**School Management System**



Dha Lhamu Lama

00172890

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT and E-Commerce

Kathmandu, Nepal

1. Introduction

The system is made with the use of PHP language with its framework called Laravel. The backend of the system is used Laravel whereas the frontend is developed by the use of JavaScript with its framework css html. For the database I have used Mysqli. The Laravel is used in this software since it is very popular and develops the website very attractive and useful with its tools. The school management system is a software that handles the database of school where all the activities of the school are included. This software is made mainly for school management system with the purpose of providing a nice website where administration, student, teacher, parents can easily access the information and find their details which makes very easy for the school instead of not having website.

2. Problem Statement

School was facing the problem in recording the attendances and the report in salary and fees and there was not clear idea of how many students are enrolled every year. To have accurate data on students, teachers and clear account of fees and salary, school management system is done so that the management can be out of problem.

3. Background of the project

Our company is running with good prestige and has served numerous companies and schools with many projects. The project is done since many schools are not implementing the latest technical methods of using the system. Since schools are having difficulty in managing the schools management. The project is introduced and it has to be done to overcome the problem faced by the schools and we can create the project by understanding and researching the requirements of the schools and after the project is ready, we have to understand whether the project is user friendly and good to use to the schools or not.

4. Aims and Objectives

**Aim:** The main aim is to meet the requirements of the users for their benefit as well as for the school management system. The required analysis of data of students, teachers, fees, salary etc. can be possible by making software in the computer where one can access easily and get the required answers wanted by them

**Objectives:** The main objectives of developing this software is

(a) Record Keeping

* It maintains the data of all the student and teacher in database system.
* It maintains the personal record as well as the academic record of the student.
* It maintains student fee record and dues record.
* Teacher personal record and salary record are also store in the software.
* It also manages class record, subject record, examination record.

(b) Reports Generating

Reports are the essential outcomes of school management system (SMS).Reports includes the given points

* Total number of student in the class
* Student admission reports
* Character certificate of the student
* Submitted due reports
* Teacher Experience reports
* Examination report
* Detail Marks Certificate

5. Features of the project

Following are the main features of our project

* Admin work: the admin can approve the request from teacher, student parent and make them access to the website to view their information
* Crud function: admin can update the information of the teacher student parent on their request.
* Thorough Attendance: It keeps the full record of the attendance of the students as well as the staff or teachers of the school.
* Maintaining Examination: The main feature of this software is also maintaining the exams of the students that keeps full record of the student record.
* Communication: It gives notice to the users, which includes the staff and the students of the institute about fees dues, holiday declaration, exam and the ongoing activities. This serves just like a notice board.

.

6. WBS

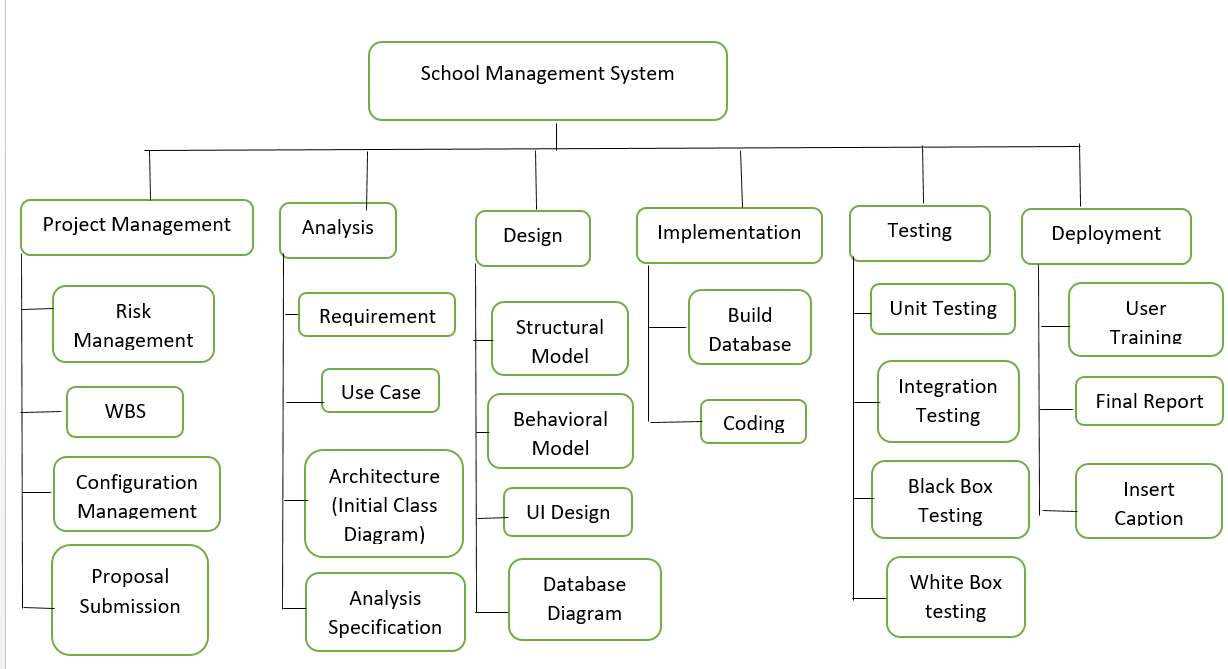
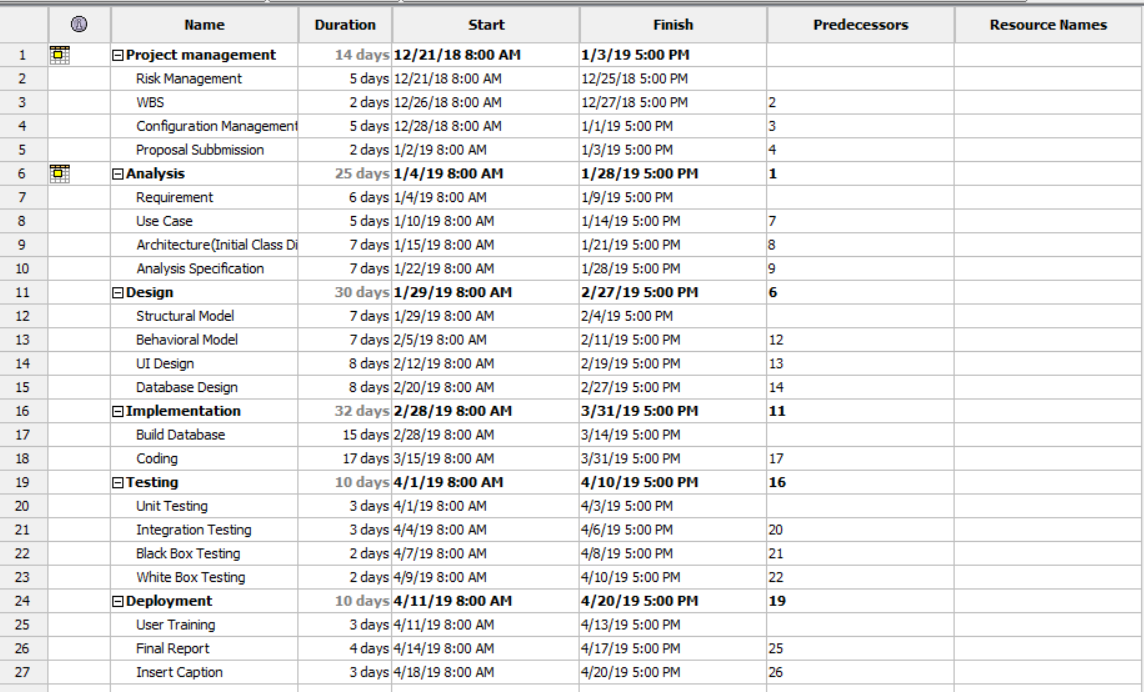
Work breakdown structure is a project management and systems where works are divided into smaller projects. It is also known as tree structure where work is divided into smaller sub projects that makes easier in completing task. I have shown the work breakdown of school management system which defines all the projects into smaller components. 

Figure 4: Work Breakdown Structure

7. Gantt chart

Schedulling is the process of managing the task by allocating the suitable time. It helps the project to complete in time and we can say that it is just like planning.

****

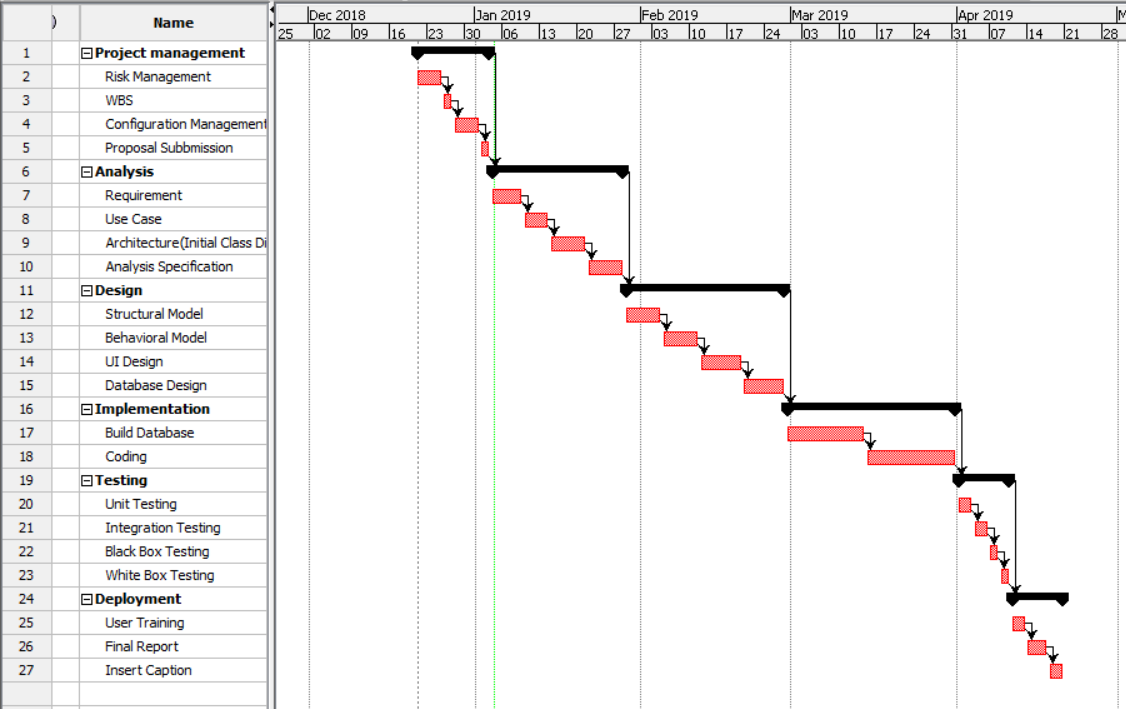
****

Figure 5: Gantt chart

8. Methodology with justification and why

Here, I have chosen Waterfall Method for the development of this software. Since it is traditional System Development Life Cycle (SDLC), this method involves a complete set of steps that a team follows. The main idea is to divide the development process into a series of phases or stages, each of which finishes before next one starts. Below is shown the diagram of waterfall module which has been used in this school management system.

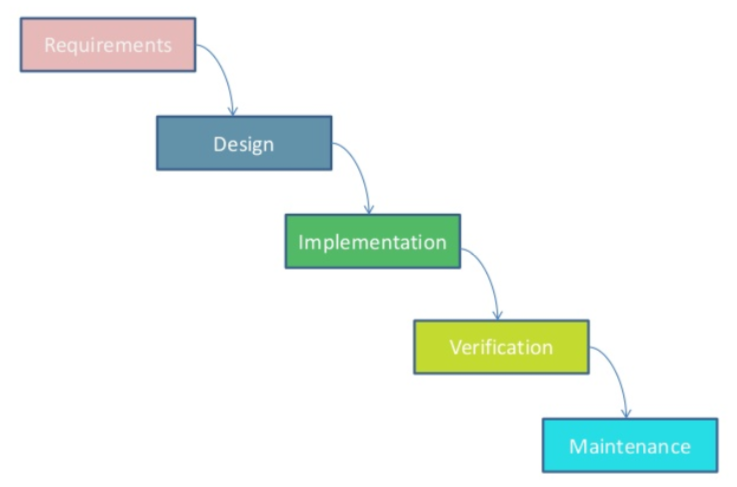


Figure 1: Waterfall Module

**Requirements:** all requirements are gathered and analyzed in this phase and documented for next phase

**Design:** After gathering the requirements, it is studied and design process takes place

**Implementation:** After designing, project start to be developed in small programs (units) and are integrated for next phase.

**Verification**: After integration process, verification of the project is needed which is done to test the faults and failures.

**Maintenance**: Once test is done, if there is any faults and failures, then comes the step of maintenance which is important

9. Architecture with diagram

System architecture is defined as conceptual model that includes the structure, behavior, and more views of a system. I am going to use 3-tier system architecture. For example internet banking system, and the diagram is given below.

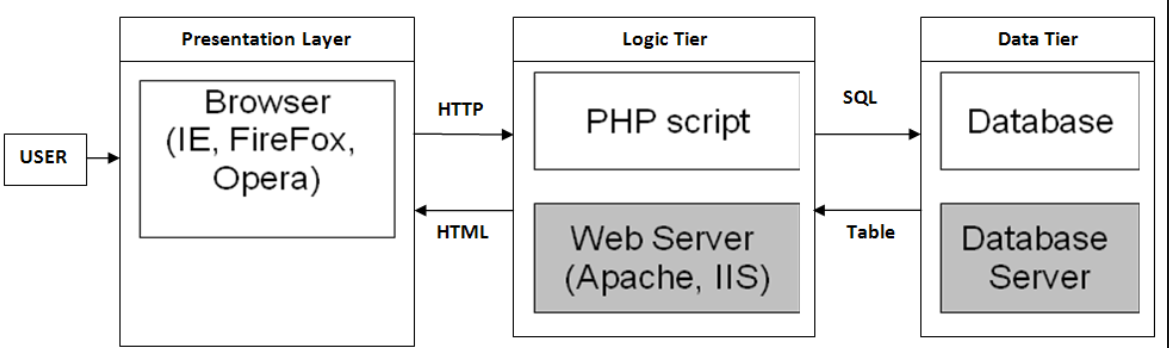


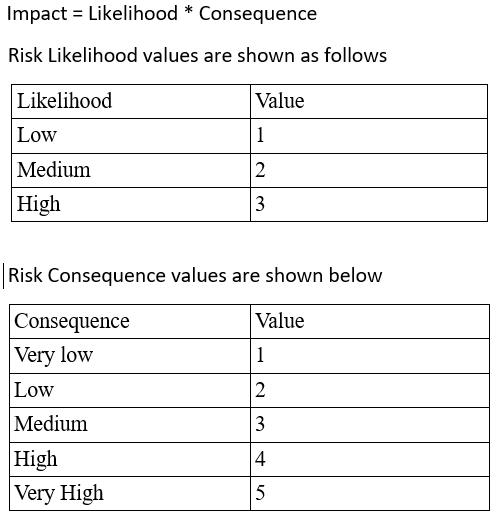
Figure 3:3-tier system architecture

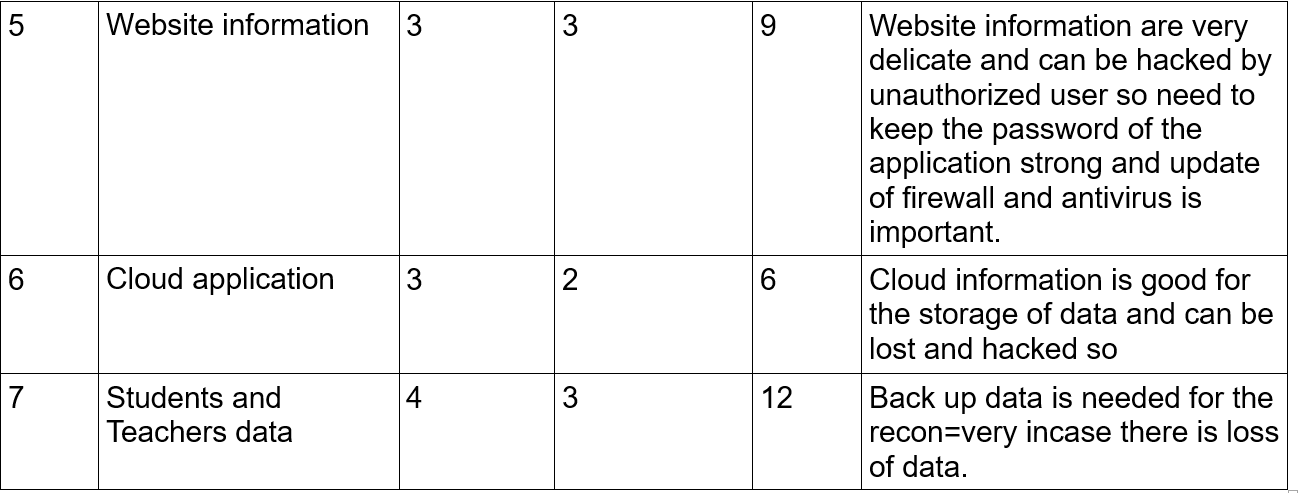
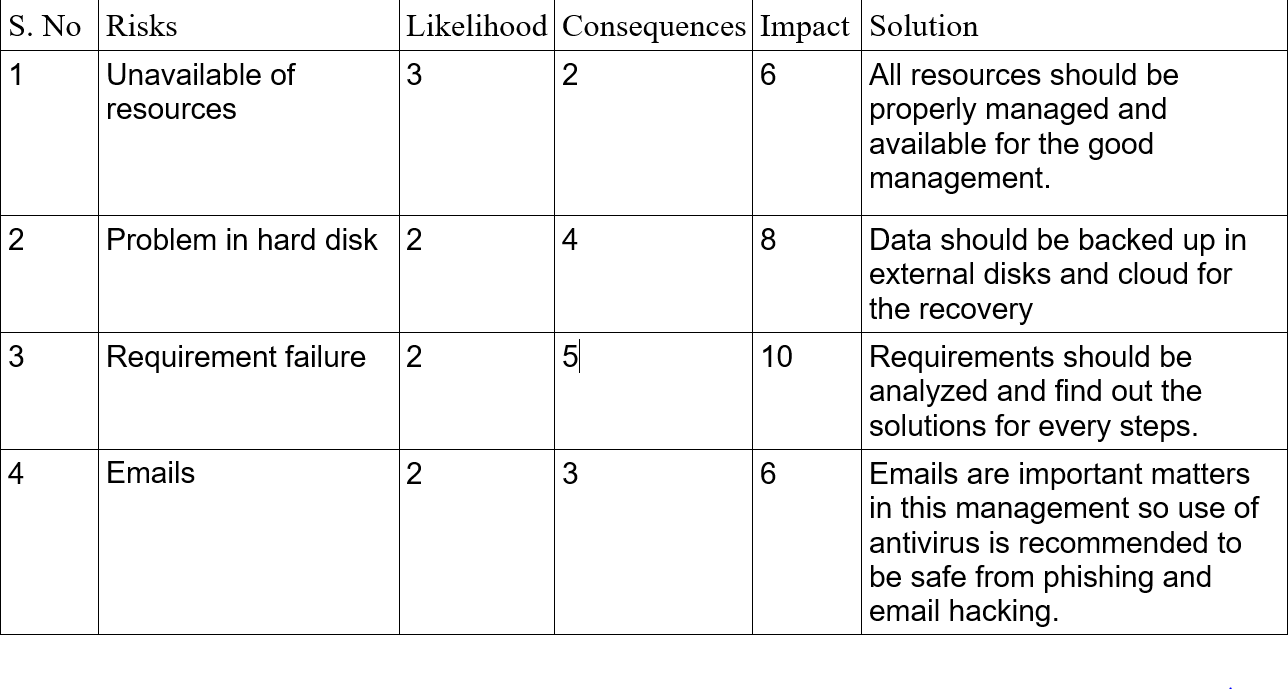
Presentation Layer: This tier communicates with other tiers by sending results to the browser and other tiers in the network.

Logic Tier: It is also called as middle tier, presentation tier, business logic or logic tier, this tier is pulled from the presentation tier. It does control application functionality by performing detailed processing.

Data Tier: Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

10. Risk and configuration management





11. Design (initial class diagram, use case with justification)

I am using here MVC design pattern in this project. MVC consists of data model, presentation information and control information and I have chosen PHP Laravel framework which is free and open source software that is widely used and famous.

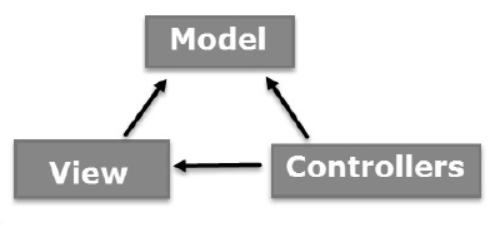


Figure 2: Design pattern for school management system

Model: It corresponded related logic data where user works on it. For example, a Customer retrieves customer information from the database, manipulates it and updates its data back to the database or uses it to render data.

View: It is used in all UI logic of the application. For example, the Customer view includes all UI components like text box, dropdown, etc. in which final user will be using it.

Controller: It plays role of interface between Model and View components which process all business logic, requested data incoming request and manipulate and interact with the Views to render the final output. For example, the Customer Controller handles all interactions and inputs from customer view and updates the database using customer model.

Use case

Definition: A use case diagram is a graphic depiction of the interactions among the elements of a system.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Website. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

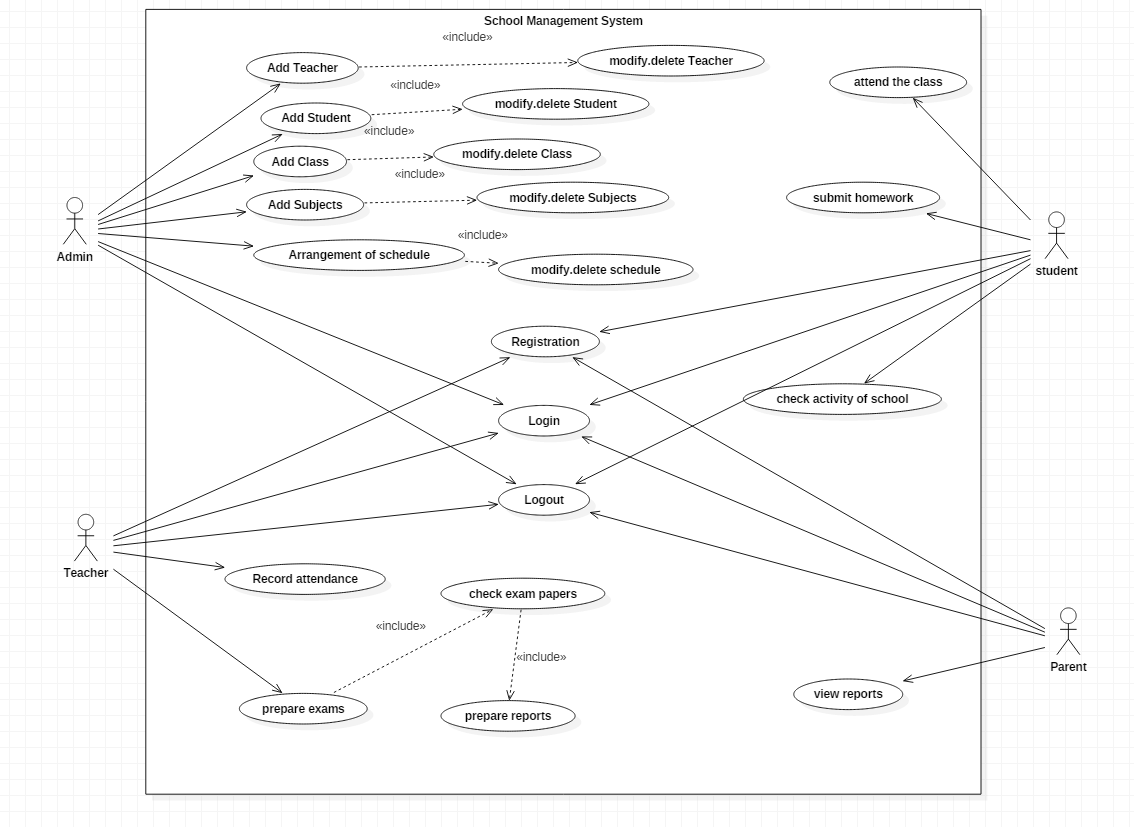


Fig: Use cases for school management system

Justification: A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

Initial Class Diagram:

Definition: class diagram to show the classes and relationships. A class diagram shows classes, the relationships between classes, constraints, and attributes of classes. The diagram is helpful in designing and building a system because it can be used as the blueprint for the final product.

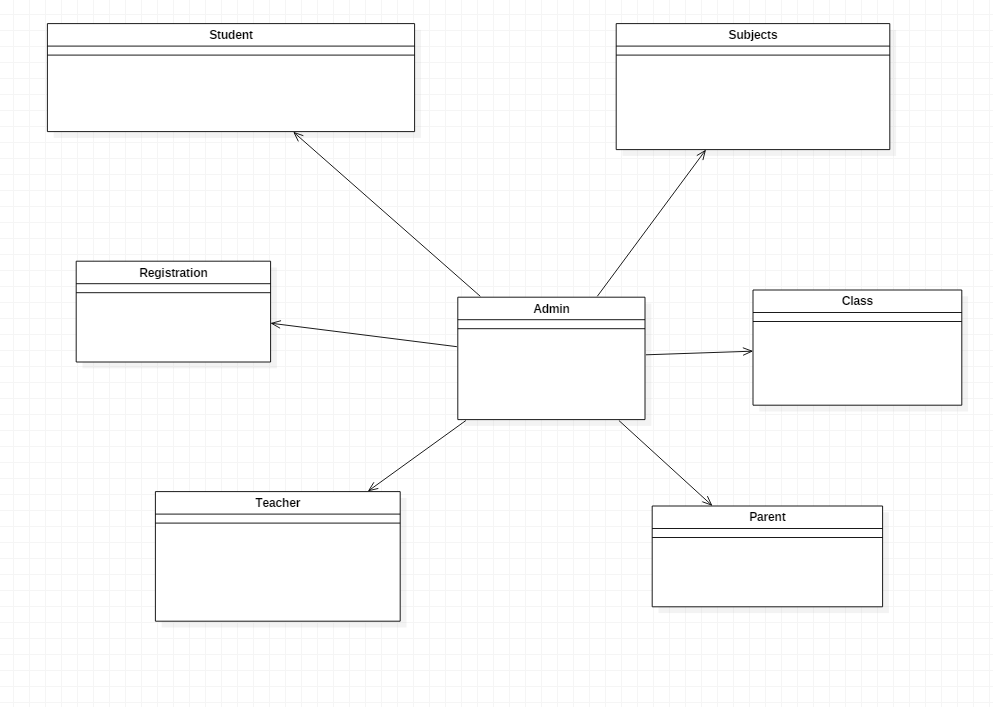


Fig: initial diagram

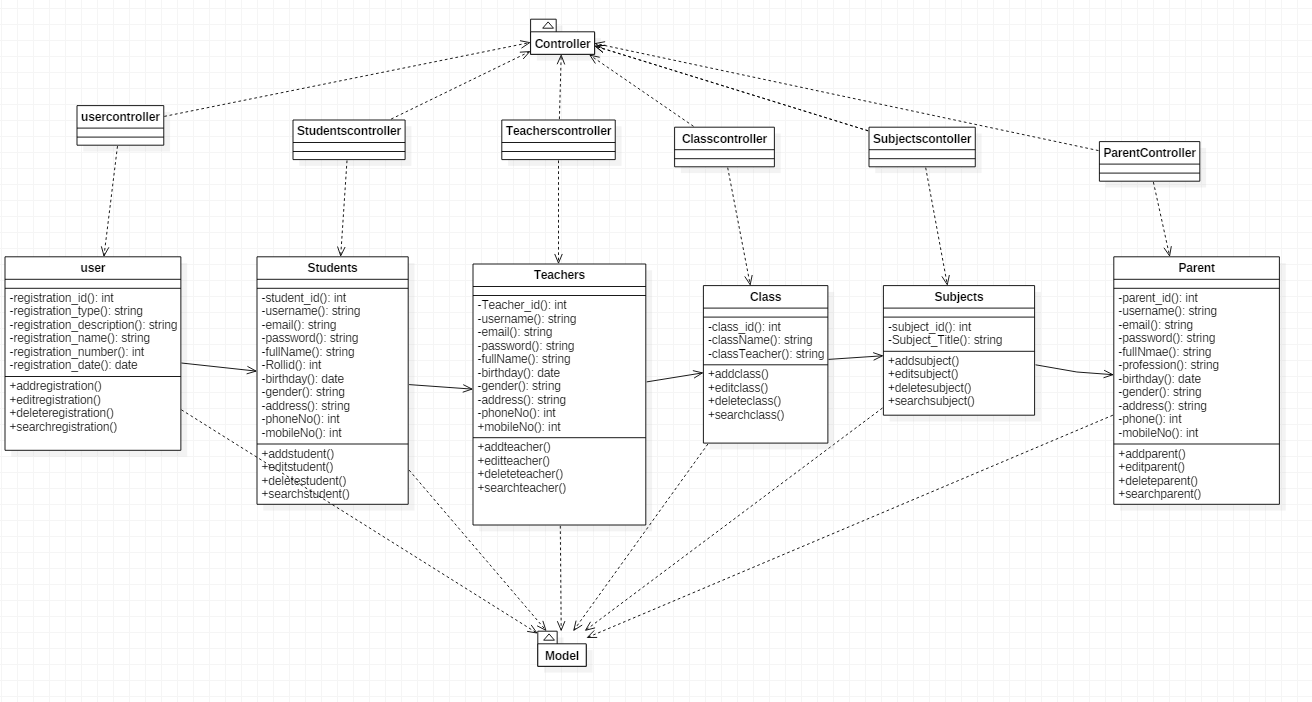


Fig: Final class diagram

Justification:

Final Class Diagram: Justification: A class diagram is an illustration of the relationships and [source code](https://searchmicroservices.techtarget.com/definition/source-code) dependencies among classes in the Unified Modeling Language (UML). In this context, a [class](https://whatis.techtarget.com/definition/class) defines the [method](https://whatis.techtarget.com/definition/method)s and [variable](https://whatis.techtarget.com/definition/variable)s in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

ER diagram for School Management System

Definition: An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

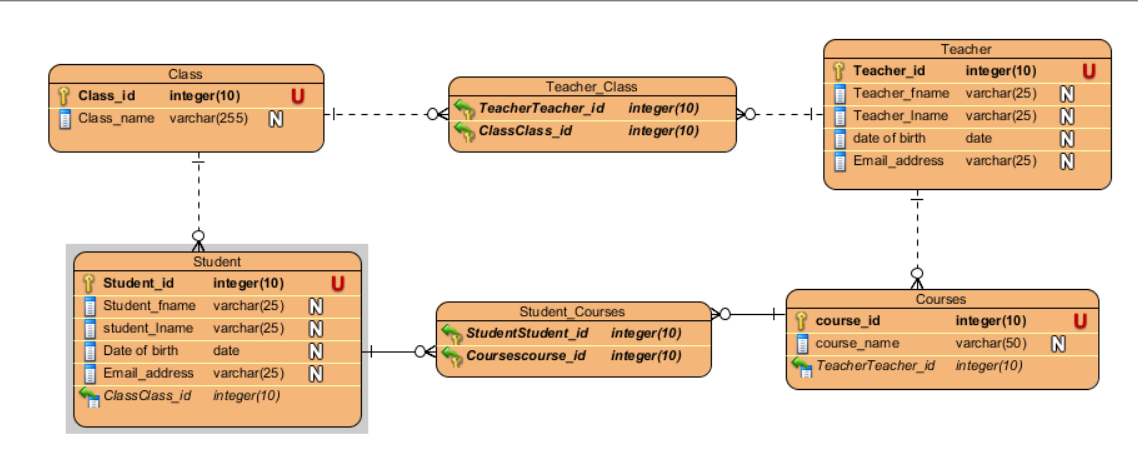


Fig: ER Diagram

Justification: ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong. Business information systems: The diagrams are used to design or analyze relational databases used in business processes.

**Class diagram**

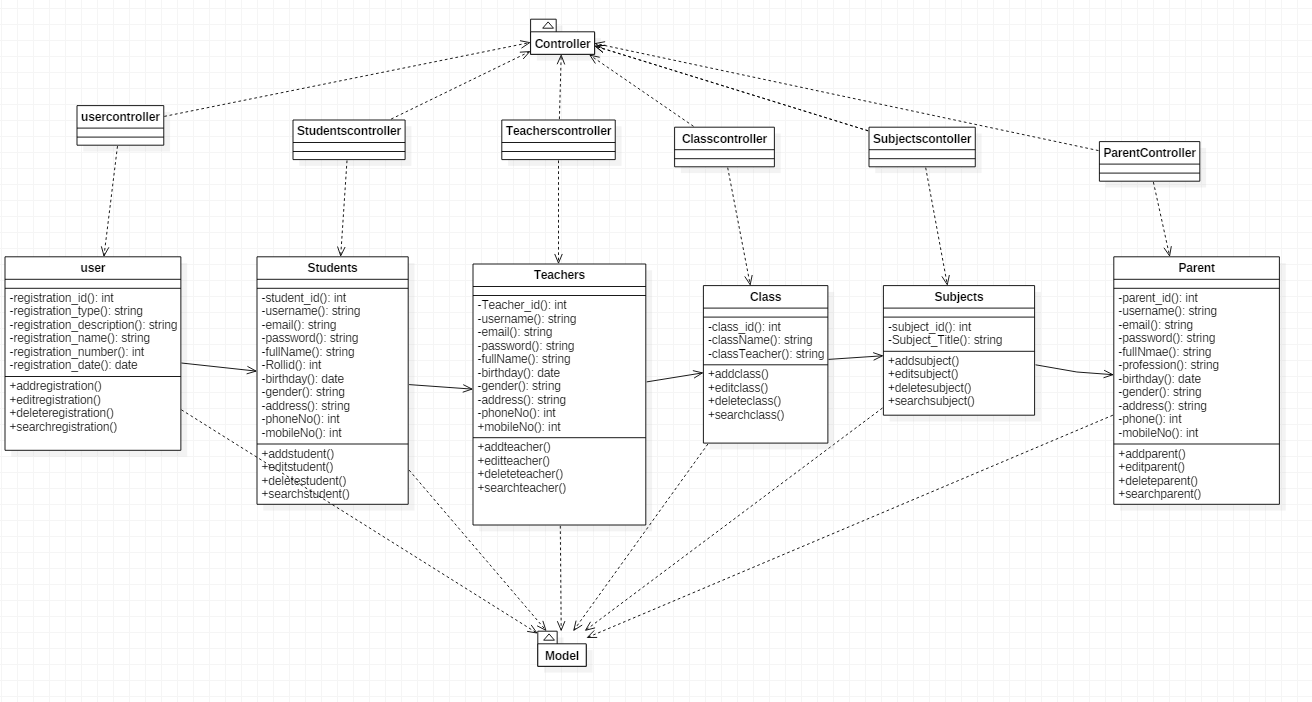


Fig: Final class diagram

Justification:

Final Class Diagram: Justification: A class diagram is an illustration of the relationships and [source code](https://searchmicroservices.techtarget.com/definition/source-code) dependencies among classes in the Unified Modeling Language (UML). In this context, a [class](https://whatis.techtarget.com/definition/class) defines the [method](https://whatis.techtarget.com/definition/method)s and [variable](https://whatis.techtarget.com/definition/variable)s in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

Activity Diagram

Definition

Activity diagram**:** An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

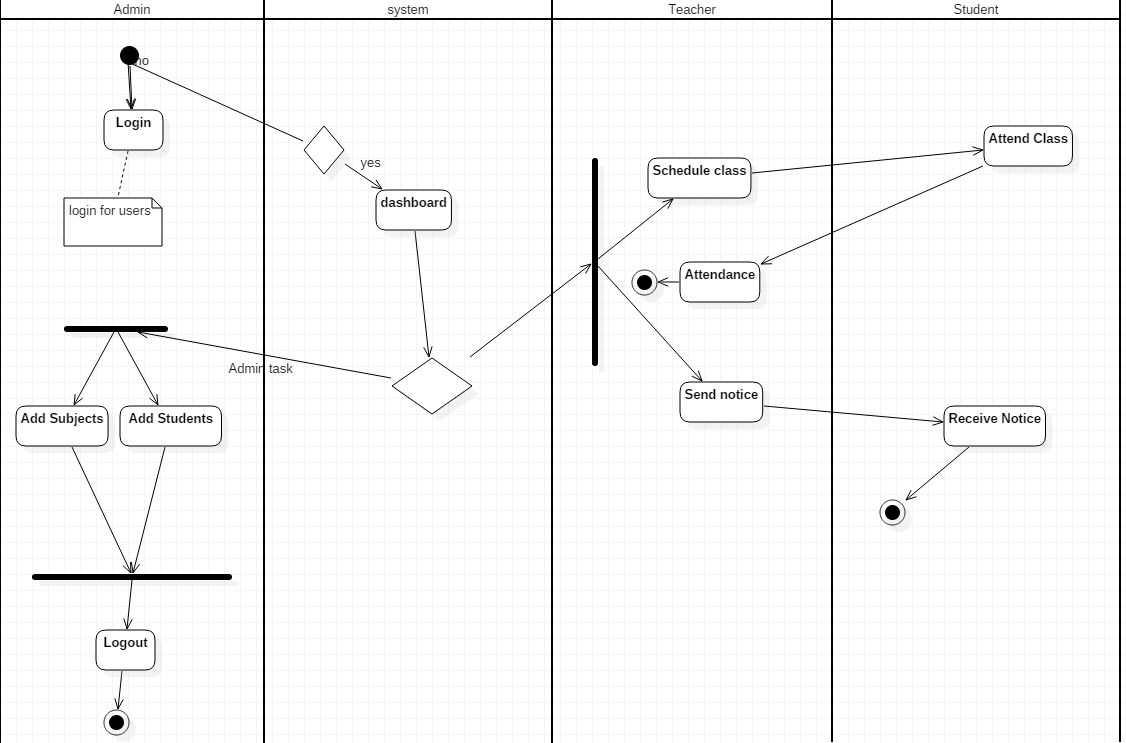


Fig: Diagram of activity diagram

Justification**:** Purpose of Activity Diagrams. The basic purposes of activity diagramsis similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another

Sequence diagram: A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequencediagrams describe how and in what order the objects in a system function.

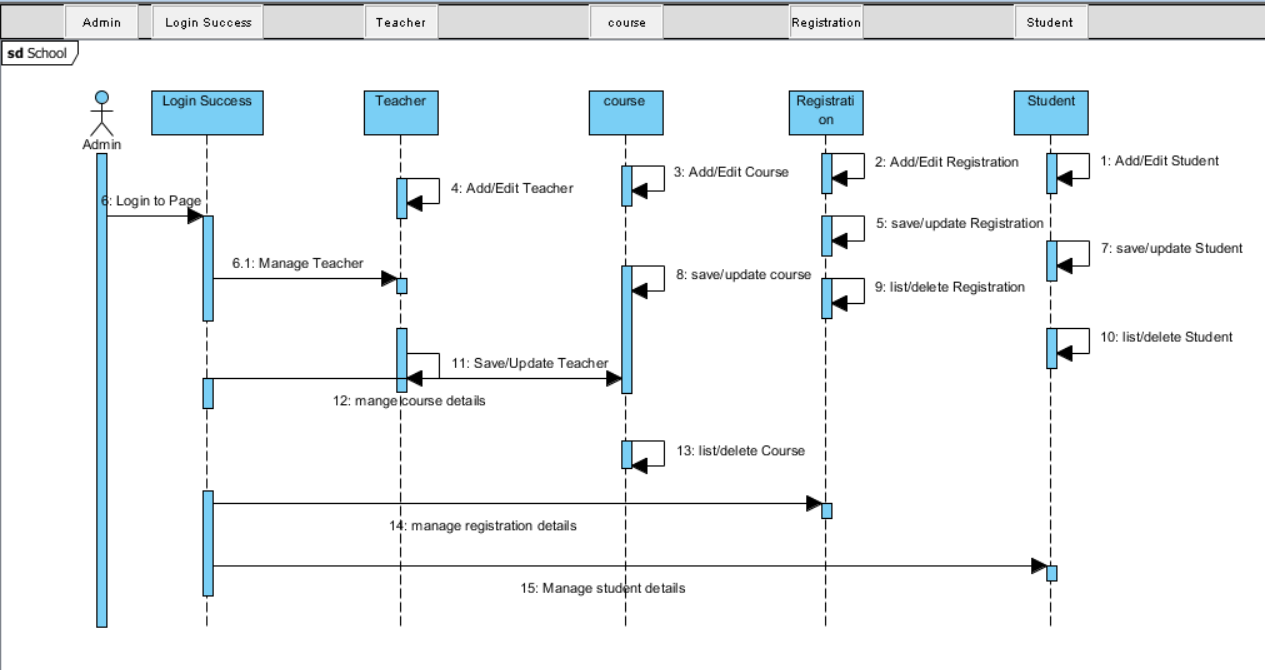
****

Fig: sequence diagram

Justification: A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called event diagrams or event scenarios.

12. Coding and UI design (minimum 3)

Login form with code

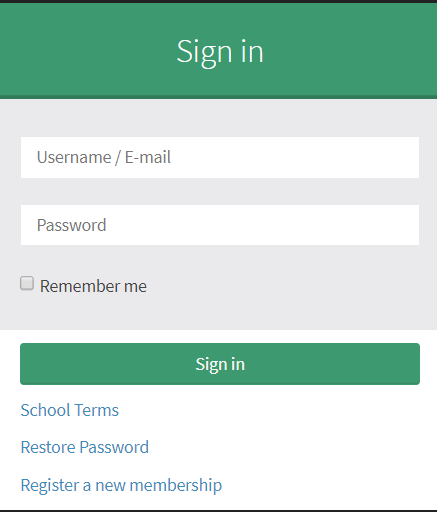
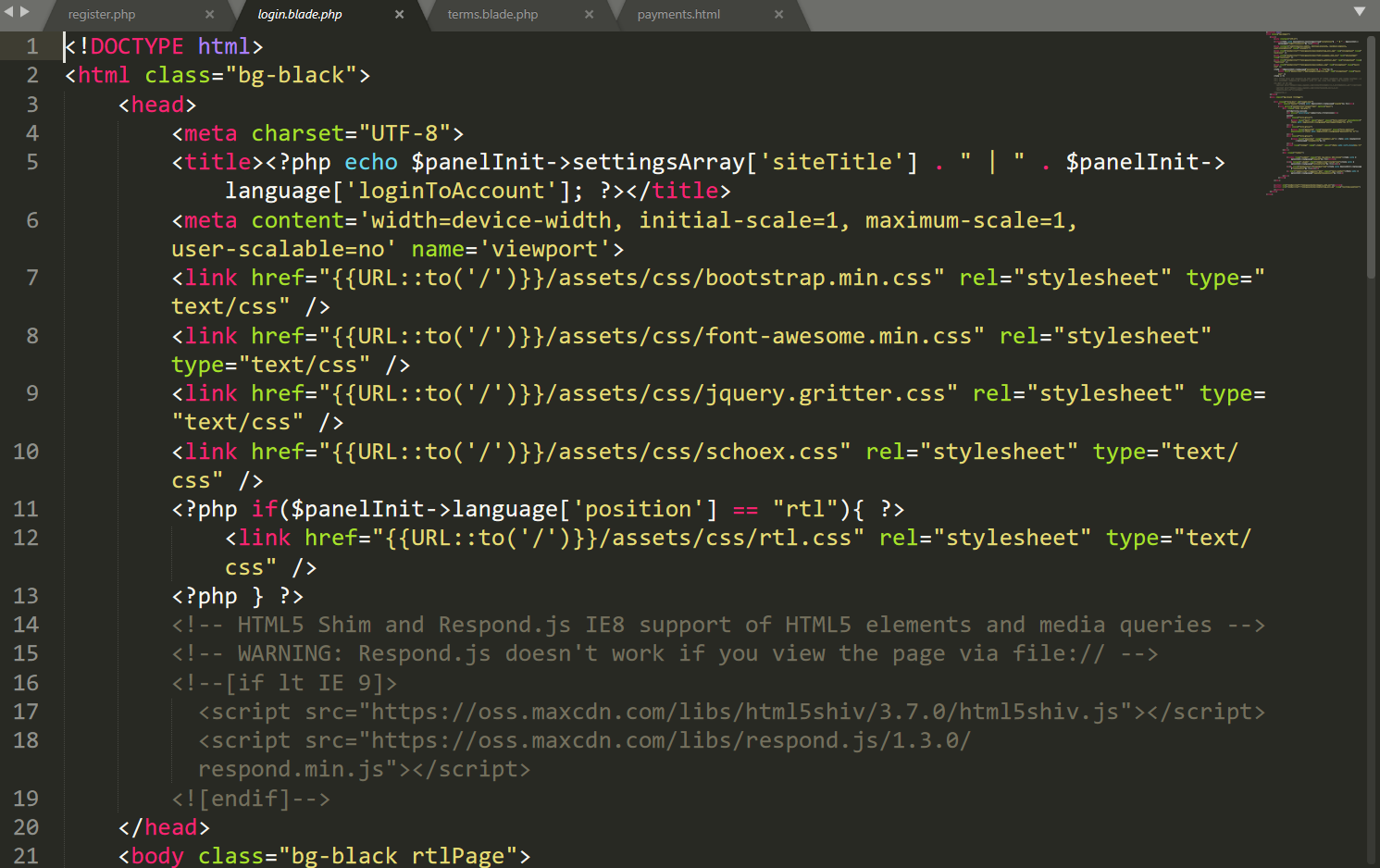


Fig: UI form for login



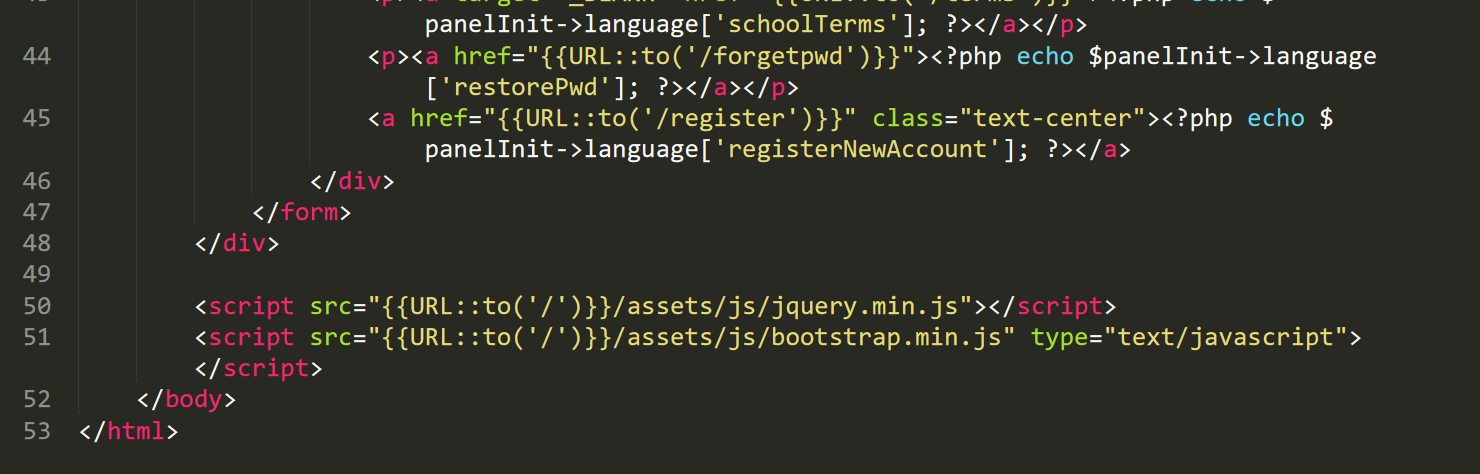
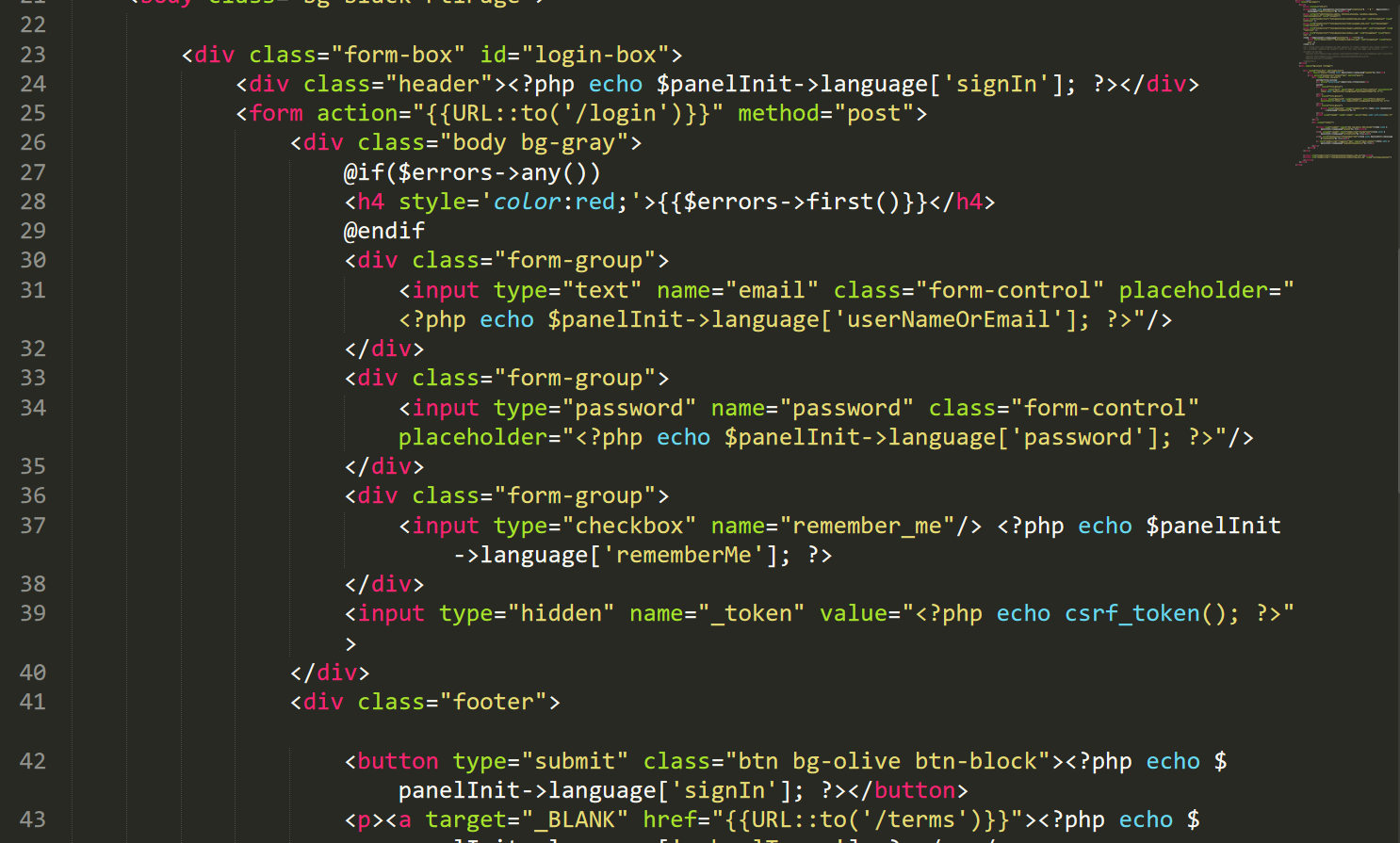


Fig: Code for login form

Registration form

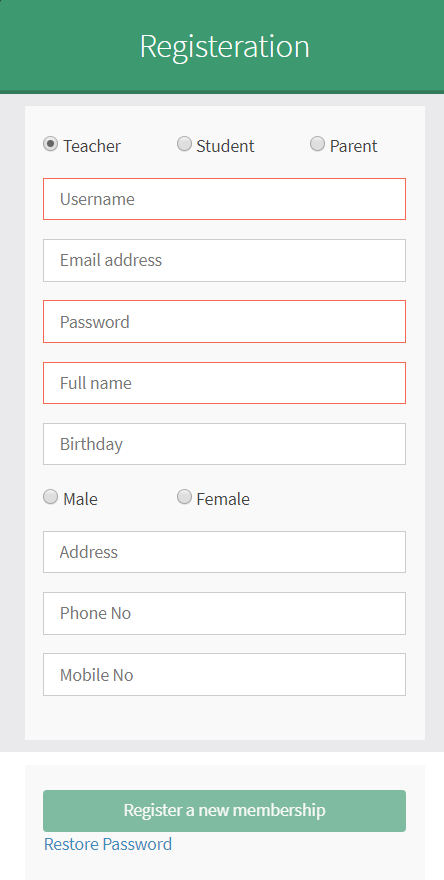
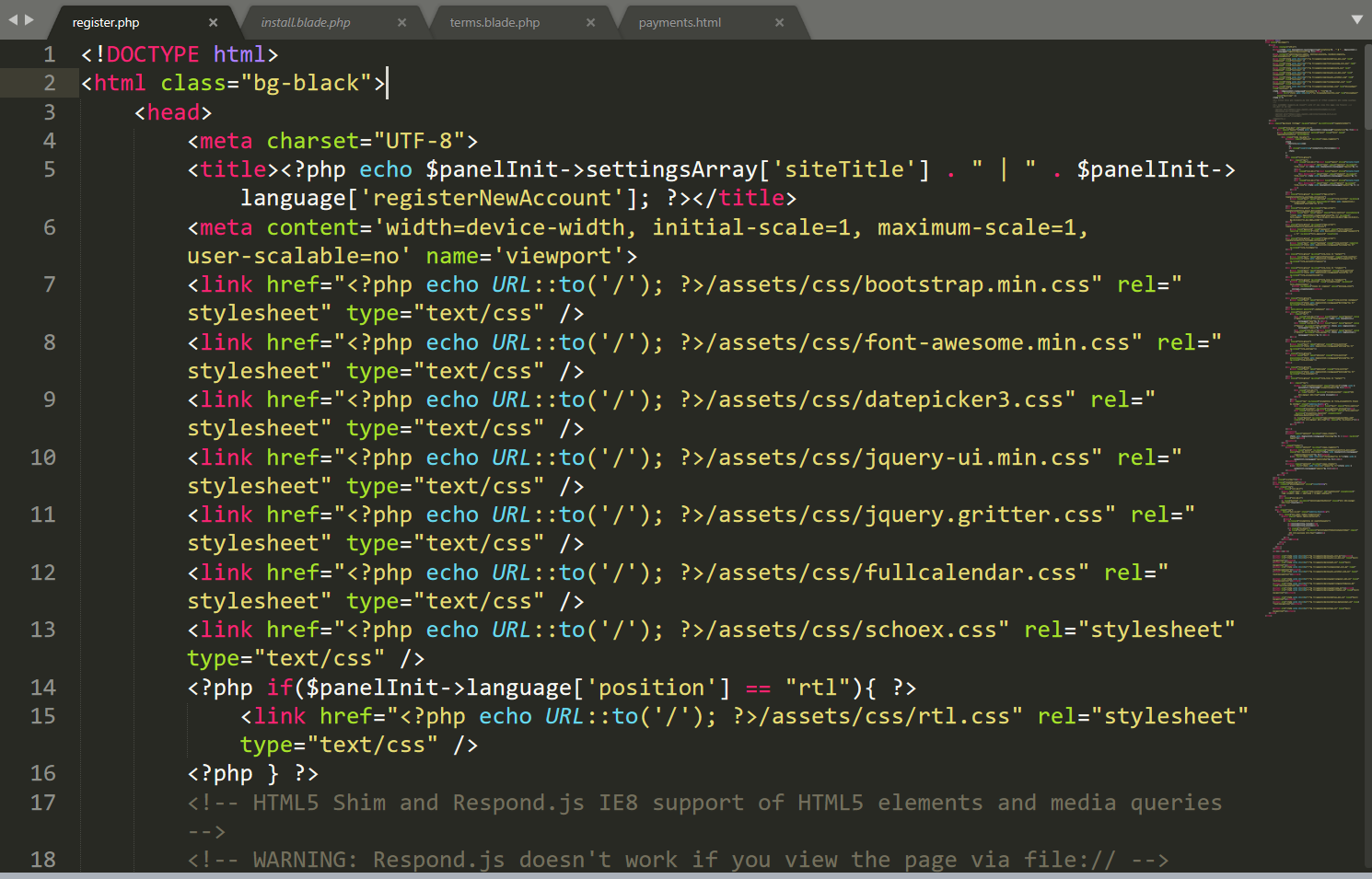
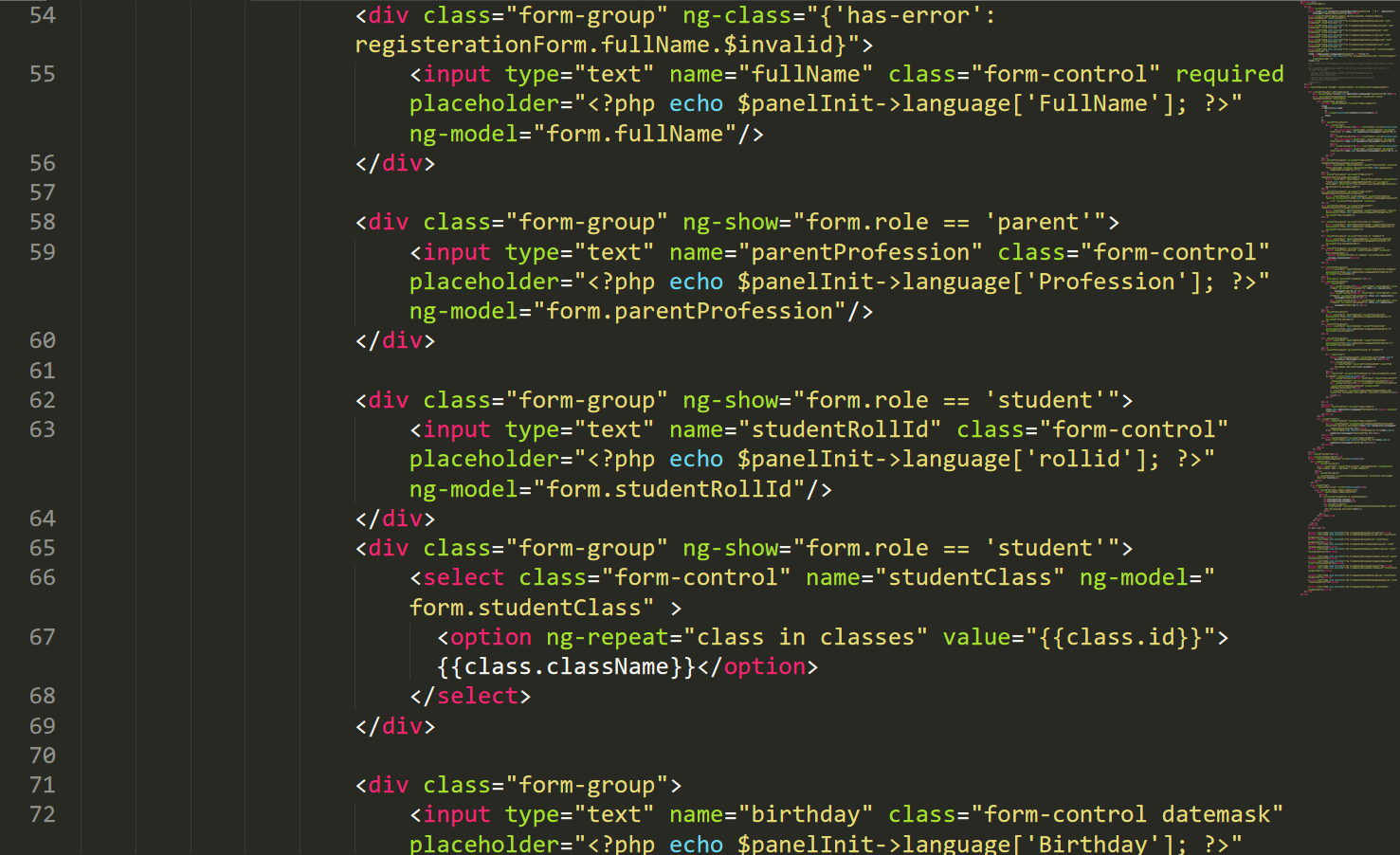


Fig: UI design for Registration form









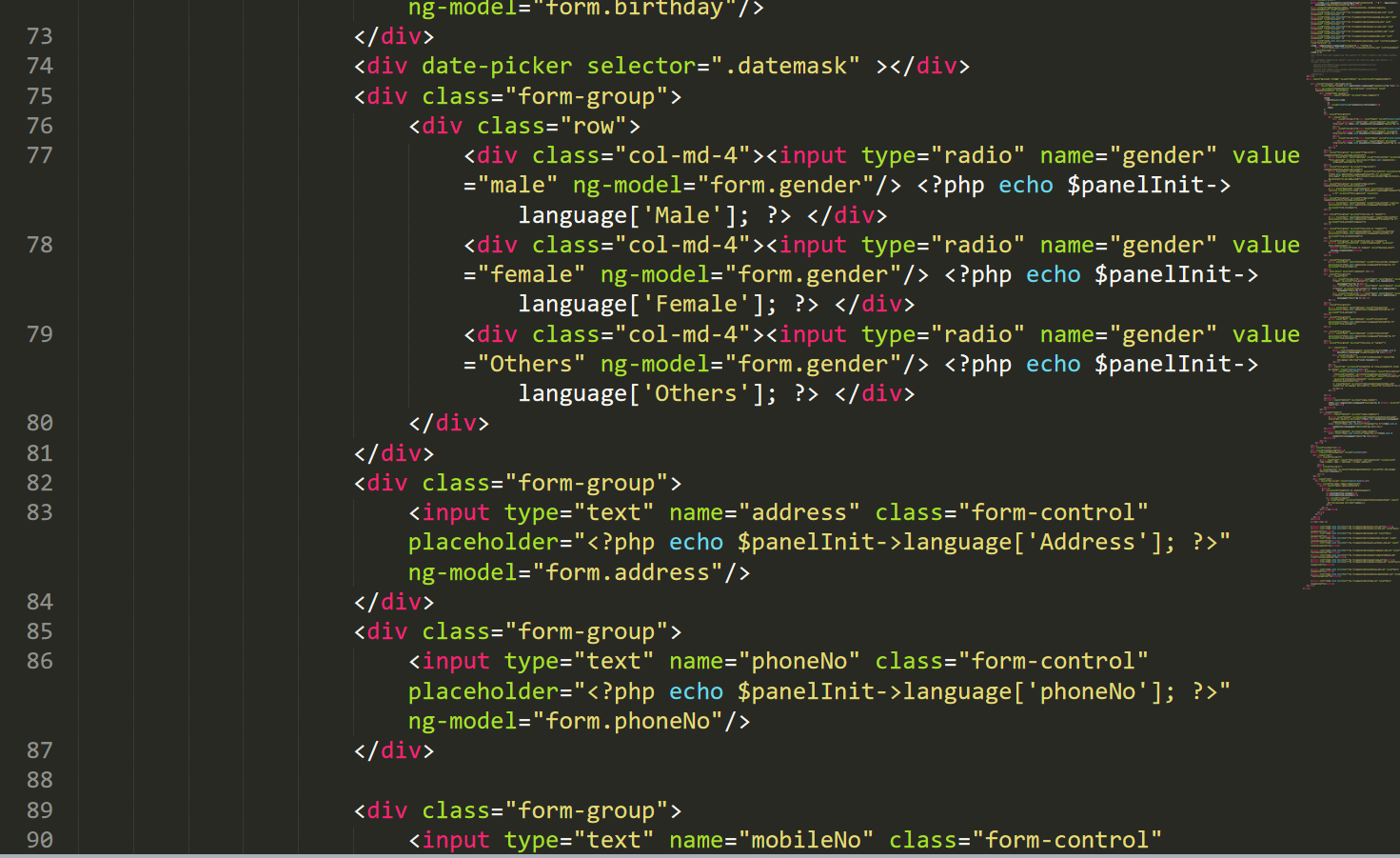




Fig: Code for registration form

Layout form



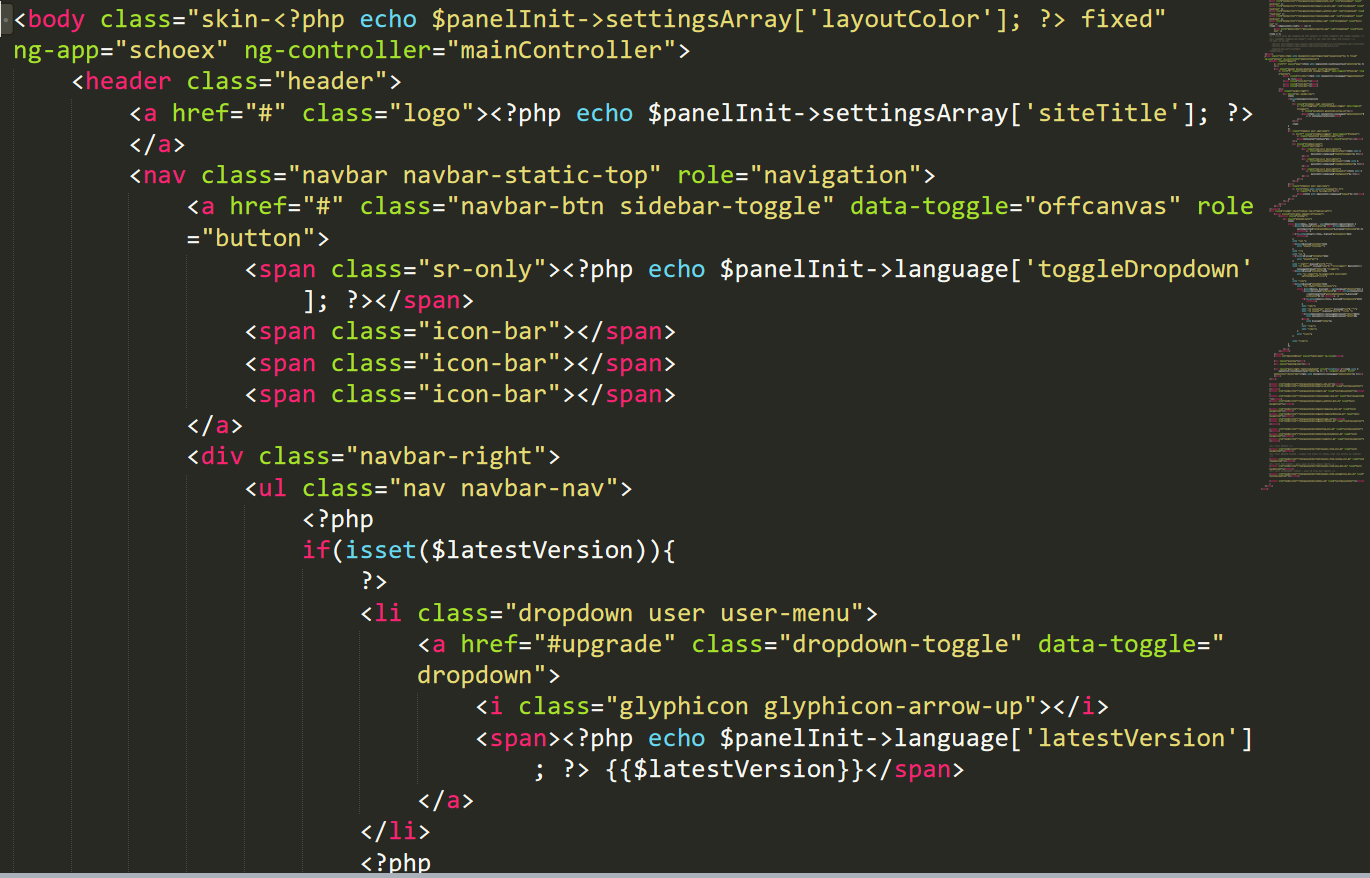


Fig: code for layout

13. Test cases (at least 10 test cases)

Test case

I have tested done two types of testing which is called white box testing and black box testing. The test I have done is Black box testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test\_id | Description | Test value | Expected result | Actual Result | Pass or Fail |
| Test\_01 | Testing: login done by admin | Emai\_id=admin Password=admin | Should be logged in | Successful login page and opens dashboard | pass |
| Test\_02 | Testing: login done by admin | Emai\_id=admin Password=asdfgg | Should not login | Failed in login with message"please check your username and password" | fail |
| Test\_03 | Testing: Registration by Teacher | [Emai\_id=babu@gmail.com](mailto:Emai_id=babu@gmail.com)  Password=babu123 | Should be registered | Successful registration and opens login page | pass |
| Test\_04 | Testing: login done by Teacher | [Emai\_id=babu@gmail.com](mailto:Emai_id=babu@gmail.com)  Password=babu123 | Should be logged in | Successful login with dashboard | pass |
| Test\_05 | Testing  Login using wrong email\_id or password by Teacher | [Email\_id=babuu@gmail.com](mailto:Email_id=babuu@gmail.com)  Password=hsdvhsab | Should not login | Shows message="Please check your username and password " | fail |
| Test\_06 | Testing  Update email\_id of teacher on request of teacher by admin | old  [email\_id=babu@yahoo.com](mailto:email_id=babu@yahoo.com)  new  [Email\_id=babu@gmail.com](mailto:Email_id=babu@gmail.com) | Should update | Successful update of teacher email\_id | pass |
| Test\_07 | Testing  Registration by student | [email\_id=maya@gmail.com](mailto:email_id=maya@gmail.com)  username=maya123  password=maya123 | Registered should be done | Successful registration with login page | pass |
| Test\_08 | Student login test | [email\_id=maya@gmail.com](mailto:email_id=maya@gmail.com)  username=maya123  password=maya123 | Should show dashboard with successful login | Successful logged in with student dashboard | pass |
| Test\_09 | Student trying to login with wrong username | [email\_id=maya@gmail.com](mailto:email_id=maya@gmail.com)  username=maya12345  password=maya123 | Should not login | Message shown"Please check your username and password" | Fail |
| Test\_010 | Deleting maya@gmail from student list | Before there is [maya@gmail.com](mailto:maya@gmail.com) | Should delete from student list | Message box"Are you sure you want to delete" and deletes successfully | Pass |
|  |  |  |  |  |  |

Test\_01 screenshots

Admin login with correct password

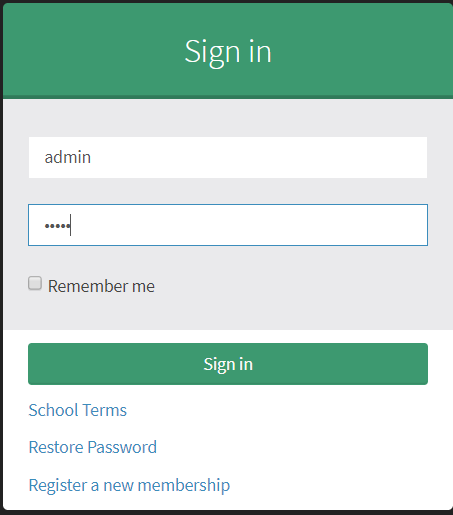


Fig:1

Dashboard

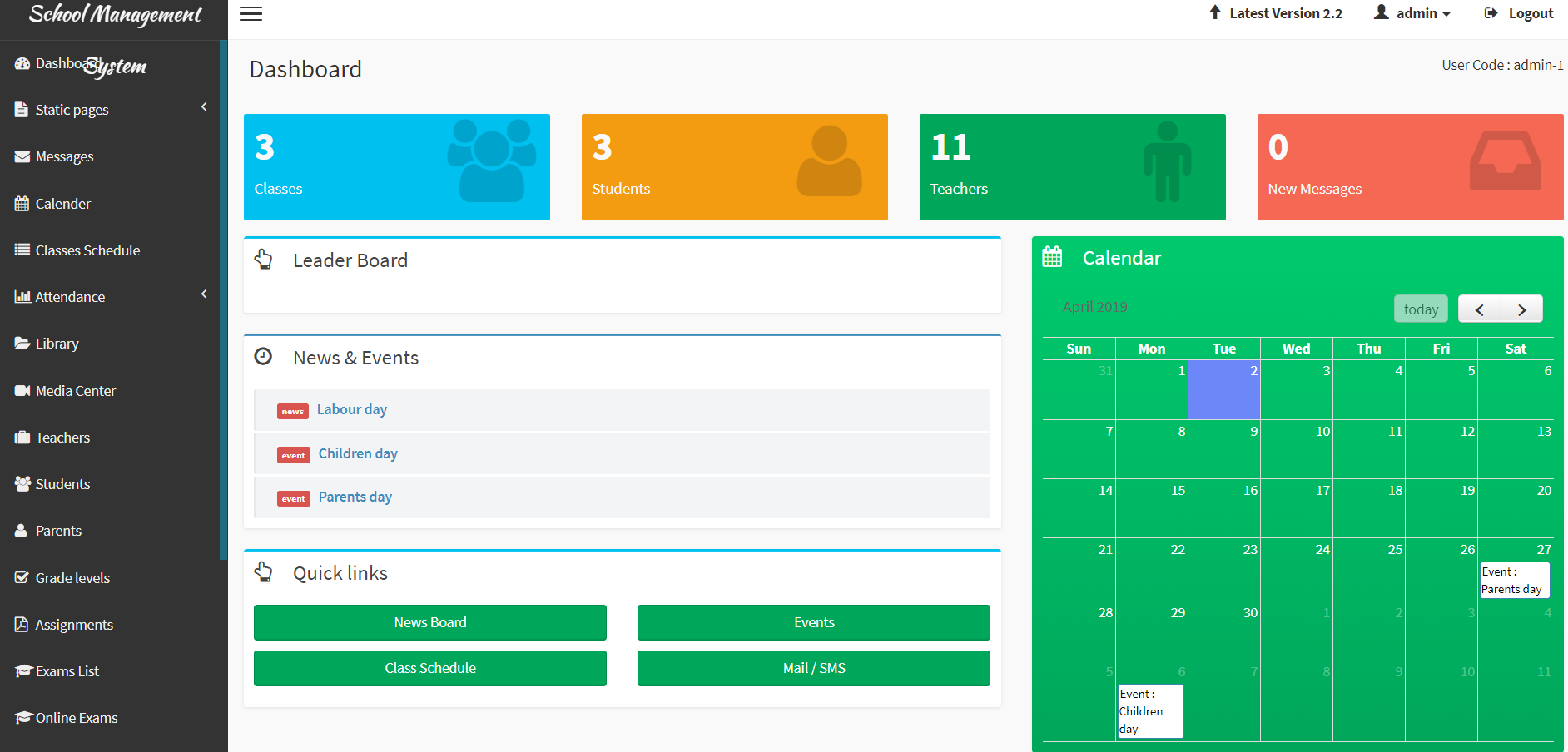


Fig:2

Test\_2 screenshots

Admin trying to login with wrong password

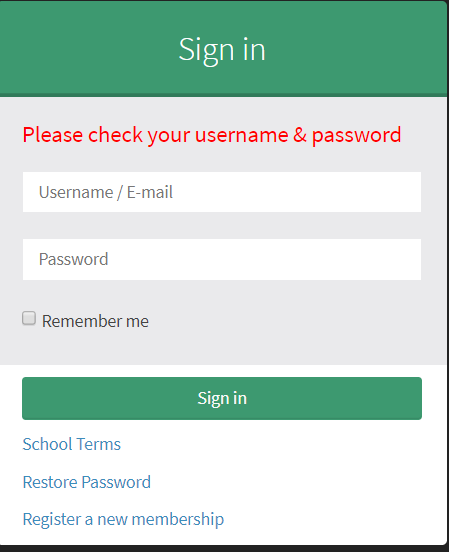


Fig: 3

Test\_03 Registration by Teacher

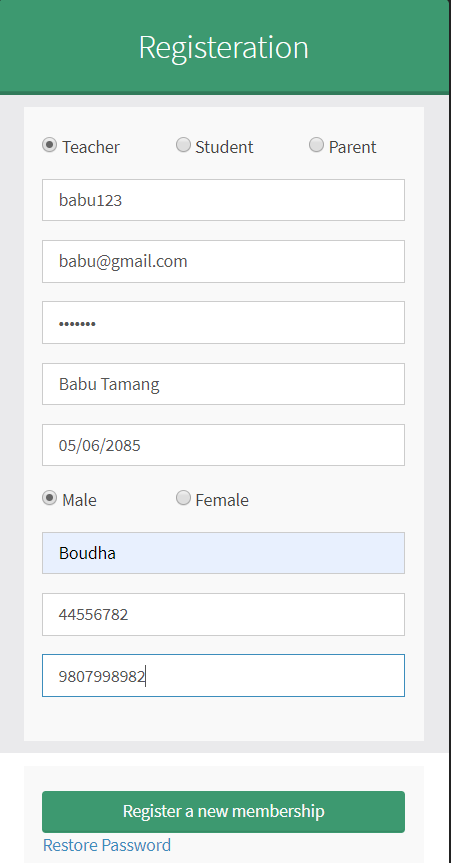


Fig:4

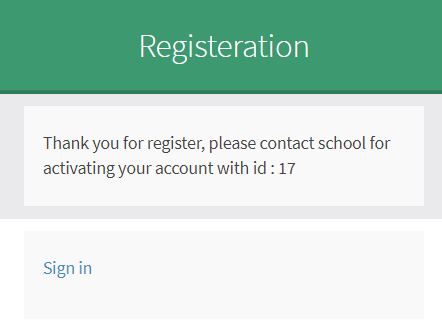


Fig: 5

Test\_04

Login by Teacher with correct password

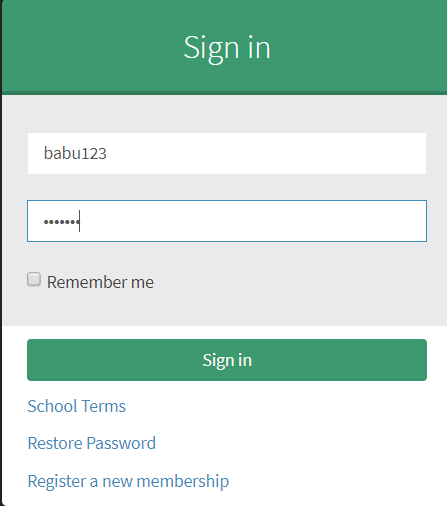


Fig: 6

Dashboard

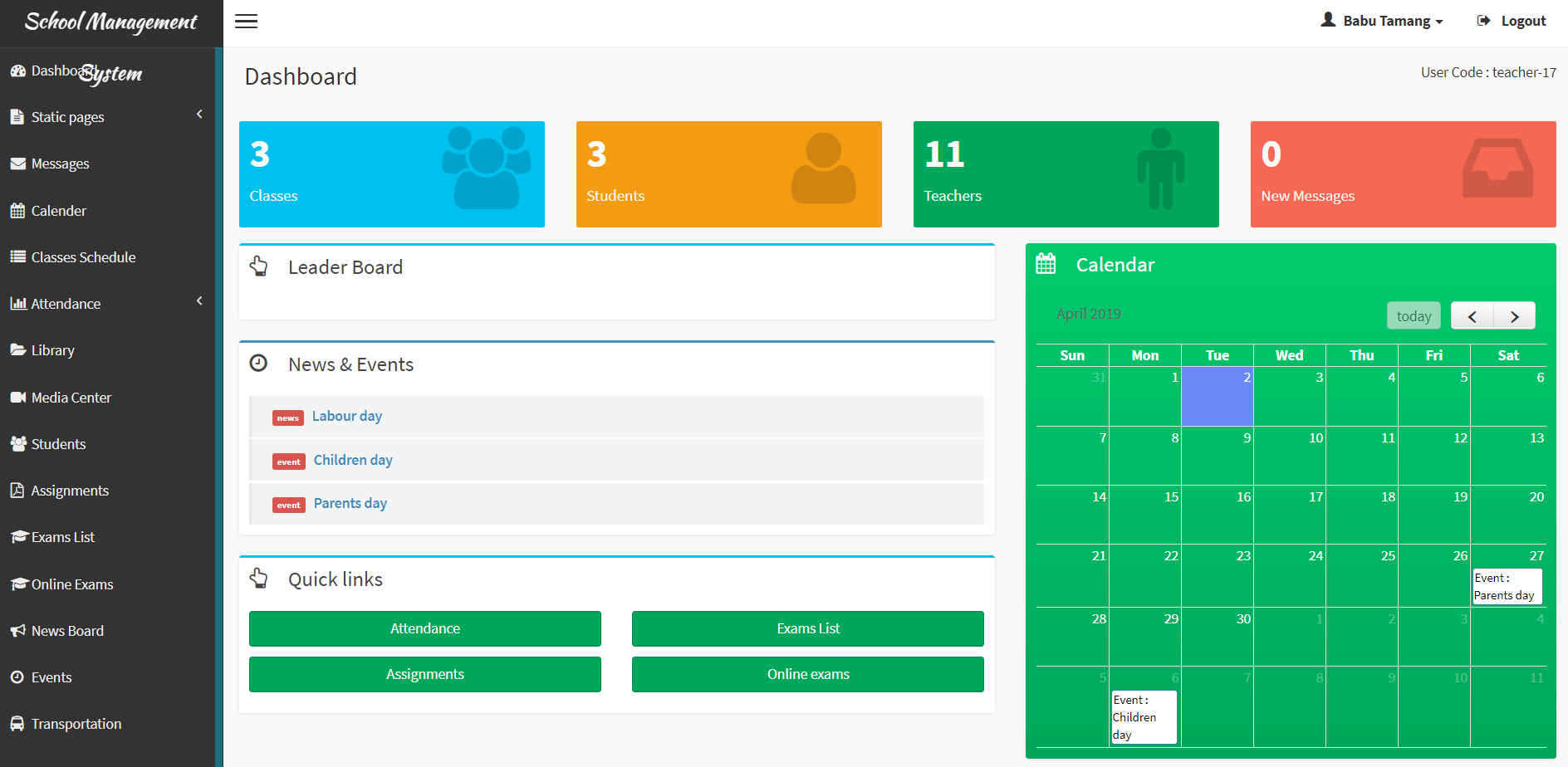
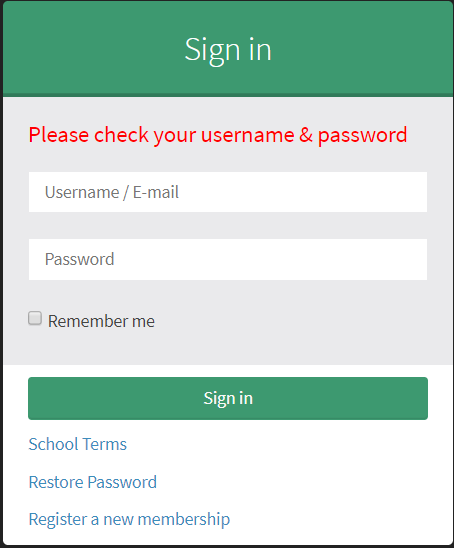


Fig: 7

Test\_05

Teacher login with wrong username and password



Test\_06

Update email of teacher by admin

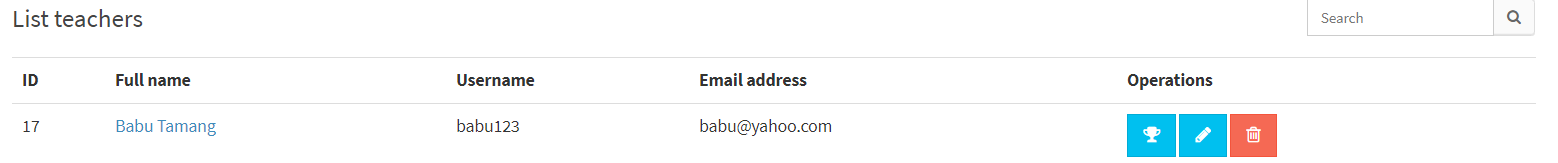


Fig: 8

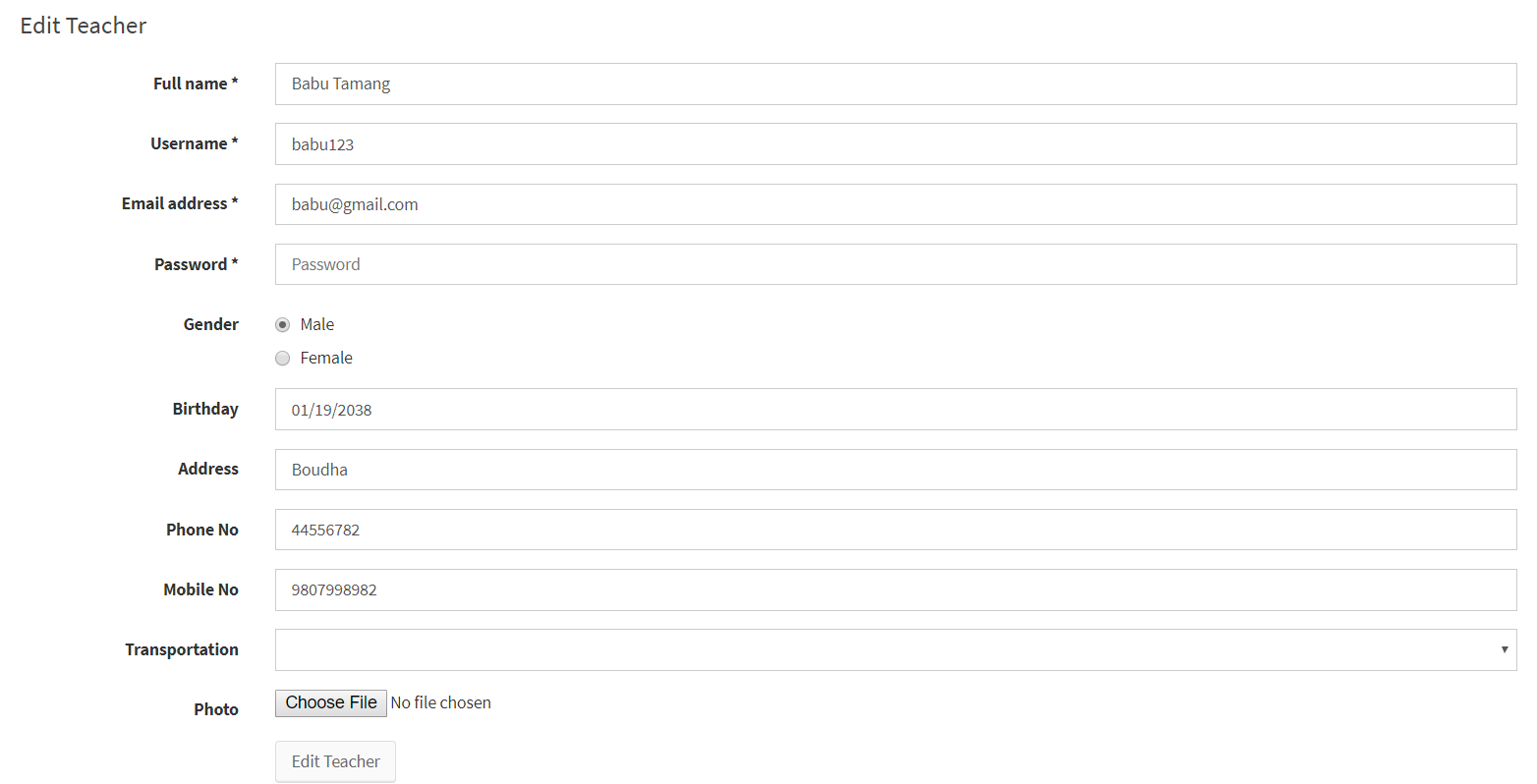


Fig: 9

Updated email id of Teacher by admin on request of teacher

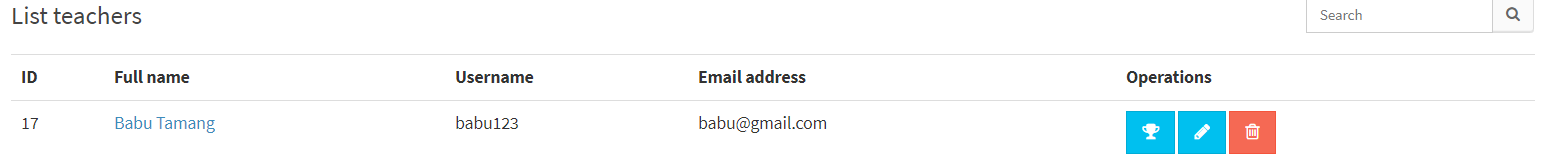
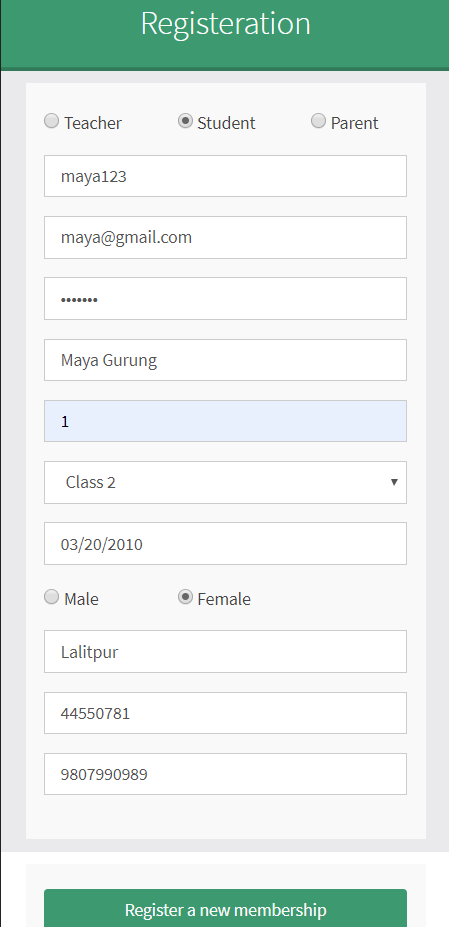


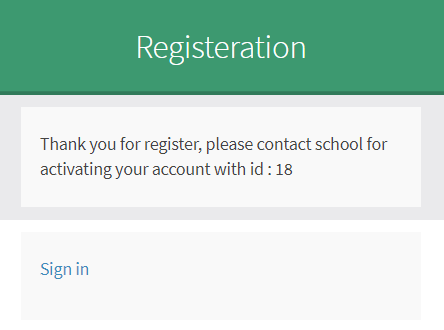
Fig: 10

**Test\_07**

**Student registration process**

****

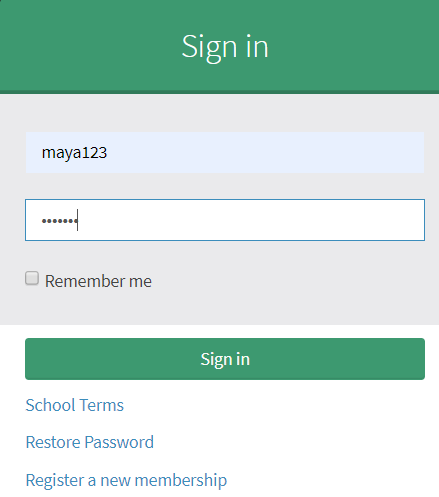
**Fig: 11**

****

**Fig: 12**

**Test\_08**

**Student login process**

****

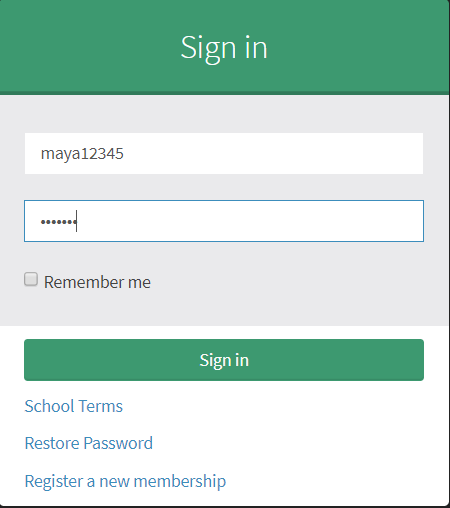
**Fig: 13**

****

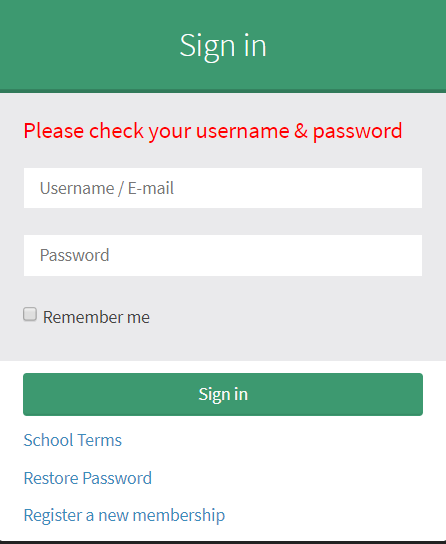
**Fig: 14**

**Test\_09**

**Student trying to login with wrong username**

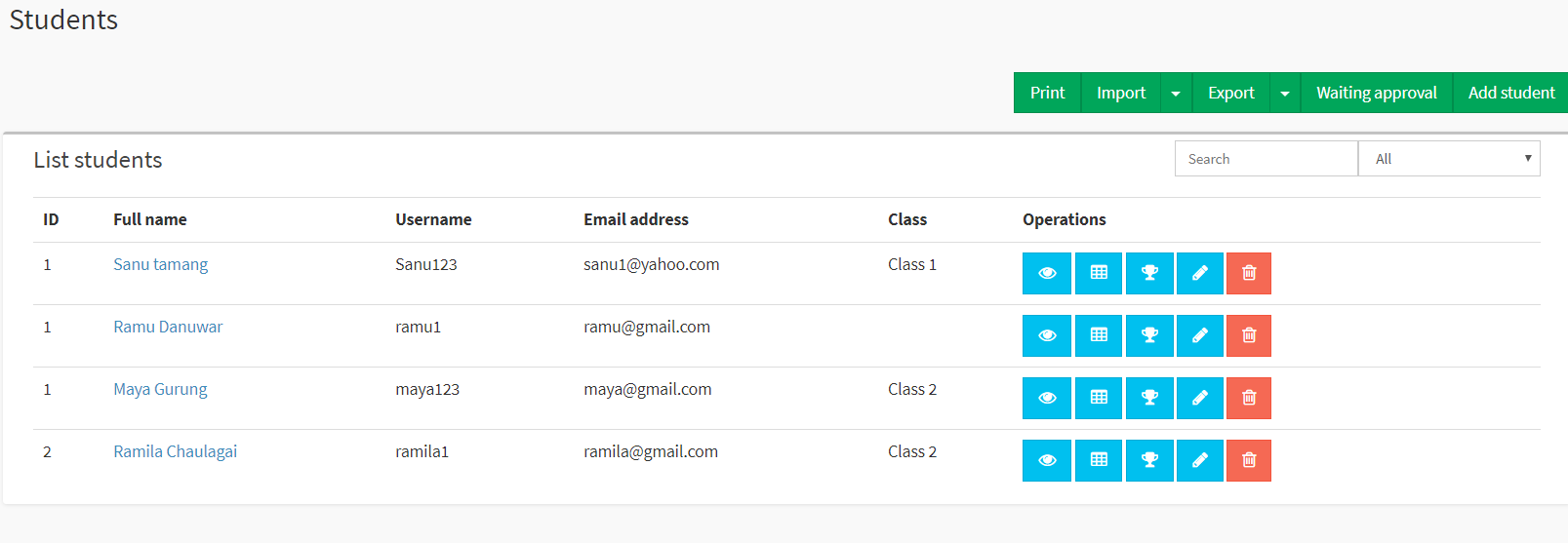
****

**fig: 15**

****

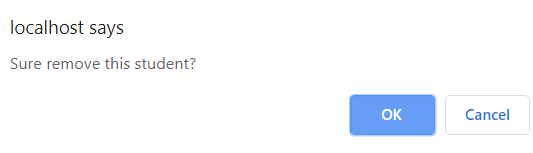
**fig:16**

**Before removing Maya Gurung from student list**

****

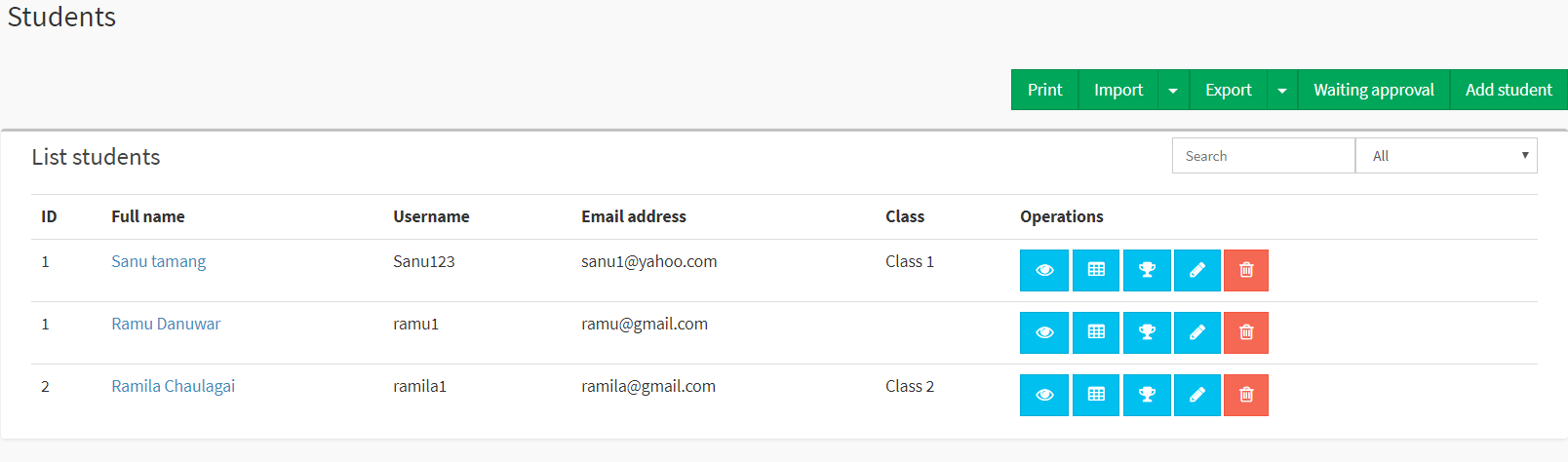
**Fig: 17**

**Confirmation on deletion of student**

****

**Fig: 18**

**After deleting student the list is like this below**

****

**Fig: 19**

14. Future work

Future work are as follows for my school management system

* Online school management system is one of my future work
* Student can pay through online their fees
* Teacher can easily get paid through online.

15. Limitation

* If the system crashes this application will be corrupted.
* Backend can be accessed for the storage but there is limit in storage.
* It is not online software.
* This application can be used only by school management.

16. User manual

17. Conclusion

So the main thing in this project is that I have used various design to develop which has made the system very reliable and user friendly and any school management can have good use of this system. Now with this system school have very much benefit in running school management.

18. References

<https://www.workbreakdownstructure.com/>

<https://en.wikipedia.org/wiki/Configuration_management>

https://www.tutorialspoint.com/mvc\_framework/mvc\_framework\_introduction.htm/

Analysis

Functional and non-functional requirement for School Management System

Functional requirement:Functional requirements are those which are related to the technical functionality of the system

Non-functional requirement: It is a requirement that specifies criteria that can be used to judge the operation of a system in particular conditions, rather than specific behaviors.

|  |  |  |
| --- | --- | --- |
| Functional(F)/Non-functional(NF) | Requirements | MoSCoW |
| NF(R1) | Login | M |
| NF(R2) | Registration | M |
| F(R3) | Admission Process | M |
| F(R4) | Update Information | S |
| F(R5) | Delete Information | C |
| F(R6) | Retrieve Information | S |
| F(R7) | Create Information | S |
| NF(R8) | Fees Information | S |
| NF(R9) | Teacher Information | S |
| NF(R10) | Student Information | M |
| F(R11) | Routines | M |
| F(R12) | Results | M |
| F(R13) | Laboratory | M |
| F(R14) | Salary | M |
| NF(R15) | Calendar | S |

Functional and Non-functional Requirements of School Management System