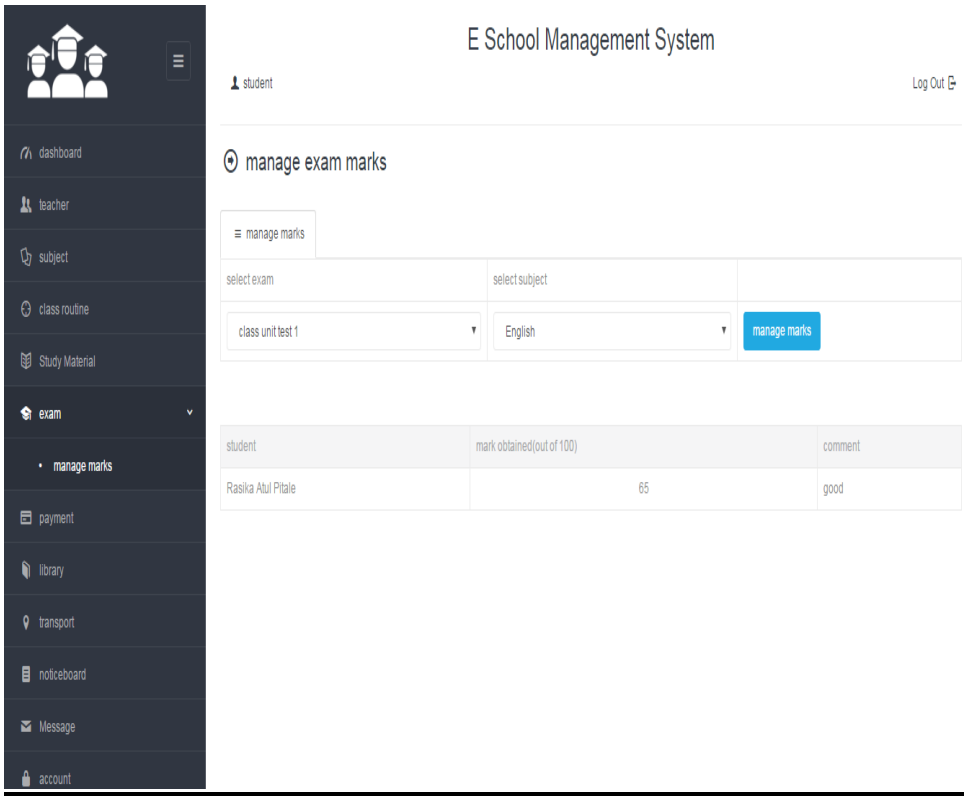
Class Routine



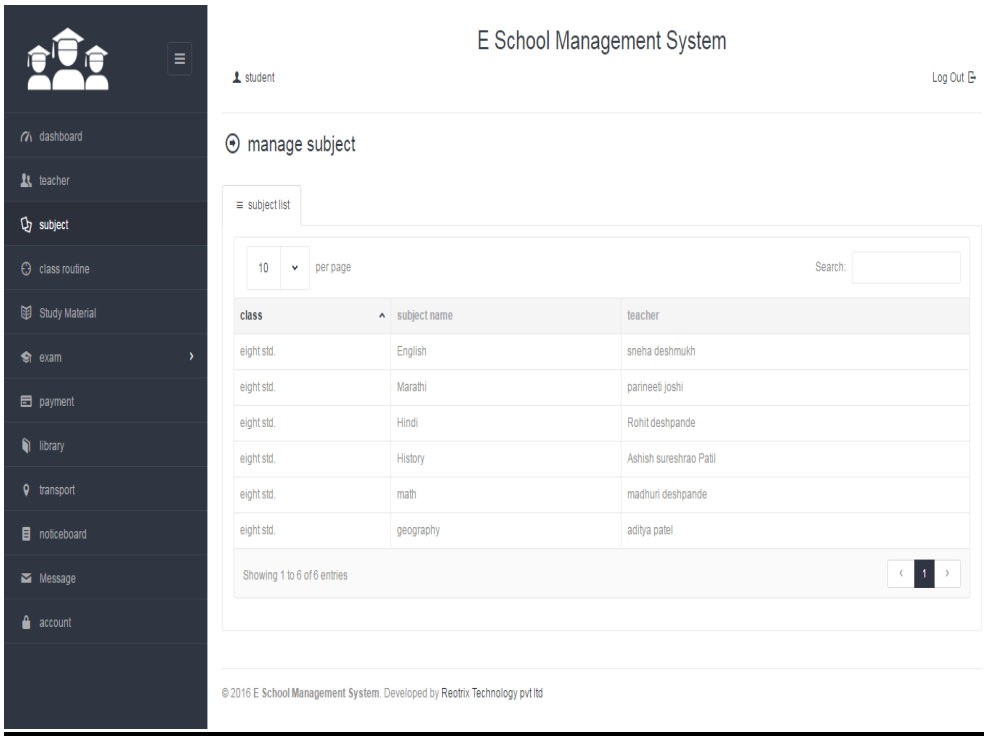
Marks of Exams



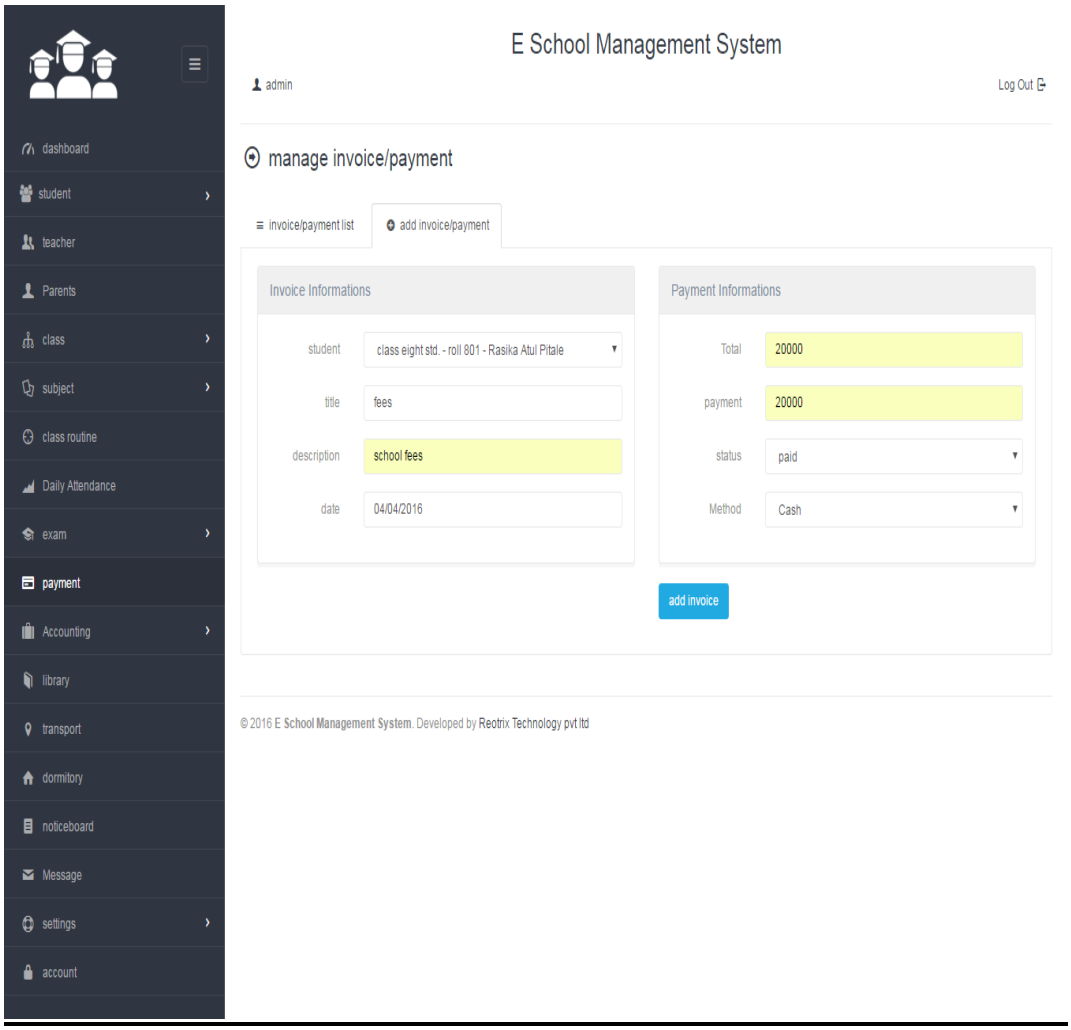
Manage Exam Marks



Subject List



Manage Invoice and Payment



ANNEXURE 2:- OUTPUT REPORTS WITH DATA

Student Information

Student Information - class eight std. Ekta School

roll	name	address	email
801	Rasika Atul Pitale	mayur colony ,kothrud	rasika.pitale@gmail.com
802	Pranita Mahadevrao Deshmukh	waarje ,pune	pranita.deshmukh@gmail.com
803	shailesh suhas patil	ekta colony,kothrud	shailesh.patil@gmail.com
804	ajinkya jadhav	shivaji nagar,pune	ajinkya.jadhav@gmail.com
805	Amit Sanket Patil	mayur colony ,kothrud	amit.patil@mail.com

Student Information - class seven std. Ekta School

roll	name	address	email
701	Prajot Adgokar	kothrud, pune	prajot.adgokar@gmail.com
702	Ishwar dhote	kothrud ,pune	ishwar.dhote@gmail.com
703	abhijjet kolhe	warje, pune	abhijeet.kolhe@gmail.com
704	prajot deshpande	karve nagar,pune	prajot.deshpande@gmail.com
705	sneha patil	shivaji nagar ,pune	sneha.patil@gmail.com

Teacher Information

manage teacher Ekta School

name	email
sneha deshmukh	sneha.deshmukh@gmail.com
madhuri deshpande	madhuri.deshpande@gmail.com
Rohit deshpande	rohit.deshpande@gmail.com
parineeti joshi	parineeti.joshi@gmail.com
Ashish sureshrao Patil	ashish.patil@gmail.com
aditya patel	aditya.patel@gmail.com

Parent Information

4/5/2016 All Parents | Ekta School

#^	name	email	phone	profession
1	Prashant Suresh Jadhav	prashant.jadhav@gmail.com	9730753785	Software Engineear
2	Ashok shamrao Adgokar	ashok.adgokar@gmail.com	9403194115	Govt. Servant
3	Shankar Mahadevrao Dhote	shankar.dhote@gmail.com	9945051245	Govt. Servant
4	Vijay Mahesh Kolhe	vijay.kolhe@gmail.com	8087164512	Teacher
5	Pravin Ashok Deshpande	pravin.deshpande@gmail.com	8087458595	Mechanical Engineear
6	Sanket Vijay Patil	sanket.patil@gmail.com	7076808752	Software Engineear
7	Atul Ramesh Pitale	atul.pitale@gmail.com	9965543345	Business
8	Mahadev Shankar Deshmukh	mahadev.deshmukh@gmail.com	7677675567	Professor
9	Shushant Ravi Shelukar	shushant.shelukar@gmail.com	9977556644	Doctor
10	Suhas Mangesh Patil	suhas.patil@gmail.com	7788234562	Teacher

Invoice Details

Creation Date : 08 Apr,2016
title : fees
description : admission fees
status : paid

payment to

E School Management System
pune, maharashtra
+8012654159

bill to

Ishwar dhote
class seven std.
roll - 702

Total Amount : 15000

Paid Amount : 15000

Payment History

date	amount	Method
08 Apr, 2016	15000	Cash

Expenses

Expenses Ekta School

title	Category	Method	amount	date
accessories	Exam Accessories	Cash	10000	08 Apr,2016
Book purchase	Inventory Purchase	Cash	6000	08 Apr,2016
annual school day	Classroom Decorations	Cash	8000	08 Apr,2016
lab equipments	Classroom Equipments	Card	10000	08 Apr,2016
salary	Teacher Salary	Check	8000	08 Mar,2016

Income Details Report

Incomes Ekta School

title	description	Method	amount	date
fees	admission fees	Cash	15000	08 Apr,2016
fees	school fees	Cash	20000	04 Apr,2016
fees	admission fees	Cash	10000	08 Mar,2016

Expenses Category

#	name
1	Teacher Salary
2	Classroom Equipments
3	Classroom Decorations
4	Inventory Purchase
5	Exam Accessories

ANNEXURE 2:- SAMPLE PROGRAM CODE

```
<body>
   <form id="form1" runat="server">
   <asp:ScriptManager ID="ScriptManager1" runat="server">
   </asp:ScriptManager>

   <div>
   <div class="panel panel-default"
         style="width: 800px; height:964px; padding: 10px; margin:
10px" >
   <div id="Tabs" role="tabpanel">
      <!-- Nav tabs -->
      <ul class="nav nav-tabs" role="tablist">
         <li class="active"><a href="#personal" aria-
controls="personal" role="tab" data-toggle="tab">
            Profile</a></li>
         <li><a href="#employment" aria-controls="employment"
role="tab" data-toggle="tab">Documents</a>

         </li>
         <li>
            <br />
         </li>
      </ul>
      <!-- Tab panes -->
      <div class="tab-content" style="padding-top: 20px; ">
         <div role="tabpanel" class="tab-pane active" id="personal">

            <table align="left" cellpadding="2" class="style1">
               <tr>
                  <td class="style8" valign="top">
                         
```

```
                    <asp:Image ID="Img_photo" runat="server"
Height="100px" Width="100px"
                      AlternateText="Student Image"
BorderColor="Black" BorderStyle="Double"
                      ToolTip="Student Image"
ImageAlign="AbsMiddle"
                      GenerateEmptyAlternateText="True"
CssClass="image" />
                     </td>
                  <td class="style7">
                       Student Profile : <asp:Label
ID="Label1" runat="server" Text="Label"></asp:Label>
                   <asp:Label ID="Label7" runat="server"
Text="Label"></asp:Label>
                   <asp:Label ID="Label8" runat="server"
Text="Label"></asp:Label>
                  </td>
                </tr>
                <tr valign="top" >
                  <td class="style3" align="justify">


                      <asp:Label ID="Label2"
runat="server" ></asp:Label>

                   <asp:Label ID="Label9" runat="server"
></asp:Label>

                    <asp:Label ID="Label10" runat="server"
></asp:Label>

                  <br />

                     <asp:Label ID="Label3" runat="server"
></asp:Label>
                  <br />
```

```
                     Alumni
                    <br />
                     Course :- <span><asp:Label ID="Label4"
runat="server" Text="Label"></asp:Label></span><br />
                     Batch   :- <span><asp:Label ID="Label5"
runat="server" Text="Label"></asp:Label></span><br />
                     Adm No  :- <span><asp:Label ID="Label6"
runat="server" Text="Label"></asp:Label></span>


                </td>
                <td class="style4" align="justify">
                     
                    <span style="color: rgb(26, 119, 1); font-family:
Arial, Helvetica, sans-serif; font-size: 12px; font-style: italic; font-
variant: normal; font-weight: bold; letter-spacing: normal; line-
height: normal; orphans: auto; text-align: start; text-indent: 0px; text-
transform: uppercase; white-space: normal; widows: 1; word-
spacing: 0px; -webkit-text-stroke-width: 0px; display: inline
!important; float: none; background-color: rgb(248, 250, 251);">
                    GENERAL</span><br />
                    <div  style="margin-left:-35px; width: 380px;">
                    <asp:DetailsView ID="DetailsView2"
runat="server" Height="50px" Width="380px"  ToolTip="Student
Profile"
                        AutoGenerateRows="False"
DataKeyNames="Admission_No"
                        DataSourceID="SqlDataSource3"
BorderColor="#CC9966" BorderStyle="None"
                        CellPadding="4" EmptyDataText="Sorry,No
Data" HorizontalAlign="Center"
                           Font-Bold="True" Font-Italic="True"
                           Font-Names="Arial" BackColor="White"
BorderWidth="1px">
                        <EditRowStyle BackColor="#FFCC66" Font-
Bold="True" ForeColor="#663399" />
```

```
                    <Fields>
                    <asp:BoundField DataField="Student_id"
HeaderText=" Student_id "  Visible="false"
                        SortExpression=" Student_id " />
                         <asp:BoundField DataField="Name"
HeaderText="Name"
                        SortExpression="Name " />
                         <asp:BoundField
DataField="Date_Of_Birth" HeaderText="Date Of Birth"
                        SortExpression="Date_Of_Birth" />
                         <asp:BoundField DataField="Gender"
HeaderText="Gender"
                        SortExpression="Gender" />

                         <asp:BoundField DataField=" Email "
                         <asp:BoundField DataField="password "
HeaderText=" password "

                        SortExpression=" password " />
                         <asp:BoundField DataField="Roll no "
HeaderText=" Roll no "
                        SortExpression=" Roll no " />
                         <asp:BoundField
DataField="Student_Category" HeaderText="Student Category"
                        SortExpression="Student_Category" />
                         <asp:BoundField DataField="Address1"
HeaderText="Address1"
                        SortExpression="Address1" />

                         <asp:BoundField DataField="City"
HeaderText="City"
                        SortExpression="City" />
                         <asp:BoundField DataField="State"
HeaderText="State"
                        SortExpression="State" />
```

```
                        <asp:BoundField DataField="Phone_No1"
HeaderText="Phone_No1
                        SortExpression="Phone_No1" />
                        <asp:BoundField DataField="Email_Id"
HeaderText="Email_ID"
                        SortExpression="Email_Id" />

                        <asp:CommandField ShowEditButton="True"
CausesValidation="false" />
                        </Fields>
                        <FooterStyle BackColor="#FFFFCC"
ForeColor="#330099" />
                        <HeaderStyle BackColor="#990000" Font-
Bold="True" ForeColor="#FFFFCC" />
                        <PagerStyle BackColor="#FFFFCC"
ForeColor="#330099" HorizontalAlign="Center" />
                        <RowStyle BackColor="White"
ForeColor="#330099" />
                    </asp:DetailsView>
                    </div>
                  <asp:SqlDataSource ID="SqlDataSource3"
runat="server"
                    ConnectionString="<%$
ConnectionStrings:StudentConnectionString %>"

                        SelectCommand="SELECT [student_id],
[Name],[Date_Of_Birth],[Gender], [Address],
[password],[Phone_No1],[Phone_No2],[Email_Id] FROM
[Student_Info] WHERE ([Admission_No] = @AdmissionNo)"
                        UpdateCommand="UPDATE Student_Info SET
name=@name, Gender= @Gender ,password= @password,,
Email=@Email WHERE student id=@student">id
                    <SelectParameters>
                        <asp:SessionParameter Name="student id "
SessionField="AdmissionNo"
                        Type="Int64" />
```

```
                    </SelectParameters>
                     <UpdateParameters>
                     <asp:Parameter Name="student" />

                        <asp:Parameter Name="Name" />
                        <asp:Parameter Name="birthday" />

                        <asp:Parameter Name="Gender" />
                        <asp:Parameter Name="Blood_Group" />
                        <asp:Parameter Name="address" />
                        <asp:Parameter Name="phone" />
                        <asp:Parameter Name="Email" />
                        <asp:Parameter Name="Password" />

                     </UpdateParameters>
                </asp:SqlDataSource>
            </td>



             </tr>
             <tr>
             <td class="style2">

                </td>
            <td width="380px">
            <br />
            <span
                    style="color: rgb(26, 119, 1); font-family: Arial,
Helvetica, sans-serif; font-size: 12px; font-style: normal; font-
variant: normal; font-weight: bold; letter-spacing: normal; line-
height: normal; orphans: auto; text-align: start; text-indent: 0px; text-
transform: uppercase; white-space: normal; widows: 1; word-
spacing: 0px; -webkit-text-stroke-width: 0px; display: inline
```