DEPT. Of Computer Science Engineering
SRM IST, Kattankulathur – 603 203
Course Code: 18CSC206J
Course Name: Software Engineering and Project Management
University Practical Examination Report

Name of candidate	**Thejaswin S**
Team Members	**Chinmaya Purohit, AP Rohith**
Register Number	**RA1911026010029**
Date of experiment	**07-06-21**

Team Members:

Sl No	Register No	Name	Role
1	RA1911026010020	Chinmaya purohit	Lead
2	RA1911026010029	Thejaswin S	Member
3	RA1911026010030	A P Rohith	Member

Contents:

EXPERIMENT- 1 A

Aim

To Frame, a project team, analyze and identify a Software project

Project Title: Attendance Management System

Project Description

By maintaining the attendance manually, then efficient reports cannot be generated. The system can generate efficient weekly, consolidated reports based on attendance. As the attendances are maintained in registers it has been a tough task for admin and staff to maintain for a long time. Instead, the software can keep long and retrieve the information when needed. This system will reduce manual work and avoid redundant data. Attendance Management System basically has two main modules for proper functioning: Admin module has rights for creating any new entry of faculty and student details. Users have the rights to make daily attendance, generating reports. Attendance reports can be taken by giving details of student details, date, and class.

Result:

Thus, the project team formed and the project is described.

EXPERIMENT- 1 B

Aim:

To create a business case and Arrive at a Problem Statement for the attendance management system.

Business Case

1. Executive Summary

The use of an attendance system is necessary for the students and staff of an institution, hence reducing the amount of time and effort required for manual attendance in the institute, and the safety will increase.

2. Strategic Business Context

2.1. Business Need

Teachers/Faculties and students are facing difficulties every day on taking/giving the attendance. Thus, the software is needed to keep track of attendance and update on a daily basis for the students in a university.

2.2. Business Outcomes

The final outcome of the project will be a well-organized record of student's attendance without any errors or technical failures and a reduction of manual work and stress. Also, the usage of paper will get reduced altogether which is beneficial for the environment.

3. Detailed Business Problem

3.1. Problem/Opportunity Statement

The problem here is the need for universities to adapt to an attendance management system where both students and faculty can securely maintain their attendance without the use of a lot of manual registers and workload and less paper wastage.

3.2. High-Level Requirements

The requirements are A secure login and password website showing their subject-wise attendance and cumulative attendance of the student and attendance of the staff and with a secure database for attendance of students and staff respectively

3.3. Assumptions

S.No	Assumptions
1	The person using this software has the necessary equipment such as a hard disk and laptop with an internet connection

3.4. Constraints

NA

S.No	Category	Constraints
1	hardware	absence of hardware
2	Network	absence of good network connection

3.5. Dependencies

N/A

Dependency Description	Critical Date	Contact
Cloud service	April 2022	Azure sales executive

3.6. Stakeholder Analysis

Name	Designation	Role in Project
Chinmaya Purohit	Back End Developer/Corporate Head for Sales & Marketing	These members will handle the work such as web servers, APIs, frameworks, and languages. The members will focus on the direct promotion of the brand and build content for the purpose of educating audiences.
AP Rohith	Project Manager	Project managers will supervise the planning, execution, and testing of the project.
Thejaswin S	Front End Developer	These members will be assigned the front-end work including the structure and design.

4. Detailed Analysis

4.1. Evaluation Criteria

The criteria for our attendance system will be a good front-end website without any glitches online and with a secure database to maintain the attendance of the students and staff

Evaluation Criteria	Deal Breakers (5)	Minimum Requirement (3)	Non-essential (1)	Score
Personal Identifiable Information (PII): Data must be encrypted 'At Rest and 'On Transit' The database must be secure and encrypted	Y	N A login page for checking attendance A separate database for students and staff necessary tools for additional looks	Hard disk	8

4.2. Cost of each Possible Options

Options (#)	One Time [CapEx]		Operational [OpEx]			Total Cost in INR
	Effort (Cost)	Infrastructure Cost	License Cost	Maintenance Cost	Infrastructure Increment	
Server hosting	0	20000	7000	12000	0	39000
Data plans	0	0	15000	11000	0	26000
Hardware	0	20000	0	0	0	20000
Software	0	0	3000	5000	0	8000

Category	Cost in INR
One Time (CapEx)	25000
Operational (OpEx)	28000

4.3. Risks

The risks could be from the database as to maintain students and staff attendance record for a prestigious university and from the front end website

Risk ID (#)	Risk Description	Risk Category [Low/Medium/High]	Risk Appetite [Accept/ Mitigate/ Transfer/Transfer]
R01	server maintenance for website hosting	High	Accept
RO2	Database for storing user information	Medium	Accept
R03	Miscommunication may lead to misfires. To stay on track and avoid wasted efforts in the wrong directions, the development team needs frequent Medium Mitigate input and feedback.	Medium	Mitigate
R04	Inadequate development or management resources can prove to be a huge loss. If the team doesn't have management-level expertise with the relevant technology, planning itself can be a serious hurdle.	High	Mitigate

5. Implementation & Governance

5.1. Required Skills

Skills	More Info
Frontend Development	Design and Develop UI and frontend layer
Backend Development	Design Database and Develop Service / API
Testing	Develop Test Cases
Project Management	Project Planning, Scheduling, Executing, Monitoring and Controlling

5.2. Milestone

Identify the significant points or events in the project. This table can also represent a high-level project schedule.

S.No	Project Milestone	Description	Expected Date
1	Planning	Requirements for the project will be planned and work will be divided among the group and each will be assigned a task.	10-1-2021
2	Design of UML diagrams	UML Diagrams describing the classes, functions, and modules will be made.	10-02-2021
3	Coding	Keeping the above layouts in mind, the coding for the project will be done.	25-03-2021
4	Testing	Testing of the project for bugs, trying with different test cases and resolving issues.	7-04-2021

5.3. Change Management

The business world is changing at a fast pace: technology keeps evolving, customer trends are changing, new market regulations are being launched on a regular basis, and businesses have to cope with unprecedented global crises.
Following steps can be taken to cope with the changes:
• Implementation of a new technology
• Change in leadership
• Change in organizational structure
The workflow is as follows:
• Review/Reporting
• Impact analysis

• Approve/Deny

5.4. Performance Measurement

Return in timeline	Return in INR	Investment (INR)	Remaining Investment
Return on 1st year	15,000	1,10,000	95000
Return on 2nd year	17000	1,10,000	50000
Return on 3rd year	23000	1,10,000	52,000
Return on 4th year	28000	1,10,000	67,000

The organizations will be deciding the implementation of the software by how organized and efficient the software is.

6. Project Charter

6.1. Simplified Project Charter

Section	Details
Project Scope	The project of the attendance management system seeks to provide a hassle-free attendance registering way for the students and staff alike. The system seeks to reduce the time and effort required for manual attendance.
Project Schedule	Updates to be shown weekly with the final project to be submitted by April 2021.
Project Cost	20,000
Constraints	Hard set deadline, absence of required hardware which may result in overhead expenses.
ROI	67,000
Intangible Benefit	Time loss reduced, safety increased, amount of effort is reduced.

6.2. Project Team Structure

Use an organizational chart to show the structure of the project team as well as the relationships between team members

6.2.1. Roles & Responsibilities

Project Role	Responsibilities	Assigned To
Software Developer	Responsible for coding and development of software. Checking bugs in the software may lead to fatal errors.	Thejaswin S,Chinmaya Purohit
Project Manager	Checking the overall progress and managing unforeseen cases.	AP Rohit
Integrator and Tester	Responsible for integrating the hardware and software and checking for any abnormalities. Making test cases in extreme conditions to check the software and hard compatibility.	AP Rohit
Business Analyst	Conducts research and analysis in order to come up with solutions to business problems and help to introduce these systems to businesses and their clients.	Chinmaya Purohit

6.3. Approval

Name	Designation	Role in Project	Signature
Ms.P.Mahalakshmi	Faculty -Incharge	Evaluator	

Reference

1. https://www.pmi.org/
2. https://www.projectmanagement.com/

EXPERIMENT-2

Aim: To identify the appropriate Process Model for the project and prepare Stakeholder and User Descriptions.

1. Executive Summary

Stakeholders are identified as part of the project initiation, and this list requires reviewing and updating as the project progresses. The process involves identifying everyone remotely connected with the project (individuals, groups, or organizations) that could influence or be impacted by a decision, activity, or outcome of the project. It also entails analyzing and documenting relevant information regarding their interests, involvement, interdependencies, influence, and potential impact on project success. Identifying all stakeholders in the initial stage itself helps in better manageable projects as it helps in factoring all stakeholder's interests at the planning stage itself. Obviously, in case any new stakeholders are identified subsequently and their requirements understood, the plans may need to be subsequently revised to accommodate those requirements. From day one on the project, we have been identifying a road map of how to integrate the stakeholders in the design process and have held several meetings and forums to review and discuss the objectives, constraints, and opportunities. Through this process, we are on track for an equitable phasing solution that allows a quality project to be constructed within the available FAA funding.

2. Selection of Methodology

This project is conducted based on the new proposed software development methodology called Agile which involves discovering requirements and developing solutions through the collaborative effort of self-organizing and cross-functional teams and their customer/end-user. Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. Agile software development refers to software development methodologies centered around the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictability, and greater aptitude to respond to change.

2.1. Roles and Methods

The roles and methods for our attendance management system would be first to assign different people such as information officer and Chief executive officer and financial head to allocate their time and money to enhance the system without any major faults

2.2. Agile Development and Testing[optional]

Throughout the building process, we may encounter numerous errors and bugs, which we can rectify with the help of iterative testing. This is how the agile model is beneficial for our project.

When a traditional system focuses on upfront planning where factors like cost, scope, and time are given importance, Agile management gives prominence to teamwork, customer collaboration, and flexibility.

3. Stakeholder Management

3.1. Identification of Stakeholders

The identification of Stakeholders will be as follows:
Chinmaya Purohit - Chief Executive Officer
AP Rohith - Chief Information Officer
Thejaswin S - Financial Head and Department executive

3.2. Interest and Influence matrix

Interest	Influence
High	High
Low	Low
Low	High
High	Low

Low Interest, High Influence Keep them satisfied as they can be 'defenders' Help them engage more	**High Interest, High Influence** Engage them closely as they are key 'drivers'
Low Interest, Low Influence Low Priority as they are 'spectators'	**High Interest, Low Influence** Keep them informed as they can be 'blockers'

Stakeholder Name	Activity / Area / Phase	Interest	Influence	Priority (High / Medium/Low)
Chief Executive Officer	Coordinating all other departments of the system	High	High	High
Chief Information Officer	Gathering the information and related enhancements of the project	High	Medium	High
Financial Head and Department executive	Assessing risk in investments and ensuring all accounting activities comply with regulations.	High	Medium	High
Frontend developer	responsible for implementing visual elements that users see and interact with within web applications.	High	High	High
Backend developer	Responsible for the server-side application logic and integration work.	High	High	High

3.3. Communication Plan for Stakeholders

Communication Goals:

- Keep stakeholders informed of the project timeline, budget, and project needs
- Provide clear insight into any decisions needed or roadblocks
- Provide structured opportunities for feedback from stakeholders
- Give stakeholders needed into to gain acceptance of the project

Weekly check-ins:

Weekly phone calls with Coordinator and Project Lead to go over progress, questions, and share any related work as it's completed

Share:

- Timeline & budget flags

- What's been completed
- What's in progress
- Any deliverables needing approval

Example:

- Walkthrough of progress to High Interest and High Influence stakeholders
- Providing single page dashboard and communicate via meetings
- Weekly report for High Interest and Low Influence

Reference

1. https://www.pmi.org/learning/library/stakeholder-analysis-pivotal-practice-projects-8905

EXPERIMENT-3

Aim

To Identify and document the Requirements of an Attendance Management Software system

1. Executive Summary

This Attendance management system is developed for students and teachers to keep track/record of daily basis of students attendance and to generate accurate monthly reports for statistical analysis and to calculate the student's eligibility criteria.
The projects have scopes like organizing the data and building the project flawless by proper testing etc.
The requirements of the project are proper broadband with optimum bandwidth, A computer with good performance, etc.
Project Deliverables:
A GitHub repository containing all of our code, commits, and documentation.

2. Project Scope

Expand on the scope definition and outline the major activities required to complete the project (for example, develop module ABC, develop requirements document, etc.). Out-of-scope activities are identified to reduce ambiguity.

S.No	Activities In Scope	Activities Out of Scope
1.	gathering Data of students	network with good bandwidth
2.	Developing and testing the data	employing an admin
3.	organizing the data	good working laptop and pc
4.	designing good UI	Compatibility on all platforms
5.	Accurate Calculations and Predictions	Workability in different mediums

2.1. In Scope

1. Gathering Data of students
   Collecting data of students such as date of enrollment, name, and year.
2. Developing and testing
   Constructing and developing the AMS with the requirements given by the client and testing it to make sure it has bug/error-free.
3. Organizing the data
   The data of a large no. of students are stored and organized well in the database to utilize it efficiently.
4. A good UI (User Interface)

By using common elements in our UI, users feel more comfortable and are able to get things done more quickly.

2.2. Out of Scope

1.Network with high bandwidth:

A good internet connection is mandatory in order to use the software and also plays an important role in managing and updating data.

2. Admin:

The admin takes care of all the methods and functions that are in the AMS.

3. Good working pc or laptop:
The machine should be able to adapt to our website and should not give any errors or bad requests that may hamper the performance of both the software and the machine

3. Epics [Major Functions]

Epic (#)	Epic Description
E1	Registration
E2	Login /Sign-up
E3	User
E4	Student details
E5	Attendance
E6	Report
E7	Logout

4. Requirements

4.1. Functional Requirements

Functional Requirements can also be expressed in the form of a “user story” which is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

Requirement (#)	Requirement Specification	Department	Name of Business User	Status
E1FR1	Authentication of the user whenever he/she logs into	web dev Team & cybersecurity	Back-end Developer	ongoing

	the Software user interface and should be safe			
E1FR2	The system should be flexible enough to edit the personal information according to the user.	web development Team	Developer	Ongoing
E1FR3	The Software must be able to display the basic information of the user such as name, roll no., the number of leaves taken and required no. of days to be present for eligibility.	web dev and database management	front-end Developer	Ongoing

4.2. Non-Functional Requirements

Requirement (#)	Category of NFR	Requirement Specification	Department	Name of Business User	Status
NFR1	Performance	All pages should load with less time	Tech Department		ongoing
NFR2	Performance	Search should bring the results in less than 10 seconds	Tech Department		ongoing
	Availability	The application should be available 24 hours and 7 days a week	Tech Department		ongoing
	Scalability	Registration Service should scale to serve 1000 request per second over 5 minutes timespan	Tech Department		ongoing
	Confidentiality	Only authorized users (students) belonging to a particular institution can have access to the software.	Tech Department		ongoing
E1NFR2	Usability	When used for a longer duration no issues are seen with respect to RAM consumption or any network problems	Tech Department		ongoing
	Security	Data security of student's personal	Tech & cloud Department		ongoing

		details and displaying what's required to the user.			
	Traceability	Users data is untraceable	cybersecurity Department		ongoing
	Flexibility	Being able to view attendance whenever we try to login to the portal	Tech Department		ongoing
E1NFR1	Extensibility	Ordering Service should be extensible to other countries within a week's time	Tech & marketing Department		ongoing
	Interoperability	Ensuring the software works on different computers other than the host	Tech Department		ongoing
	Reliability	The quality of the software is trustworthy and its performance is consistent.	Tech & management Department		ongoing
	Rapidity	To get the student's details in less time without any warnings or errors	Tech Department		ongoing

4.3. Infrastructure Requirements

Requirement (#)	**Requirement Specification**	**Department**	**Name of Business User / Project Team Member**	**Status**
IR1	Development Machine with min 4 GB Ram and 4 Cores	hardware development	system user	ongoing
IR2	Code Repository	web dev team	Thejaswin S	ongoing
IR3	UI of the AMS	web dev team	A P Rohith	ongoing
IR4	Proper databases and servers to manage and store data	web dev team	Chinmaya Purohit	ongoing
IR5	IDE-vscode	Technical team	AP Rohith	ongoing

4.4. Requirement definition in Agile [Optional … Use according to methodology chosen by student]

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective.

How to write a user story…

- We are building our software for the students to check their attendance subject-wise and cumulative attendance for a semester or a calendar year. The user for our system are the students themselves
- We are building an Attendance Management System for the students of a prestigious institution and the intention being a hassle-free management system with proper data security and reduction of manual work in huge registers which makes it expensive and difficult to maintain
- We are building our AMS for the students to check their attendance because attendance is a valuable criterion which institutions take for checking student's academic record and evaluating performance and the value it brings to the user is a satisfaction to check whether their attendance is regular without any technical issues or manual errors in maintaining attendance and to reduce labor and stress for the employees of the institution.

User Story	Acceptance Criteria	Size of User Story
As a student, I can view my attendance in various subjects individually and cumulative attendance for the entire semester or calendar year	View my attendance on a daily basis to ensure real-time updation Allow having my user details safe and secure and able to view my attendance without any errors The allowing of generating an update request in case there's a mistake in the attendance percentage of a subject or the cumulative as a whole	Medium
As a student,I can get a detailed monthly report and eligibility criteria for writing exam based on attendance	the software keeps track of details of attendance and calculates and updates data in the database on a daily basis.	Medium

EXPERIMENT-4

Aim

To Prepare Project Plan based on scope, Find Job roles and responsibilities, Calculate Project effort based on resources

1. Executive Summary

The plan is to create an AMS with excellent data security and a reduction of manual labor. In order for this to work, we divide the work into teams of people who are good at their respective departments and estimation of requirements will be a good database and a well running website online and a working PC or laptop with a stable network and the cost estimation will be around 20000 INR. The skills required are obviously knowledge in programming languages such as JS, HTML and CSS and backend development languages such as SQL and to coordinate the whole project we need good business and technical knowledge to present it to different institutions and stakeholders.

2. Project Management Plan

Focus Area	Details
Integration Management	Governance Framework Project Team Structure Roles & Responsibilities of Team Chinmaya Purohit: Technical lead(Responsible for server-side web application logic,backend developer) AP Rohith: Project Manager(Monitor the overall progress of the project) Thejaswin S : Developer (designing UI, implementing, and managing software programs.) Tester: execute all levels of testing (System, Integration, and Regression) and analyze user's stories. Project sponsor: Negotiate to fund for the project and Review changes to the project environment, including schedules, priorities, tasks. Change Management

	minimize risks of changes while developing the AMS system. Goal is to conduct a formal, standardized methodology in the handling of all Changes in order to be transparent in our work, prevent change-related incidents, minimize negative impact on delivery of services to our users and clients. (Change Control, Issue Management) Project Closure
Scope Management	Scope Statement Our project aims to provide the user with a user-friendly interface. The user is asked to enter the details of students such as date of enrollment, year, etc. According to the data entered the software provides accurate calculation of attendance and leaves taken . Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Gathering: collecting required data from students and storing it in a database. control: Vscode, XAMPP server assumption: stable internet connection with good bandwidth accuracy(accurate calculations) efficient database constraints: Security: The Software can only be accessed by authorized users of the organization. Deadline: Project must be completed within 90 days. quick remediation- fix it before it breaks Budget: The whole project should be under 10000 Deliverable Software that will be useful for both students and teachers in keeping track of attendance and calculation instead of manual calculation which takes time. Requirement Change Control The project should be flexible enough to accept change requests by client or customer and make changes. Activities and Sub-Tasks

	track attendance login page registration page print report
Schedule Management	Milestones expected date of completion UI/UX frontend - 16/4/21 web hosting and database - 1/5/21 project completion & testing - 25/5/21 Schedule Control Re-estimate the remaining components of project identify the need for change request measure variances against schedule conduct performance reviews
Cost Management	Effort min 5 hours/day per person required min 4 person Team Chinmaya Purohit: Team lead /backend developer AP Rohith: project manager, UI/UX Thejaswin S: Frontend developer, designer Budget Control expected budget - Rs10000 internet cost -Rs 1000
Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques, and reporting Regular inspection of software for identification of bugs a qualified tester tests the software. Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products Improved operational excellence and customer satisfaction verify if the quality of the parameters are met Inspection and user feedback
Resource Management	Estimate and Manage the need People: Frontend developer,backend developer, tester, project manager,HR Finance: budget- RS 10000 (including all)

	Physical: Visual studio IDE
Stakeholder	Identifying, Analyzing, Engaging Stakeholders CEO Supervision of all the departments i.e., Verifying technical aspects of the project, taking care of companies' status, making sure that everything in the company goes right CIO responsible for planning, implementing and managing the overall use of Information technologies within companies ensuring support and security. Finance head Plays a crucial role in controlling the cost of a project while maintaining the quality and standards of the project. Calculate cost estimates for building AMS and also implement precautionary measures for AMS. project sponsor ensure the resources are in place, promote the project, and hold overall responsibility for the project's success. They represent the business side of the project. frontend developer backend developer client A person or organization which is going to use this software investors competitors project manager The work is to supervise the plan and progress of AMS. Keep the process monitored that the milestones are reached as per the schedule and no mistake is made.
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient] Collecting information related to work performance, analyzing it, creating reports, and sending them to respective stakeholders are involved in reporting the performance of the project.

	Requirements documents should also be validated by the sponsor. Project status report Review project status and discuss potential issues and delays Team standup Discuss what each team member did yesterday, what they'll do today. Project review Present project deliverables, gather feedback, and discuss next steps Post marten meeting Assess what worked and what did not work and discuss actionable takeaways.
Risk Management	Steps to identify risks Risk repository: The risk repository is the history data containing the list of risks identified for completed projects. The risk repository can be used to arrive at a list of potential risks for the project. Checklist analysis: The risk identification checklist is a questionnaire that helps identify gaps and potential risks. It is developed based on experience and project type. Expert judgment: Risk identification is also done by brainstorming with or interviewing experienced project participants, stakeholders, and subject matter experts. Identifying, analysing, and prioritizing project risks Tight Schedules Often, project managers face the pressure of having to deliver the project earlier than anticipated. This can happen due to many different reasons – lack of resources, poor planning, or even technical glitches. Data security, information privacy, large-scale system implementation, software integration, and compliance.
Procurement Management	Adhering to the organization procurement process Getting the best quality from the outside vendors employed by the company to assist in its doing business.

	The software involves 3rd party access to sensitive university data (eg. Student).

3. Estimation

3.1. Effort and Cost Estimation

WBS	Activity	Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
E1FR1	E1R1A1	Design the user interface for the AMS	E1R1A1T1	attendance calculation	2 hours	1000
			E1R1A1T2	Login page and authentication. Ask the user their email and password and confirm the authenticity.	approx 3-4 hours.	1000
			E1R1A1T3	Make the UI user friendly	4-5 hours.	2000
	E1R1A2	gathering of data from the particular organization(eg. college or school)	E1R1A2T1	collect data and document/store data in a database.	less than 10	2000
			E1R1A2T2	Email/phone number confirmation for the first time to confirm the login.	2-3 hours.	500
E1FR2	E1R2A1	profile page	E1R2A1T1	display student details and statistics of attendance to date.	3 hrs	500
			E1R2A1T2	DIsplay and suggest required classes to attend for each subject.	1 hour	1000
	E1R2A2	Get to know the client's requirements	E1R2A2T1	confirm the user requirements acceptance criteria.	2 hours	0
E1FR3	E1R3A1	server hosting and setting database	E1R3A1T1	Storing login credentials of the user	3 hours	2500

Effort (hr)	Cost (INR)

1	2500

3.2. Infrastructure/Resource Cost [CapEx]

Infrastructure includes the hardware needed to be programmed. Since buying hardware will be only be used in the initial phase, it comes under CapEx

Infrastructure Requirement	Qty	Cost per qty	Cost per item
IR1(computer)	4	248.76	2000
IR2(hard drive)	2	501.23	1000
IDE	1	0	0
PhpMyAdmin	1	0	0

4. Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	developer tester team members DB admin	5	1,80,000	60000
License	Operating system Database IDE	0	0	0
Infrastructures	database and server Storage vs code ide development machine network	8 GB ram and 4 cores	0	2000

5. Project Team Formation

5.1. Identification Team members

Name	Role	Responsibilities
User(Institution)	Key Business User	Provide clear business and user requirements
AP Rohith	Project Manager	Manage the project
A P Rohith	Business Analyst	Discuss and Document Requirements

Chinmaya Purohit	Technical Lead	Design the end-to-end architecture
Thejaswin S	UX Designer	Design the user experience
Thejaswin S	Frontend Developer	Develop user interface
Chinmaya Purohit	Backend Developer	Design, Develop and Unit Test Services/API/DB
Team as a whole	Integrator	Integrate the hardware with software
Team as a whole	Bug fixing	Fixing errors if or any present
Team as a whole	Tester	Define Test Cases and Perform Testing

5.2. Responsibility Assignment Matrix

RACI Matrix	**Team Members**			
Activity	**Chinmaya Purohit (project lead & backend developer)**	**Thejaswin S (Frontend Developer & UI/UX designer)**	**AP Rohith (Project Manager & business analyst & Tester)**	**Key Business User(client)**
User Requirement Documentation	A/I	R/I	R/I	C/I
Design the user interface	I/A	R	I	C/I
develop and unit test services	I/R	I/R	A	C/I

A	Accountable
R	Responsible
C	Consult
I	Inform

EXPERIMENT-5

Aim

To Prepare the Work, Breakdown Structure based on timelines, Risk Identification and Plan.

1. Executive Summary

The plan is to create an AMS with excellent data security and reduction of manual labor. In order for this to work, we divide the work to teams of people who are good at their respective departments and estimation of requirements will be a good database and a well running website online and a working PC or laptop with a stable network. The skills required are obviously knowledge in programming languages such as JS, HTML and CSS and backend development languages such as SQL and to coordinate the whole project we need good business and technical knowledge to present it to different institutions and stakeholders.

In this document we shall discuss our Work Breakdown Structure with Project Schedule, followed by Risk identification and risk management.

2. WBS With Project Schedule

Module (#)	Activity (#)	Sub-Task(#)	Assignee(s)	Planned Start Date	Planned End Date	Actual Start Date	Actual End Date	Status
1.	1.Planning	1.Discuss work plan	Chinmaya Purohit	15-02-21	20-02-21	15-02-21	20-02-21	completed
		2.Assigning roles and task	A P Rohith	5-02-21	7-02-21	5-02-21	7-02-21	completed
2.	1.dataset	1.Collecting data	Thejaswin S	2-03-21	5-03-21	2-03-21	2-03-21	completed
		2.Organizing data	AP Rohith	5-03-21	10-03-21	5-03-21	5-03-21	completed
3.	1.Designing the UI	1.Login page	Chinmaya Purohit, Thejaswin S	16-03-21	20-03-21	20-03-21	20-03-21	completed
		2.attendance page						
4.	Attendance tracking	1.algorithm	A P rohith, Chinmaya purohit	28-03-21	2-04-21	28-03-21	28-03-21	completed
		2.Database						

5.	frontend coding	1.login and register 2.attendance page 3. profile page	Thejaswin S, A P Rohith	20-03-21	26-03-21	21-03-21	21-03-21	completed
6.	Authentication using firebase	Adding security to project and authentication	Chinmaya purohit	3-04-21	8-04-21	5-04-21	5-04-21	completed
7.	Testing	Rechecking complete software with all end to end test cases	A P rohith	26-04-21	1-05-21	26-04-21	26-04-21	completed

3. Risk Identification

- Structured Brainstorming with team and stakeholders
- The checklist is a list of actions/points to be considered [Information can be used from similar previous projects]
- Risk can be identified from
    - Assumption-Constraint analysis
    - SWOT Analysis [Strength/Weakness/Opportunity/Threat]

3.1. List (Describe) Register

Risk ID (#)	Risk Description	Impact Description
R01	Server Breakdown	The website fails to load
R02	Denial of Service Attack malware attack	Server overload may lead to system failure or the data to be erased or to be changed.
R03	Insufficient database	improper attendance tracking
R04	Lack of ownership	project failure
	Budget Risk	
R05	Market competitor	less no. of users

3.2. Managing Risk

Risk ID (#)	Status [Open / Closed]	Risk Appetite [Accept/ Mitigate/ Transfer/Avoid]	Action	Action Owner	Target Date	Remarks
R01	Open	Transfer	Choosing the best server providers providing 24/7 service	Thejaswin S	29-04-21	Handling server issues will be done by the server providing service itself
R02	Open	Mitigate	Preventing hackers to overload the servers/hack the website	Chinmaya Purohit	30-04-21	Reduce the likelihood of a DOS / DNS attack on the website by cybersecurity tools.
R03	Open	Avoid	Frequent database updation	A P Rohith	continuous	-
R04	closed	Avoid	see to that everyone is engaged in making the software and Keep Stakeholders Updated and Keep to Planned Scope and avoid scope creep.	AP Rohith	continuous	-
R05	Open	Accept	marketing	Marketing team	Post-publication	Advertising to be done by the marketing team

EXPERIMENT-6

Aim

To prepare the architecture and design of the system

Software Used

Star UML, Lucid Chart

Architecture Diagram with description

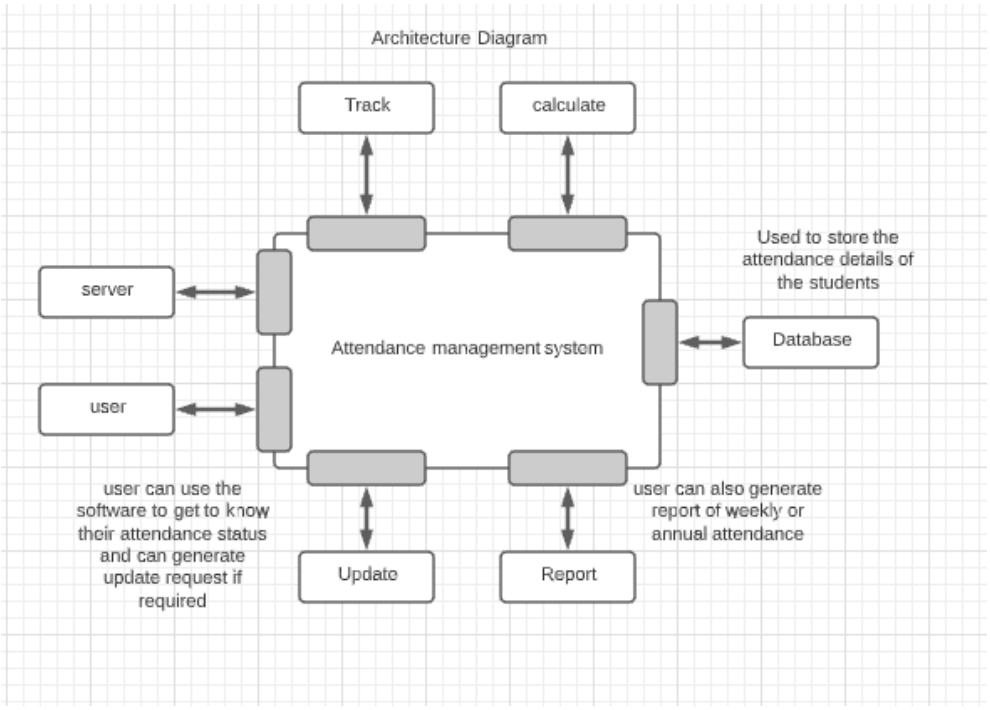
An architecture diagram is a graphical representation of a set of concepts that are part of the architecture, including their principles, elements, and components.

It is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components.

The Architectural context is used to identify and define the external entities(other systems, devices, and people that the software interacts with (say actor).

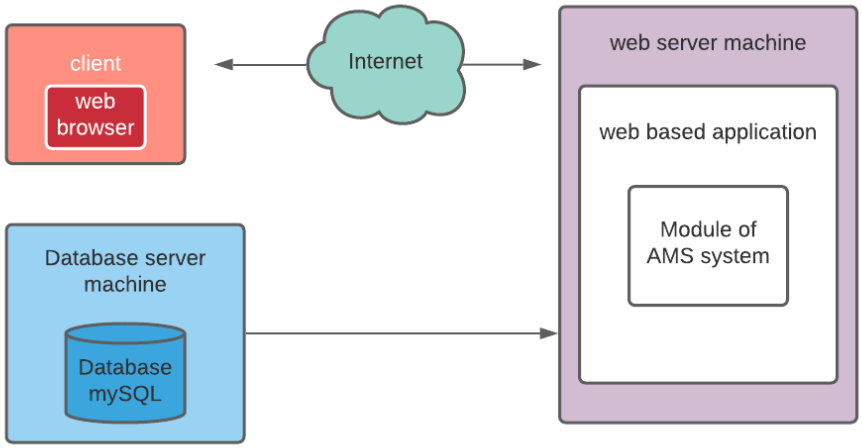
The external entities that may involve in the project Attendance management system are:

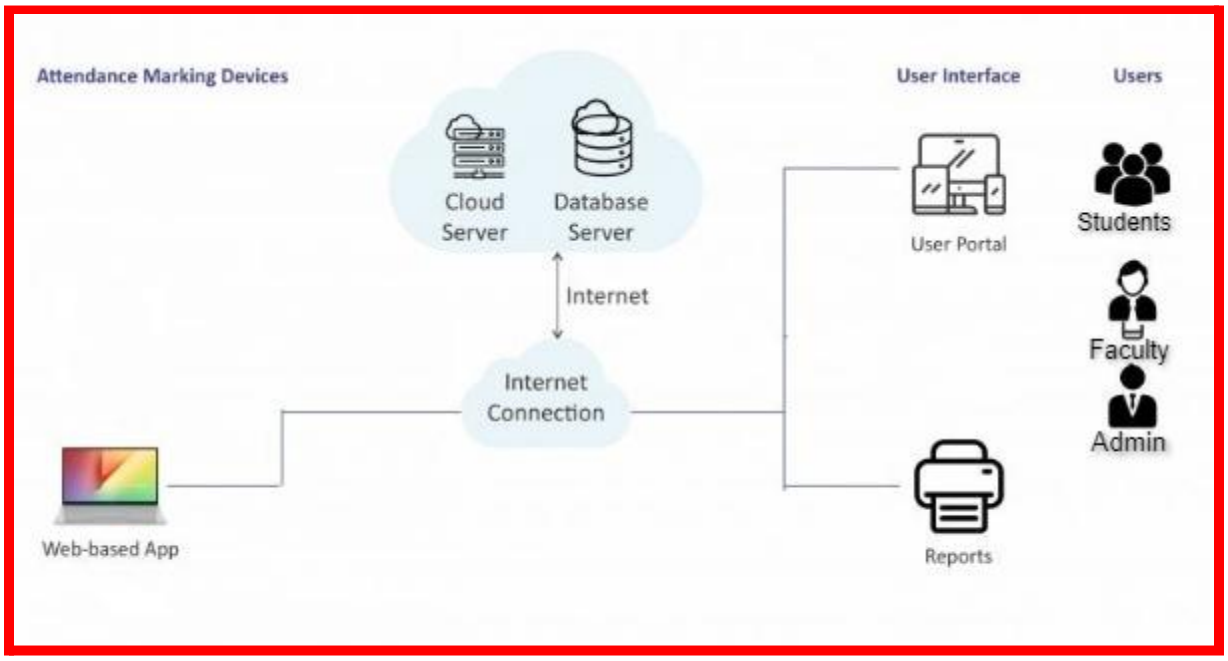
1. User
2. Server
3.Track
4.Calculate attendance
5.Database
6.Update attendance
7.Report generation



In the architecture diagram, the user interacts with the system through the attendance login page. The user here is the students or faculty and the server is used to get logged in the system. There is a tracking process to help trace the daily and cumulative attendance of a student in a class and the calculation process is used to calculate the percentage of his attendance in a semester or on a daily basis as well. The database process is used to store the attendance details of the student on a daily basis excluding leaves and holidays.

Finally, the Update process is used so that the student can use the software to get to know their attendance status and in case of a mistake or error, he/she can generate an update request which will be fulfilled later, and the Report process is used to generate a final report of the weekly or annual attendance of the student in the university.


client
web browser
Internet
web server machine
web based application
Module of AMS system
Database server machine
Database mySQL


Attendance Marking Devices
User Interface
Users
Cloud Server
Database Server
User Portal
Students
Internet
Faculty
Internet Connection
Admin
Web-based App
Reports

In this design technique, the total application is divided into three parts, so it is named Three Layer Architecture. They are:

1. Presentation Layer (User Interface)
2. Business Logic Layer (BLL)
3. Persistence Layer

Presentation Layer :

The presentation layer is the front-end of the Application Design Architecture which provides the user interface to either programmer or end-user. The presentation layer is responsible for

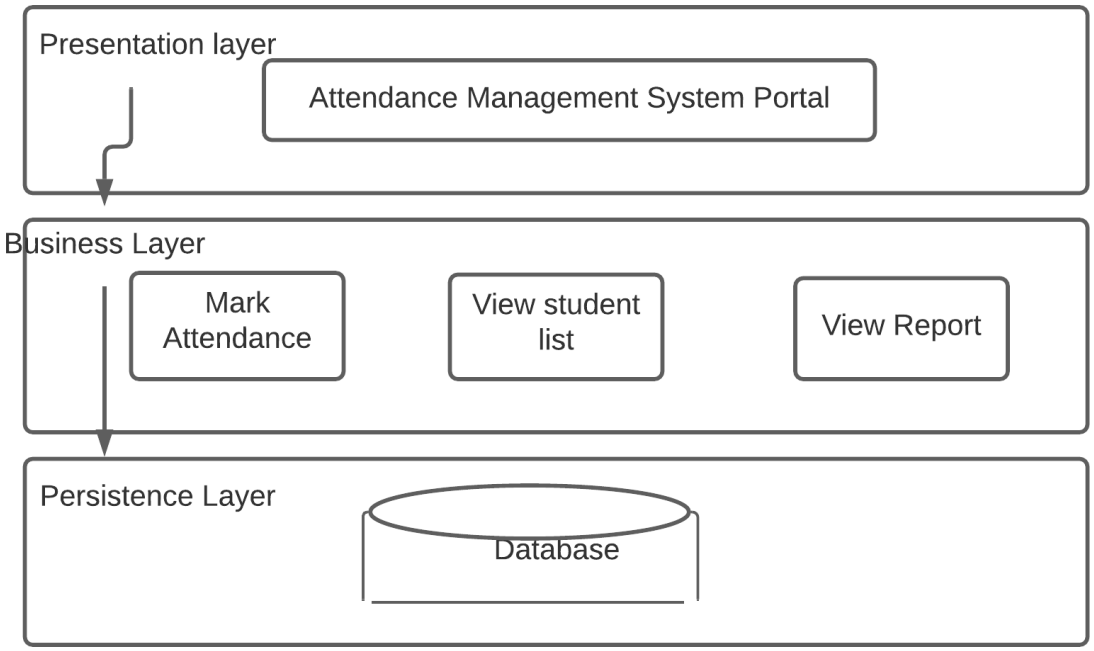
- Data encryption/decryption
- Character/string conversion
- Data compression
- Graphic handling

Business Logic Layer

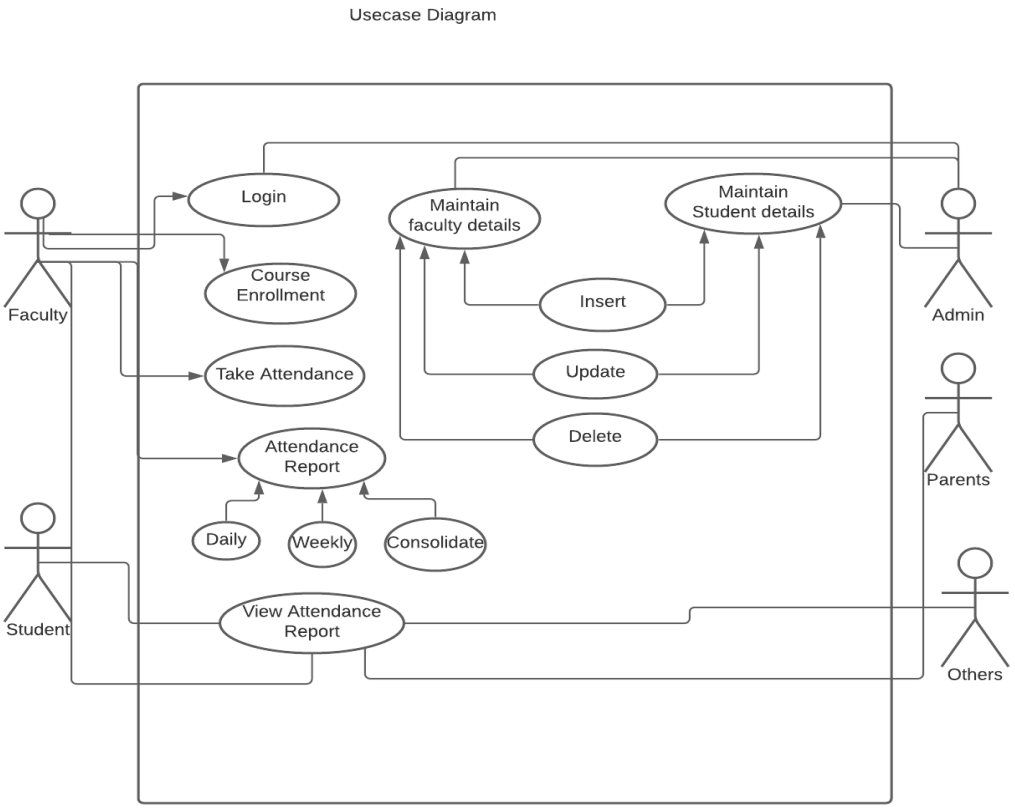
The business logic layer is the middle layer of the Application Design Architecture which makes the bridge between the front-end and backend. All the business logic is implemented in this layer. This layer is a class that we use to write the function which works as a mediator to transfer the data from the Application or presentation layer data layer. In the three-tier architecture, we never let the data access layer interact with the presentation layer.

Persistence Layer:

This is the backend of the Application Design Architecture which is concerned about the access, retrieval, update, and storage of data. The layer is also a class that we use to get or set the data to the database back and forth. This layer only interacts with the database. We write the database queries or use stored procedures to access the data from the database or to perform any operation to the database


Presentation layer
Attendance Management System Portal
Business Layer
Mark Attendance
View student list
View Report
Persistence Layer
Database

Use Case Diagram

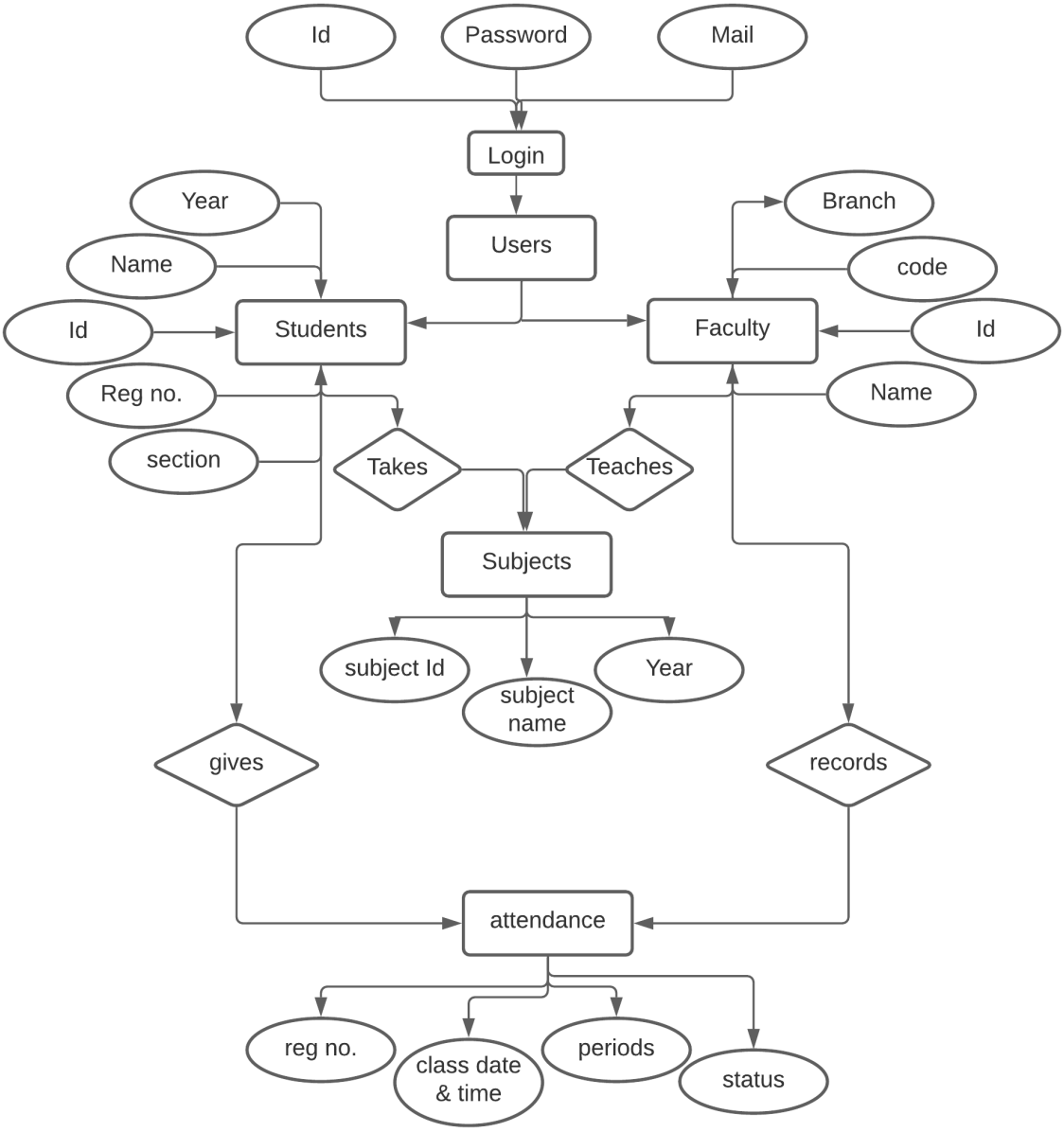


A use case diagram is the primary form of system or software requirements for a new software program under development. It consists of 3 active participants: Actors, Use Cases and Boundary of the System. The actors in the diagram drawn below are Faculty, students, Admin, Parents, and Others. The use cases are Login, Courses Enrollment, Take Attendance, Attendance Report, View Attendance Report, Maintain Faculty Details, and Maintain Student Details. The rectangular box representing the system boundary is potentially the entire system as defined in the requirements document. An actor is a person and an external entity who interacts with the use case. He has responsibility towards the system i.e. the input and has expectations from the system i.e. the output. Use case is a system function which can be processed manually or automated to do some specific work. In our project it is automated. The actors communicate with the use cases by a solid link, while some use cases may not be connected with the actors. Some use cases also have functionalities attached to them. In this use case diagram the first actor

is the faculty who logs in to the system by giving a valid user id and password. Then he chooses a particular course among the 3 courses i.e. MCA, MBA, and MTech. After that, he can view the list of students enrolled in a particular course and take attendance. Then he can generate the attendance report on the basis of daily, weekly and consolidated entry owing to which he can view the Attendance report. Daily, weekly, and consolidate are the 3 functionalities attached to the attendance report. The other 3 actors, students, parents and others have only to view the attendance report.

Last but not the least, admin comes into the picture by maintaining the faculty and student details. The admin can insert, delete and update the student and faculty details. Again these are the functionalities attached to the two use cases by dotted lines.

ER Diagram

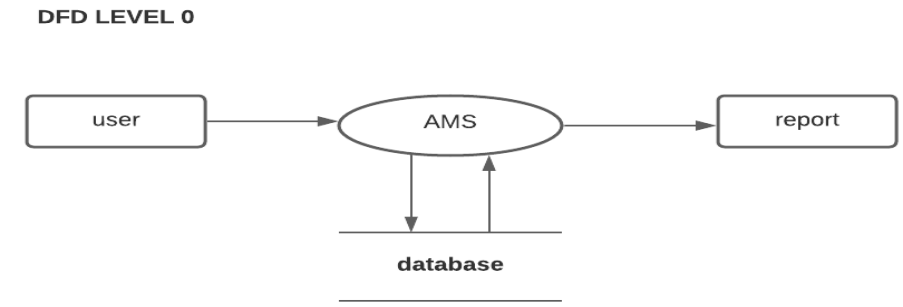


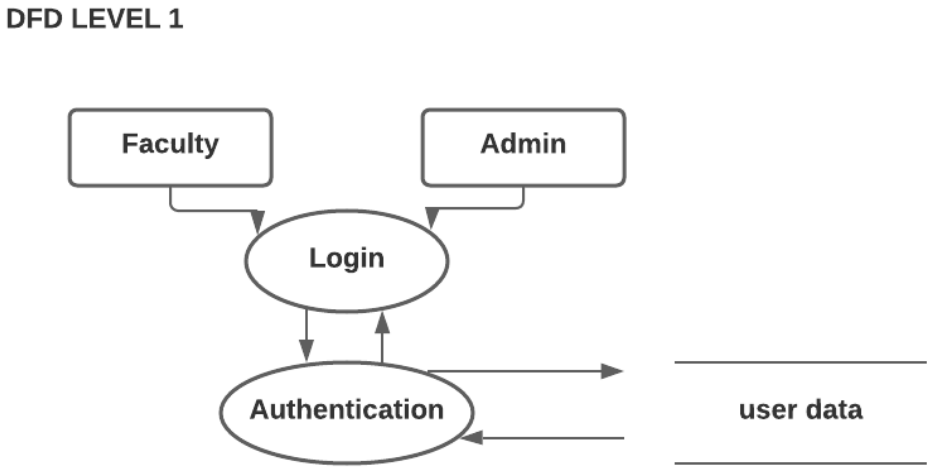
E-R (**Entity-Relationship**) Diagram is used to represent the relationship between entities in a table.

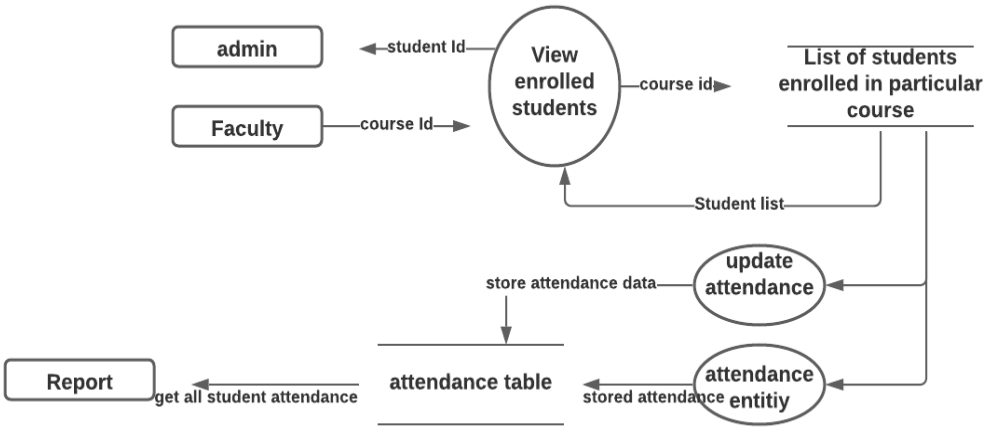
E-R diagram means Entity Relationship diagram. Entity means object of the system, generally, we refer to the entity as a database table, the e-r diagram represents the relationship between each table of the database. E-R diagram represents an entity with attributes, attributes are property of an entity. If we assume an entity is a database table then all the columns of the table are treated as attributes.

In this AMS, we have various attributes such as reg no, periods, status, and ID and so on with the entities such as Users, Students, Faculty, and Login. So once he/she submits their credentials on the system and logged in successfully, There is a Users entity which directs to either Students or Faculty depending on the user and attributes such as Name, ID, section are stored and for faculty, code, branch and ID attributes are stored. After this, they are directed towards the subject's entity with the attributes such as subject name, ID. This is separate as the student takes subjects for the academic year and the faculty is responsible for teaching said subjects. After this, Students can check their attendance which is present in the attendance entity with attributes such as reg no, date & time and status. The Faculty is responsible for giving attendance to students so they are separated from the student's entity.

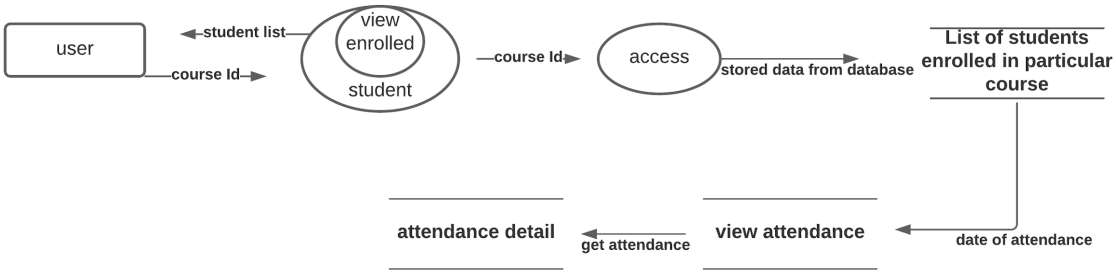
DFD Diagram (process)



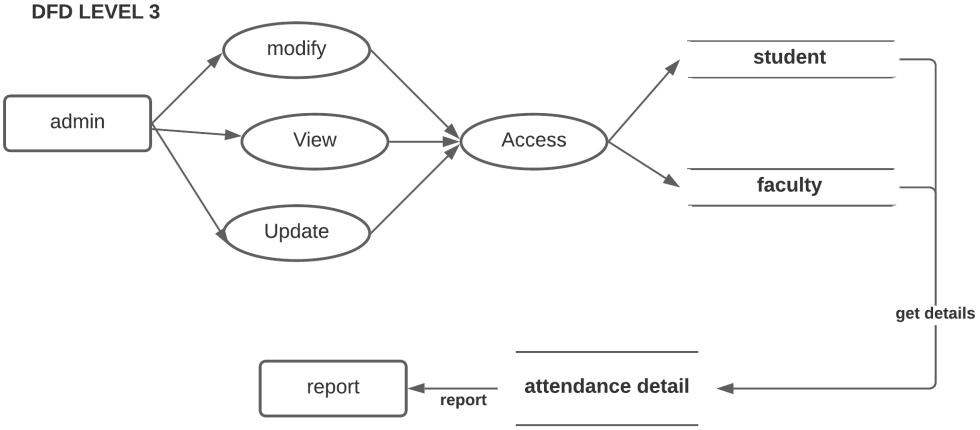




DFD LEVEL 2



DFD LEVEL 3



A data flow diagram is a graphical representation of the data flow through an information system that is used to model the process aspects of the system. DFD is the preliminary step used to create an overview of the system. DFD is used for structural design.

LEVEL 1 DIAGRAM

The context-level DFD is then exploded to produce a level 1 DFD which models the details of the system. The level 1 DFD shows how the system is divided into subsystems (processes), and how each process deals with one or more of the data flows to or from an external entity, and how the processes together provide all the functionalities of the system. The level 1 DFD also justifies the internal data stores which must be there for the system to do its job and shows the data flow between the various parts of the system. In the below level 1 of the DFD, the attendance system has been decomposed into 5 processes which are namely Login, Admin, View courses taken, View enrolled student list, Upload attendance. Each process is accessed by the faculty and the admin. There are data storage namely attendance tables, enrolled student lists, and course lists, and user details which are used in the system.

LEVEL 2 DIAGRAM

The level 2 DFD is the further decomposition of level 1 processes into sub-processes (sub-systems) which give a detailed description of the data flow in each process. Here we have decomposed the view design in the sub-process view attendance. In this, the user may be faculty, admin, parents, students, and others select the course ID and get the details of the students enrolled and then he can view the attendance for that particular course. The data storage used in this level is the attendance details, course list, and enrolled student list.

LEVEL 3 DFD

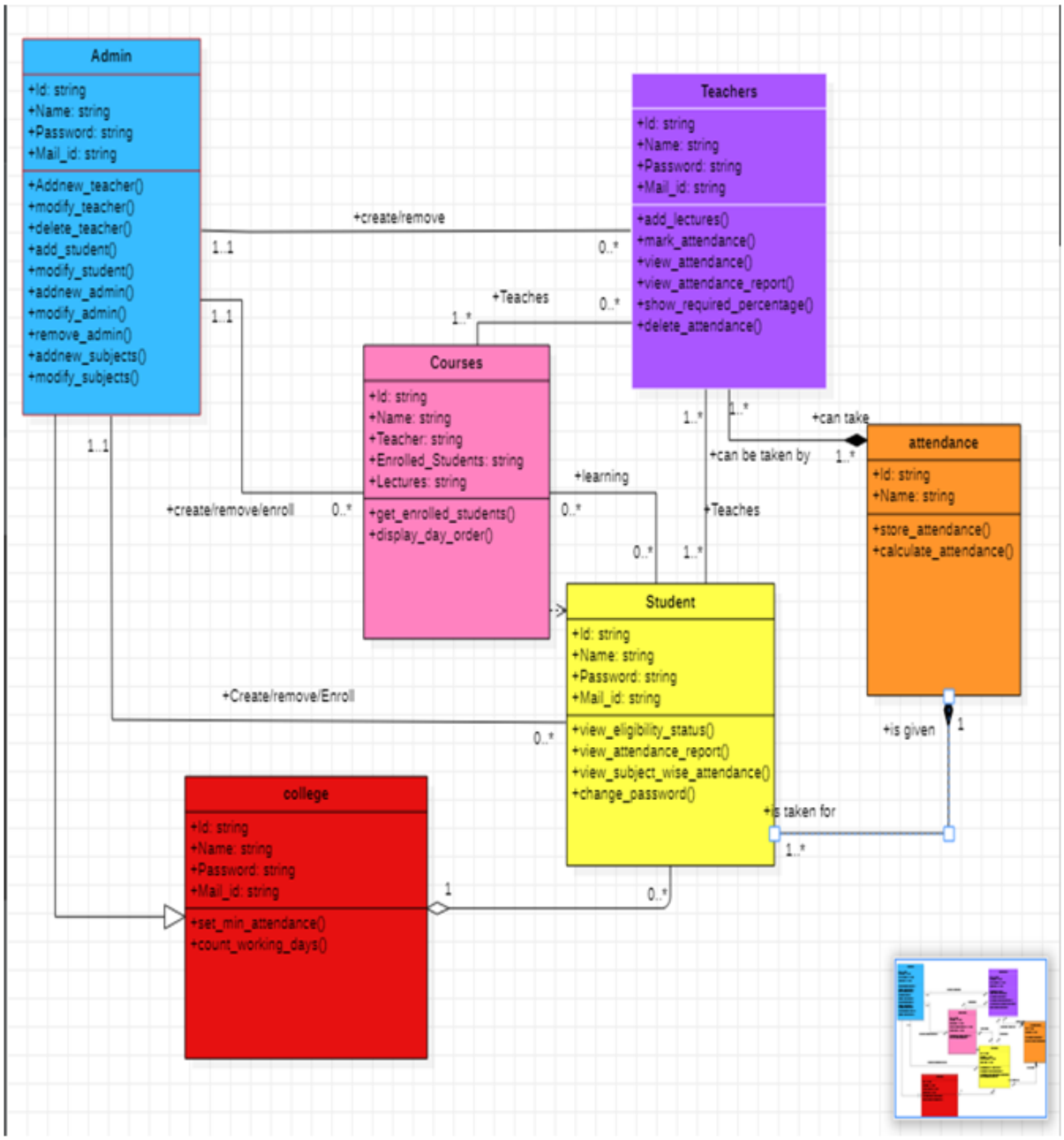
The level 3 DFD is the further decomposition of level 2 processes into sub-processes which again give the detailed description of the data flow in each of the processes shown below. Here the third module of the attendance management system the admin has been decomposed with its functionalities. The admin here can either modify, view, or update the faculties or the students in the institution. After that, he can get data stored from the respective faculty database and the student database and thereby get the details of the attendance and view the attendance report.

Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

1. Shows the static structure of classifiers in a system
2. Diagram provides a basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from a business perspective



The Class named teacher and class attendance is connected by composition as Class teacher is a part of class attendance for taking attendance and these classes have strong ownership between them. The multiplicity order here denotes that teachers can take attendance for one or more students and attendance can be taken by one or more teachers. Similarly, class attendance and student are connected by composition. The multiplicity order here denotes that students can give one attendance per period and attendance can be taken for multiple students. Class admin and teacher and student are associated. Admin can create/remove/enroll 0 or more students as well as teachers. And teachers and students data

are controlled by 1 admin. Class students and teachers are associated as teachers can teach one or more students and students are taught by one or more teachers. Class courses are associated with all classes except attendance and college. Class admin and Class College have generalization as college is superclass and admin is base class. Class College and class students have aggregation since Class College can work even if class students are not there but still it's a part of Class College. College can set min attendance required for semester/year for 1 or more students.

Sequence Diagram (Applied For OOPS based Project)

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects 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. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

USES OF SEQUENCE DIAGRAM

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

Given below are sequence diagram respectively:

LIFELINE USED:

For login validity:

Login page, forgot password, database, verification, authenticate page

For Administrator:

Login, student, teacher, attendance management

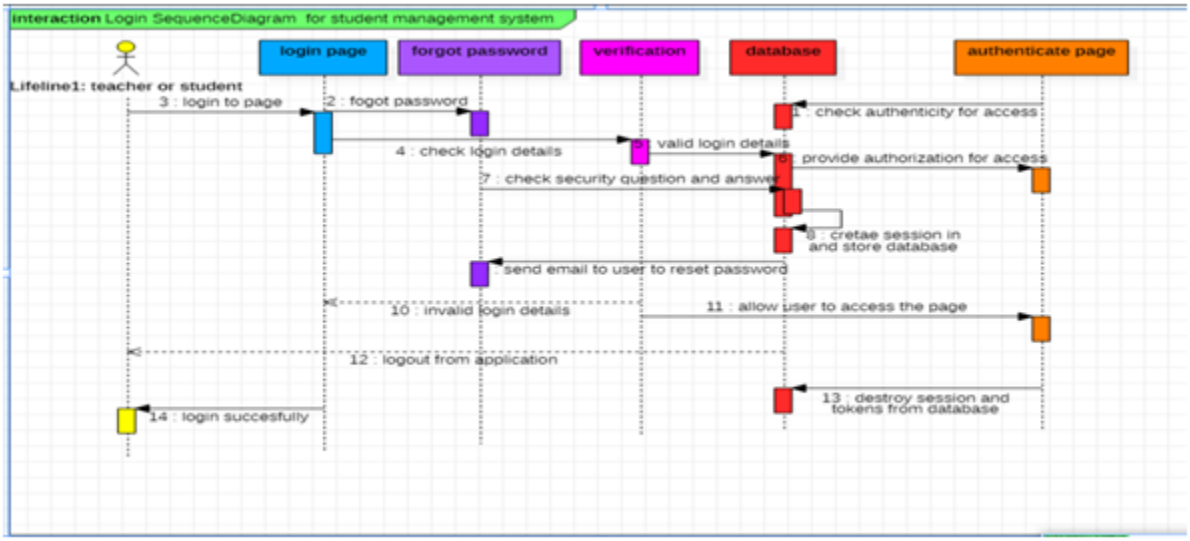
For Student:

Login, database, student homepage, timetable, report

For Teacher:

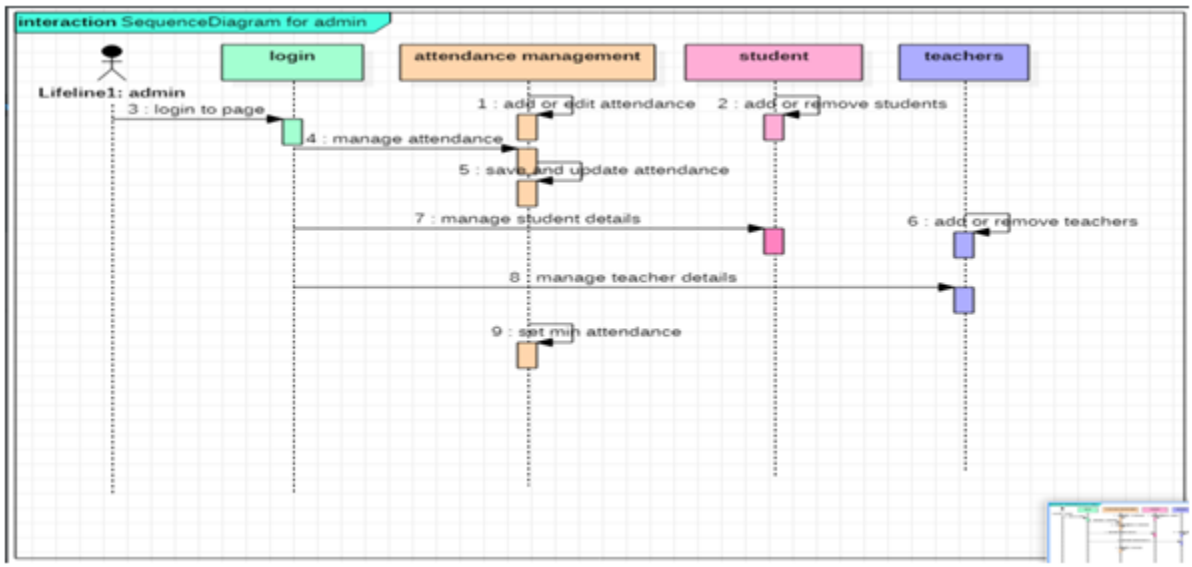
Teacher, interface, report, database

For login validity:



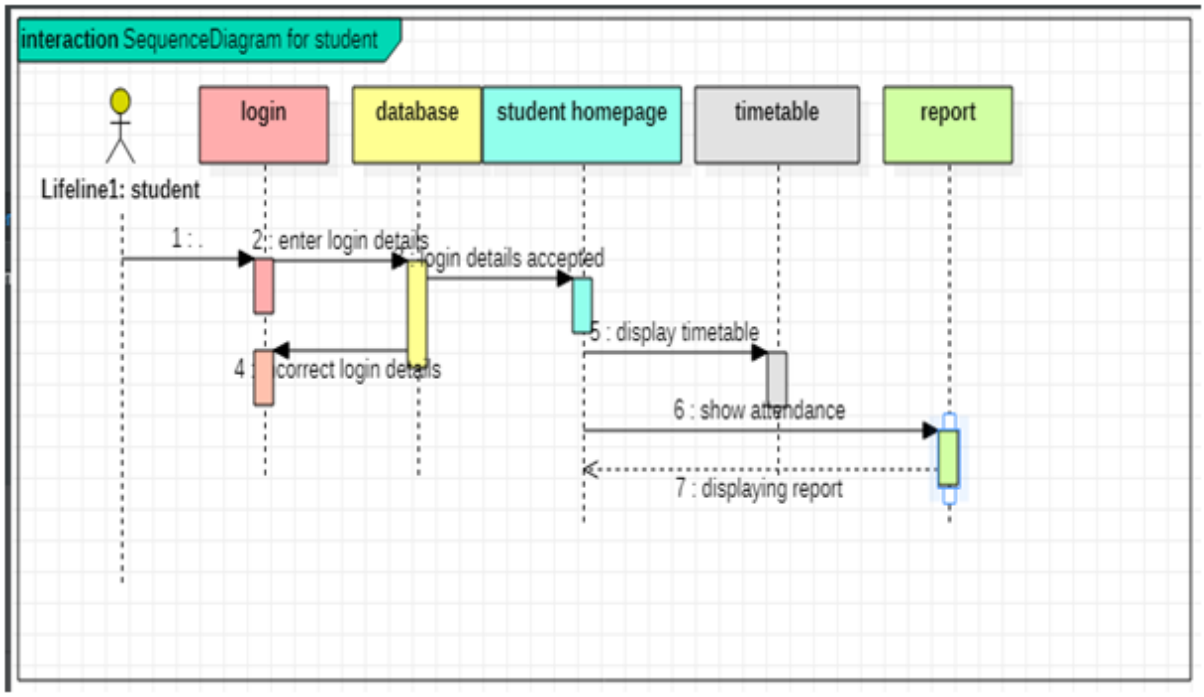
This is a sequence diagram for the login part and the teacher with the necessary information required such as the login page, verification, database, and operations such as verifying login details, checking authenticity for access.

For Administrator:



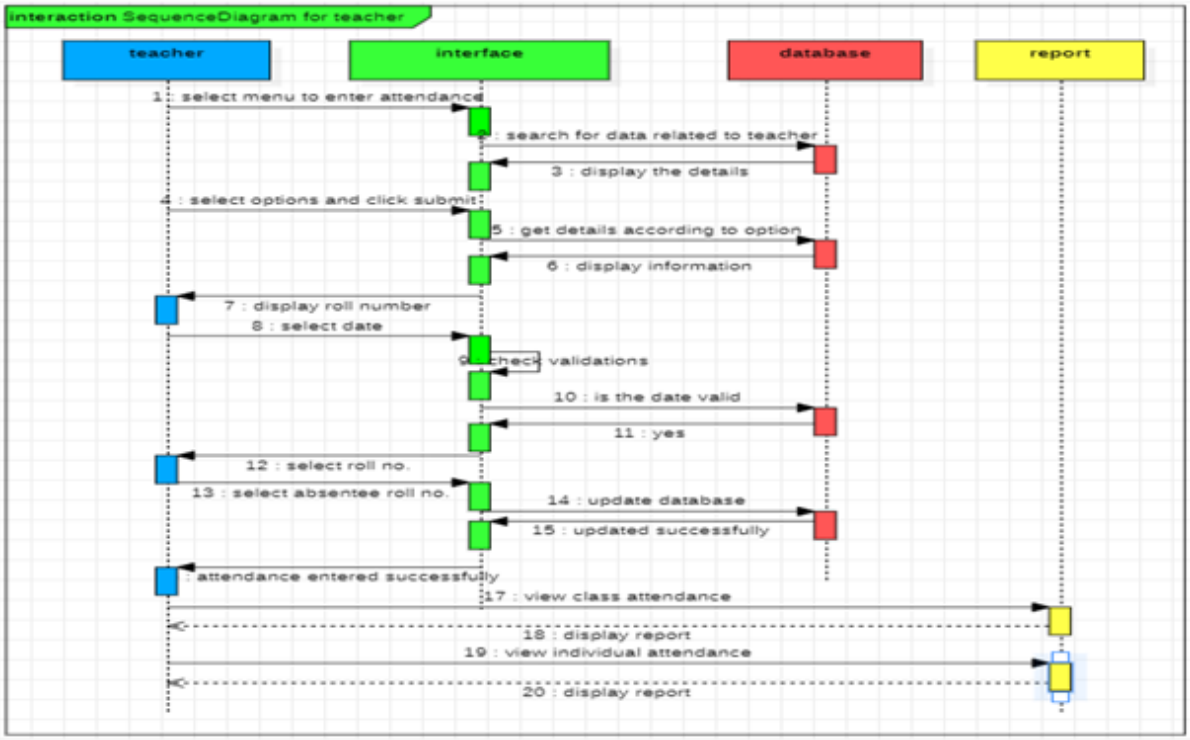
This is a sequence diagram for the admin which takes care of the whole college activities such as employing teachers and admissions of students. The necessary operations are done such as attendance management of students and setting min attendance for students and adding or removing teachers and students alike.

For Student:



This is a sequence diagram for students who are able to check their credentials through their login page and accept it in the student homepage database, viewing their timetable and getting the report of their attendance per course handled by the teacher.

For Teachers:



Result:

Thus, the architecture and design of the system were documented successfully.

EXPERIMENT-7

Aim

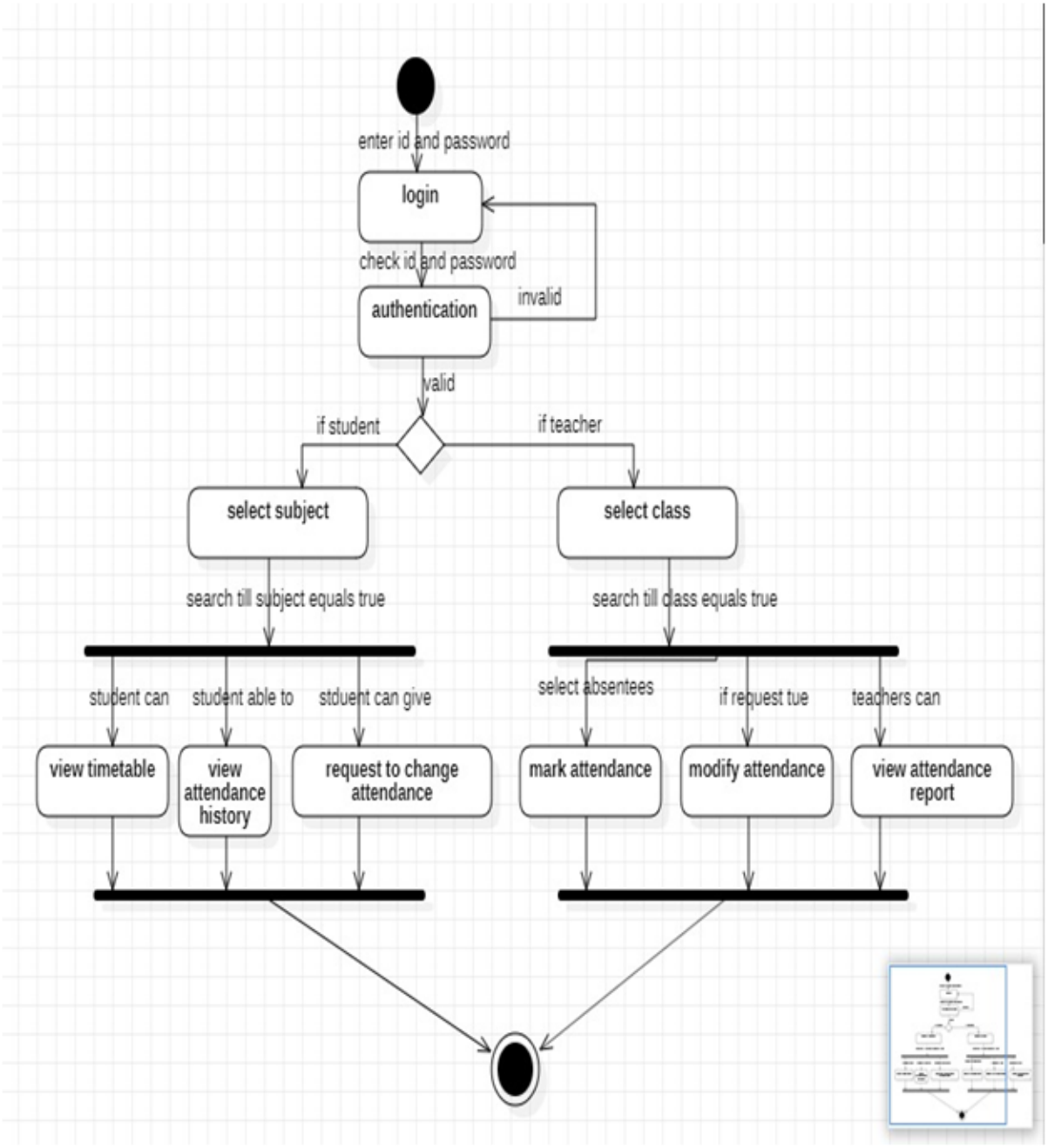
To Design State, Collaboration, Deployment Diagram, Sample Frontend Design (UI/UX) for the project.

State Diagram with Description

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. We prefer to model the states with three or more states.

USES OF STATE DIAGRAM

- We use it to state the events responsible for the change in state (we do not show what processes cause those events).
- We use it to model the dynamic behavior of the system.
- To understand the reaction of objects/classes to internal or external stimuli.



This is the State-chart diagram for our system where the student and teacher can do different tasks hence it avoids any unwanted overlaps. For the teacher after verified login, he/she can select class and mark attendance for the absentees as well as people actually present and modify and view attendance reports. The student can only see the timetable per day order and view attendance and request to change attendance if there is any fault and after finishing the work, they log out of the system.

Collaboration Diagram with Description

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

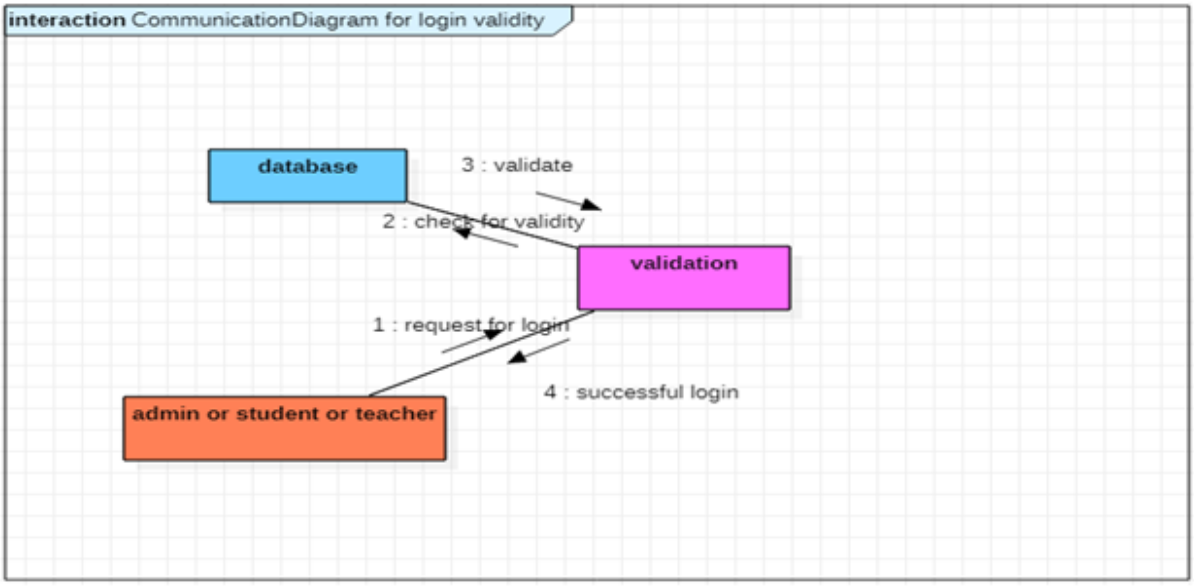
There is a collaboration diagram for the validity of the details given by the admin or student or teacher alike. There is a database which checks for the given login details and if it gets validated then the person can get access to the necessary information.

There are separate collaboration diagrams for Teacher, Student and Admin alike. For Teachers, A database is maintained and their login credentials are uploaded in that. If they gain access through necessary verification, then they are able to mark and update attendance for the students. After getting logged in through the necessary verifications, they can view their attendance and get the eligibility report for the same.

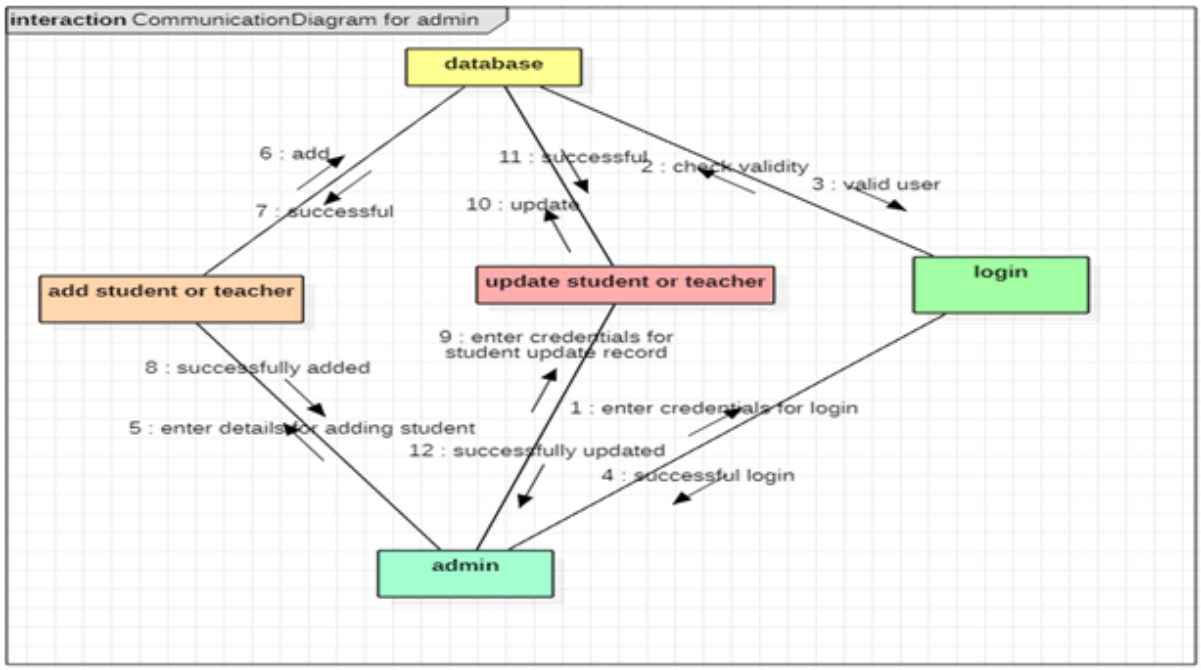
For the Admin, There is a database of their necessary login details. After getting verified from the database, they can be able to access the teacher's record and the student's attendance update and adding or removing the teachers and students alike.

Given below are the collaboration diagrams respectively:

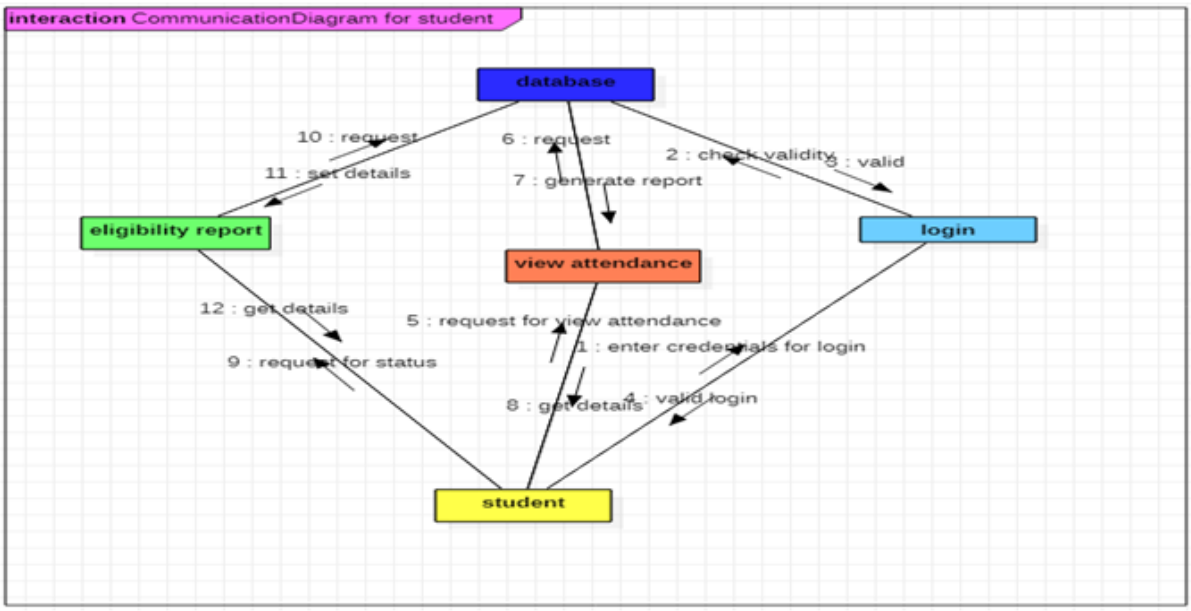
For login validity:



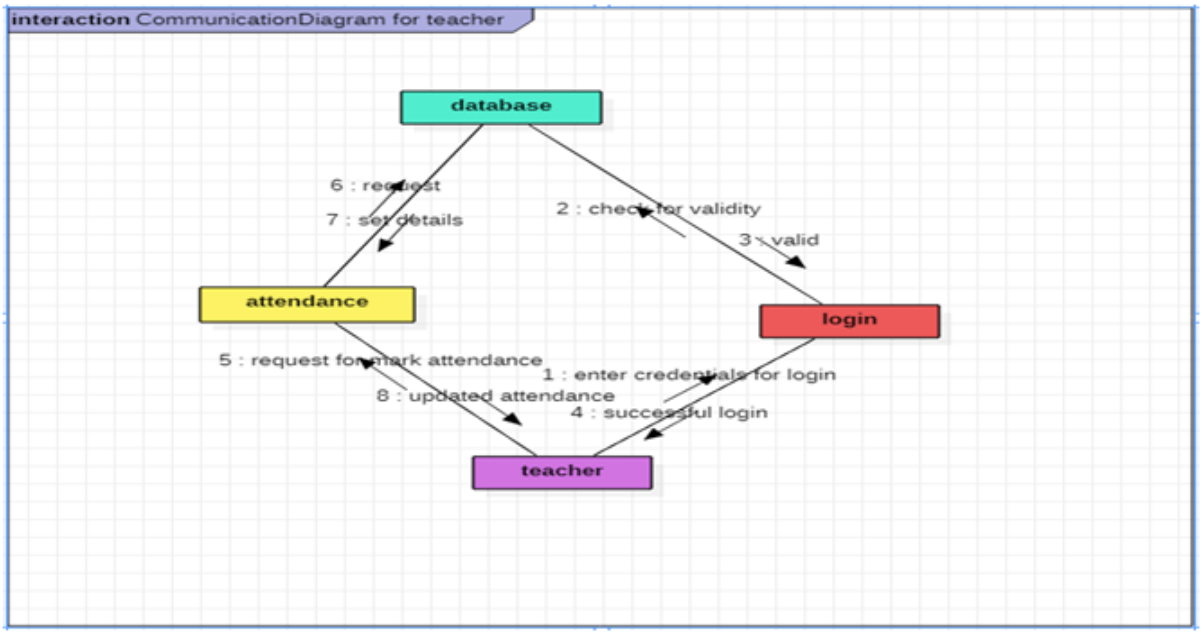
For Admin:



For Student:



For teacher:



Deployment Diagram with Description

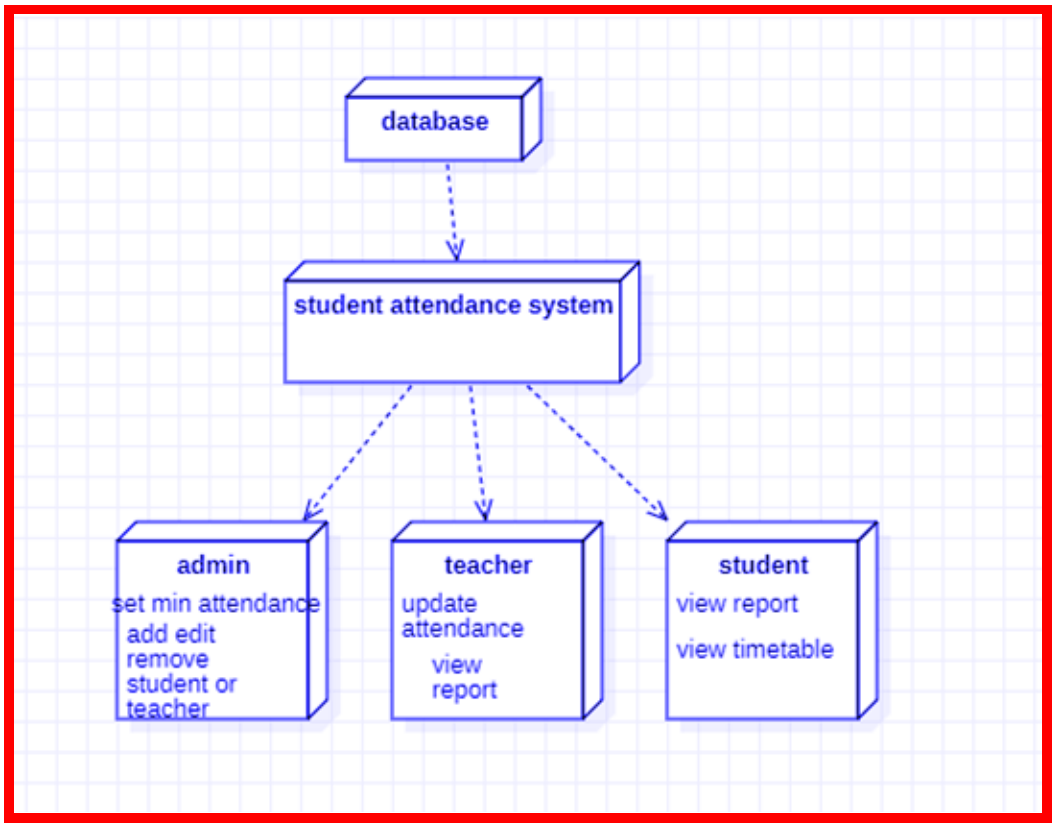
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

PURPOSE

The purpose of deployment diagrams can be described as −

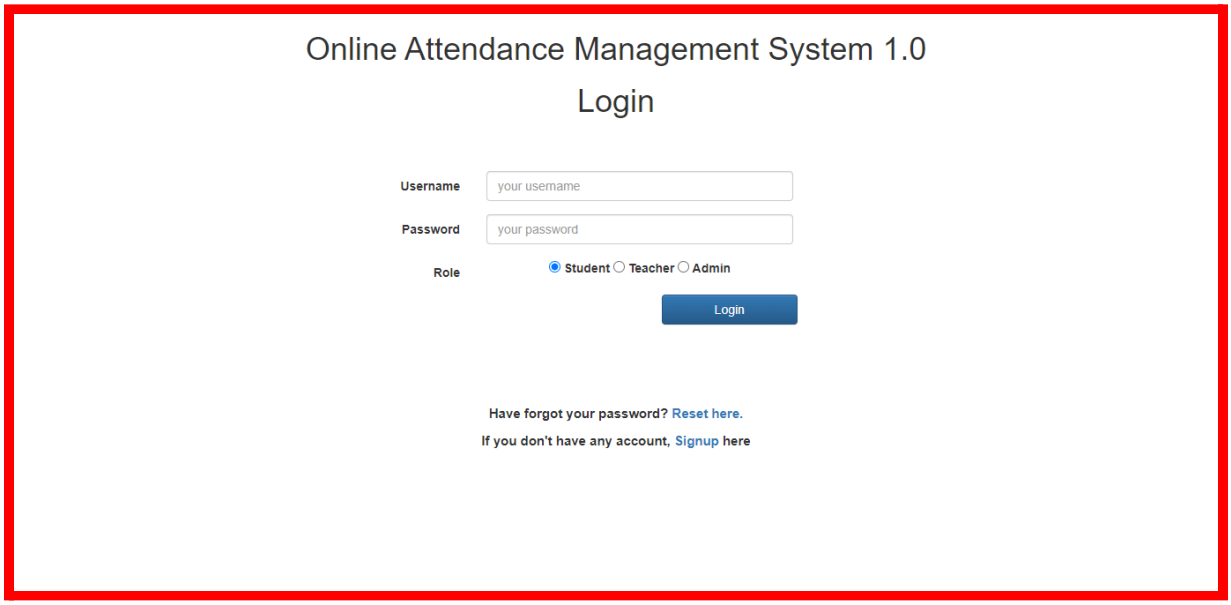
- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.



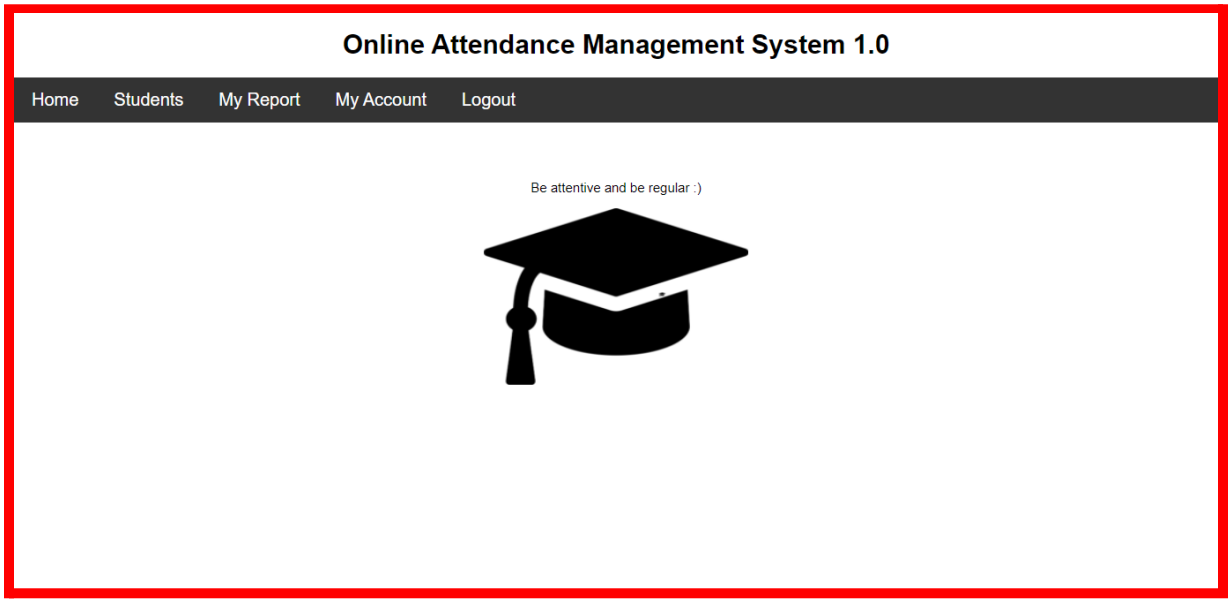
In this deployment diagram, first is the database where the student details are noted and updated regularly. For the student attendance system, the admin node contains the following operations such as set minimum attendance, add or remove student and teacher respectively. In the teacher node, they can update the attendance of the student and view the report whereas, in the student node, they can view their individual report after proper updating and their daily timetable respectively.

Sample Frontend design

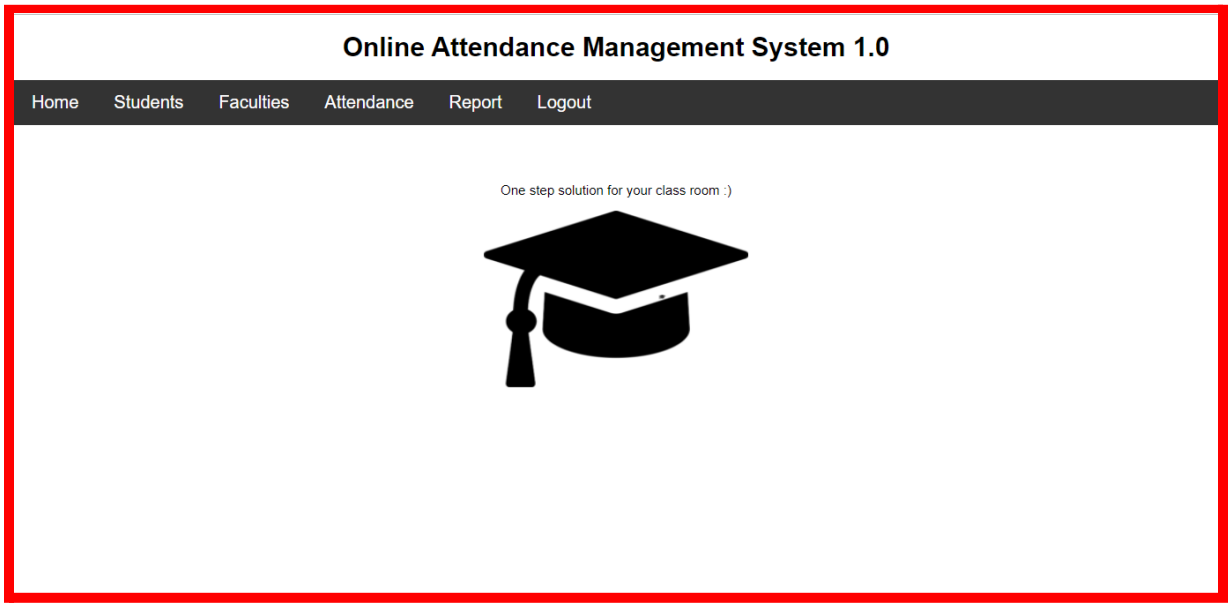
Main Page/Login Page



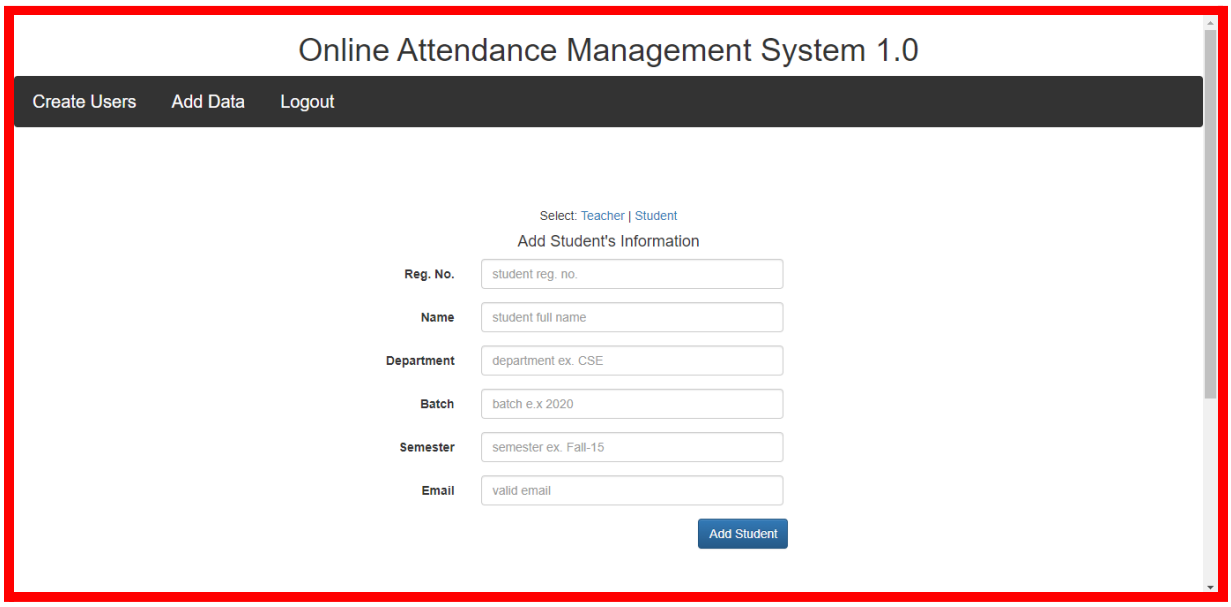
Student Page



Faculty Page



Admin Page



Result:

Thus, above mentioned designs of the system were documented successfully.

EXPERIMENT-8

Aim

To describe modules and implement Module1.

Code of Module 1

index.php

```
<?php

if(isset($_POST['login']))
{
    //start of try block

    try{

        //checking empty fields
        if(empty($_POST['username'])){
            throw new Exception("Username is required!");

        }
        if(empty($_POST['password'])){
            throw new Exception("Password is required!");

        }
        //establishing connection with db and things
        include ('connect.php');

        //checking login info into database
        $row=0;
        $result=mysql_query("select * from admininfo where
username='$_POST[username]' and password='$_POST[password]' and
type='$_POST[type]'");

        $row=mysql_num_rows($result);

        if($row>0 && $_POST["type"] == 'teacher'){
            session_start();
```

```
            $_SESSION['name']="oasis";
            header('location: teacher/index.php');
        }

        else if($row>0 &&  $_POST["type"] == 'student'){
            session_start();
            $_SESSION['name']="oasis";
            header('location: student/index.php');
        }

        else if($row>0 && $_POST["type"] == 'admin'){
            session_start();
            $_SESSION['name']="oasis";
            header('location: admin/index.php');
        }

        else{
            throw new Exception("Username,Password or Role is wrong, try
again!");

            header('location: login.php');
        }
    }

    //end of try block
    catch(Exception $e){
        $error_msg=$e->getMessage();
    }
    //end of try-catch
}

?>

<!DOCTYPE html>
<html>
<head>

    <title>Online Attendance Management System</title>
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <!-- Latest compiled and minified CSS -->
```

```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

    <!-- Optional theme -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

    <link rel="stylesheet" href="styles.css" >

    <!-- Latest compiled and minified JavaScript -->
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>

</head>

<body>
    <center>

<header>

  <h1>Online Attendance Management System 1.0</h1>

</header>

<h1>Login</h1>

<?php
//printing error message
if(isset($error_msg))
{
    echo $error_msg;
}
?>
<div class="content">
    <div class="row">
```

```
        <form method="post" class="form-horizontal col-md-6
col-md-offset-3">
            <div class="form-group">
                <label for="input1" class="col-sm-3
control-label">Username</label>
                <div class="col-sm-7">
                  <input type="text" name="username"  class="form-control"
id="input1" placeholder="your username" />
                </div>
            </div>

            <div class="form-group">
                <label for="input1" class="col-sm-3
control-label">Password</label>
                <div class="col-sm-7">
                  <input type="password" name="password"
class="form-control" id="input1" placeholder="your password" />
                </div>
            </div>


            <div class="form-group" class="radio">
            <label for="input1" class="col-sm-3
control-label">Role</label>
            <div class="col-sm-7">
              <label>
                <input type="radio" name="type" id="optionsRadios1"
value="student" checked> Student
              </label>
                  <label>
                <input type="radio" name="type" id="optionsRadios1"
value="teacher"> Teacher
              </label>
              <label>
                <input type="radio" name="type" id="optionsRadios1"
value="admin"> Admin
              </label>
            </div>
            </div>
```

```
            <input type="submit" class="btn btn-primary col-md-3
col-md-offset-7" value="Login" name="login" />
        </form>
    </div>
</div>



<br><br>
<p><strong>Have forgot your password? <a href="reset.php">Reset
here.</a></strong></p>
<p><strong>If you don't have any account, <a href="signup.php">Signup</a>
here</strong></p>

</center>
</body>
</html>
```

signup

```
<?php

include('connect.php');

  try{

      if(isset($_POST['signup'])){

        if(empty($_POST['email'])){
          throw new Exception("Email cann't be empty.");
        }

        if(empty($_POST['uname'])){
           throw new Exception("Username cann't be empty.");
        }
```

```
        if(empty($_POST['pass'])){
            throw new Exception("Password cann't be empty.");
        }

        if(empty($_POST['fname'])){
            throw new Exception("Username cann't be empty.");
        }
        if(empty($_POST['phone'])){
            throw new Exception("Username cann't be empty.");
        }
        if(empty($_POST['type'])){
            throw new Exception("Username cann't be empty.");
        }

        $result = mysql_query("insert into
admininfo(username,password,email,fname,phone,type)
values('$_POST[uname]','$_POST[pass]','$_POST[email]','$_POST[fname]','$_P
OST[phone]','$_POST[type]')");
        $success_msg="Signup Successfully!";


  }
}
  catch(Exception $e){
    $error_msg =$e->getMessage();
  }

?>

<!DOCTYPE html>
<html lang="en">
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="css/main.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >
```

```
  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
</head>
<body>

<header>

  <h1>Online Attendance Management System 1.0</h1>

</header>
<center>
<h1>Signup</h1>
<div class="content">

  <div class="row">
    <?php
    if(isset($success_msg)) echo $success_msg;
    if(isset($error_msg)) echo $error_msg;
     ?>
    <!-- Old version started -->
    <!--<form action="" method="post">

      <table>

        <tr>
          <td>Email</td>
          <td><input type="text" name="email"></td>
        </tr>
        <tr>
          <td>Username</td>
```

```
        <td><input type="text" name="uname"></td>

      </tr>
      <tr>
        <td>Password</td>
        <td><input type="Password" name="pass"></td>
      </tr>

      <tr>
        <td>Full Name</td>
        <td><input type="text" name="fname"></td>
      </tr>

      <tr>
        <td>Phone Number</td>
        <td><input type="text" name="phone"></td>
      </tr>

      <tr>
        <td>Type</td>
        <td>      <select name="type">
      <option name="teacher" value="teacher">Teacher</option>
      <option name="student" value="student">Student</option>
    </select></td>
      </tr>

      <tr><td><br></td></tr>
      <tr>
        <td></td>
        <td><input type="submit" name="signup" value="Signup"></td>
      </tr>

    </table>
  </form>--><!-- Old version ended -->

  <form method="post" class="form-horizontal col-md-6 col-md-offset-3">

    <div class="form-group">
        <label for="input1" class="col-sm-3 control-label">Email</label>
        <div class="col-sm-7">
```

```
            <input type="text" name="email"  class="form-control"
id="input1" placeholder="your email" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3
control-label">Username</label>
          <div class="col-sm-7">
            <input type="text" name="uname"  class="form-control"
id="input1" placeholder="choose username" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3
control-label">Password</label>
          <div class="col-sm-7">
            <input type="password" name="pass"  class="form-control"
id="input1" placeholder="choose a strong password" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Full
Name</label>
          <div class="col-sm-7">
            <input type="text" name="fname"  class="form-control"
id="input1" placeholder="your full name" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Phone
Number</label>
          <div class="col-sm-7">
            <input type="text" name="phone"  class="form-control"
id="input1" placeholder="your phone number" />
          </div>
      </div>
```

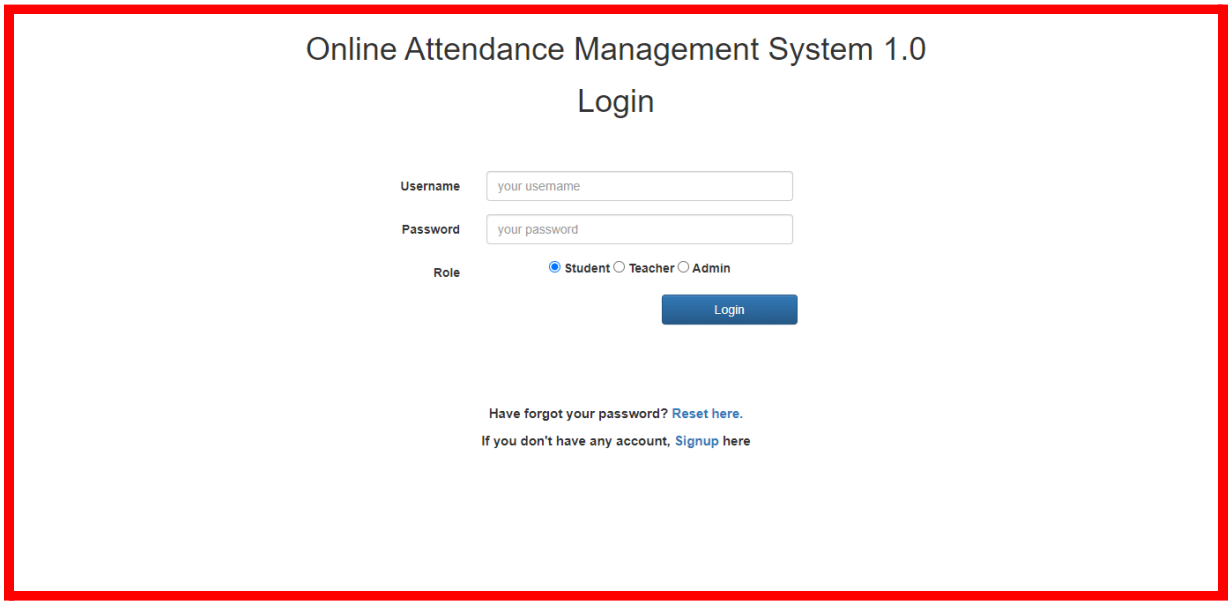
```
      <div class="form-group" class="radio">
      <label for="input1" class="col-sm-3 control-label">Role</label>
      <div class="col-sm-7">
        <label>
          <input type="radio" name="type" id="optionsRadios1"
value="student" checked> Student
        </label>
           <label>
          <input type="radio" name="type" id="optionsRadios1"
value="teacher"> Teacher
        </label>
        <!-- <label>
          <input type="radio" name="type" id="optionsRadios1"
value="admin"> Admin
        </label> -->
      </div>
      </div>

      <input type="submit" class="btn btn-primary col-md-2
col-md-offset-8" value="Signup" name="signup" />
    </form>
  </div>
    <br>
    <p><strong>Already have an account? <a href="index.php">Login</a>
here.</strong></p>

</div>

</center>

</body>
</html>
```

connect.php

```
<?php
```

```
mysql_connect('localhost','root','') or die('Cannot connect to server');
mysql_select_db('attsystem') or die ('Cannot found database');

?>
```

Result of Module 1



Result:

Thus, modules are described, Module 1 was implemented and documented successfully.

EXPERIMENT-9

Aim

To implement Module 2 of the project and display the output of the module with new requirements may be assimilated.

Code of Module 2

student

index.php

```
<?php

ob_start();
session_start();

if($_SESSION['name']!='oasis')
{
  header('location: ../index.php');
}
?>

<!DOCTYPE html>
<html lang="en">

<!-- head started -->
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="../css/main.css">

</head>
<!-- head ended -->

<!-- body started -->
<body>

<!-- Menus started-->
<header>
```

```
  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
  <a href="report.php">My Report</a>
  <a href="account.php">My Account</a>
  <a href="../logout.php">Logout</a>

</div>

</header>
<!-- Menus ended -->

<center>

<!-- Content, Tables, Forms, Texts, Images started -->
<div class="row">
    <div class="content">
      <p>Be attentive and be regular :)</p>
    <img src="../img/tcr.png" height="200px" width="300px" />

  </div>

</div>
<!-- Contents, Tables, Forms, Images ended -->

</center>

</body>
<!-- Body ended  -->

</html>
```

students.php

```
<?php

ob_start();
session_start();

if($_SESSION['name']!='oasis')
```

```
{
  header('location: login.php');
}
?>
<?php include('connect.php');?>

<!DOCTYPE html>
<html lang="en">
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="../css/main.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>


</head>
<body>

<header>

  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
```

```
  <a href="report.php">My Report</a>
  <a href="account.php">My Account</a>
  <a href="../logout.php">Logout</a>

</div>

</header>

<center>

<div class="row">

  <div class="content">
    <h3>Student List</h3>
    <br>

   <form method="post" action="" class="form-horizontal col-md-6
col-md-offset-3">
      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Batch</label>
            <div class="col-sm-7">
                <input type="text" name="sr_batch"  class="form-control"
id="input1" placeholder="Only 2020" />

            </div>

      </div>
      <input type="submit" class="btn btn-primary col-md-3
col-md-offset-7" value="Go!" name="sr_btn" />

   </form>

   <div class="content"></div>
    <table class="table table-striped">

      <thead>
      <tr>
        <th scope="col">Registration No.</th>
        <th scope="col">Name</th>
        <th scope="col">Department</th>
```

```
        <th scope="col">Batch</th>
        <th scope="col">Semester</th>
        <th scope="col">Email</th>
      </tr>
      </thead>
   <?php

    if(isset($_POST['sr_btn'])){

     $srbatch = 2020;
     $i=0;

     $all_query = mysql_query("select * from students where
students.st_batch = '$srbatch' order by st_id asc");

     while ($data = mysql_fetch_array($all_query)) {
       $i++;

     ?>

     <tr>
       <td><?php echo $data['st_id']; ?></td>
       <td><?php echo $data['st_name']; ?></td>
       <td><?php echo $data['st_dept']; ?></td>
       <td><?php echo $data['st_batch']; ?></td>
       <td><?php echo $data['st_sem']; ?></td>
       <td><?php echo $data['st_email']; ?></td>
     </tr>

     <?php
          }
              }
      ?>
    </table>

  </div>

</div>

</center>
```

```
</body>
</html>
```

accounts.php

```
  <?php

  ob_start();
  session_start();

  //checking if the session is valid
  if($_SESSION['name']!='oasis')
  {
    header('location: ../login.php');
  }
  ?>

  <?php include('connect.php');?>


<?php
try{

         //checking form data and empty fields
          if(isset($_POST['done'])){

          if (empty($_POST['name'])) {
            throw new Exception("Name cannot be empty");

          }
              if (empty($_POST['dept'])) {

                throw new Exception("Department cannot be empty");

              }
                  if(empty($_POST['batch']))
                  {
```

```
                    throw new Exception("Batch cannot be empty");

                }
                    if(empty($_POST['email']))
                    {
                      throw new Exception("Email cannot be empty");

                    }

  //initializing the student id
  $sid = $_POST['id'];

  //udating students information to database table "students"
  $result = mysql_query("update students set
st_name='$_POST[name]',st_dept='$_POST[dept]',st_batch='$_POST[batch]',st_
sem='$_POST[semester]', st_email = '$_POST[email]' where st_id='$sid'");
  $success_msg = 'Updated  successfully';

  }

}
catch(Exception $e){

  $error_msg =$e->getMessage();
}


?>



<!DOCTYPE html>
<html lang="en">

<!-- head started -->
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="../css/main.css">
```

```
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>


</head>
<!-- head ended -->


<!-- body started -->
<body>

<!-- Menus started-->
<header>

  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
  <a href="report.php">My Report</a>
  <a href="account.php">My Account</a>
  <a href="../logout.php">Logout</a>

</div>

</header>
<!-- Menus ended -->
```

```
<!-- Content, Tables, Forms, Texts, Images started -->
<center>

<div class="row">
    <div class="content">

          <h3>Update Account</h3>
          <br>

          <!-- Error or Success Message printint started --><p>
      <?php

          if(isset($success_msg))
          {
            echo $success_msg;
          }
          if(isset($error_msg))
          {
            echo $error_msg;
          }

        ?>
          </p><!-- Error or Success Message printint ended -->

          <br>

          <form method="post" action="" class="form-horizontal col-md-6
col-md-offset-3">
            <div class="form-group">
                <label for="input1" class="col-sm-3
control-label">Registration No.</label>
                <div class="col-sm-7">
                  <input type="text" name="sr_id"  class="form-control"
id="input1" placeholder="enter your reg. no. to continue" />
                </div>
            </div>
            <input type="submit" class="btn btn-primary col-md-3
col-md-offset-7" value="Go!" name="sr_btn" />
          </form>
```

```
          <div class="content"></div>


      <?php

      if(isset($_POST['sr_btn'])){

      //initializing student ID from form data
       $sr_id = $_POST['sr_id'];

       $i=0;

       //searching students information respected to the particular ID
       $all_query = mysql_query("select * from students where
students.st_id='$sr_id'");
       while ($data = mysql_fetch_array($all_query)) {
         $i++;

       ?>
<form action="" method="post" class="form-horizontal col-md-6
col-md-offset-3">
   <table class="table table-striped">

          <tr>
            <td>Registration No.:</td>
            <td><?php echo $data['st_id']; ?></td>
          </tr>

          <tr>
              <td>Student's Name:</td>
              <td><input type="text" name="name" value="<?php echo
$data['st_name']; ?>"></input></td>
          </tr>

          <tr>
              <td>Department:</td>
              <td><input type="text" name="dept" value="<?php echo
$data['st_dept']; ?>"></input></td>
          </tr>
```

```
          <tr>
              <td>Batch:</td>
              <td><input type="text" name="batch" value="<?php echo
$data['st_batch']; ?>"></input></td>
          </tr>

          <tr>
              <td>Semester:</td>
              <td><input type="text" name="semester" value="<?php echo
$data['st_sem']; ?>"></input></td>
          </tr>

          <tr>
              <td>Email:</td>
              <td><input type="text" name="email" value="<?php echo
$data['st_email']; ?>"></input></td>
          </tr>
          <input type="hidden" name="id" value="<?php echo $sr_id; ?>">

          <tr><td></td></tr>
          <tr>
                <td></td>
                <td><input type="submit" class="btn btn-primary col-md-3
col-md-offset-7" value="Update" name="done" /></td>

          </tr>

    </table>
</form>
     <?php
   }
     }
     ?>


      </div>

  </div>

  </center>
```

```
<!-- Contents, Tables, Forms, Images ended -->

</body>
<!-- Menus ended -->

</html>
```

teacher

index.php

```
<?php

ob_start();
session_start();

if($_SESSION['name']!='oasis')
{
  header('location: ../index.php');
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="../css/main.css">

</head>
<body>

<header>

  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
  <a href="teachers.php">Faculties</a>
```

```
  <a href="attendance.php">Attendance</a>
  <a href="report.php">Report</a>
  <a href="../logout.php">Logout</a>

</div>

</header>

<center>

<div class="row">
    <div class="content">
      <p>One step solution for your class room :)</p>
    <img src="../img/tcr.png" height="200px" width="300px" />

  </div>

</div>

</center>

</body>
</html>
```

teacher.php

```
<?php

ob_start();
session_start();

if($_SESSION['name']!='oasis')
{
  header('location: login.php');
}
?>
<?php include('connect.php');?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="../css/main.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
</style>

</head>
<body>

<header>

  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
  <a href="teachers.php">Faculties</a>
  <a href="attendance.php">Attendance</a>
  <a href="report.php">Report</a>
  <a href="../logout.php">Logout</a>

</div>
```

```
</header>

<center>

<div class="row">

  <div class="content">
    <h3>Teacher List</h3>

    <table class="table table=stripped">
        <thead>
          <tr>
            <th scope="col">Teacher ID</th>
            <th scope="col">Name</th>
            <th scope="col">Department</th>
            <th scope="col">Email</th>
            <th scope="col">Course</th>
          </tr>
        </thead>

      <?php

        $i=0;
        $tcr_query = mysql_query("select * from teachers order by tc_id
asc");
        while($tcr_data = mysql_fetch_array($tcr_query)){
          $i++;

        ?>
          <tbody>
              <tr>
                <td><?php echo $tcr_data['tc_id']; ?></td>
                <td><?php echo $tcr_data['tc_name']; ?></td>
                <td><?php echo $tcr_data['tc_dept']; ?></td>
                <td><?php echo $tcr_data['tc_email']; ?></td>
                <td><?php echo $tcr_data['tc_course']; ?></td>
              </tr>
          </tbody>
```

```
          <?php } ?>

    </table>

  </div>

</div>

</center>

</body>
</html>
```

attendance.php

```
<?php

ob_start();
session_start();

if($_SESSION['name']!='oasis')
{
  header('location: login.php');
}
?>

<?php
    include('connect.php');
    try{

    if(isset($_POST['att'])){

      $course = $_POST['whichcourse'];

      foreach ($_POST['st_status'] as $i => $st_status) {

        $stat_id = $_POST['stat_id'][$i];
        $dp = date('Y-m-d');
        $course = $_POST['whichcourse'];
```

```
        $stat = mysql_query("insert into
attendance(stat_id,course,st_status,stat_date)
values('$stat_id','$course','$st_status','$dp')");

        $att_msg = "Attendance Recorded.";

      }



    }
  }
  catch(Execption $e){
    $error_msg = $e->$getMessage();
  }
 ?>

<!DOCTYPE html>
<html lang="en">
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="../css/main.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
```

```
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>


<style type="text/css">
  .status{
    font-size: 10px;
  }

</style>

</head>
<body>

<header>

  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
  <a href="teachers.php">Faculties</a>
  <a href="attendance.php">Attendance</a>
  <a href="report.php">Report</a>
  <a href="../logout.php">Logout</a>

</div>

</header>

<center>

<div class="row">

  <div class="content">
    <h3>Attendance of <?php echo date('Y-m-d'); ?></h3>
    <br>

    <center><p><?php if(isset($att_msg)) echo $att_msg;
if(isset($error_msg)) echo $error_msg; ?></p></center>
```

```
    <form action="" method="post" class="form-horizontal col-md-6
col-md-offset-3">

     <div class="form-group">

               <!-- <label>Select Batch</label>

                <select name="whichbatch" id="input1">
                      <option name="eight" value="38">38</option>
                      <option name="seven" value="37">37</option>
                </select>-->


                <label>Enter Batch</label>
                <input type="text" name="whichbatch" id="input2"
placeholder="Only 2020">
              </div>

     <input type="submit" class="btn btn-primary col-md-2 col-md-offset-5"
value="Show!" name="batch" />

    </form>

    <div class="content"></div>
    <form action="" method="post">

      <div class="form-group">

        <label >Select Subject</label>
              <select name="whichcourse" id="input1">
              <option  value="algo">Analysis of Algorithms</option>
         <option  value="algolab">Analysis of Algorithms Lab</option>
        <option  value="dbms">Database Management System</option>
        <option  value="dbmslab">Database Management System Lab</option>
        <option  value="weblab">Web Programming Lab</option>
        <option  value="os">Operating System</option>
        <option  value="oslab">Operating System Lab</option>
        <option  value="obm">Object Based Modeling</option>
        <option  value="softcomp">Soft Computing</option>
```

```
            </select>

      </div>

    <table class="table table-stripped">
      <thead>
        <tr>
          <th scope="col">Reg. No.</th>
          <th scope="col">Name</th>
          <th scope="col">Department</th>
          <th scope="col">Batch</th>
          <th scope="col">Semester</th>
          <th scope="col">Email</th>
          <th scope="col">Status</th>
        </tr>
      </thead>
   <?php

    if(isset($_POST['batch'])){

     $i=0;
     $radio = 0;
     $batch = 2020;
     $all_query = mysql_query("select * from students where
st_batch='$batch' order by st_id asc");

     while ($data = mysql_fetch_array($all_query)) {
       $i++;
     ?>
  <body>
     <tr>
       <td><?php echo $data['st_id']; ?> <input type="hidden"
name="stat_id[]" value="<?php echo $data['st_id']; ?>"> </td>
       <td><?php echo $data['st_name']; ?></td>
       <td><?php echo $data['st_dept']; ?></td>
       <td><?php echo $data['st_batch']; ?></td>
       <td><?php echo $data['st_sem']; ?></td>
       <td><?php echo $data['st_email']; ?></td>
       <td>
         <label>Present</label>
```

```
        <input type="radio" name="st_status[<?php echo $radio; ?>]"
value="Present" >
        <label>Absent </label>
        <input type="radio" name="st_status[<?php echo $radio; ?>]"
value="Absent" checked>
      </td>
    </tr>
  </body>

    <?php

       $radio++;
     }
}
     ?>
   </table>

   <center><br>
   <input type="submit" class="btn btn-primary col-md-2 col-md-offset-10"
value="Save!" name="att" />
  </center>

</form>
  </div>

</div>

</center>

</body>
</html>
```

admin

```
<?php

ob_start();
session_start();
```

```php
if($_SESSION['name']!='oasis')
{

  header('location: ../index.php');
}
?>

<?php

include('connect.php');

//data insertion
  try{

    //checking if the data comes from students form
    if(isset($_POST['std'])){

      //students data insertion to database table "students"
        $result = mysql_query("insert into
students(st_id,st_name,st_dept,st_batch,st_sem,st_email)
values('$_POST[st_id]','$_POST[st_name]','$_POST[st_dept]','$_POST[st_batc
h]','$_POST[st_sem]','$_POST[st_email]')");
        $success_msg = "Student added successfully.";

    }

        //checking if the data comes from teachers form
        if(isset($_POST['tcr'])){

          //teachers data insertion to the database table "teachers"
          $res = mysql_query("insert into
teachers(tc_id,tc_name,tc_dept,tc_email,tc_course)
values('$_POST[tc_id]','$_POST[tc_name]','$_POST[tc_dept]','$_POST[tc_emai
l]','$_POST[tc_course]')");
          $success_msg = "Teacher added successfully.";
    }

  }
  catch(Execption $e){
    $error_msg =$e->getMessage();
```

```
  }

 ?>



<!DOCTYPE html>
<html lang="en">
<!-- head started -->
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">

  <link rel="stylesheet" type="text/css" href="../css/main.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
<style type="text/css">

  .message{
    padding: 10px;
    font-size: 15px;
    font-style: bold;
    color: black;
  }
</style>
</head>
```

```
<!-- head ended -->

<!-- body started -->
<body>

    <!-- Menus started-->
    <header>

      <h1>Online Attendance Management System 1.0</h1>
      <div class="navbar">
      <a href="signup.php">Create Users</a>
      <a href="index.php">Add Data</a>
      <a href="../logout.php">Logout</a>

    </div>

    </header>
    <!-- Menus ended -->

<center>
<!-- Error or Success Message printint started -->
<div class="message">
        <?php if(isset($success_msg)) echo $success_msg;
if(isset($error_msg)) echo $error_msg; ?>
</div>
<!-- Error or Success Message printint ended -->

<!-- Content, Tables, Forms, Texts, Images started -->
<div class="content">

  <center> Select: <a href="#teacher">Teacher</a> | <a href="">Student</a>
<br></center>

  <div class="row" id="student">



      <form method="post" class="form-horizontal col-md-6
col-md-offset-3">
      <h4>Add Student's Information</h4>
```

```
      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Reg.
No.</label>
          <div class="col-sm-7">
            <input type="text" name="st_id"  class="form-control"
id="input1" placeholder="student reg. no." />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Name</label>
          <div class="col-sm-7">
            <input type="text" name="st_name"  class="form-control"
id="input1" placeholder="student full name" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3
control-label">Department</label>
          <div class="col-sm-7">
            <input type="text" name="st_dept"  class="form-control"
id="input1" placeholder="department ex. CSE" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Batch</label>
          <div class="col-sm-7">
            <input type="text" name="st_batch"  class="form-control"
id="input1" placeholder="batch e.x 2020" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3
control-label">Semester</label>
          <div class="col-sm-7">
            <input type="text" name="st_sem"  class="form-control"
id="input1" placeholder="semester ex. Fall-15" />
```

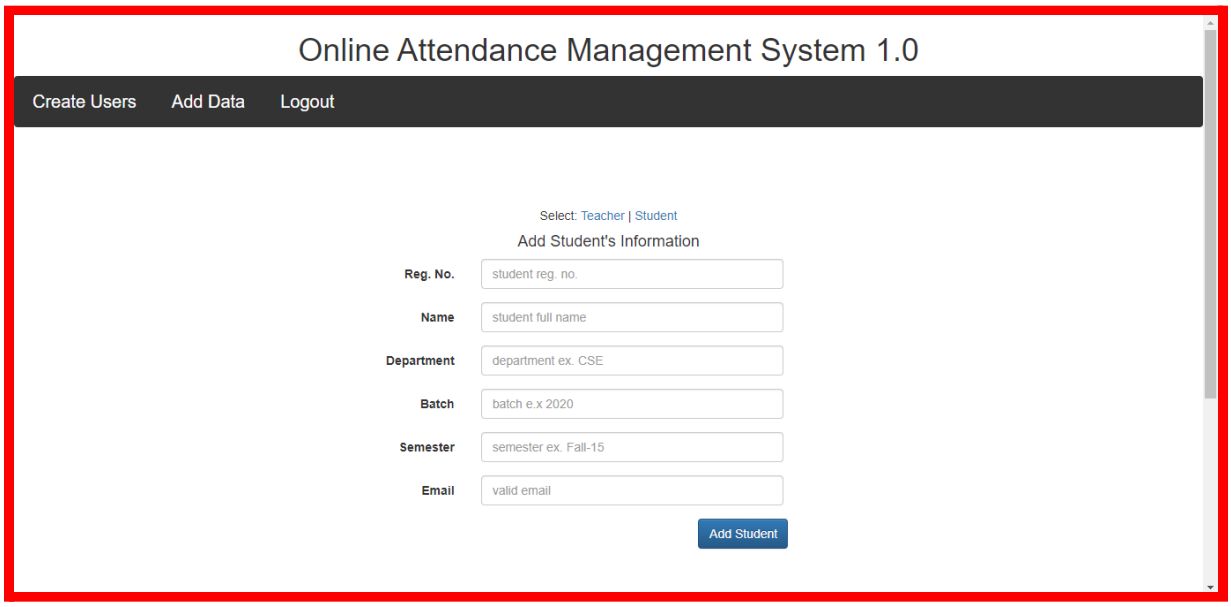
```
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Email</label>
          <div class="col-sm-7">
            <input type="email" name="st_email"  class="form-control"
id="input1" placeholder="valid email" />
          </div>
      </div>


      <input type="submit" class="btn btn-primary col-md-2
col-md-offset-8" value="Add Student" name="std" />
    </form>

  </div>
<br><br><br>
  <div class="rowtwo" id="teacher">


       <form method="post" class="form-horizontal col-md-6
col-md-offset-3">
        <h4>Add Teacher's Information</h4>
      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Teacher
ID</label>
          <div class="col-sm-7">
            <input type="text" name="tc_id"  class="form-control"
id="input1" placeholder="teacher's id" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Name</label>
          <div class="col-sm-7">
            <input type="text" name="tc_name"  class="form-control"
id="input1" placeholder="teacher full name" />
          </div>
      </div>
```

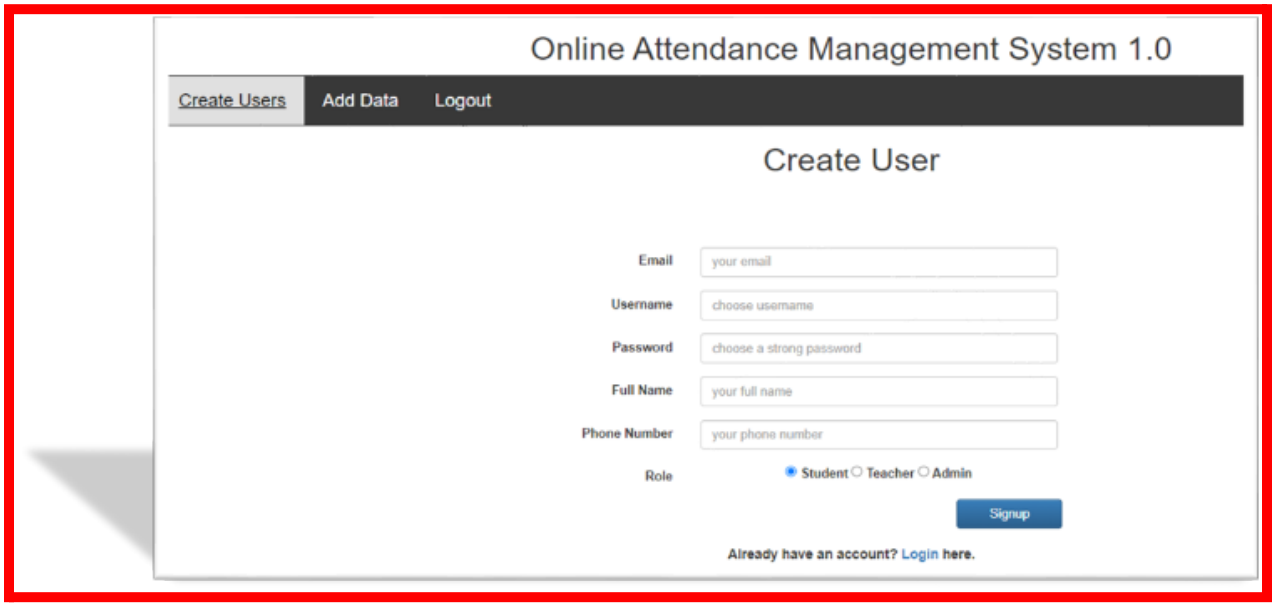
```
      <div class="form-group">
          <label for="input1" class="col-sm-3
control-label">Department</label>
          <div class="col-sm-7">
            <input type="text" name="tc_dept"  class="form-control"
id="input1" placeholder="department ex. CSE" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Email</label>
          <div class="col-sm-7">
            <input type="email" name="tc_email"  class="form-control"
id="input1" placeholder="valid email" />
          </div>
      </div>

      <div class="form-group">
          <label for="input1" class="col-sm-3 control-label">Subject
Name</label>
          <div class="col-sm-7">
            <input type="text" name="tc_course"  class="form-control"
id="input1" placeholder="subject ex. Software Engineering" />
          </div>
      </div>

      <input type="submit" class="btn btn-primary col-md-2
col-md-offset-8" value="Add Teacher" name="tcr" />
    </form>

  </div>


</div><br>
<!-- Contents, Tables, Forms, Images ended -->

</center>
</body>
<!-- Body ended  -->
```

```
</html>
```

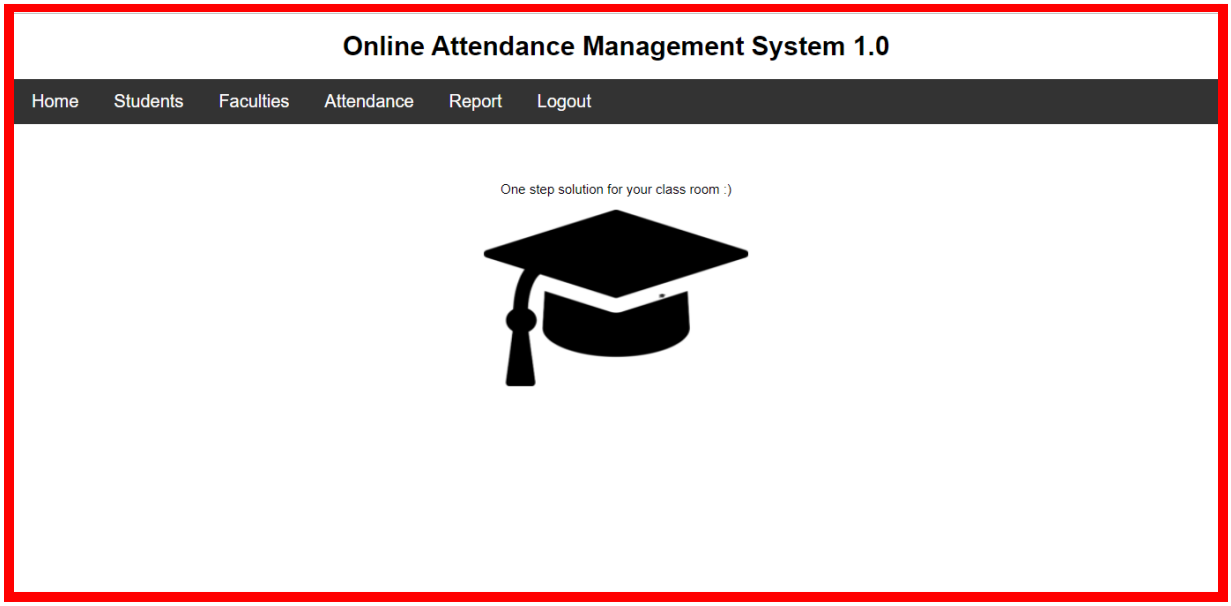
Result of Module 2

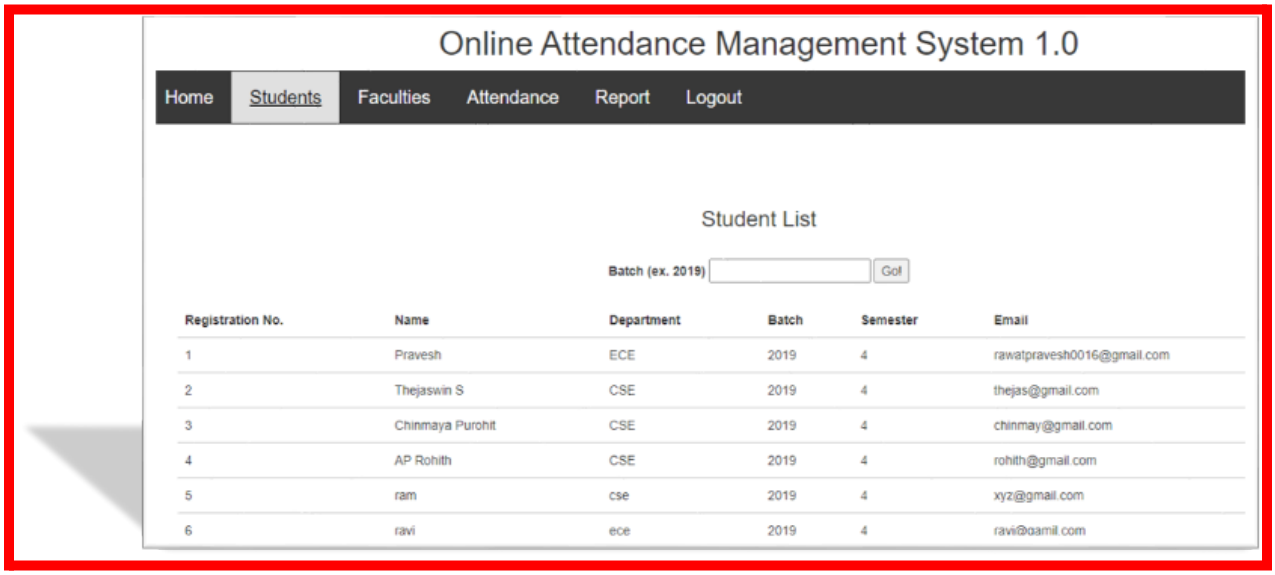
Admin

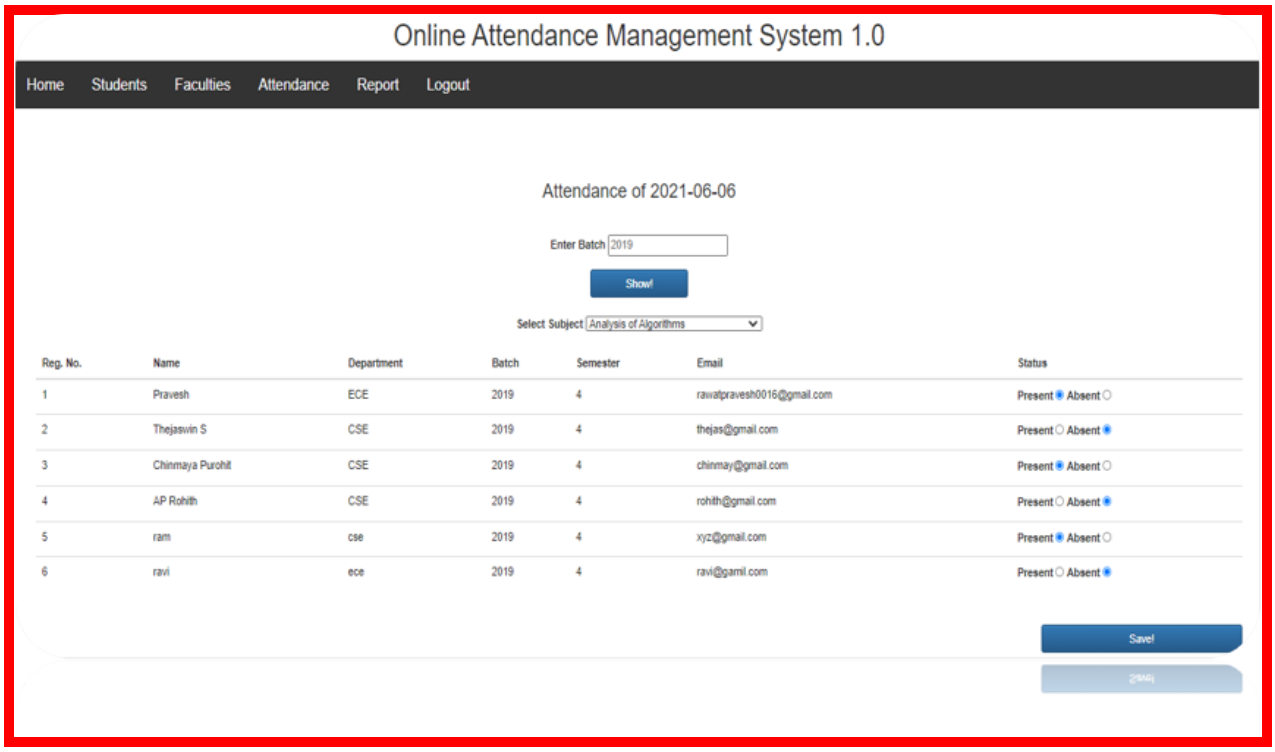




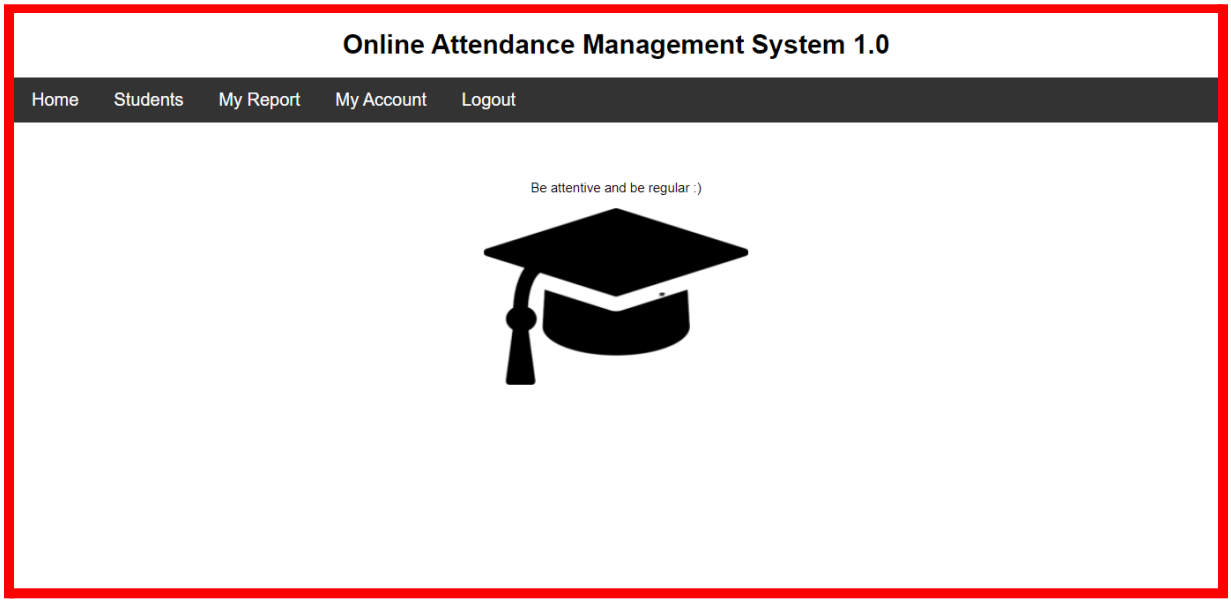
Teacher

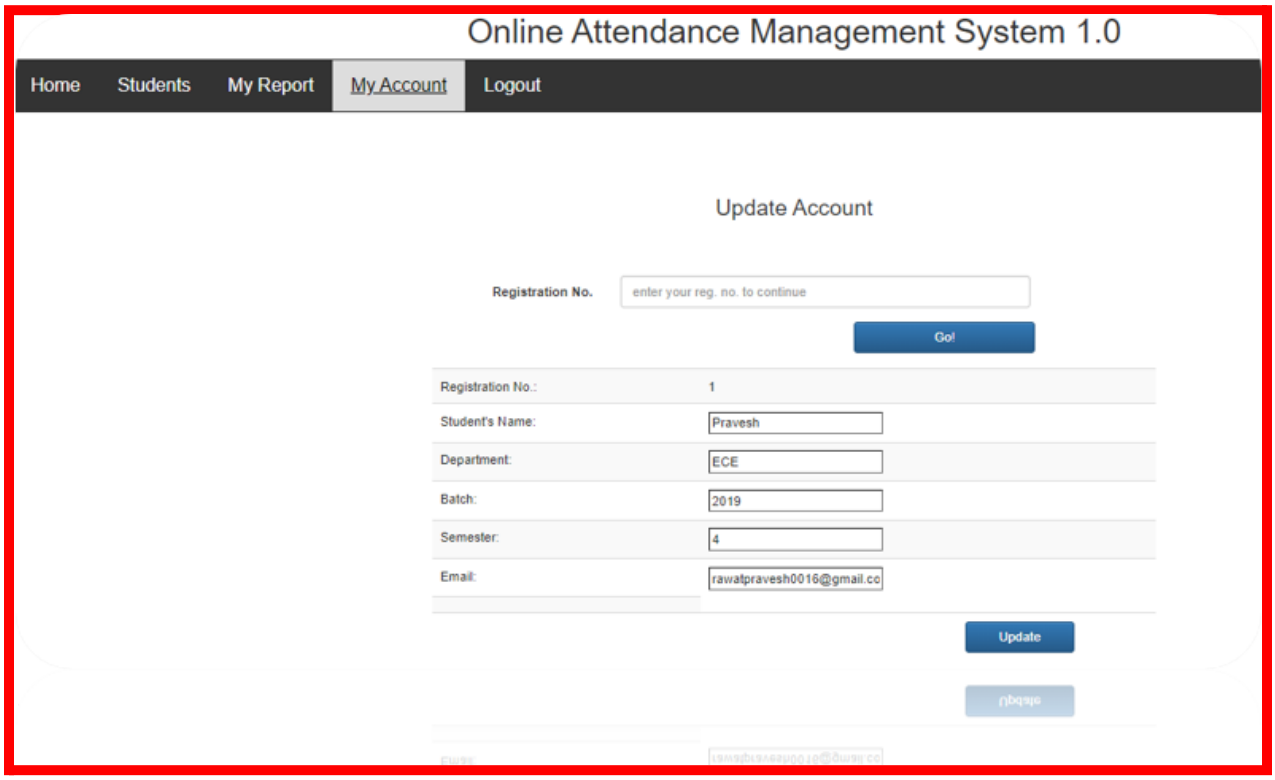

Online Attendance Management System 1.0
Home
Students
Faculties
Attendance
Report
Logout
One step solution for your class room :)


Online Attendance Management System 1.0
Home
Students
Faculties
Attendance
Report
Logout
Student List
Batch (ex. 2019)
Go!
Registration No.
Name
Department
Batch
Semester
Email
1
Pravesh
ECE
2019
4
rawatpravesh0016@gmail.com
2
Thejaswin S
CSE
2019
4
thejas@gmail.com
3
Chinmaya Purohit
CSE
2019
4
chinmay@gmail.com
4
AP Rohith
CSE
2019
4
rohith@gmail.com
5
ram
cse
2019
4
xyz@gmail.com
6
ravi
ece
2019
4
ravi@gamil.com


Online Attendance Management System 1.0
Home
Students
Faculties
Attendance
Report
Logout
Attendance of 2021-06-06
Enter Batch 2019
Show!
Select Subject Analysis of Algorithms
Reg. No. Name Department Batch Semester Email Status
1 Pravesh ECE 2019 4 rawatpravesh0016@gmail.com Present Absent
2 Thejaswin S CSE 2019 4 thejas@gmail.com Present Absent
3 Chinmaya Purohit CSE 2019 4 chinmay@gmail.com Present Absent
4 AP Rohith CSE 2019 4 rohith@gmail.com Present Absent
5 ram cse 2019 4 xyz@gmail.com Present Absent
6 ravi ece 2019 4 ravi@gamil.com Present Absent
Save!

Student


Online Attendance Management System 1.0
Home
Students
My Report
My Account
Logout
Be attentive and be regular :)



Result:

Thus, module2 was implemented and documented successfully.

EXPERIMENT-10

Aim

To implement Module 3 of the project and display the output of the module with solving New Issues.

Code of Module 3

logout.php

```php
<?php
session_start();
session_destroy();
header('location: index.php');
?>
```

attsystem.sql

```
-- phpMyAdmin SQL Dump
-- version 4.7.0
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Server version: 10.1.25-MariaDB
-- PHP Version: 5.6.31

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";




--
-- Database: `attsystem`
--

-- --------------------------------------------------------
```

```
--
-- Table structure for table `admininfo`
--

CREATE TABLE `admininfo` (
  `username` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL,
  `email` varchar(30) NOT NULL,
  `fname` varchar(20) NOT NULL,
  `phone` varchar(10) NOT NULL,
  `type` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admininfo`
--

INSERT INTO `admininfo` (`username`, `password`, `email`, `fname`,
`phone`, `type`) VALUES
('admin', 'admin', 'admin@gmail.com', 'admin', '2147483647', 'admin'),
('pravesh', 'pravesh', 'rawatpravesh0016@gmail.com', 'Pravesh Rawat',
'0992642003', 'student'),
('sumit', 'sumit', 'sumitbangar59@gmail.com', 'sumit bangar', '988766363',
'teacher');

-- --------------------------------------------------------

--
-- Table structure for table `attendance`
--

CREATE TABLE `attendance` (
  `stat_id` varchar(20) NOT NULL,
  `course` varchar(20) NOT NULL,
  `st_status` varchar(10) NOT NULL,
  `stat_date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```sql
-- Dumping data for table `attendance`
--

INSERT INTO `attendance` (`stat_id`, `course`, `st_status`, `stat_date`)
VALUES
('1', 'algo', 'Present', '2018-11-14'),
('2', 'algo', 'Present', '2018-11-13'),
('1', 'algo', 'Absent', '2018-11-13');

-- --------------------------------------------------------

--
-- Table structure for table `reports`
--

CREATE TABLE `reports` (
  `st_id` varchar(30) NOT NULL,
  `course` varchar(30) NOT NULL,
  `st_status` varchar(30) NOT NULL,
  `st_name` varchar(30) NOT NULL,
  `st_dept` varchar(30) NOT NULL,
  `st_batch` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `students`
--

CREATE TABLE `students` (
  `st_id` varchar(20) NOT NULL,
  `st_name` varchar(20) NOT NULL,
  `st_dept` varchar(20) NOT NULL,
  `st_batch` int(4) NOT NULL,
  `st_sem` int(11) NOT NULL,
  `st_email` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```
-- Dumping data for table `students`
--

INSERT INTO `students` (`st_id`, `st_name`, `st_dept`, `st_batch`,
`st_sem`, `st_email`) VALUES
('1', 'Pravesh', 'CSE', 2020, 2, 'rawatpravesh0016@gmail.com'),
('2', 'Nitish Sihmar', 'CSE', 2020, 3, 'sihmar.nitish@gmail.com'),
('3', 'Shivam Singh', 'CSE', 2020, 3, 'shivam@gmail.com'),
('4', 'Tushar Garg', 'CSE', 2020, 3, 'tushar@gmail.com');

-- --------------------------------------------------------

--
-- Table structure for table `teachers`
--

CREATE TABLE `teachers` (
  `tc_id` varchar(20) NOT NULL,
  `tc_name` varchar(20) NOT NULL,
  `tc_dept` varchar(20) NOT NULL,
  `tc_email` varchar(30) NOT NULL,
  `tc_course` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `teachers`
--

INSERT INTO `teachers` (`tc_id`, `tc_name`, `tc_dept`, `tc_email`,
`tc_course`) VALUES
('1', 'Sumit Bangar', 'cse', 'sumit@gmail.com', 'SE');

--
-- Indexes for dumped tables
--


--
-- Indexes for table `admininfo`
--
ALTER TABLE `admininfo`
```

```
  ADD PRIMARY KEY (`username`);

--
-- Indexes for table `attendance`
--
ALTER TABLE `attendance`
  ADD KEY `stat_id` (`stat_id`);

--
-- Indexes for table `reports`
--
ALTER TABLE `reports`
  ADD PRIMARY KEY (`st_id`);

--
-- Indexes for table `students`
--
ALTER TABLE `students`
  ADD PRIMARY KEY (`st_id`);

--
-- Indexes for table `teachers`
--
ALTER TABLE `teachers`
  ADD PRIMARY KEY (`tc_id`);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `attendance`
--
ALTER TABLE `attendance`
  ADD CONSTRAINT `attendance_ibfk_1` FOREIGN KEY (`stat_id`) REFERENCES
`students` (`st_id`);
COMMIT;
```

report.php

```
<?php

ob_start();
session_start();

if($_SESSION['name']!='oasis')
{
  header('location: login.php');
}
?>
<?php include('connect.php');?>

<!DOCTYPE html>
<html lang="en">

<!-- head started -->
<head>
<title>Online Attendance Management System 1.0</title>
<meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="../css/main.css">
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s" >

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.
min.css" >

  <link rel="stylesheet" href="styles.css" >

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>
```

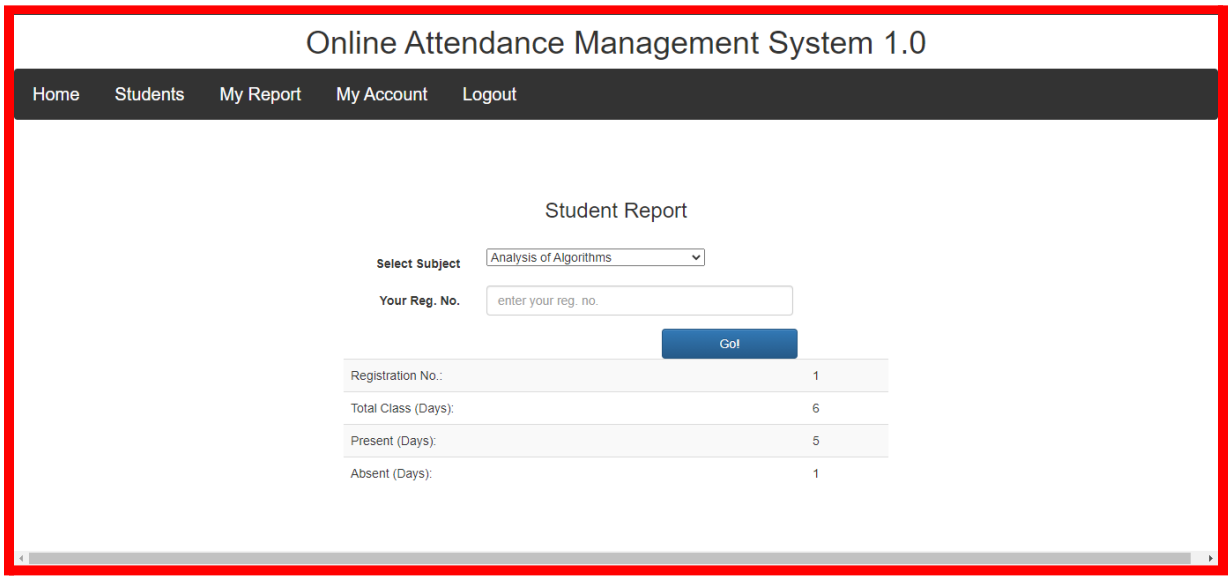
```
</head>
<!-- head ended -->

<!-- body started -->
<body>

<!-- Menus started-->
<header>

  <h1>Online Attendance Management System 1.0</h1>
  <div class="navbar">
  <a href="index.php">Home</a>
  <a href="students.php">Students</a>
  <a href="report.php">My Report</a>
  <a href="account.php">My Account</a>
  <a href="../logout.php">Logout</a>

</div>

</header>
<!-- Menus ended -->

<center>

<!-- Content, Tables, Forms, Texts, Images started -->
<div class="row">

  <div class="content">
    <h3>Student Report</h3>
    <br>
    <form method="post" action="" class="form-horizontal col-md-6
col-md-offset-3">

  <div class="form-group">

    <label  for="input1" class="col-sm-3 control-label">Select
Subject</label>
      <div class="col-sm-4">
      <select name="whichcourse" id="input1">
         <option  value="algo">Analysis of Algorithms</option>
```

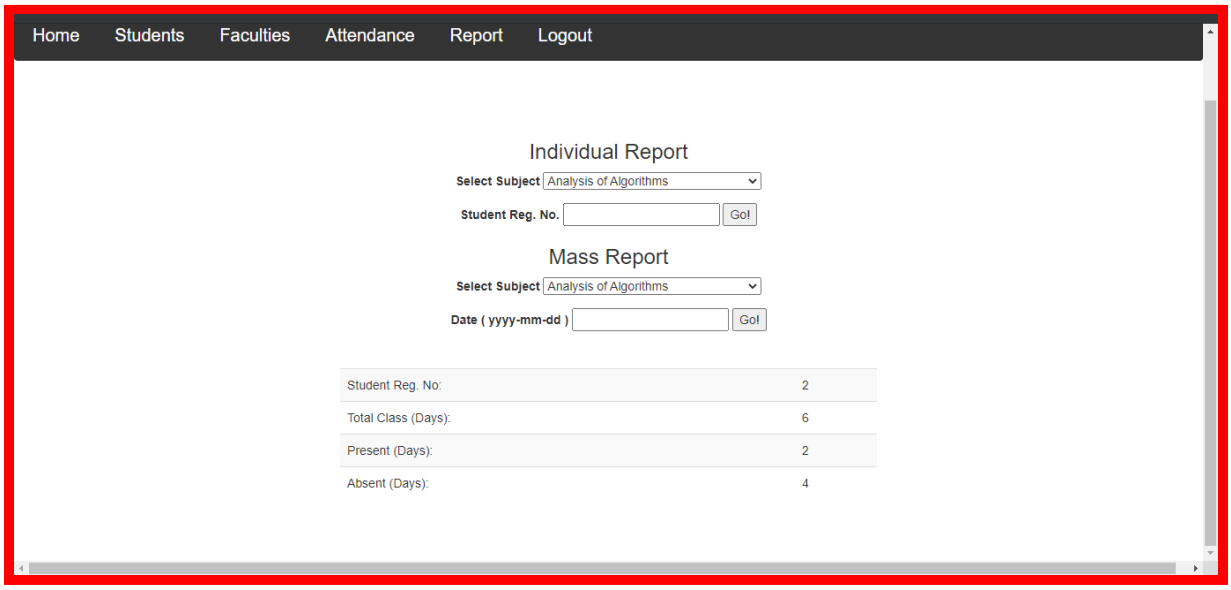
```
         <option  value="algolab">Analysis of Algorithms Lab</option>
        <option  value="dbms">Database Management System</option>
        <option  value="dbmslab">Database Management System Lab</option>
        <option  value="weblab">Web Programming Lab</option>
        <option  value="os">Operating System</option>
        <option  value="oslab">Operating System Lab</option>
        <option  value="obm">Object Based Modeling</option>
        <option  value="softcomp">Soft Computing</option>

      </select>
      </div>

  </div>

        <div class="form-group">
           <label for="input1" class="col-sm-3 control-label">Your Reg.
No.</label>
              <div class="col-sm-7">
                  <input type="text" name="sr_id"  class="form-control"
id="input1" placeholder="enter your reg. no." />
              </div>
        </div>
        <input type="submit" class="btn btn-primary col-md-3
col-md-offset-7" value="Go!" name="sr_btn" />
    </form>

    <div class="content"><br></div>

    <form method="post" action="" class="form-horizontal col-md-6
col-md-offset-3">
    <table class="table table-striped">

   <?php

    //checking the form for ID
    if(isset($_POST['sr_btn'])){

    //initializing ID
     $sr_id = $_POST['sr_id'];
     $course = $_POST['whichcourse'];
```

```
    $i=0;
    $count_pre = 0;

    //query for searching respective ID
   //  $all_query = mysql_query("select * from reports where
reports.st_id='$sr_id' and reports.course = '$course'");
   //  $count_tot = mysql_num_rows($all_query);
    $all_query = mysql_query("select stat_id,count(*) as countP from
attendance where attendance.stat_id='$sr_id' and attendance.course =
'$course' and attendance.st_status='Present'");
    $singleT= mysql_query("select count(*) as countT from attendance
where attendance.stat_id='$sr_id' and attendance.course = '$course'");
    $count_tot;
    if ($row=mysql_fetch_row($singleT))
    {
    $count_tot=$row[0];
    }

    while ($data = mysql_fetch_array($all_query)) {
      $i++;
     //  if($data['st_status'] == "Present"){
     //     $count_pre++;
     //  }
      if($i <= 1){
    ?>


    <tbody>
     <tr>
         <td>Registration No.: </td>
         <td><?php echo $data['stat_id']; ?></td>
     </tr>

     <tr>
       <td>Total Class (Days): </td>
       <td><?php echo $count_tot; ?> </td>
     </tr>

     <tr>
```

```
        <td>Present (Days): </td>
        <td><?php echo $data[1]; ?> </td>
      </tr>

      <tr>
        <td>Absent (Days): </td>
        <td><?php echo $count_tot -  $data[1]; ?> </td>
      </tr>

    </tbody>

   <?php

     }
    }}
     ?>
    </table>
  </form>
  </div>

</div>
<!-- Contents, Tables, Forms, Images ended -->

</center>

</body>


</html>
```

Result of Module 3





Result:

Thus, module3 was implemented and documented successfully.

EXPERIMENT-11

Aim

To Prepare a master test plan and Test cases for testing the project.

1. Executive Summary

This Attendance management system is developed for students and teachers to keep track/record of daily basis of student's attendance and to generate accurate monthly reports for statistical analysis and to calculate the student's eligibility criteria.
The projects have scopes like organizing the data and building the project flawless by proper testing etc.

2. Test Plan

2.1. Scope of Testing

Functional: Are all modules covered? Any exceptions for any modules? Does automation cover all functional test cases or Regression – Critical Path Test Cases?
Yes, all the modules mentioned like User registration and login are being covered. The exception is present in the attendance module for the class as only individual attendance of the student can be seen but not the class-wise attendance. We have not used automation-based testing and focused on manual testing

Non-Functional: Are all NFR (Non-Functional Requirements) covered?
Yes, Non-Functional requirements are covered in our system like Performance and scalability and confidentiality of the system where the webpage loads in less than 7 seconds and registration of over 1000 requests in 5 minutes, and proper authorization of users in the system respectively.

2.2. Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
Functional Requirements	Manual	Excel Template, word template, AMS Software

Non-Functional Requirements	Manual	AMS Software, Excel Template, word template

2.3. Test Deliverables

The deliverables are mentioned below with the defect log present in another document

TEST CASE DOCUMENTATION:

i.) PROJECT INFORMATION:
1.Project Name: Attendance management system
ii.) TEST OBJECTIVE:
1.Type of Testing: Unit Testing
Objective: In order to check whether the particular module (login/signup) has any bugs, we perform unit testing
2.Type of Testing: System Testing
Objective: The purpose of system testing is to ensure that a system meets its specification and any non-functional requirements (such as stability and throughput) that have been agreed upon with its users. In our Project, we performed System testing, to ensure functional and non-functional requirements.
iii) TEST REPORTING:
For AMS software Number of total test cases executed = 15 No. of test cases passed = 13 No. of test cases failed = 2 Pass percentage = 86.66% Fail Percentage = 13.33%
iv)DEFECT LOG: Program crashed when more than 1000 logins or 1000 request per second over 5 mins time span-In progress Not Accept valid email and phone number-In progress

3. Test Case

3.1. Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1.	Verify User Login	Accept Valid Username and Password on Page#1 and not accept empty fields.	1. User choose the category based on student, teacher, admin 2. User clicks on the text area of username and password 3. Enter the username and password on the text box 4. Click Login button	Users should be taken to the next page to use various features and check details and if left fields blanks and messages should be shown to fill the details.	The user is directed to another page that contains several options and if not filled, a message is shown.	Pass	success
2.	Add User Registration and sync of user details on the database.	Accept email, username and phone number on the registration page	1. User clicks on signup 2. User enters the required details. 3. User clicks Signup button	Users should be directed to the registration page and after adding successfully the login option is shown and data should be added to the database.	Users are directed to the registration page on clicking signup and can go back to the login page on clicking login and data is updated in	Pass	Success

					the database.		
3.	Verify user registration	Not Accept empty fields for registration	1. User clicks on signup 2. Users do not enter the required details. 3. User clicks Signup button	Messages should be shown to tell the user to enter details.	The message is shown that fields cannot be empty	Pass	Success
4.	Verify user registration	Accept valid email and phone number	1. User clicks on signup 2. The user enters the invalid details. 3. User clicks Signup button	A prompt message to tell user invalid.	Accept invalid mail and phone number.	Fail	NIL
5.	Updating details	Change of details if provided incorrectly previously.	1. Click my account and reg. Number. 2. Enter correct details 3. Click update	Should be able to re-enter the details again and updated details should get reflected in the database.	Students can update and make changes in details by entering again and the update is done in the database.	Pass	Success
6.	Update attendance on a daily basis of each subject	Select present and absent options for each student based on the presence of students in class daily.	1. Select the attendanc e option on the teacher's page and enter the date and subject in the respective fields.	Faculties will be able to smoothly enter attendance daily and are updated in the database.	Faculties can enter present or absent and update the attendance on a daily basis for each subject.	Pass	Success

			2. Click present or absent. 3. Click save.				
7.	Attendance Report	View and display the individual report	1. Click the report option in the student's or teacher's page 2. Select a subject and enter reg no. 3. Click Go	Able to display and view the attendance stats e.g., No. Of days absent, present, class.	Can display and view the attendance of student by both students and teachers on the website	Pass	Success
		View and display Mass report	1. Click the report option on the teacher's page 2. Select a subject and enter reg no. 3. Click Go	Should be able to view a mass report of students	Not able to view and display mass reports.	Fail	NIL
8.	Logout	Logout out of the session and account.	1. Click the logout option.	Users should be able to log out from their accounts after using the website.	Users can log out from the account on the website by clicking logout.	Pass	Success

4.1. Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks

1.	Performance	Computed average time to operate the program (150 Trails)	1.Open the website and run it. 2.Do the same for 100 Trails and calculate the average time taken	Less Than 6 sec of computing Time	Approximately less than or equal to 3 sec	Pass	Success
2.	Availability	The program Should be able to run for more than 24 hours continuously without Crashing.	Open website and run the program for around 45 hours	The program should work without Crashing for more than 24hrs	The program worked with no problem for up to 45 hours	Pass	Success
3.	Scalability	The program should not crash when more than 1000 logins or 1000 request per second over a 5mins time span	1.Open website on multiple devices at the same time 2.Try logging in at the same time	The program should not crash	Program crashed at approximately 500 logins and requests	Fail	Failure
4.	Usability	When used for a longer duration no issues are seen with respect to RAM consumpti	1. Open the website and run it. 2.Measure the CPU performance and temperature.	CPU usage should be minimized	Optimum CPU usage recorded	Pass	Success

		on or any network problems					
5.	Security	The website should be secured form wrong Logins, Glitches and Data Leaks	1.Alpha version of Program is given to Testers 2.Testing for glitches and Data Leaks performed	Program Should pass at least 98% Of total trails	Program Executed without Glitches and without security Breaches for 99.9%	Pass	Success
6.	Flexibility	User should be able to get the expected result and not directed to many pages when clicked different options and able to change easily in response to different user	1. Allow a set of people to use the website and get their feedback. 2. check manually all the options available in the software.	Websites should be comfortable to use and easily accessible.	Flexibility in software is assured.	Pass	Success

Result:

Thus, the test plan and test cases are documented successfully

EXPERIMENT-12

Aim

To conduct the manual test using Test cases and prepare test report for the project

1. Executive Summary

This AMS is developed for students and teachers to keep track/record of daily basis of student's attendance and to generate accurate monthly reports for statistical analysis and to calculate the student's eligibility criteria.
The projects have scopes like organizing the data and building the project flawlessly by proper testing etc.

2. Test Plan

2.1. Scope of Testing

Functional: Are all modules covered? Any exceptions for any modules? Does automation cover all functional test cases or Regression – Critical Path Test Cases?
Yes, all the modules mentioned like User registration and login are being covered. The exception is present in the attendance module for the class as only individual attendance of the student can be seen but not the class- wise attendance. We have not used automation-based testing and focused on manual testing

Non-Functional: Are all NFR (Non-Functional Requirements) covered?

Yes, Non-Functional requirements are covered in our system like Performance and scalability, and confidentiality of the system where the webpage loads in less than 7 seconds and registration of over 1000 requests in 5 minutes, and proper authorization of users in the system respectively.

2.2. Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
Functional Requirements	Manual	Word Template , AMS software
Non-Functional Requirements	Manual	AMS software, word template

2.3. Test Deliverables

TEST CASE DOCUMENTATION:

i.) PROJECT INFORMATION:
1.Project Name: Attendance management system
ii.) TEST OBJECTIVE:
1.Type of Testing: Unit Testing
Objective: In order to check whether the particular module (login/signup) has any bugs, we perform unit testing
2.Type of Testing: System Testing
Objective: The purpose of system testing is to ensure that a system meets its specification and any non-functional requirements (such as stability and throughput) that have been agreed upon with its users. In our Project, we performed System testing, to ensure functional and non-functional requirements.
iii) TEST REPORTING:
For AMS software Number of total test cases executed = 15 No. of test cases passed = 13 No. of test cases failed = 2 Pass percentage = 86.66% Fail Percentage = 13.33%
iv)DEFECT LOG: Program crashed when more than 1000 logins or 1000 request per second over 5mins time span-In progress

Not Accept valid email and phone number-In progress

3. Test Case

3.1. Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1.	Verify User Login	Accept Valid Username and Password on Page#1 and not accept empty fields.	1. User choose the category based on student, teacher, admin 2. User clicks on text area of username and password 3. Enter the username and password on the text box 4. Click Login button	Users should be taken to the next page to use various features and check details and if left fields blanks and messages should be shown to fill the details.	The user is directed to another page which contains several options and if not filled, a message is shown.	Pass	success
2.	Add User Registration and sync of user details on the database.	Accept email, username and phone number on the registration page	1. User clicks on signup 2. The user enters the required details. 3. User clicks Signup button	Users should be directed to the registration page and after adding successfully the login option is shown and data should	Users are directed to the registration page on clicking signup and can go back to the login page on clicking login and data is	Pass	Success

				be added to the database.	updated in the database.		
3.	Verify user registration	Not Accept empty fields for registration	1. User clicks on signup 2. Users do not enter the required details. 3. User clicks Signup button	Messages should be shown to tell the user to enter details.	The message is shown that fields cannot be empty	Pass	Success
4.	Verify user registration	Accept valid email and phone number	1. User clicks on signup 2. User enters the invalid details. 3. User clicks Signup button	Prompt message to tell user invalid.	Accept invalid mail and phone number.	Fail	NIL
5.	Updating details	Change of details if provided incorrectly previously.	1. Click my account and reg. Number. 2. Enter correct details 3. Click update	Should be able to re-enter the details again and updated details should get reflected in the database.	Students can update and make changes in details by entering again and the update is done in the database.	Pass	Success
6.	Update attendance on daily basis of each subject	Select present and absent options for each student based on the presence of students in class daily.	1. Select the attendance option on the teacher's page and enter the date and subject in the respective fields. 2. Click present or absent. 3. Click Save.	Faculties will be able to smoothly enter attendance daily and are updated in the database.	Faculties can enter present or absent and update the attendance on a daily basis for each subject.	Pass	Success

7.	Attendance Report	View and display the individual report	1. Click the report option on the student's or teacher's page 2. Select a subject and enter reg no. 3. Click Go	Able to display and view the attendance stats eg. No. Of days absent, present, class.	Can display and view the attendance of student by both students and teachers on the website	Pass	Success
		View and display Mass report	1. Click report option in the teacher's page 2. Select subject and enter reg no. 3. Click Go	Should be able to view mass report of students	Not able to view and display mass reports.	Fail	NIL
8.	Logout	Logout out of the session and account.	1. Click the logout option.	Users should be able to log out from their accounts after using the website.	Users can log out from the account on the website by clicking logout.	Pass	Success

4.1. Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1.	Performance	Computed average time to operate the program (150 Trails)	1. Open the website and run it. 2. Do the same for 100 Trails and calculate the	Less Than 6 sec of computing Time	Approximately less than or equal to 3 sec	Pass	Success

			average time taken				
2.	Availability	The program Should be able to run for more than 24 hours continuously without Crashing.	Open website and run the program for around 45 hours	Program should work without Crashing for more than 24hrs	Program worked with no problem for up to 45 hours	Pass	Success
3.	Scalability	The program should not crash when more than 1000 logins or 1000 request per second over 5 mins time span	1.Open website in multiple devices at the same time 2.Try logging in at the same time	The program should not crash	Program crashed at approximately 500 logins and requests	Fail	Failure
4.	Usability	When used for longer duration no issues are seen with respect to RAM consumption or any network problems	1. Open the website and run it. 2.Measure the CPU performance and temperature.	CPU usage should be minimized	Optimum CPU usage recorded	Pass	Success
5.	Security	The website should be secured from wrong	1.Alpha version of Program is given to Testers	Program Should pass at least 98% Of total trails	Program Executed without Glitches and without security	Pass	Success

		Logins, Glitches and Data Leaks	2.Testing for glitches and Data Leaks performed		Breaches for 99.9%		
6.	Flexibility	User should be able to get the expected result and not directed to many pages when clicked different options and able to change easily in response to different user	1. Allow a set of people to use the website and get their feedback. 2. check manually all the options available in software.	Websites should be comfortable to use and easily accessible.	Flexibility in software is assured.	Pass	Success

5. Defect Log

Requirement #	Defect ID #	Defect Description	Assignee	Status
M1R1	1	Program crashed when more than 1000 logins or 1000 request per second over 5 mins time span	Software testing team (AP Rohith)	In-progress
M3R1	2	Not Accept valid email and phone number	Coding Team (Chinmaya Purohit and Thejaswin)	Not started

6. Test Report

The system is fully functional with the students being able to view their attendance and teachers are able to submit attendance to the student subject-wise and they can generate report
The only obstacle being that the mass report of the student is not being generated due to technical issues. Apart from this issue, everything is functional and ready to go.
We have sought help from our stakeholders related to the constraint to remove the obstacles

Category	Progress Against Plan	Status
Functional Testing(login)	Green	Completed
Functional Testing(Signup)	Green	Completed
Non-Functional Testing	Green	In progress

Functional	Test Case Coverage (%)	Status
Test ID 1	100%	Completed
Test ID 2	100%	Completed
Test ID 3	100%	Completed
Test ID 4	50%	In progress
Test ID 5	100%	Completed
Test ID 6	100%	Completed
Test ID 7	100%	Completed
Test ID 8	100%	Completed

Result:

Thus, the software test conducted and documented the report successfully