A
PROJECT REPORT
ON

ATTENDANCE MANAGEMENT SYSTEM

SUBJECT: SYSTEM DESIGN PRACTICE

DHARAMSINH DESAI UNIVERSITY

FACULTY OF TECHNOLOGY, NADIAD

PREPARED BY:

Parth Panchal (CE-73)

Rutvik Panchal (CE-75)

GUIDED BY:

Prof. Bhavika. M. Gambhava

(Assistant Professor.)

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of System Design Practice titled "Attendance Management System" and recorded in this report is a work of

Name	Roll No.	Student Id
Parth Panchal	CE-073	15CEUBS100
Rutvik Panchal	CE-075	15CEUBS063

Department of Computer Engineering, semester VI. They were involved in Project developing during the period Dec-17 to Apr-2018.

Guide:

Prof. Bhavika M. Gambhava
Assistant Professor,
CE Department,
Faculty of Technology,
Dharmsinh Desai University.

Dr. C. K. Bhensdadia,
Head, CE Department,
Faculty of Technology,
Dharmsinh Desai University.

CONTENTS

Abstract

For Colleges/Institute who are still using pen-paper based attendance for students. The Attendance Management System provides facility and ease of taking attendance in digital way over extra efforts for manage attendance for each students and support to go green. It mainly covers student attendance and also provide interface to view attendances.

AMS-Stands for Attendance Management System provide feature of Attendance management with each student's record based on courses. It also has feature to view report based on course taken by student. On Student View, Student can track his Attendance. Whole System is manage by Administrator.

1. Introduction:

1.1 Project Details: Broad Specification

Attendance Management System is a software developed for automatic student attendance management in colleges or institutes. It facilitates faculty to take attendance in digital way instead of manage it paper-wise.
It also access to view attendance information for students. The System provide admin panel through which head of department/admin allocate faculty on basis of course. This system will also help in evaluating attendance eligibility criteria of a student.

1.2 Technology/Platform Used:

Front-End: HTML, CSS and Bootstrap.

For Presentation logic we have used HTML, CSS and Bootstrap v-3.3.7. We build responsive mobile first web application with the help of front end component library-Bootstrap.

Bootstrap is an open source and most popular toolkit for developing with HTML, CSS, and JS. It provides building a responsive Mobile-First site.

Back-End: PHP

PHP is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

We have used PHP scripting language for developing Business logic.

Database: MySQL

MySQL is an open-source relational database management system. It easily modelled with PHP scripting language.

Diagram Tool: Draw.io, Umlet

All the UML diagrams of this project is made in Umlet and ER-diagram is made in Draw.io.

2. Software Requirement Specification:

2.1 Introduction

2.1.1. Purpose

The aim of this document is to specify complete description and understanding of Attendance Management System. It will explain the interface of system, features of system, request and response of system, constraint under which it Operate and how the system will react to different users. Main purpose of the attendance management system is to computerized the traditional way of taking attendance. Another purpose for developing this software is to generate the report automatically at the end of the session or in the between of the session.

2.1.2. Scope of Project

This software system will be Attendance Management System for colleges/institution who are still using pen-paper based attendance for students. This system is designed to take attendance in digital way over extra efforts of faculty for manage attendance for each students on hard-copy and it's support to go green. It leads to increase institution productivity by efficient and accurate management of student's attendance in digital way which reduce time and efforts for faculty, Instead of take attendance record manually on hardcopy. It is maximizing faculty's work efficiency and productivity.

Mainly system covers Attendance management for students taken online through web. System manages student attendance record for each students and prepare date wise report for student attendance. System manages students and Faculties information. System also contain relational database. System will manages attendance for student branch-wise, so that each branch or department of colleges are provided different connection.

2.1.3 Intended Audience and Reading Suggestions

This document is to be read by the development team, the project managers, department staff, testers and documentation writers. Our stakeholders, company manufacturing associated hardware, company providing embedded operating system, Stack-holders and distributors who markets the finished product, may review the document to learn about the project and to understand the requirements. The SRS has been organized approximately in order of increasing specificity. The developers and project managers need to become intimately familiar with the SRS.

2.1.4 Glossary:

Users	Faculty and Students
Admin	Head of Department Who manages whole system in digital form.

2.1.5 References

IEEE Standards 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.Contact Money Lenders for Information about Real Time Requirements.

2.1.6 Overview of Document

The purpose this documents is to present a detailed description of the Attendance Management System. It will explain the purpose and features of the software, the interfaces of the software, what the software will do, the constraints under which it must operates. The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

2.2 Overall Description

2.2.1 Product Perspective

The product Attendance Management system, is an independent product and does not depend on any other product or system. The product will automate various tasks associated with handling attendance details and better organizing the stored information and optimum performance, thus helping the Colleges to ensure smooth working of these processes.

2.2.2 Product Functions

Our system has two types of accessing modes,

1. **Administrator**
2. **Student**
3. **Faculty**

- **Administrator:**

  Attendance Management System is managed by Administrator. Administrator has to update and monitor the student attendance details, add a new faculty and assign course to him, provide unique ID number for all students and faculty, and view course which is taken by student. Administrator can update his profile, and also can give help to the faculty and students.

- **Student:**

  Student can only view their personal details, course assigned, and edit their assigned course and can view their attendance.

- **Faculty:**

  Faculty can request for account to administrator. After approval of admin, faculty can take attendance of students and view report of student attendance. Faculty can also view allocated students and subjects to himself.

2.2.3 User Characteristics

This software gives access to two kinds of users.

1. Administrator: College administrator will have administrator access through admin login to add, delete and modify information.

2. Authorized User: Teaching staff will have access to only view the information and can update the student's attendance in the form of formatted reports.

2.2.4 Operating Environment

Attendance Management System should run on GUI based Operating System like Windows 2000, Windows Vista, Windows 7/8/8.1/10, Linux Based System and Mac also which has Chrome, Firefox etc. as browser support .

2.2.5 Design and implementation Constraints

Attendance Management System is open source version. Everyone can use but who want to use the product first they will contact to admin. Attendance Management System requires only basic browser and also requires internet connection. Users are comfortable with English language. Users not require much memory on hard drive. Everything stores in server db. Users information is safe so don't worry.

2.2.6 User Documentation

If user want to learn about use of web application, we also provide help section in application. User can easily refer from help section and user guide. We also provide tutorial of the use of application.

2.2.7 Assumption and Dependencies

- It is assumed that the hardware designed will work correctly with the third-party operating system and the developed software. It is assumed that browser is supported PHP and MySQL platform.
- Users with administrator access should be careful in deleting or modifying any information knowingly or unknowingly which will lead to inconsistency of the database.
- The end users of this software are assumed to have basic level of computer knowledge i.e. point and click.

2.3 Specific Functional Requirements

1. **R1: System Authenticate users (faculty) through Secure Login.**
   Description: User provide system access after once user is authenticated through secure login feature.
   Precondition: Before this use case can be initiated, the user (Faculty or student) has already accessed the online Attendance Management System.

   R1.1: Login form contain field to take user-name and password which is provided by Admin for authentication of system.

   R1.2: System verify and authenticate login credentials of user.

   Input: Enter login credentials by user. (As per R1.1)

   Output: provide access to authenticated user.

   Description: System read login credentials filled in login form and authenticate it.

2. **R2: System provide interface for admin to add student records and allocate courses to faculty.**

   Description: Admin creates users. He also updates user details, manage database. Here system can maintain all required details of users and admin can also manipulate them.

Precondition: Database administrator must have access of admin panel.

R2.1: Admin add student records branch-wise.

Description: System provide feature for add student records, here admin add students based on his convenience.

R2.1.1: Student are allocated as per course taken by student in semester.

Description: System give feature to choose course to students and system allocate student as per subject chosen by him.

R2.1.2: Student are allowed to take optional Courses based on his choice per Semester.

Description: Here student choose elective course and system manage those student according to students.

R2.2: Admin Add Faculty to system as per course in department.

R2.2.1: Admin allocate Faculty on basis of no of courses as per branch.

R2.2.2: Admin is allowed to allocate multiple courses to faculty branch.

R2.2.3: Admin can allocate sections of courses to multiple faculty.

3. **R3: System Provide Simple Login for Student to view Attendance**

R3.1: Student can view Attendance as per chosen course.

R3.2: System calculate total attendance of student base on course out of total lectures.

R3.3: Student can view number of lecture present per total and percentage of attendance on each course as well as

total percentage of attendance based on no of lecture attendance per total lecture taken.

4. **R4: System Manages Attendance**
   Description: System provide interface where faculty choose appropriate semester and subject then he take attendance for appropriate date.

   R4.1: System provide interface to users to take Attendance.

   Description: Faculty can take attendance from interface which provided by system. Faculty can select semester and subject and date for take attendance.

   R4.1.1: System provide feature for select all attendance.

   Description: System provide feature for mark all student attendance

   R4.1.2: System provide feature for save attendance.

   Input: Take all student attendance by check the box.

   Output: Attendance is saved.

   Description: System provide feature for save attendance and faculty can verify it.

   R4.2: System get permit to user to update attendance.

   Input: Select appropriate date and subject for update attendance.

   Output: Retake attendance saved successfully.

   R4.3: System provide feature to allow to take past days attendance or pending attendance for faculty.

   Description: System permit to faculty for take past attendance or any remaining attendance. Here system permit to faculty because there is some reason behind for attendance not taken.

R4.3.1: System provide facility to show current date attendance and past pending attendance and faculty also take attendance.

R4.3.2: System show to faculty the subjects which are allocated to him and students which are take those subjects.

5. **R5: Dashboard Facility**

   Description: System provide dashboard facility to faculty where he/she easily seen current date attendance and also previous pending attendance and also take attendance. Faculty also seen subjects allocated to him and students who take that subject.

6. **R6: System generate report of student attendance and show to everyone when it require.**

   Description: Faculty can generate report and view attendance of each student

2.4 Non-Functional Requirements:

2.4.1 Performance:

Easy tracking of records and updating can be done. All the requirements relating to performance characteristics of the system are specified in the section below. There are two types of requirements.

A. Static Requirements:

These requirements do not impose any constraints on the execution characteristics of the system. They are:

1) Number of Terminals:

The software makes use of an underlying database that will reside at the server, while the front end will be available online to the administrative as well as students and faculty.

2) **Number of Users:**

The number of users may vary, as this software finds applications in almost all department of the organization.

B. Dynamic Requirements:

These specify constraints on the execution characteristics of the system. They typically include response time and throughout of the system. Since these factors are not applicable to the proposed software, it will suffice if the response time is high and the transactions are carried out precisely and quickly.

2.4.2 Reliability:

The software will not be able to connect to the centralized database in the event that the internet connection fails or in the event of the server being down due to a hardware or software failure.

2.4.3 Availability:

The software will be available only to authorized users of the colleges like faculties to mark the student attendance, student to view their enrolled course, admin to add an update faculty record and course assign.

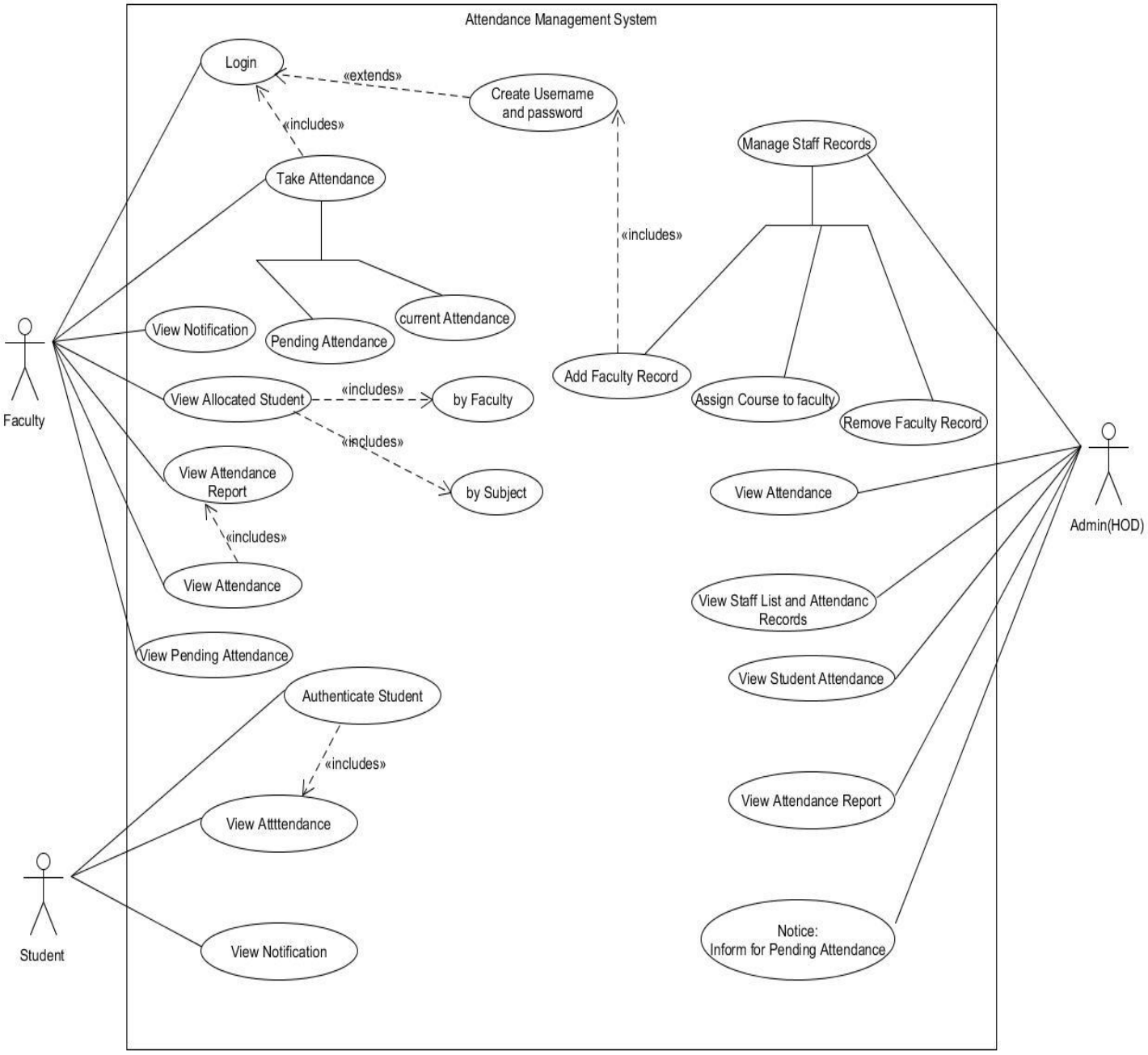
2.4.4 Security:

The security requirements deal with the primary security. The software should be handled only by the administrator and authorized users. Only the administrator has right to assign permission like creating new accounts and generating password. Only authorized users can access the system with username and password.

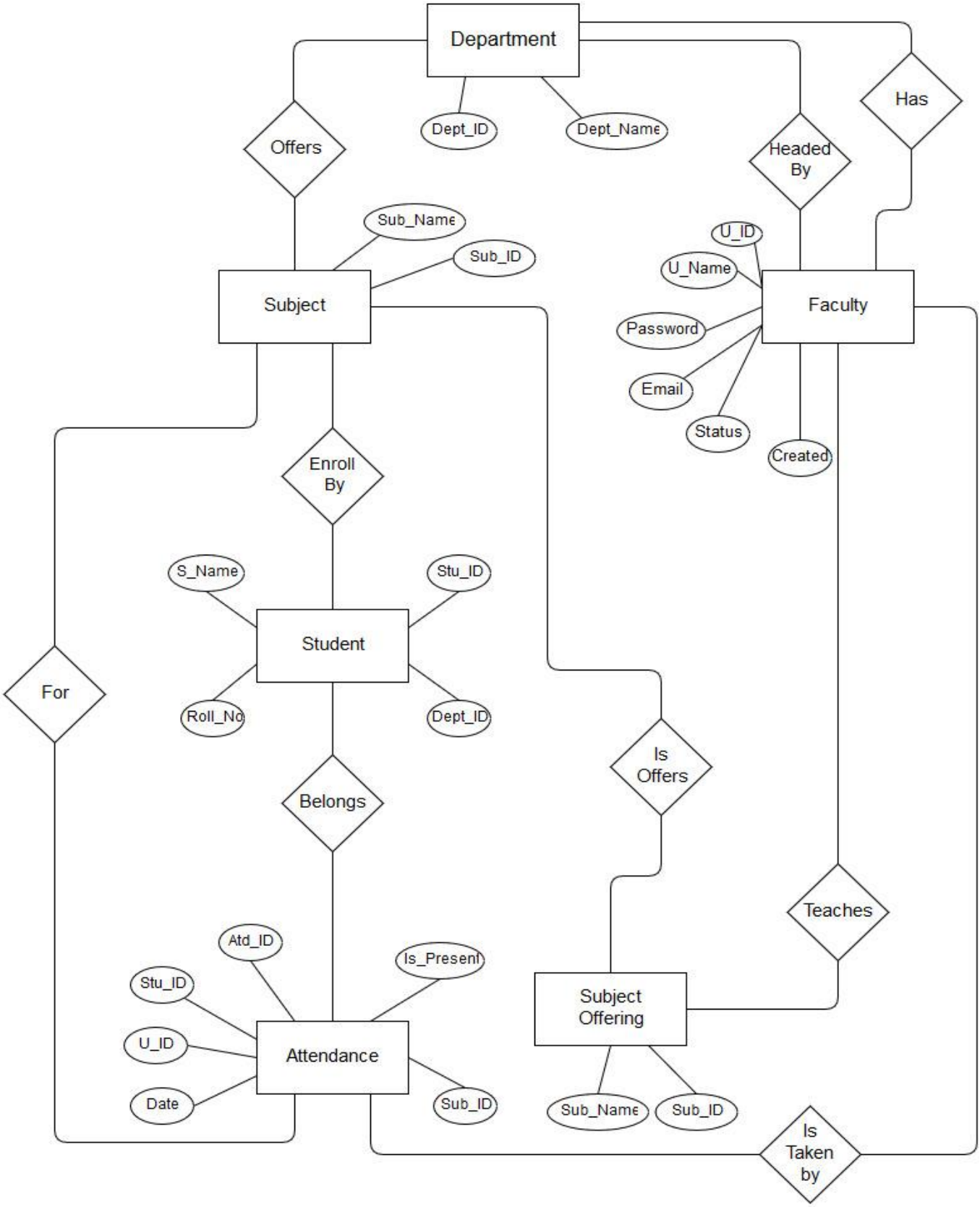
2.4.5 Portability:

The Software is a web-based application and is built in PHP and MYSQL so it is platform independent and is independent of operating system.

3. Use Case Diagram:



4. ER-Diagram:


Department
Dept_ID
Dept_Name
Offers
Has
Headed By
Sub_Name
Sub_ID
Subject
U_ID
U_Name
Faculty
Password
Email
Status
Created
Enroll By
S_Name
Stu_ID
Student
For
Roll_No
Dept_ID
Is Offers
Belongs
Teaches
Atd_ID
Is_Present
Stu_ID
U_ID
Attendance
Date
Sub_ID
Subject Offering
Sub_Name
Sub_ID
Is Taken by

5. Data Dictionary:

- **Faculty:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**uid**	int(11)			No	*None*		AUTO_INCREMENT
2	**uname**	varchar(255)	latin1_swedish_ci		No	*None*		
3	**password**	varchar(255)	latin1_swedish_ci		No	*None*		
4	**did**	int(11)			Yes	*NULL*		
5	**email**	varchar(255)	latin1_swedish_ci		No	*None*		
6	**status**	tinyint(4)			No	*None*		
7	**created**	int(11)			No	*None*		

- **Student:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**sid**	int(11)			No	*None*		
2	**name**	varchar(255)	latin1_swedish_ci		No	*None*		
3	**rollno**	int(11)			No	*None*		
4	**did**	int(11)			Yes	*NULL*		

- **Subject:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**name**	varchar(255)	latin1_swedish_ci		No	*None*		

- **Department:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(255)			No	*None*		AUTO_INCREMENT
2	**did**	int(255)			No	*None*		
3	**dname**	varchar(255)	latin1_swedish_ci		No	*None*		

- **Attendance:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**aid** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**sid**	int(11)			No	*None*		
3	**date**	int(11)			No	*None*		
4	**ispresent**	tinyint(4)			No	*None*		
5	**uid**	int(11)			No	*None*		
6	**id**	int(11)			No	*None*		

- **Department-Subject:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**did**	int(11)			No	*None*		
2	**id**	int(11)			No	*None*		

- **Department-User:**

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	**did**	int(11)			No	*None*	
2	**uid**	int(11)			No	*None*	

- **User-Subject:**

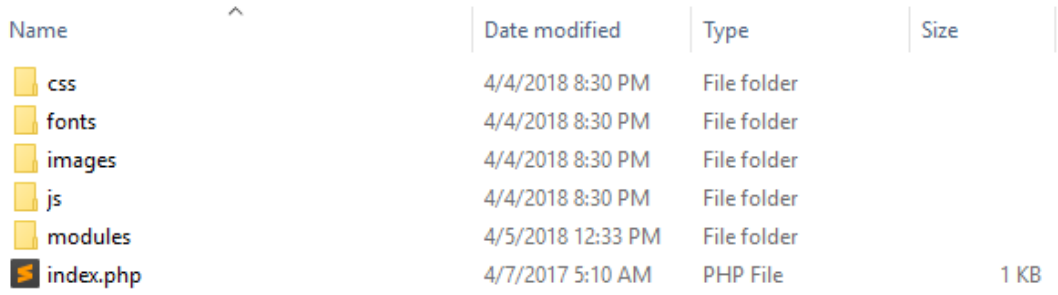
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**uid**	int(11)			No	*None*		
2	**id**	int(11)			No	*None*		

- **Student-Subject:**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**sid**	int(11)			No	*None*		
2	**id**	int(11)			No	*None*		
3	**ssid** 🔑	int(11)			No	*None*		AUTO_INCREMENT

6. Implementation:

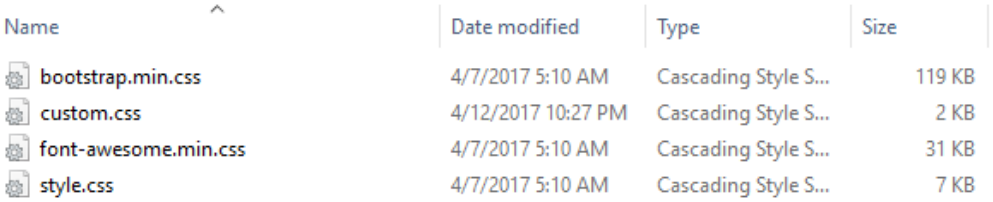
Folder Structure:



Name	Date modified	Type	Size
css	4/4/2018 8:30 PM	File folder	
fonts	4/4/2018 8:30 PM	File folder	
images	4/4/2018 8:30 PM	File folder	
js	4/4/2018 8:30 PM	File folder	
modules	4/5/2018 12:33 PM	File folder	
index.php	4/7/2017 5:10 AM	PHP File	1 KB

CSS:

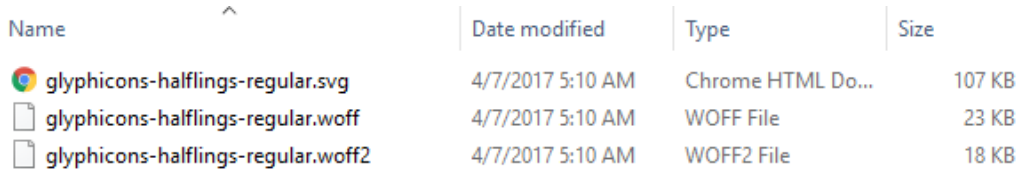
CSS folder contains all needed CSS files for presentation purpose.



Name	Date modified	Type	Size
bootstrap.min.css	4/7/2017 5:10 AM	Cascading Style S...	119 KB
custom.css	4/12/2017 10:27 PM	Cascading Style S...	2 KB
font-awesome.min.css	4/7/2017 5:10 AM	Cascading Style S...	31 KB
style.css	4/7/2017 5:10 AM	Cascading Style S...	7 KB

Fonts:

Font folder contains all Font related files for presentation purpose.



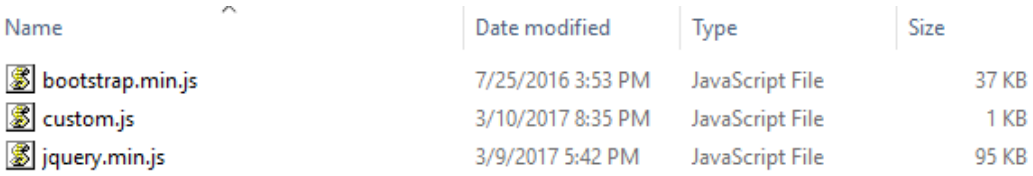
Name	Date modified	Type	Size
glyphicons-halflings-regular.svg	4/7/2017 5:10 AM	Chrome HTML Do...	107 KB
glyphicons-halflings-regular.woff	4/7/2017 5:10 AM	WOFF File	23 KB
glyphicons-halflings-regular.woff2	4/7/2017 5:10 AM	WOFF2 File	18 KB

Images:

Images folder contains **Logo** and **other images** files for presentation purpose.

Js:

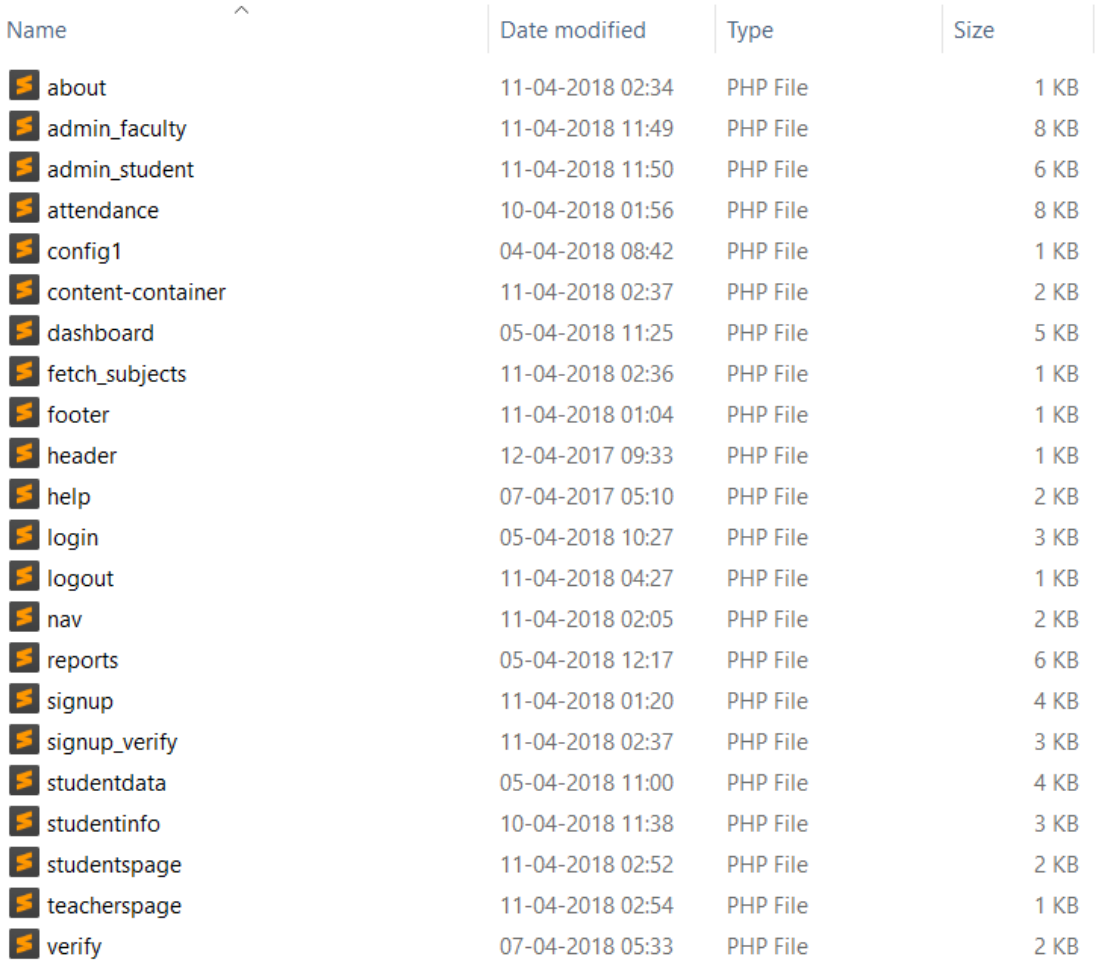
Js folder contains **bootstrap, jQuery** and **custom js** files for presentation purpose.



Name	Date modified	Type	Size
bootstrap.min.js	7/25/2016 3:53 PM	JavaScript File	37 KB
custom.js	3/10/2017 8:35 PM	JavaScript File	1 KB
jquery.min.js	3/9/2017 5:42 PM	JavaScript File	95 KB

Modules:

Modules folder contains all modules of system.



Name	Date modified	Type	Size
about	11-04-2018 02:34	PHP File	1 KB
admin_faculty	11-04-2018 11:49	PHP File	8 KB
admin_student	11-04-2018 11:50	PHP File	6 KB
attendance	10-04-2018 01:56	PHP File	8 KB
config1	04-04-2018 08:42	PHP File	1 KB
content-container	11-04-2018 02:37	PHP File	2 KB
dashboard	05-04-2018 11:25	PHP File	5 KB
fetch_subjects	11-04-2018 02:36	PHP File	1 KB
footer	11-04-2018 01:04	PHP File	1 KB
header	12-04-2017 09:33	PHP File	1 KB
help	07-04-2017 05:10	PHP File	2 KB
login	05-04-2018 10:27	PHP File	3 KB
logout	11-04-2018 04:27	PHP File	1 KB
nav	11-04-2018 02:05	PHP File	2 KB
reports	05-04-2018 12:17	PHP File	6 KB
signup	11-04-2018 01:20	PHP File	4 KB
signup_verify	11-04-2018 02:37	PHP File	3 KB
studentdata	05-04-2018 11:00	PHP File	4 KB
studentinfo	10-04-2018 11:38	PHP File	3 KB
studentspage	11-04-2018 02:52	PHP File	2 KB
teacherspage	11-04-2018 02:54	PHP File	1 KB
verify	07-04-2018 05:33	PHP File	2 KB

Index.php:

- Index.php is master page which include header, navigation, content-container and footer php files.
- Content-container.php has routing logic for modules file and work as process service.

❖ Implementation Environment

- Any Client/Hosting Server Which Provides Apache, PHP, MySQL Services.
- It is compatible on all operating system with Chrome, Firefox, and IE as browser support.

• Modules Description

• Log - In Module

Login Module is used to Authenticate Faculty (User) Based on Faculty Data store in user table. User table contain verify status field, which is set after verify by Head of Department. After that faculty is allowed to login.

It Contain login.php and verify.php page.

Input: Faculty (User's) information or credentials

Output: Verified successfully

Processing: Authenticate by checking user's credentials in the database with entered Credential, if it is verified successfully user will redirect to dashboard.

• Dashboard Module:

Once After Successful Login User is redirect to Dashboard page.

It mainly include three parts

1. Pending Attendance Status: Where User View Last 5 days Attendance Status.
2. Today's Attendance: Take Todays Attendance
3. Student And Subject Info: Numbers of Subject and Student Allocated per Subject.

It contains dashboard.php

Input: User Credential (After Successful Login)

Output: Dashboard View

- **Attendance Module:**

  Attendance is the main module for the system. It involves the functionality to take Attendance, View Attendance and Update taken Attendance.

  Navigation:

  Once after successful login user redirected to Dashboard Page.

  - User Selects To take Today's Attendance

    Input: Mark Present Student Roll No.

    Output: Attendance Taken Successfully

    Processing: First Faculty select his subject from list and Load Students for that subject, then Faculty takes Attendance of Students. This is done by selecting checkbox corresponding to present student.

    After successfully taking Attendance Faculty Click to Save Attendance.

- **Student and Subject Info Module**

  Navigation:

  Once After Successful Login User redirected to Dashboard Page.

  - User Selects To View his Subject and Students per Subject

  Description: This Module includes

  1. List of allocated subject for faculty and
  2. Students List For subjects allocated to faculty.

- **Report Module:**

This module is useful for faculty to view date-wise attendance for students who are enrol for subject (of faculty).

Description: With help of report module Faculty can update student attendance and take pending attendance if any pending attendance exist. Faculty can view/track each student attendance percentage out of total Lecture conducted for subject.

- **Admin Module:**

Admin module includes two parts:

1. Faculty Panel: Add/Update/Delete Operation for faculty data
2. Student Panel: Add/Update/Delete Operation for Student data

Input: Admin credentials

Output: Verified successfully

Processing: Authenticate admin credentials in the database with entered Credential, if it is verified successfully and user has role of admin then user will redirect to admin dashboard.

Description: Admin Module is used to perform crud operation on student and faculty data via user interface. Admin user is also assigned as faculty or user in database.

- **Faculty Panel:**
  - Admin can add new faculty and assigned him subject according to department.
  - Admin can activate and deactivate user or faculty account by edit faculty account.
  - Admin can delete faculty data (account).

- **Student Panel:**
  - Admin can add new student by branch id and roll no in student table. By default all the subjects in their

department which are mandatory are assigned to newly created student.

- Admin can edit student info which includes name, branch id and roll no.
- Admin can delete Student info if student no longer exist.

- **Student-data Module:**

  This Module is available for student to view his attendance for enrolled subject.

  Input: Student Id

  Output: Subject Wise Attendance Report.

  Description: Student can view lecture attended to total lecture ratio and its corresponding percentage of his attendance for each subject. Student can also view present days out of total working days and his overall percentage of attendance (enrolled subject).

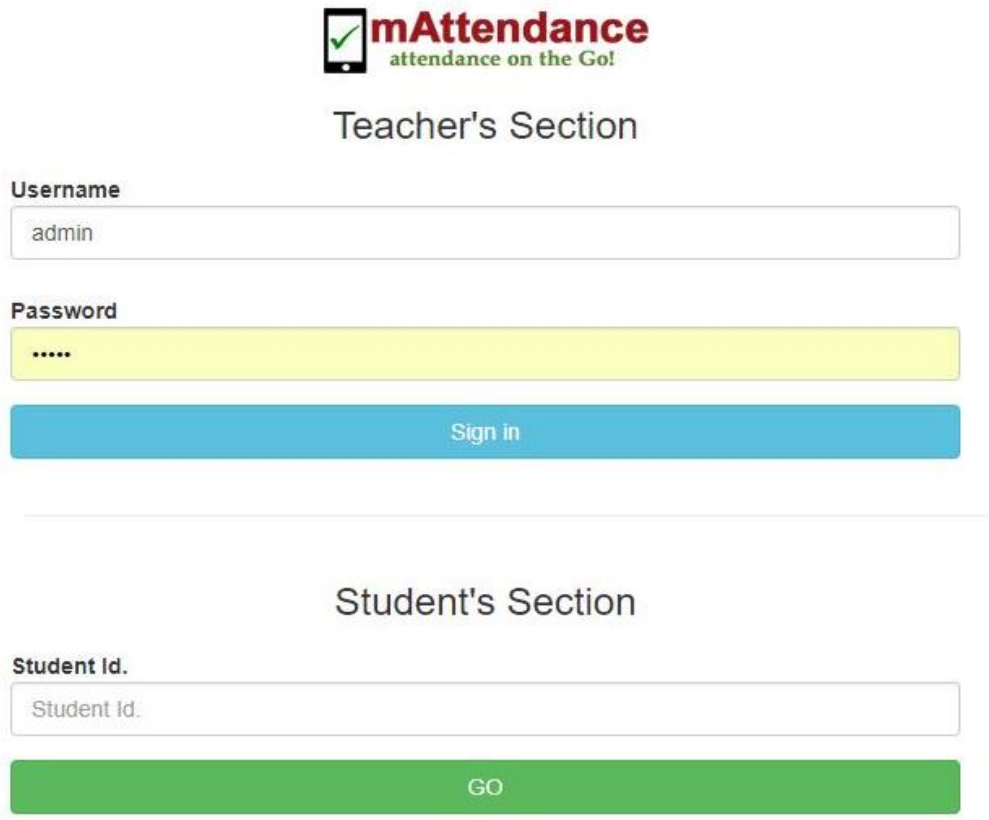
7. Test Cases:

Sr. No.	Test Case	Input	Expected/Actual Output	Remark Test pass or fail?
1	User(Faculty) Log In	Valid User Credentials: Username: apv Password: apv	Logged In Successfully	pass
2	User Log In	Invalid User Credentials Username:user1 Password:user1	Invalid Credential -Try Again	pass
3	User Log In	Empty Username OR Password Or Both	Invalid Credential -Try Again	pass
4	Take Attendance	Select Subject and Load Student -No Student	No Student List displayed	pass
5	Take Attendance	Select Subject and Load Student - Students Present	Student list displayed	pass
6	Take Attendance	Select Future Date To take Attendance	Invalid Date Selection	pass
7	Take Attendance	Select Current Date To take Attendance	Display Student List	pass
8	Take Attendance	Select Past Date To take Attendance	Display Taken Attendance or Take Attendance	pass
9	Report	From Date and To Date From Date less than current date To Date up to current date	View Report	pass
10	Report	From Date and To Date From Date less than current date To Date is future date	Invalid Date Selection	pass
11	Report	From Date and To Date From Date is Future date To Date is current date(or any)	Invalid Date Selection	pass
12	Assigned To You	Select Subject or Student Link	Display Allocated Subject and Students	pass
13	Student Login	Valid Student Id	Display Student Attendance Report	pass
14	Student Login	Invalid Student Id	Invalid Student Id	pass
15	Admin Login	Valid Admin credentials	Login Successfully	pass
16	Add Faculty details	Faculty Details	Faculty added Successfully	pass
17	Edit Faculty Details	Valid Faculty details, Constraints: dept_ID btw 1 to 4 Status: 0 or 1	Faculty edited successfully	pass
18	Edit Faculty Details	Invalid Faculty details	Invalid faculty credentials	pass

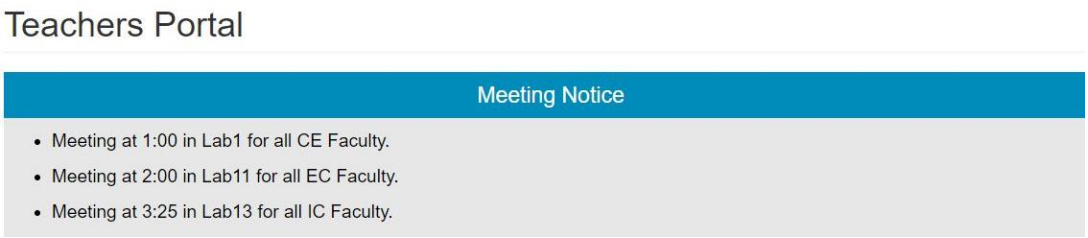
19	Add Student details	Valid Student details Constraints: dept_ID btw 1 to 4, Roll no is not negative	Student added successfully	pass
20	Edit student details	Valid student details	Student edited successfully	pass

8. SCREENSHOTS

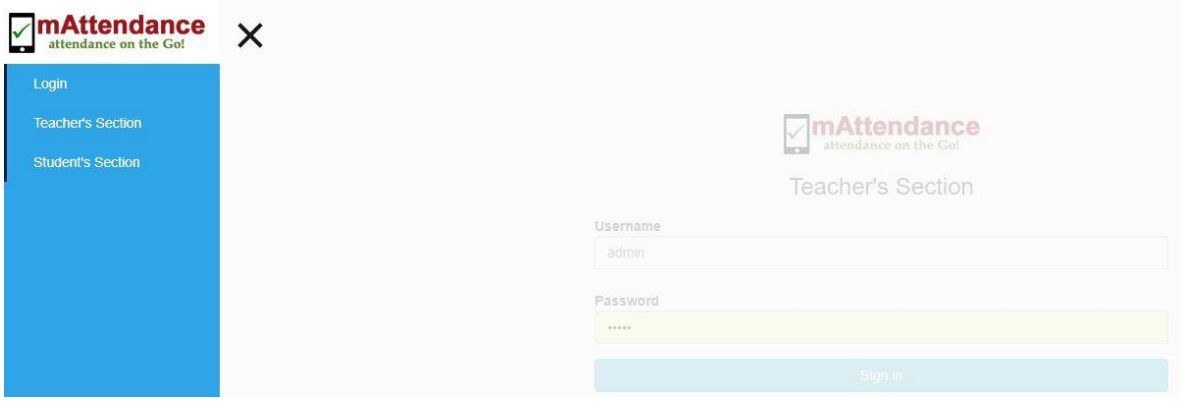
- **Faculty/Student Log-In Page (Front Page)**



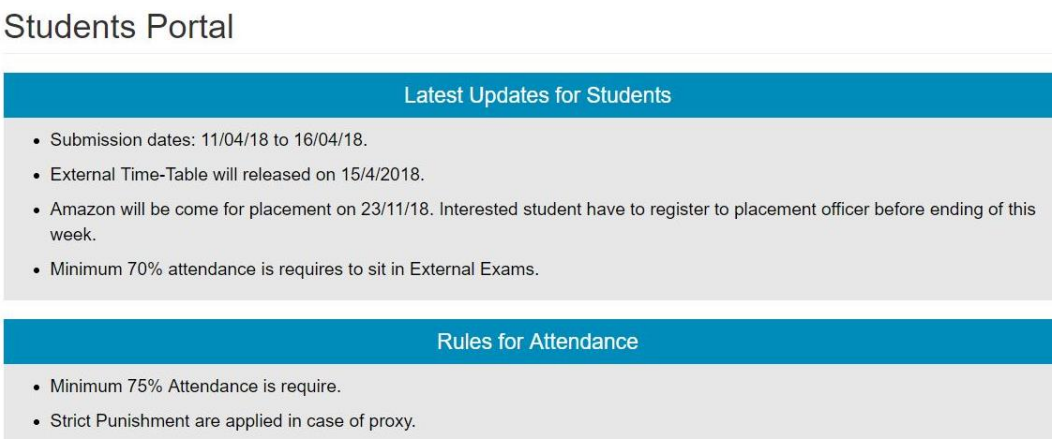
- **Navigation Without Login**



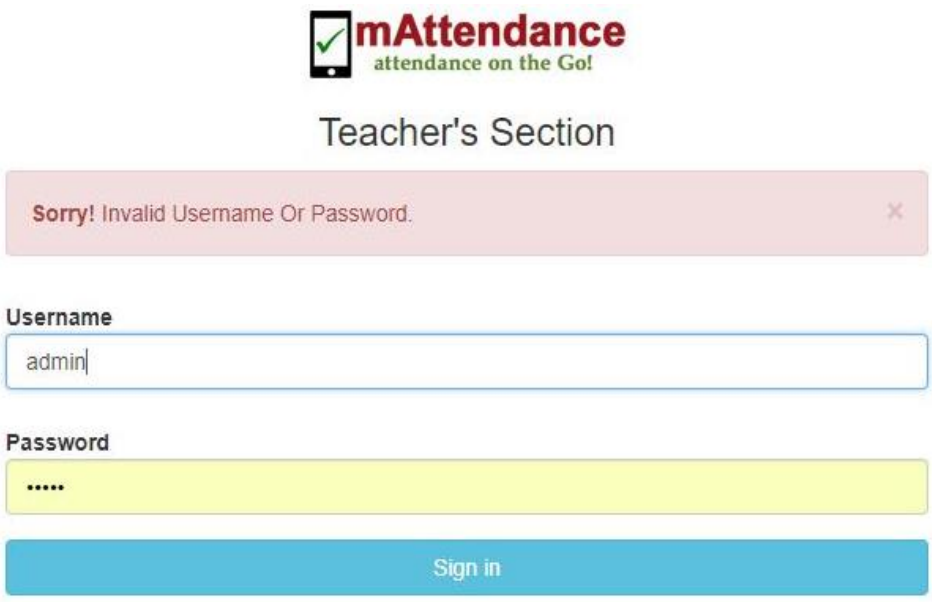
- **Teacher's Section**



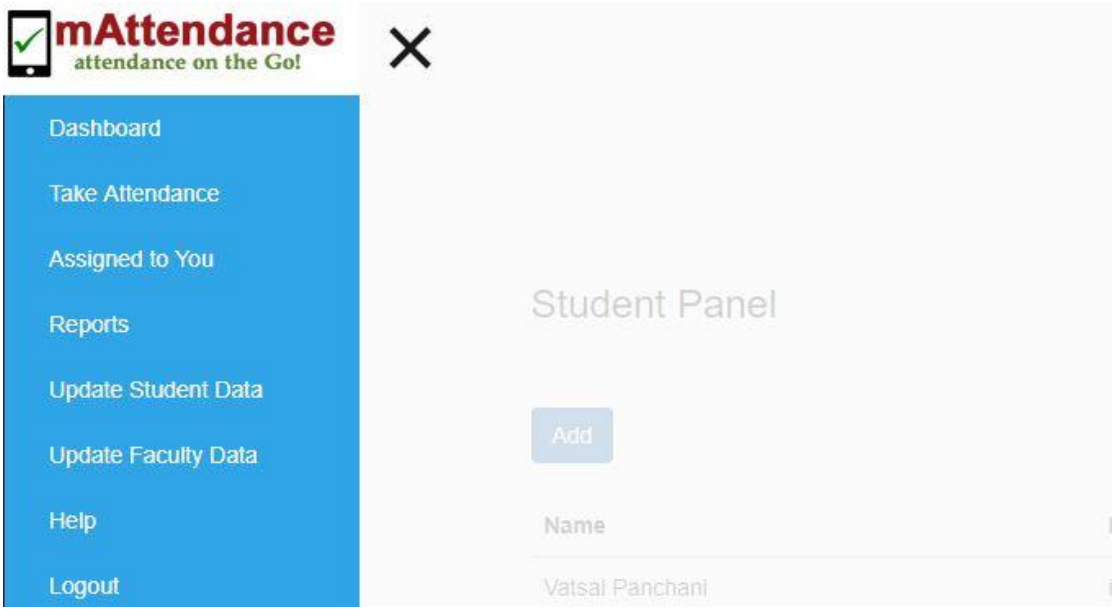
- **Student's Section**



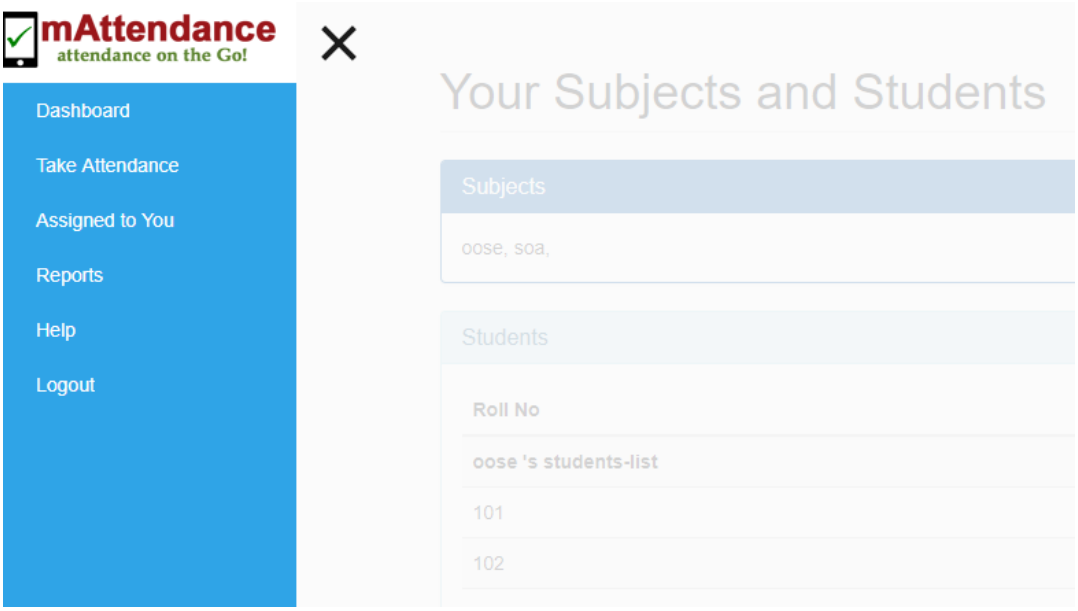
- **Invalid Username or Password**



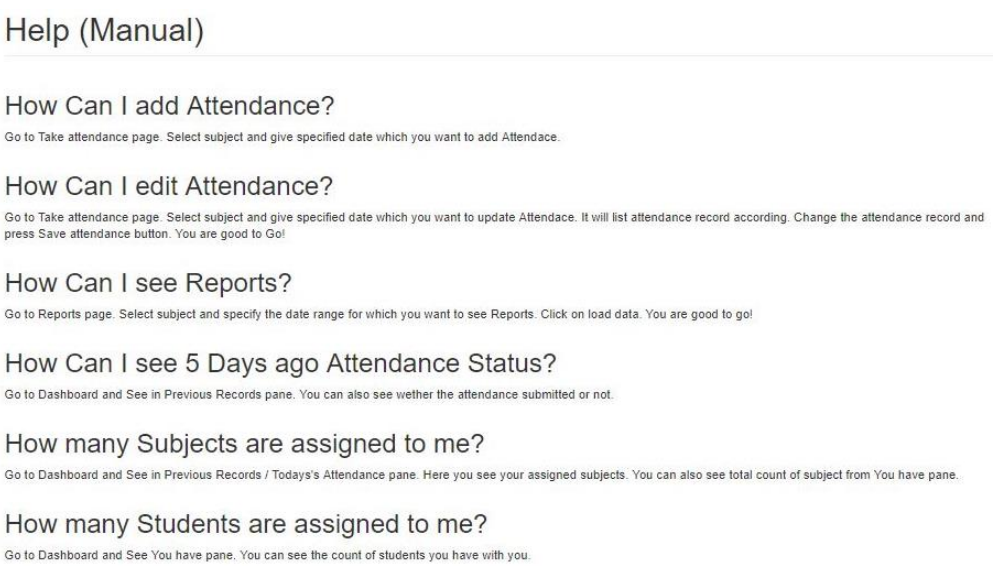
- **Navigation Bar on Admin Login**



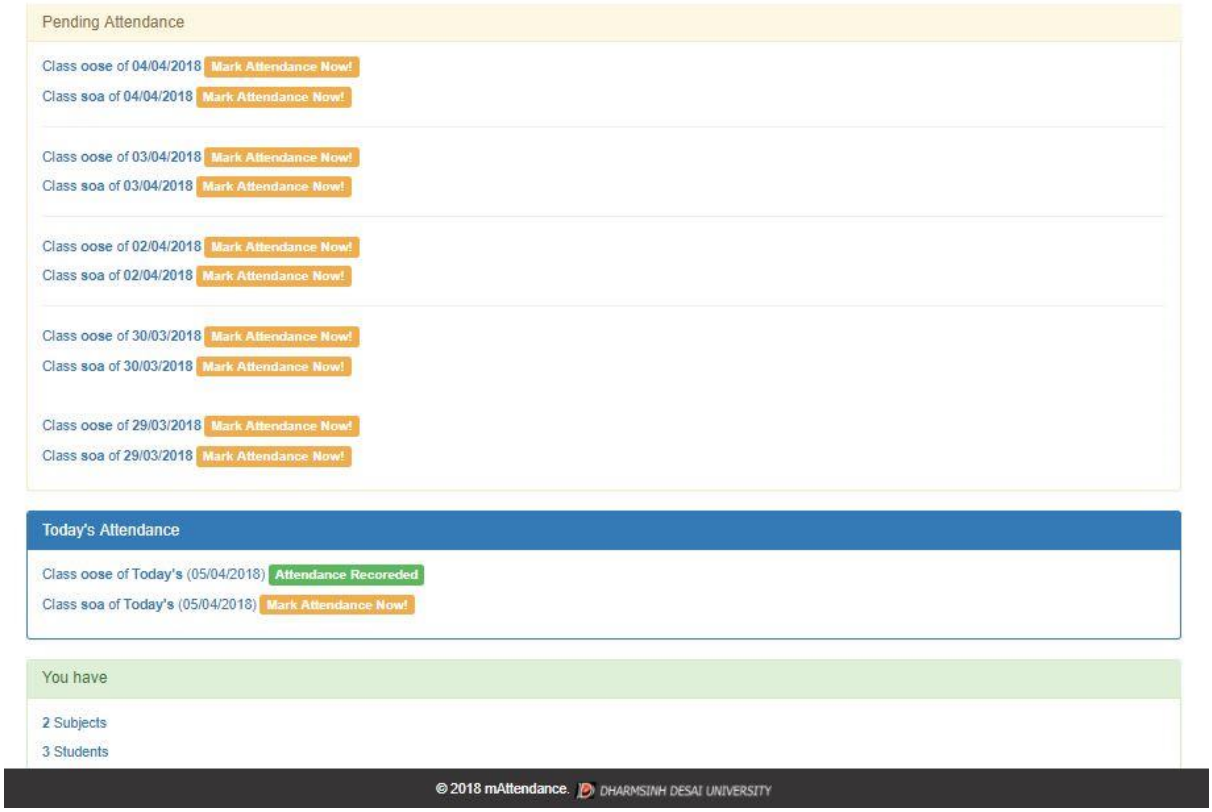
- **Navigation On User Login**



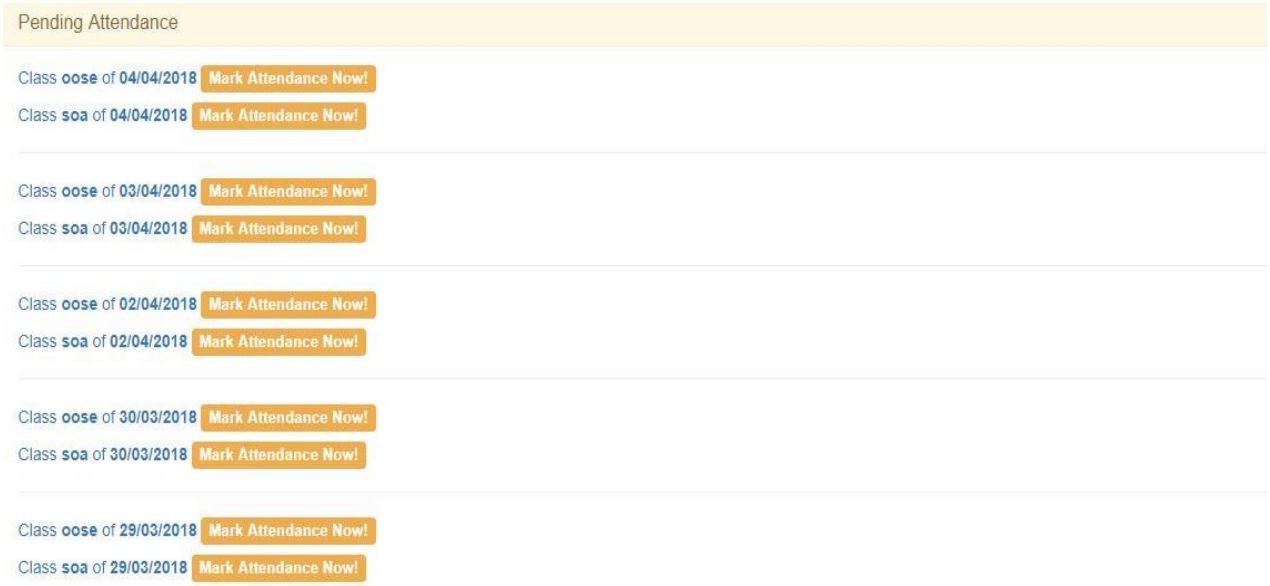
- **Help**



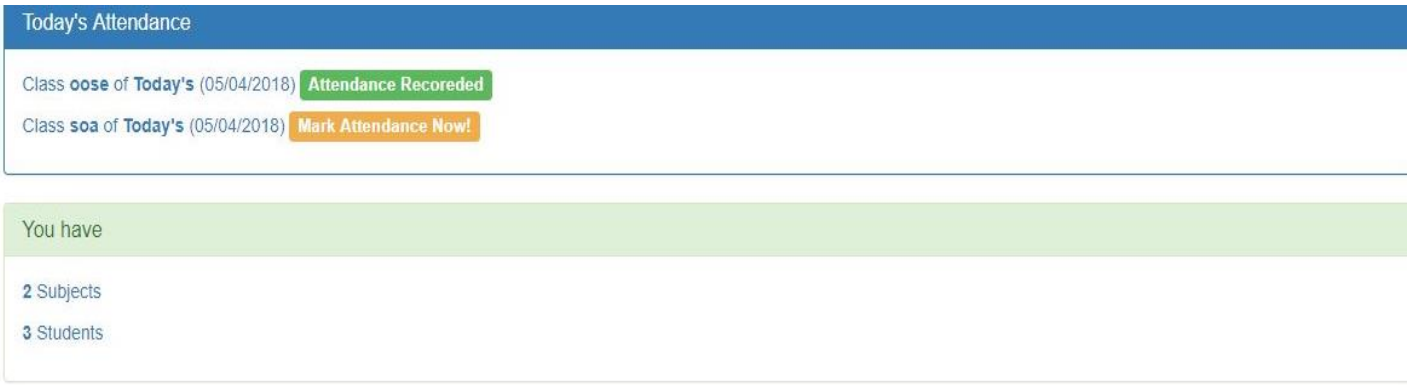
- **Dashboard**



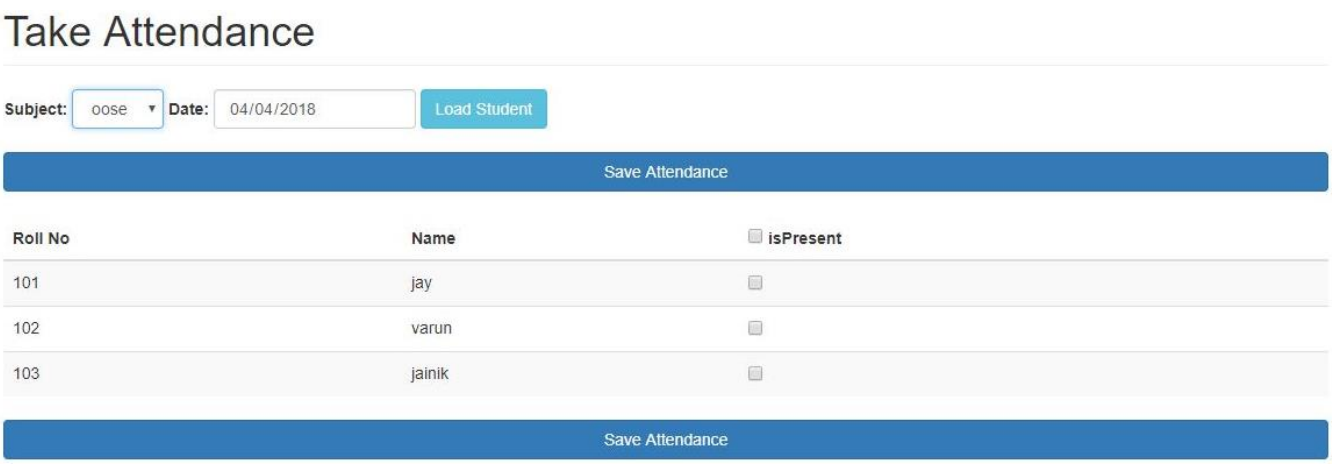
- **Last 5 Pending Attendance According to User-Subject**



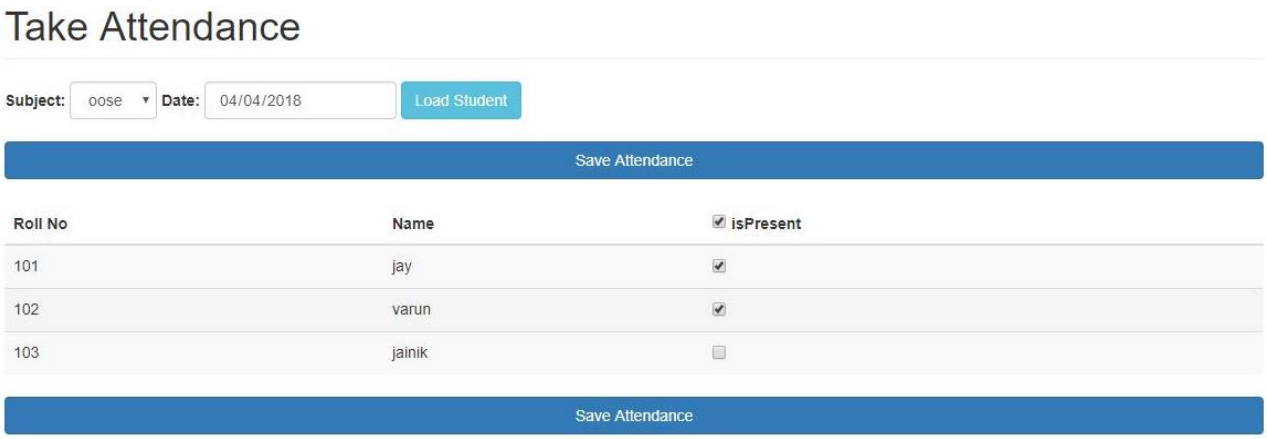
- **Take Today's Attendance**



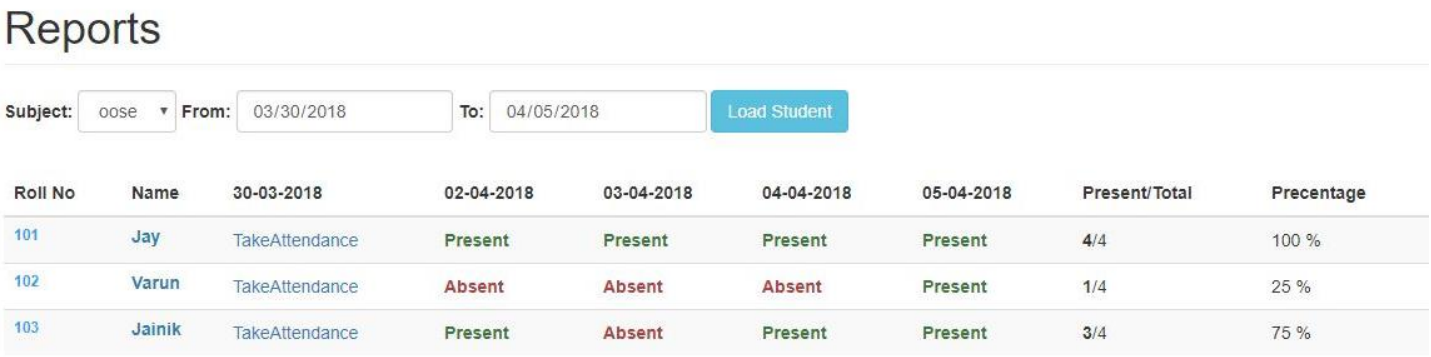
- **Take Attendance Page**



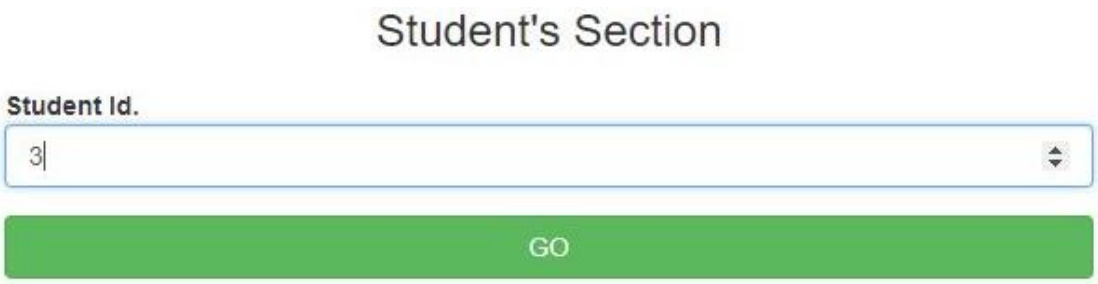
- **Take Attendance Page(Select All)**



- **Report Page**



- **Student Login**



- **Student Attendance View (Subject-Wise)**

Jainik (103) 'S Attendance

Subject	29-03-2018	02-04-2018	03-04-2018	04-04-2018	05-04-2018	Total	%
Sdp	Not Taken	Not Taken	Not Taken	Not Taken	Not Taken	0/0	0 %
Oose	Present	Present	Absent	Present	Present	4/5	80 %
Cn	Not Taken	Not Taken	Present	Absent	Present	2/3	66.67 %
Nis	Not Taken	Not Taken	Not Taken	Not Taken	Not Taken	0/0	0 %
Tafl	Not Taken	Not Taken	Not Taken	Not Taken	Present	1/1	100 %
Soa	Not Taken	Present	Not Taken	Present	Not Taken	2/2	100 %

Summary Of Attendance

Present Days out of Working Days: **9/11**

Attendance Percentage: **81.82 %**

- **Update Faculty Panel**

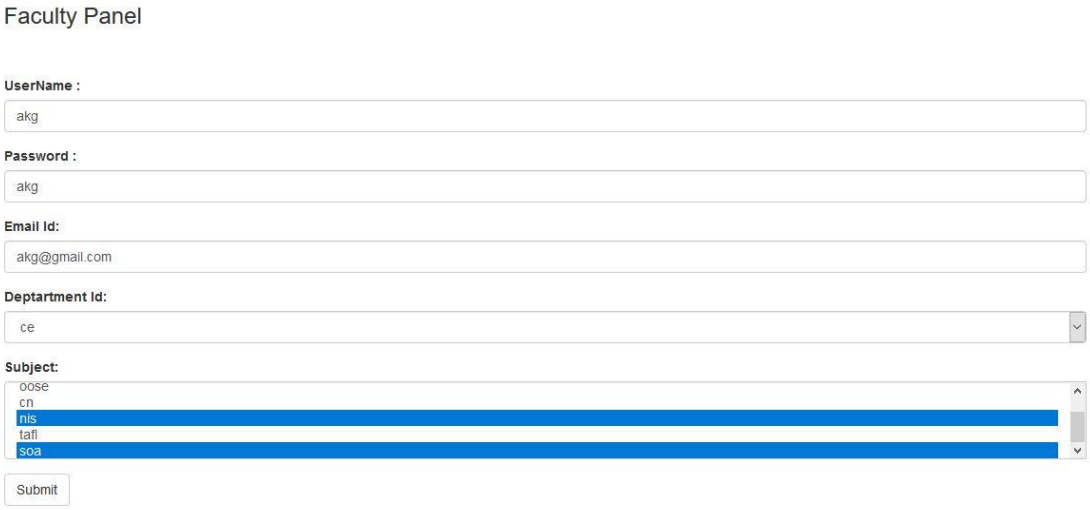
Faculty Panel

Add

Name	dept id	email	status	subjects	
itf1	it	itfac1@gmail.com	active	adv. java,	Edit Delete
admin	it	admin@gmail.com	active	daa, dsa,	Edit Delete
itf2	it	itf2@fgmail.com	active	mapi,	Edit Delete
mhf1	mh	mh1@gmail.com	active	fluid mech., thermodynamics,	Edit Delete
chf1	ch	ch1@gmail.com	active	mass transfer op.,	Edit Delete
chf2	ch	ch2@gmail.com	active	thermal engg.,	Edit Delete
apv	ce	apv@gmail.com	active	oose, soa,	Edit Delete
jhb	ce	jhb@gmail.com	active	oose,	Edit Delete
spm	ce	spm@gmail.com	active	cn,	Edit Delete

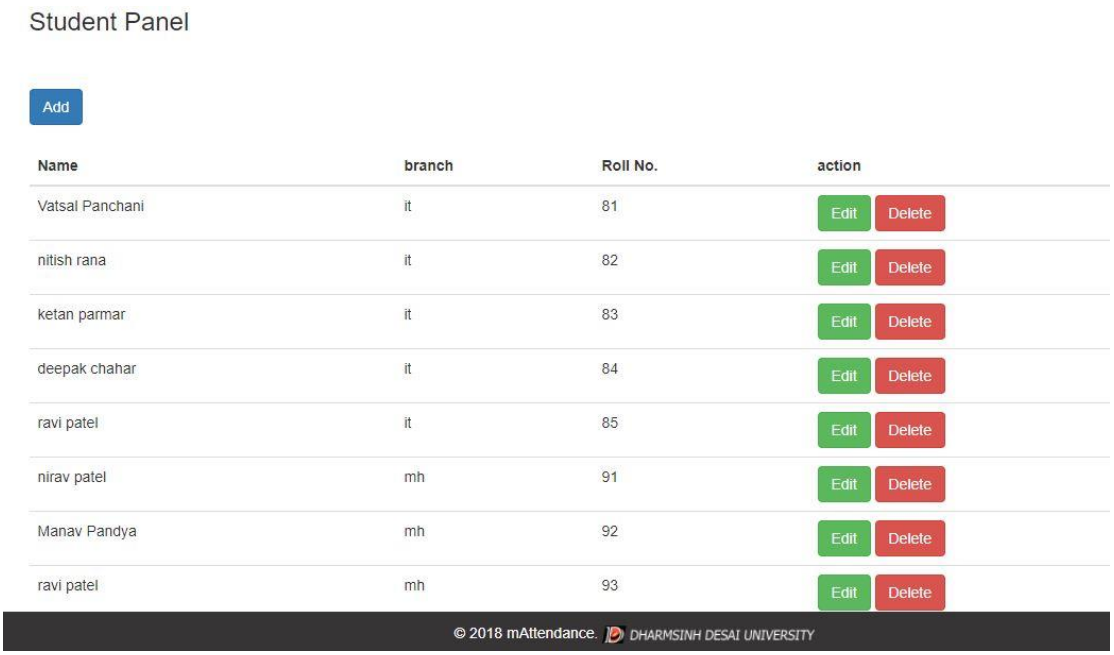
- **Add Faculty Member details**

- **Add Faculty Member details**



- **Edit Faculty details**

- **Update Student Panel**



- **Add Student**

- **Edit Student Details**

- **About**

Goal:

For Institute and Colleges who are still using pen-paper based attendance for students and manually manage Attendance which is time-consuming,not well managed,waste of paper and there may be chance of mistake ,To Solution to that The Attendance Management System provides facility and ease of taking attendance in digital way over extra efforts for manage attendance for each students and support to go green.

9. Deployment Steps:

1. First of all create Faculty authentication tables.
    - Faculty table
    - Student table
    - Manually insert all Faculty and Student credentials in table.

2. Create Department table for add data of department and create relationship between faculty-department and subject-department.
    - Create Department table and add department data.
    - Create Department-Faculty relationship table.
      [For add faculty to department]
    - Create Department-Subject relationship table.
      [For add subject to department]

3. Now create Subject table for add subject and create relationship between subject and faculty.
    - First create Subject table to add subject.
    - Create Faculty-Subject relationship table.
      [For allocating subject to faculty]
    - Create Student-Subject relationship table.
      [For subject taken by student]
    - Create Attendance table for store attendance data.

4. After following ordering of above step while creating table using SQL file of database.
    - The database should be now up and running, based on Faculty registration faculty can able to login and use System Functionality.
    - Student can able to login and use System functionality using his ID.

5. Deployment is complete.

10. Conclusion

Hereby, we declare that the functionality implemented in this system was performed by understanding all the modules.

This project aims to provide online taking attendance mechanism which was fulfilled to a greater extent. All the UML designs where prepared before implementation and the coding was done accordingly. After the coding was completed, comprehensive testing was performed and the results were provided in the report. Unit testing of all modules were done and later, Integration Testing was also performed.

11. Limitations and Future Enhancements

➢ Limitations

- Administrator required for add Faculty, Student and Subject information.
- Security issues as only that student view attendance, only providing ID.
- JavaScript framework must be supported by browser to run application.

➢ Future Enhancements

- Admin user interface for add information about faculty and student.
- Link application to bio-metric based attendance system.

12. Bibliography:

❖ **Web Tutorials:**

- www.w3schools.com
- http://php.net/manual/en/tutorial.php
- https://www.mysql.com/
- https://jquery.com/

❖ **For Bootstrap**

- https://getbootstrap.com/
- https://bootswatch.com/