**A Project report**

**on**

***Online Teacher Diary Management System***

Submitted in partial fulfillment of the requirements of the award of the   
degree of Bachelor of Computer Application



Guide Submitted by

(Dr. Nitin Deepak) Ravi Pandey

Preeti Kushwaha

Rajkumar Gupta

Sapna Bhatt

Satyam Shrivastav

Submitted to

Amrapali Institute of Applied Sciences, Haldwani

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Dr Nitin Deepak Sir for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my project team & member of Amrapali group of institute for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to my team members (Ravi Pandey, Preeti Kushwaha, Rajkumar Gupta and Satyam Shrivastav) for giving me such attention and time.

CERTIFICATE

This is to certify that this project entitled ” Online Teacher Diary Management System” Submitted in partial fulfillment of the degree of Bachelor of Computer Application to the Dr. Nitin Deepak, Associate Professor, Amrapali group of institute, done by Sapna Bhatt, Roll No. 1661330119 is an authentic work carried out by her at amrapali group of institute under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

DECLARATION

We hereby declare that the project work entitled “TEACHER’S DIARY” is being submitted to Faculty of Computer Science and Application is the authentic record of our own project work done under the guidance of Mr. Nitin Deepak (Astt. Professor) FCSA, Amrapali Group of Institutes, Haldwani.

Team: Signature of Team

Ravi Pandey

Sapna Bhatt

Preeti Kumari Kushwaha

Rajkumar Gupta

Satyam Kumar Shrivastav

CERTIFICATE

This is to certify that the project titled “TEACHER’S DIARY” is the bonafide work carried out by Ravi Pandey, Preeti Kumari Kushwaha, Sapna Bhatt, Rajkumar Gupta, Satyam Kumar Shrivastav students of Bachelor of Computer Applications, Amrapali Institute of Applied Sciences, Haldwani affiliated to Kumaun University, Nainital, India, during the academic year 2018-19, and that the project has not formed on the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

**Signature of the Guide**

**Table of Contents**

|  |
| --- |
| Title Page |
| Declaration of the Student |
| Certificate of the Guide |

1. Introduction 1

1.1 Overview 3

2. The Overall Description 3

2.1 Product Perspective 4

2.1.1 System Interfaces 4

2.1.2 Interfaces 4

2.1.3 Hardware Interfaces 4

2.1.4 Software Interfaces 5

2.1.5 Communications Interfaces 5

2.1.6 Memory Constraints 5

2.1.7 Operations 5

2.1.8 Site Adaptation Requirements 5

2.2 Product Function 5-7

2.3 User Characteristics 7 2.4 Constraints 7

3. Specific Requirements 7

3.1 External Interfaces 8

3.2 Functions 9

3.3 Performance Requirements 9

3.4 Logical Database Requirements 9

3.5 Software System Attributes 10

3.5.1 Reliability 10

3.5.2 Availability 10

3.5.3 Security 11

3.5.4 Maintainability 11

3.5.5 Portability 11

4. Change Management Process

5. Document Approvals

6. Supporting Information

INTRODUCTION

A teacher diary is a diary where a teacher records what happens in their classes and their thoughts about it. Teacher diaries are a good way to start a course of development, by collecting information about what goes on in our classroom, and by analyzing and evaluating this information, we identify and explore our own practices and underlying beliefs. This may then lead to changes and improvements in our teaching.

Teachers play vital roles in the lives of the students in their classrooms. Teachers are best known for the role of educating the students that are placed in their care. Beyond that, teachers serve many other roles in the classroom. Teachers set the tone of their classrooms, build a warm environment, mentor and nurture students, become role models, and listen and look for signs of trouble.

Teacher’s diary is an automated version of manual teacher diary. The design and implementation of a comprehensive teacher diary and user interface is to replace the current paper records. College Students and faculty are able to directly access all aspects of a student’s academic progress through a secure, online interface embedded in the college’s website.

The system utilizes user authentication, displaying only information necessary for an individual’s duties. Additionally, each sub-system has authentication to allow authorized users to create or update information in that subsystem. All data is stored securely on SQL servers managed by the college administrator and ensures highest possible level of security.

1.1Purpose:

1. We have developed the Teacher’s diary. In this portal all details of students are handling

by faculty.

2. This portal will be most useful for Teacher’s for keeping track record of various students.

3. Each teacher must maintain a teacher’s diary which will serve as a comprehensive

Record of his/her work throughout the year. The diary should contain the syllabus

of the subjects taught, Lesson/Lecture plan, Internal test, The weekly plan of work

Attendance of students, The projects undertaken, The response from the students,

the contacts with parents, etc.

4. The work for each week should be pre-planned and entered in the diary and this should

Be submitted to HOD/ V. principal/ principal on the last working day of each week day.

5. The data once entered will be able to edited/deleted as per required when there will be

Vast entries of data.

Scope

* The system "Teacher’s Diary" can be used by any school who wants to make the

Communication between parent and teacher effective. Using this system parent

take care of his child’s study. Parent can also get the regular details of the school.

* There are no banners for advertisement on the site.
* This system would be able to run only in Windows platform.
* Supported only by My SQL database.
* There will be no security of data.
* Supportive language is only English.

1.2 Definition, Acronyms and Abbreviations:

ADMIN: Administrator.

DEO: Data entry operator.

RAM: Random access memory.

ROM: Read only memory.

GUI: Graphics user interface.

GB: Giga Byte.

SRS: Software Requirement and Specification.

DFD: Data Flow Diagram.

E-RD: Entity Relationship Diagram.

E-Mail: Electronic Mail.

OS: Operating System.

DBMS: Database Management System.

PHP: Hypertext preprocessor.

SQL: Structured Query Language.

WWW: World Wide Web.

HTML: Hyper Text Markup Language.

CSS: Cascade Style Sheet.

JS: Java Script.

CI: Code Igniter

HDD: Hard Disk Drive.

DD: Data Dictionary.

PERT: Program Evaluation Review Technique.

DOC: Date of Commencement

ADOC: Actual Date of Commencement

EDOC: Expected Date of Commencement

Reference

<https://www.teachingenglish.org.uk/article/teacher-diary>

[www.google.com](http://www.google.com)

[www.academia.edu/23567721/Teaching\_Diary\_A\_Tool\_For\_an](http://www.academia.edu/23567721/Teaching_Diary_A_Tool_For_an) Effective Teaching

Overview

This project is aimed at developing a web application of teacher’s diary for the faculty of college. The system is a web application that can be accessed throughout the organization with proper login provided. This system can be used as a web application for the Teacher’s of the college to manage the student information. The key feature of this project is that it is a onetime registration. Our project provides the facility of maintaining the details of the students. Administrator logging in may also search any information put up by the students. This project will aid colleges to practice full IT deployment. This will also help in fast access procedures in student and faculty related activities.

2. THE OVERALL DESCRIPTION

The role of the school and teachers has always been vital in the all round personality development of the students. In fact, a teacher is a role model influencing every facet of the student’s growth and developing their innate potentials, in addition to being a motivator, guide and friend.

Besides, the teacher of today is also responsible to enable and empower the learner to emerge as a competent youth, ready to take on the challenges of the rapidly changing world. Hence, it is imperative that the teacher continuously upgrades his/her knowledge and methodology in order to enhance the quality of teaching. If the quality of teaching is good and the commitment of teachers is high, the standard of the Institution is bound to rise.

2.1 Product Perspective

The software is a self-contained and an independent product with proper user interface.

The application uses the centralized databases which contains all the information.

2.1.1 System Interface

Apache will be used as web server. The user inputs data via the web server

Using HTML forms. The actual program that will perform the operations is

written in PHP.

2.1.2 Interfaces

The new system shall provide a very intuitive and simple interface to the user

and the administrator, so that the user can easily navigate through pages,

assignments, groups and sub- groups, start discussion threads, blogs, survey,

upload assignments, share data, old papers sharing and the administrator

can easily manage groups and revoke user permission.

2.1.3 Hardware Interfaces

a) Server side the web application will be hosted on a web server which is

listening on the web standard port, port 80.

b) Client side Monitor screen – the software shall display information to

the user via the monitor screen Mouse – the software shall interact

with the movement of the mouse and the mouse buttons. The mouse

shall activate areas for data input, command buttons and select options

from menus. Keyboard – the software shall interact with the keystrokes

of the keyboard. The keyboard will input data into the active area of the

database

(1) Processor: Intel Core I 3  
 (2) RAM: 2 GB   
 (3) Hard Disc: 20 GB

2.1.4 Software Interfaces

Server side- An Apache web server will accept all requests from the client

and forward it. Accordingly a database will be hosted centrally using

My SQL.

Client side- An OS which is capable of running a modern web browser

which supports JavaScript and HTML5.

* + 1. Front End Tool: PHP
    2. Back End Tool: My SQL
    3. Development Tools: XAMPP Server
    4. Technology: Model Control View technology
    5. Browser: Any browser

vi. DocumentationTool:MicrosoftOffice2016

2.1.5 Communication Interfaces

The HTTP or HTTPS protocol(s) will be used to facilitate communication

between the Client & Server.

2.1.6 Memory Constraints

Memory constraints will come into play when the size of My SQL grows

to a considerable size**.**

2.1.7 Operations

The product shall have operations to protect the database from being

corrupted or accidentally altered during a system failure.

2.1.8 Site Adaptation Requirements

The component will be adapted to the overarching system at the conclusion

of the system creation

.

2.2 Product Function

**Data entry operator:**

* On clicking the Teacher’s diary, a sign-in screen is displayed. If the user is a

new user, he/she will click the sign-up button on the screens; otherwise will

sign in with the specified details.

* On successfully filling all the fields in signup, a mail with the password is

sent at user’s email id. A user can exit anytime during sign-in/sign-up.

* Now, user has signed in and a home screen is displayed with the some buttons

on it: Student details, Faculty details, Attendance, Internal marks, Assignment,

Time table, Generate report, etc.

* Data entry operators maintain the student details and faculty details.

From, any screen user can switch to home screen and can log out.

**Student:**

* On clicking the Teacher’s diary, a sign-in screen is displayed. If the user is a new user, he/she will click the sign-up button on the screen; otherwise will

sign in with the specified details.

* On successfully filling all the fields in signup, a mail with the password is

sent at user’s email id. A user can exit anytime during sign-in/sign-up.

* Now, user has signed in and a home screen is displayed with the some

buttons on it: Attendance, Internal marks, Assignment, Time table etc.

* Attendance will display attendance of the student.
* Internal marks will display internal marks of the student.
* Assignment button will show the assignment screen with fields: assignment topics, course, semester, and submission date.
* Time table will display time table.
* From, any screen user can switch to home screen and can log out.

**Faculty:**

On clicking the Teacher’s diary, a sign-in screen is displayed. If the user is a new user, he/she will click the sign-up button on the screen; otherwise will sign in with the specified details.

* On successfully filling all the fields in signup, a mail with the password is sent at user’s email id. A user can exit anytime during sign-in/sign-up.
* Now, user has signed in and a home screen is displayed with the some buttons on it: Attendance, Internal marks, Assignment, Time table, Weekly report etc.
* Attendance button will take the user to the attendance screen showing fields: students name, roll no., status, etc.
* Internal marks will take the user to the internal marks screen showing fields Course name, semester, student name, roll no., student marks etc.
* Assignment button will show the assignment screen with fields: assignment topics, course, semester, and submission date. By clicking the upload button automatic message is sent to the respective batch students. This uploading will not be shown after the submission date.
* From, any screen user can switch to home screen and can log out.

**Administrator:**

Since, admin is a user too. He will sign-in like any other user.

* On clicking the Teacher’s diary, a sign-in screen is displayed. User sign-in.

(He does not need to signup).

* Now, user has signed in and a home screen is displayed with the some

buttons on it: Attendance, Internal marks, Assignment, Time table,

* Manage staff, Manage Feedback etc.
* Manage staff will take the user to the staff screen.
* Manage Feedback will take the user to the feedback screen.
* From, any screen user can switch to home screen and log out.

2.3 User Characteristics

* Education level: At least graduate should be comfortable with

English language.

* ­­­­­­­­­­­­­­­­­­Technical expertise: Should be comfortable using

general-purpose applications on a computer.

2.4 Constraints:

* Sign-in and password are used for the identification of user.
* Admin needs to update after every session.

3. SPECIFIC REQUIREMENTS

To overcome the disadvantages of current study system we are going to propose an

application name as "School Diary". Some of the Features that are included into our system are:

1. Announcement of events and circular.

2. Display holiday list.

3. Check Student details.

4. Check faculty details.

5. Only authenticate person can enter into the system.

6. Parent can send the feedback.

7. Check timetable, subject material.

8. Display result and performance of the student.

9. Maintain lecture plan, lesson plan and weekly plan.

10. Maintain attendance details.

11. Maintain internal test details.

3.1 External Interface:

The following screen will be provided:

**Login Screen/Sign-in Screen:**

This module is for the Signing in of the existing user. This will be first screen that will be displayed. It will allow user to access different screens based upon the user’s role. Various fields available o this screen will be

* User ID
* Email ID/ Phone number
* Password
* Role: Will have the following values:

Administrator, Data Entry operator, Faculty, Student.

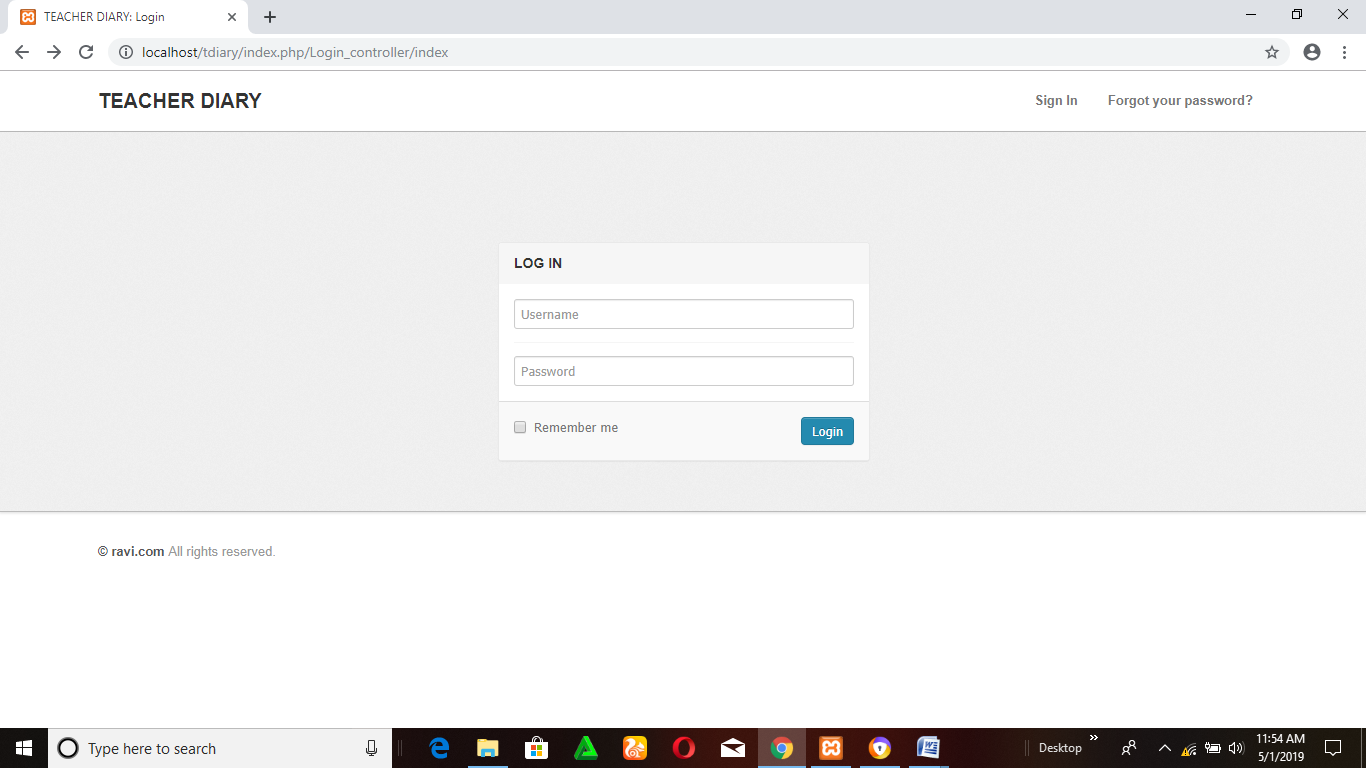


Figure 1: Login page

**Sign-up screen:**

This module is for the Signing up for the new user. Its fields are:

* User ID
* Email ID/Phone number
* Username
* Password
* Confirm Password



Figure 2: signup page

**Reset Screen**

This module deals with the changing of the password. Its fields are:

* Old Password
* New Password
* Confirm Password
* Reset

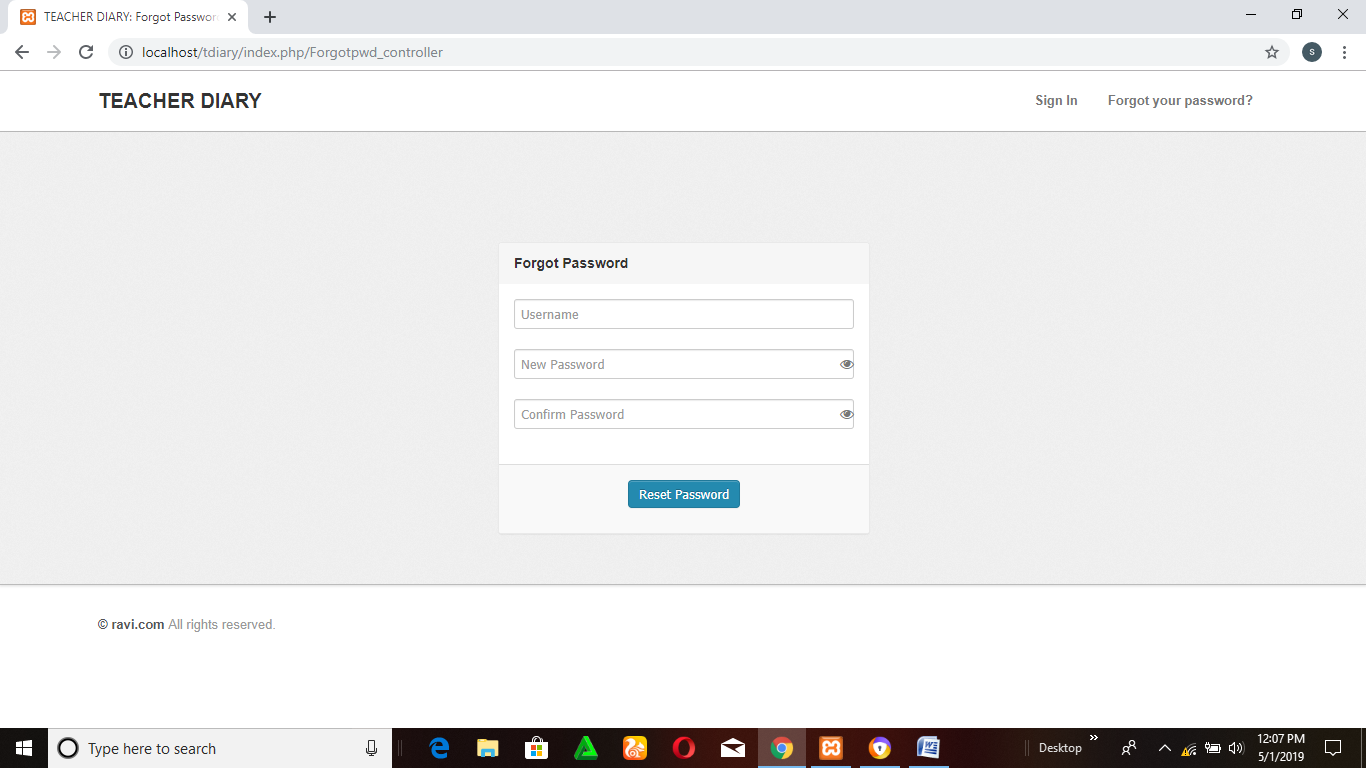


Figure 3: Reset page

3.2 Functions:

Planning is the best antidote for the nerves that many people feel when teaching a Subject for the first time or meeting a new group of students. It is also the only way to ensure that your educational objectives are achieved. Planning begins with thinking about how you would like your students to approach their learning in your subject, and what you would like them to understand, know or be able to do by the end of the session. Whether you are planning a subject for the first time, or reviewing an existing subject it is important to consider the effects of your teaching and assessment on students' learning.

3.3 Performance Requirement

Basically there are four types of users who deal with the system. The users are listed below

* + - Admin.
    - Data entry operator.
    - Faculty.
    - Student.

3.4 Logical Database Requirements

The following information will be placed in a database:

(1) **Student info:** Student name, Student id, Course, Semester, etc.

(2) **Faculty info:** Faculty name, Faculty id, etc

(3) **Subject info:** Subject name, Subject code, Semester, etc.

(4) **Marks info:** Student enrollment no., Semester, internal marks in each subject,

External Marks in each subject.

(5) **Lecture info:** Lecture no., date of commencement (DOC), Actual date of completion,

expected date of completion etc.

(6) **User account info:** User name, User ID, Password, Role.

3.5 Software System Attributes

3.5.1 Reliability

The “Teacher’s diary” software will be available for user only by their login details. During this time, it shall be operational for as long as is possible. Administrators will have 24-hour access to the system but admin also must have to login to the system.

3.5.2 Availability

The entire system should be available 99.9%. Redundancy in clustered hardware will be used to obtain web servers availability.

The system shall not leave any cookies on the client’s computer containing the user’s password. The system shall not leave any cookies on the client’s computer containing any of the user’s confidential information.

Data Storage The client’s web browser shall never display a client’s password. It shall always be echoed with special characters representing typed characters. The client’s web browser shall never display a client’s password after retrieving from the database. It shall always be shown with Boolean values true or false. The software system’s back-end servers shall never display a client’s password. The client’s password may be reset but never shown. The system’s back-end servers shall only be accessible to authenticated administrators. The system’s back-end databases shall be encrypted.

Maintainability System administrators will be given an administrative interface to manage news, category and users. Additionally, user interfaces will allow the configuration of news processing, and the application “look and feel”. Hardware maintenance on the server infrastructure will be maintained by the developer.

3.5.3 Security

1. Passwords will be saved encrypted in the database in order to ensure the user's privacy.

2. The user's IP will be logged.

3. The system will be protected against vulnerabilities such as SQL injection attacks.

3.5.4 Maintainability

My SQL is used for maintaining the database and the Apache server takes care of the site

. In case of a failure, a re-initialization of the program is recommended.

3.5.5 Portability

The application is Windows-based and should be compatible with other systems.

Apache PHP and My SQL programs are practically independent of the OS-system

which they communicate with. The end-user part is fully portable and any system

using any web.

OBJECTIVES

The objective of our application is to provide an effective and efficient way which enables Parent to check the performance regarding detail of their children, Teacher feed the details like Announcement of event and circular, exam schedule, attendance report, assignment shared, Timetable, Leave application, Parent note to Teacher, Teacher note to Parent, List of Holidays, Faculty Detail, Result and performance of Student. The primary goal of our application is to provide effective and smart system to the college using which parents can view their child performance, students can view the information shared by teacher and teacher can feed the information.

SCOPE

The system "online teacher diary management" can be used by any college who wants to make the sharing techniques cost effective and instant between parent, teacher and student. Using this system parent take care of his child’s study and their performance and also get their regular details of the college. Parent can also get the regular details of the college. Due to the use of "online teacher Diary application" they can make their child’s performance at a good level. Using feedback console teacher can also send complains of the student to the respective parent at that time, so it leads to the improvements of that student. By considering these entire factors one can make a college to a smart college.

1. There are no banners for advertisement on the site.

2. This system would be able to run only in windows platform.

3. Supported only by My SQL database.

4. There will be no security of data.

5. Supportive language is only English.

Theoretical Background Definition of Problem

**HARDWARE CONFIGURATION**

This system is developed on the following hardware configuration

1. Pentium IV processor
2. 128 Ram
3. Hard disk 80 GB
4. Microsoft Compatible Key Board
5. Scroll Mouse

**DESCRIPTION OF SOFTWARE**

This system is developed using the following software

1. Operating System: Windows XP
2. Technology: ASP.NET
3. Data Access Component ADO.NET
4. Back End: MS Access 2003

**Overview of the .NET Framework Web services**

Web services provide a web enabled user interface with tools that include various hypertext markup language (HTML) controls and web controls. Web services also handle various web protocols, security, when code is targeted for .NET, it is called managed code, which means that the code automatically runs under a “contract of cooperation” with the CLR. Managed code supplies the information necessary for the runtime to provide services such as memory management, cross-language integration, code access security, and the automatic lifetime control of all of our objects.

METHODOLOGY ADOPTED

(1)REQUIREMENT ANALYSIS:

1.1) Study of manual/existing system

In order to understand the requirements of system we visited five colleges. All the colleges doesn’t use any effective way for storing the performance detail of students and sharing the information among students and their regarding parents. Students are orally informed all the information by the teacher so the parents are not aware of that information. College related information like Circulars, Event Notification, Attendance record, Unit test, Results are written by teachers manually in the hard copy of the teacher dairy. Some parents doesn’t have the time to go college to check the college dairy regularly so the effective information gathering does not performed between parent and teacher. Sometimes it may happen that the parents are not aware about the performance of their child. Parents are also not aware about the regular activities of the college, till that there are no chances of improvement.

1.2) Analysis of problem and weaknesses of manual/existing system

After study of existing system we found following problems and weaknesses in the

System. Following are the some of the problems and weaknesses of current system:

(1) Poor Parent Teacher communication.

(2) Manually work.

(3) Parents are not aware about regular details of their child.

(4) Wait for the Parent Teacher Meeting to discuss with the teacher.

(2)REQUIREMENT GATHERING:

Gather requirements of new system

To overcome the disadvantages of current study system we are going to propose an application name as “"Online Teacher Diary Management". Some of the Features that are included into our system are:

* Announcement of events & circular.
* Display holiday list.
* Check faculty detail.
* Only authenticate person can enter into the system.
* Instant messaging.
* Display result and performance of the student.
* Check timetable, subject material.
* Parent can send the feedback.
* Fun with learning.
* One can view the achievement of the college.

Hardware and Software to be used:

Minimum Hardware Requirements

1. Intel core
2. RAM: 2GB
3. Hard Disc: 20GB

Minimum Software Requirement

1. Front End Tool: PHP,
2. Back End Tool: My SQL
3. Developments Tool: Wamp Server, Android Studio, JDK
4. Browser: Any browser
5. Documentation Tool: Microsoft Office 2010
6. Design Tool:

Maintenance

My SQL is used for maintaining the database and the Apache server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

Evaluation

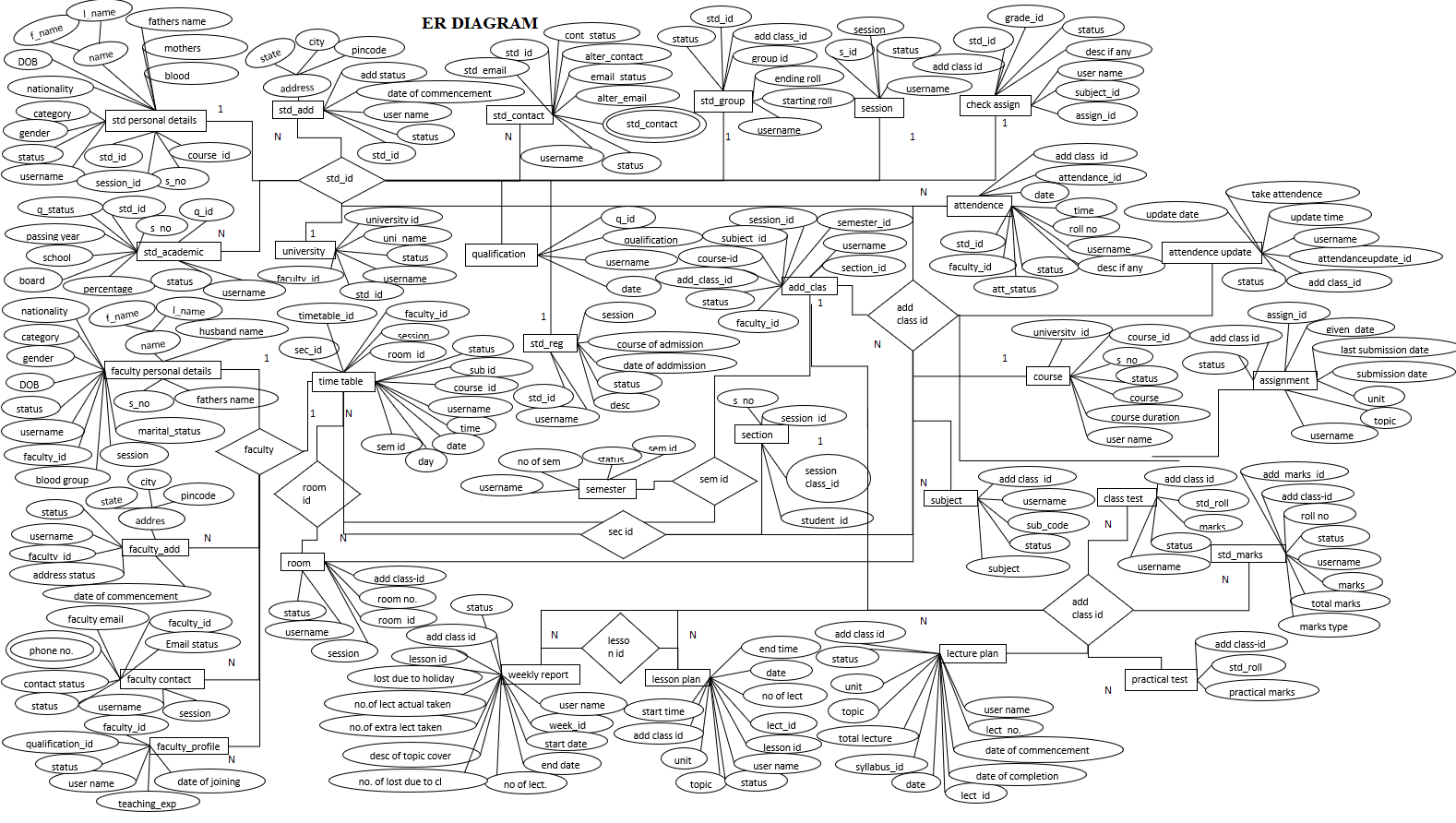
The teacher can capture the assessments per student and subject, both in numerical scale or assigning customized scales. This information can be consulted later to generate reports by student or manager of the college.

The software manages everything including maintenance, data, security, updates, etc.., and no to hire exclusive staff for the above functions.

COST AND BENEFIT ANALYSIS

It is a cost-effective solution with cheaper software licenses which are completely subscription based.

ER DIAGRAM



DATA DICTIONARY-

Add\_class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **extra** |
| **1** | **Add\_class\_id** | **Int** | **No** | **none** | **Auto\_increment** |
| **2** | **Faculty\_id** | **Varchar** | **No** | **None** |  |
| **3** | **Session\_id** | **Varchar** | **No** | **None** |  |
| **4** | **Course\_id** | **Varchar** | **No** | **None** |  |
| **5** | **Semester\_id** | **varchar** | **No** | **None** |  |
| **6** | **Subject\_id** | **Varchar** | **No** | **None** |  |
| **7** | **Section\_id** | **Varchar** | **No** | **None** |  |
| **8** | **Date\_of\_commencement** | **Date** | **No** | **None** |  |
| **9** | **Date\_of\_completion** | **Date** | **No** | **None** |  |
| **10** | **Syllabus\_pdf** | **Varchar** | **No** | **None** |  |
| **11** | **Username** | **Varchar** | **No** | **none** |  |
| **12** | **Status** | **boolean** | **no** | **None** |  |

Assignment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **default** | **Extra** |
| **1** | **Assignment\_id** | **Int** | **No** | **None** | **Auto\_increament** |
| **2** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **3** | **Assignment\_no** | **Int** | **No** | **None** |  |
| **4** | **Given\_date** | **Date** | **No** | **None** |  |
| **5** | **Submission\_date** | **Date** | **no** | **None** |  |
| **6** | **Unit** | **Int** | **No** | **None** |  |
| **7** | **Topic** | **Text** | **No** | **None** |  |
| **8** | **Status** | **Boolean** | **No** | **None** |  |
| **9** | **Username** | **varchar** | **no** | **none** |  |

Assignment\_checker

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Assignment\_checker\_id** | **Int** | **No** | **None** | **Auto\_increament** |
| **2** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **3** | **Assignment\_id** | **Int** | **No** | **None** |  |
| **4** | **Roll\_no** | **Int** | **No** | **None** |  |
| **5** | **Assignment\_status** | **Tinyint** | **No** | **None** |  |
| **6** | **Checker\_date** | **Date** | **No** | **None** |  |
| **7** | **Grade** | **Varchar** | **No** | **None** |  |
| **8** | **Status** | **Tinyint** | **No** | **None** |  |
| **9** | **Username** | **varchar** | **No** | **None** |  |

Assign\_subject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Assign\_subject\_id** | **Int** | **No** | **None** | **Auto\_increament** |
| **2** | **Subject\_id** | **varchar** | **No** | **None** |  |
| **3** | **Faculty\_id** | **Int** | **No** | **None** |  |
| **4** | **Session\_id** | **Int** | **No** | **None** |  |
| **5** | **Status** | **Tinyint** | **No** | **None** |  |
| **6** | **Username** | **Varchar** | **No** | **None** |  |

Attendance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Add\_class\_id** | **Int** | **No** | **None** | **Auto\_increament** |
| **2** | **Attendance\_id** | **Int** | **No** | **None** |  |
| **3** | **Date** | **Date** | **No** | **None** |  |
| **4** | **Time** | **Time** | **No** | **None** |  |
| **5** | **Roll\_no** | **Int** | **No** | **None** |  |
| **6** | **Attendance\_status** | **Varchar** | **No** | **None** |  |
| **7** | **Username** | **Varchar** | **No** | **None** |  |
| **8** | **Status** | **Boolean** | **No** | **None** |  |
| **9** | **Student\_id** | **Varchar** | **No** | **None** |  |
| **10** | **Faculty\_id** | **varchar** | **No** | **None** |  |
| **11** | **Description\_if\_any** | **text** | **No** | **None** |  |

Attendance\_update

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Attendancve\_update\_id** | **Int** | **No** | **None** | **Auto\_increament** |
| **2** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **3** | **Take\_attendance\_id** | **Int** | **No** | **None** |  |
| **4** | **Update\_date** | **Date** | **No** | **None** |  |
| **5** | **Update\_time** | **Timestamp** | **No** | **Current\_timestamp** |  |
| **6** | **Status** | **Boolean** | **No** | **None** |  |
| **7** | **Username** | **Varchar** | **No** | **None** |  |

Batch

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Batch\_id** | **Int** | **No** | **None** | **Auto\_increament** |
| **2** | **Batch\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Username** | **Varchar** | **No** | **None** |  |
| **4** | **Status** | **boolean** | **No** | **None** |  |

Course\_table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **University\_id** | **Varchar** | **No** | **None** |  |
| **2** | **Course\_id** | **Varchar** | **No** | **None** |  |
| **3** | **Name\_of\_course** | **Varchar** | **No** | **None** |  |
| **4** | **No\_of\_years** | **Int** | **No** | **None** |  |
| **5** | **Status** | **Boolean** | **No** | **None** |  |
| **6** | **Username** | **Varchar** | **No** | **None** |  |

Faculty\_address

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Faculty\_address\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Address** | **Varchar** | **No** | **None** |  |
| **3** | **Status** | **Varchar** | **No** | **None** |  |
| **4** | **City** | **Varchar** | **No** | **None** |  |
| **5** | **Pincode** | **Int** | **No** | **None** |  |
| **6** | **Address\_status** | **Boolean** | **No** | **None** |  |
| **7** | **Status** | **Boolean** | **No** | **None** |  |
| **8** | **Date\_of\_commencement** | **Date** | **No** | **None** |  |
| **9** | **Username** | **Varchar** | **No** | **None** |  |
| **10** | **Faculty\_id** | **Varchar** | **No** | **None** |  |

Faculty\_contact

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Faculty\_contact\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Faculty\_email** | **Varchar** | **No** | **None** |  |
| **3** | **Email\_status** | **Boolean** | **No** | **None** |  |
| **4** | **Faculty\_contact** | **Varchar** | **No** | **None** |  |
| **5** | **Faculty\_status** | **Boolean** | **No** | **None** |  |
| **6** | **Contact\_status** | **Boolean** | **No** | **None** |  |
| **7** | **Status** | **Int** | **No** | **None** |  |
| **8** | **Username** | **Varchar** | **No** | **None** |  |
| **9** | **Faculty\_id** | **varchar** | **No** | **None** |  |

Faculty\_personal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Faculty\_personal\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **First\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Last\_name** | **Varchar** | **No** | **None** |  |
| **4** | **Date\_of\_birth** | **Date** | **No** | **None** |  |
| **5** | **Gender** | **Text** | **No** | **None** |  |
| **6** | **Marital\_status** | **Varchar** | **No** | **None** |  |
| **7** | **Father\_name** | **Varchar** | **No** | **None** |  |
| **8** | **Husband\_name** | **Varchar** | **No** | **None** |  |
| **9** | **Nationality** | **Varchar** | **No** | **None** |  |
| **10** | **Category** | **Varchar** | **No** | **None** |  |
| **11** | **Bloodgroup** | **Varchar** | **No** | **None** |  |
| **12** | **Status** | **Boolean** | **No** | **None** |  |
| **13** | **Username** | **Varchar** | **No** | **None** |  |
| **14** | **Session** | **Year** | **No** | **None** |  |
| **15** | **Faculty\_id** | **Varchar** | **No** | **None** |  |

Faculty\_profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Faculty\_profile\_id** | **int** | **No** | **None** | **Auto\_increment** |
| **2** | **Qualification\_id** | **Varchar** | **No** | **None** |  |
| **3** | **Teacher\_experience** | **Int** | **No** | **None** |  |
| **4** | **Date\_of\_joining** | **Date** | **No** | **None** |  |
| **5** | **Status** | **Boolean** | **No** | **None** |  |
| **6** | **Username** | **Varchar** | **No** | **None** |  |
| **7** | **Faculty\_id** | **Varchar** | **No** | **None** |  |

Form\_type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **type** | **Null** | **Default** | **Extra** |
| **1** | **Form\_type\_id** | **int** | **No** | **None** | **Auto\_increment** |
| **2** | **Form\_type\_name** | **varchar** | **No** | **None** |  |
| **3** | **Icon** | **text** | **No** | **None** |  |
| **4** | **Status** | **boolean** | **No** | **None** |  |
| **5** | **Username** | **varchar** | **No** | **None** |  |

Lecture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Lecture\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Faculty\_id** | **Varchar** | **No** | **None** |  |
| **3** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **4** | **Unit** | **Int** | **No** | **None** |  |
| **5** | **Topic** | **Text** | **No** | **None** |  |
| **6** | **No\_of\_lecture** | **Int** | **No** | **None** |  |
| **7** | **Status** | **Boolean** | **No** | **None** |  |
| **8** | **Username** | **Varchar** | **No** | **None** |  |

Lesson

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Lesson\_id** | **Int** | **No** | **None** |  |
| **2** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **3** | **Lecture\_id** | **Varchar** | **No** | **None** |  |
| **4** | **Start\_time** | **Time** | **No** | **None** |  |
| **5** | **End\_time** | **Time** | **No** | **None** |  |
| **6** | **Unit** | **Varchar** | **No** | **None** |  |
| **7** | **No\_of\_lecture** | **Varchar** | **No** | **None** |  |
| **8** | **Topic** | **Varchar** | **No** | **None** |  |
| **9** | **Date** | **Text** | **No** | **None** |  |
| **10** | **Status** | **Date** | **No** | **None** |  |
| **11** | **Lecture\_type** | **Boolean** | **No** | **None** |  |
| **12** | **Username** | **Varchar** | **No** | **None** |  |

Marks\_type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Marks\_type\_id** | **Int** | **No** | **None** |  |
| **2** | **Marks\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Status** | **Boolean** | **No** | **None** |  |
| **4** | **Username** | **Varchar** | **No** | **None** |  |

Qualification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Q\_id** | **Int** | **No** | **none** |  |
| **2** | **Qualification** | **Varchar** | **No** | **None** |  |
| **3** | **Username** | **Varchar** | **No** | **None** |  |
| **4** | **Date** | **Date** | **No** | **None** |  |

Room

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Room\_id** | **Varchar** | **No** | **None** |  |
| **2** | **Room\_no** | **Int** | **No** | **None** |  |
| **3** | **Username** | **Varchar** | **No** | **None** |  |
| **4** | **Status** | **Boolean** | **No** | **None** |  |
| **5** | **Session\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |

Section

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **S\_no** | **Int** | **No** | **None** |  |
| **2** | **Session\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **3** | **Session\_class\_id** | **Varchar** | **No** | **None** |  |
| **4** | **Student\_id** | **Varchar** | **No** | **None** |  |

Semester

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Semester\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **No\_of\_semester** | **Int** | **No** | **None** |  |
| **3** | **Status** | **Boolean** | **No** | **None** |  |
| **4** | **Username** | **Varchar** | **No** | **None** |  |

Session

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **S\_id** | **Int** | **No** | **None** |  |
| **2** | **Session** | **Int** | **No** | **None** | **Auto\_increment** |
| **3** | **Status** | **Boolean** | **No** | **None** |  |
| **4** | **Username** | **Varchar** | **No** | **None** |  |

Sidebar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **extra** |
| **1** | **Sidebar\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Sidebar\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Path** | **Text** | **No** | **None** |  |
| **4** | **Status** | **Boolean** | **No** | **None** |  |
| **5** | **Username** | **Varchar** | **No** | **None** |  |

Sign\_up

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **S\_no** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Username** | **Varchar** | **No** | **None** |  |
| **3** | **First\_name** | **Varchar** | **No** | **None** |  |
| **4** | **Last\_name** | **Varchar** | **No** | **None** |  |
| **5** | **Email** | **Varchar** | **No** | **None** |  |
| **6** | **Password** | **Varchar** | **No** | **None** |  |
| **7** | **Phone\_no** | **Varchar** | **No** | **None** |  |
| **8** | **Date\_of\_birth** | **Date** | **No** | **None** |  |
| **9** | **Question** | **Longtext** | **No** | **None** |  |
| **10** | **Answer** | **Mediumtext** | **No** | **None** |  |
| **11** | **Category** | **Varchar** | **No** | **None** |  |
| **12** | **Status** | **Varchar** | **No** | **None** |  |

Std\_academic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Std\_academic\_id** | **Int** | **No** | **None** |  |
| **2** | **Student\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **3** | **Qualification\_status** | **Varchar** | **No** | **None** |  |
| **4** | **Passing\_year** | **Year** | **No** | **None** |  |
| **5** | **School** | **Varchar** | **No** | **None** |  |
| **6** | **Board** | **Varchar** | **No** | **None** |  |
| **7** | **Percentage** | **Int** | **No** | **None** |  |
| **8** | **Status** | **Boolean** | **No** | **None** |  |
| **9** | **Username** | **Varchar** | **No** | **None** |  |

Std\_address

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Std\_address\_id** | **Int** | **No** | **None** |  |
| **2** | **Student\_id** | **Varchar** | **No** | **None** |  |
| **3** | **Address** | **Varchar** | **No** | **None** |  |
| **4** | **State** | **Varchar** | **No** | **None** |  |
| **5** | **City** | **Varchar** | **No** | **None** |  |
| **6** | **Pinecode** | **Int** | **No** | **None** |  |
| 7 | Address\_status | Boolean | No | None |  |
| **8** | **Status** | **Boolean** | **No** | **None** |  |
| **9** | **Date\_of\_commencement** | **Date** | **No** | **None** |  |
| **10** | **username** | **Varchar** | **No** | **None** |  |

Std\_contact

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Std\_contact\_id** | **Int** | **No** | **None** |  |
| **2** | **Std\_email** | **Varchar** | **No** | **None** |  |
| **3** | **Std\_contact** | **Varchar** | **No** | **None** |  |
| **4** | **Alternate\_contact** | **Int** | **No** | **None** |  |
| **5** | **Contact\_status** | **Boolean** | **No** | **None** |  |
| **6** | **Email\_status** | **Boolean** | **No** | **None** |  |
| **7** | **Status** | **Boolean** | **No** | **None** |  |
| **8** | **Alternate\_email** | **Varchar** | **No** | **None** |  |
| **9** | **Student\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **10** | **Username** | **Varchar** | **No** | **None** |  |

Std\_group

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Session\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **2** | **Course\_id** | **Text** | **No** | **None** |  |
| **3** | **Subject\_id** | **Varchar** | **No** | **None** |  |
| **4** | **Group\_id** | **Varchar** | **No** | **None** |  |
| **5** | **Semester\_id** | **Varchar** | **No** | **None** |  |
| **6** | **Section\_id** | **Varchar** | **No** | **None** |  |
| **7** | **Starting\_roll** | **Int** | **No** | **None** |  |
| **8** | **Ending\_roll** | **Int** | **No** | **None** |  |
| **9** | **Std\_id** | **Varchar** | **No** | **None** |  |
| **10** | **Status** | **Boolean** | **No** | **None** |  |
| **11** | **Username** | **Varchar** | **No** | **None** |  |

Std\_personal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **S\_no** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **First\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Last\_name** | **Varchar** | **No** | **None** |  |
| **4** | **Date\_of\_birth** | **Date** | **No** | **None** |  |
| **5** | **Father\_name** | **Varchar** | **No** | **None** |  |
| **6** | **Mother\_name** | **Varchar** | **No** | **None** |  |
| **7** | **Gender** | **Varchar** | **No** | **None** |  |
| **8** | **Nationality** | **Varchar** | **No** | **None** |  |
| **9** | **Category** | **Varchar** | **No** | **None** |  |
| **10** | **Blood\_group** | **Varchar** | **No** | **None** |  |
| **11** | **Status** | **Boolean** | **No** | **None** |  |
| **12** | **Username** | **Varchar** | **No** | **None** |  |
| **13** | **Student\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **14** | **Session\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **15** | **Course\_id** | **Varchar** | **No** | **None** |  |

Std\_reg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Session** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **2** | **Course\_of\_admission** | **Varchar** | **No** | **None** |  |
| **3** | **Date\_of\_admission** | **Date** | **No** | **None** |  |
| **4** | **Status** | **Boolean** | **No** | **None** |  |
| **5** | **Description** | **Text** | **No** | **None** |  |
| **6** | **Username** | **Varchar** | **No** | **None** |  |
| **7** | **Std\_id** | **Varchar** | **No** | **None** |  |

Studentmarks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **null** | **Default** | **Extra** |
| **1** | **Add\_marks\_id** | **Int** | **No** | **None** |  |
| **2** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **3** | **Roll\_no** | **Int** | **No** | **None** |  |
| **4** | **Marks\_type\_id** | **Int** | **No** | **None** |  |
| **5** | **Marks** | **Int** | **No** | **None** |  |
| **6** | **Totalmarks** | **Int** | **No** | **None** |  |
| **7** | **Status** | **Boolean** | **No** | **None** |  |
| **8** | **Username** | **Varchar** | **No** | **None** |  |

Student\_time\_table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Fid** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Course\_id** | **Varchar** | **Nono** | **None** |  |
| **3** | **Semester** | **Int** | **No** | **None** |  |
| **4** | **Session** | **Varchar** | **No** | **None** |  |
| **5** | **Section** | **Varchar** | **No** | **None** |  |
| **6** | **Choose\_file** | **Longblob** | **No** | **None** |  |

Subject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Course\_id** | **Varchar** | **No** | **None** |  |
| **2** | **Semester\_id** | **Varchar** | **No** | **None** | **Auto\_increment** |
| **3** | **Subject\_id** | **Varchar** | **No** | **None** |  |
| **4** | **Subject\_name** | **Varchar** | **No** | **None** |  |
| **5** | **Status** | **Boolean** | **No** | **None** |  |
| **6** | **Username** | **Varchar** | **No** | **None** |  |

Sub\_sidebar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Sub\_sidebar\_id** | **int** | **No** | **None** | **Auto\_increment** |
| **2** | **Sidebar\_id** | **Int** | **No** | **None** |  |
| **3** | **Sub\_sidebar\_name** | **Varchar** | **No** | **None** |  |
| **4** | **Path** | **Text** | **No** | **None** |  |
| **5** | **Status** | **Boolean** | **No** | **None** |  |
| **6** | **Username** | **Varchar** | **No** | **None** |  |

Time\_table\_edit

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Tid** | **Int** | **No** | **None** | **Auto\_increment** |
| **2** | **Course\_id** | **Int** | **No** | **None** |  |
| **3** | **Day** | **Varchar** | **No** | **None** |  |
| **4** | **Semester** | **Int** | **No** | **None** |  |
| **5** | **Section** | **Varchar** | **No** | **None** |  |
| **6** | **Subject\_id** | **Varchar** | **No** | **None** |  |
| **7** | **Room** | **Int** | **No** | **None** |  |
| **8** | **Time** | **Varchar** | **No** | **None** |  |
| **9** | **Session** | **Varchar** | **No** | **None** |  |

University

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **University\_id** | **Varchar** | **No** | **None** |  |
| **2** | **University\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Status** | **Boolean** | **No** | **None** |  |
| **4** | **Username** | **Varchar** | **No** | **None** |  |

Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Users\_id** | **Int** | **No** | **None** |  |
| **2** | **Users\_name** | **Varchar** | **No** | **None** |  |
| **3** | **Status** | **Boolean** | **No** | **None** |  |
| **4** | **Username** | **Varchar** | **No** | **None** |  |

Users\_menu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Users\_menu\_id** | **Int** | **No** | **None** |  |
| **2** | **Users\_id** | **Int** | **No** | **None** |  |
| **3** | **Sidebar\_id** | **Int** | **No** | **None** |  |
| **4** | **Status** | **Boolean** | **No** | **None** |  |
| **5** | **Username** | **Varchar** | **No** | **None** |  |

Weekly

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Null** | **Default** | **Extra** |
| **1** | **Add\_class\_id** | **Int** | **No** | **None** |  |
| **2** | **Lectureid** | **Varchar** | **No** | **None** |  |
| **3** | **Lesson\_id** | **Varchar** | **No** | **None** |  |
| **4** | **Week\_id** | **Int** | **No** | **None** | **Auto\_increment** |
| **5** | **Start\_date** | **Date** | **No** | **None** |  |
| **6** | **End\_date** | **Date** | **No** | **None** |  |
| **7** | **No\_of\_lecture\_schedule** | **Int** | **No** | **None** |  |
| **8** | **No\_of\_lost\_due\_to\_holiday** | **Int** | **No** | **None** |  |
| **9** | **No\_of\_lost\_due\_to\_cl** | **Int** | **No** | **None** |  |
| **10** | **No\_extra\_taken** | **Int** | **No** | **None** |  |
| **11** | **No\_of\_lecture\_actual\_taken** | **Int** | **No** | **None** |  |
| **12** | **Description\_of\_topic** | **text** | **No** | **None** |  |

**USE CASE DIAGRAM FOR ADMIN AND DATA ENTRY OPERATOR**

D.E.O

ADMIN

* **USE CASE DIAGRAM FOR STUDENT AND FACULTY**

Faculty

Student

1. **LOGIN**

**ACTORS:** The following actors interact and participate in this use case:

Administrator, Data entry operator, Student, Faculty.

**BASIC FLOW:** This case starts when the actor Login to the Teacher’s diary.

1. The system requests that the actor enter his/her name and password.
2. The actor enters his/her name and password.
3. The system validates the entered name, password and logs the actor into the system.

**ALTERNATIVE FLOW:**

* Invalid Name/Password

In this basic flow, if the actor enters an invalid name, password then the system displays an error message. The actor can choose to either return to the beginning of the basic flow or cancel the login, at which point the use case ends.

**PRE-CONDITION:** All user must have a User Account (i.e., User id, password)

created for them in the system (through the administrator), prior to executing the use

cases.

**POST-CONDITIONS:** If the use case was successful, the actor is logged into the

system. If not, the system state is unchanged.

1. **MAINTAIN FACULTY DETAILS**

This use case allows the actor with role ‘Data entry operator(DEO)’ to maintain faculty

information. This includes adding, changing and deleting faculty information from the

system.

**ACTORS:** The following actors interact and participant in this use case:

Data entry operator (DEO).

**BASIC FLOW:** This use case starts when the data entry operator (DEO) wishes to add,

change, or delete faculty details from the system.

1. Login.
2. Once the Administrator provides the requested information, one of the sub-flows is executed.
3. Add faculty personal details, faculty profile, faculty contact details, faculty address.
4. Update faculty personal details, faculty profile, faculty contact details, faculty address.
5. Delete faculty personal details, faculty profile, faculty contact details, faculty address.

**ALTERNATIVE FLOWS:**

1. Faculty details not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The Data entry operator (DEO) must be logged onto the system

before this use case begins.

**POST-CONDITION:** If the use case is successful, the faculty information is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **MAINTAIN STUDENT DETAILS**

This use case allows the actor with role ‘Data entry operator (DEO)’ to maintain student

Information. This includes adding, changing and deleting student information from the

system.

**ACTORS:** The following actors interact and participant in this use case:

Data entry operator (DEO).

**BASIC FLOW:** This use case starts when the data entry operator (DEO) wishes to

add, change, or delete student details from the system.

* + 1. Login.
    2. Once the data entry operator (DEO) provides the requested information, one of the sub flows is executed.

1. Add student personal details, student profile, student contact details, student address.
2. Update student personal details, student profile, student contact details, student address.
3. Delete student personal details, student profile, student contact details, student address.

**ALTERNATIVE FLOWS:**

1. Student details not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The Data entry operator (DEO) must be logged onto the system

before this use case begins.

**POST-CONDITION:** If the use case is successful, the student information is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **MAINTAIN TIME TABLE DETAILS**

This use case allows the actor with role ‘Data entry operator (DEO)’ to maintain time

table information. This includes adding, changing and deleting time table related

information from the system.

**ACTORS:** The following actor interact and participant in this use case:

Data entry operator (DEO).

**BASIC FLOW:** This use case starts when the data entry operator (DEO) wishes to add,

change, or delete time table details from the system.

* + 1. Login.
    2. Once the data entry operator (DEO) provides the requested information, one of the sub-flows is executed.
  1. Add details in time table.
  2. Update time table.
  3. Delete time table.

**ALTERNATIVE FLOWS:**

1. Time table details not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The Data entry operator (DEO) must be logged onto the system

before this use case begins.

**POST-CONDITION:** If the use case is successful, the time table information is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **GENERATE REPORT**

This use case allows the actor with role ‘Data entry operator (DEO)’ to generate report.

This includes adding, changing and deleting report information from the system.

**ACTORS:** The following actor interacts and participant in this use case:

Data entry operator (DEO).

**BASIC FLOW:** This use case starts when the data entry operator (DEO) wishes to

add, change, or delete report from the system.

* + 1. Login.
    2. Once the data entry operator (DEO) provides the requested information, one of the sub-flows is executed.
    3. Generate report.
    4. Update report.
    5. Delete report.

**ALTERNATIVE FLOWS:**

1. Report not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The Data entry operator (DEO) must be logged onto the system

Before this use case begins.

**POST-CONDITION:** If the use case is successful, the report will be generated, updated,

or deleted from the system. Otherwise, the system state is unchanged.

1. **MAINTAIN STAFF DETAILS**

This use case allows the actor with role ‘Administrator’ to maintain staff information.

This includes adding, changing and deleting staff related information from the system.

**ACTORS:** The following actors interact and participant in this use case:

Administrator.

**BASIC FLOW:** This use case starts when the administrator wishes to add, change,

Or delete staff details from the system.

1. Login.
2. Once the administrator provides the requested information, one of the sub-flows is executed.
3. Add staff.
4. Delete staff.

**ALTERNATIVE FLOWS:**

1. Staff not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The administrator must be logged onto the system before this

use case begins.

**POST-CONDITION:** If the use case is successful, the Staff information is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **MANAGE FEEDBACK**

This use case allows the actor with role ‘Administrator’ to manage feedback. This

includes adding, changing and deleting feedback related information from the system.

**ACTORS:** The following actors interact and participant in this use case:

Administrator.

**BASIC FLOW:** This use case starts when the administrator wishes to add, change,

or delete feedback from the system.

1. Login.
2. Once the administrator provides the requested information, one of the sub-flows is executed.
3. Add feedback.
4. Update feedback.
5. Delete feedback.

**ALTERNATIVE FLOWS:**

1. Feedback not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The administrator must be logged onto the system before this

use case begins.

**POST-CONDITION:** If the use case is successful, the staff information is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **MAINTAIN LECTURE/LESSON PLAN**

This use case allows the actor with role ‘Faculty’ to maintain Lecture/Lesson

information. This includes adding, changing and deleting lecture/lesson related

information from the system.

**ACTORS:** The following actors interact and participant in this use case:

Faculty.

**BASIC FLOW:** This use case starts when the faculty wishes to add, change, or

delete lecture and lesson plan from the system.

1. Login.
2. Once the faculty provides the requested information, one of the sub-flows is executed.
3. Add lecture plan.
4. Update lecture plan.
5. Delete lecture plan.

**ALTERNATIVE FLOWS:**

1. Lecture/Lesson plan not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The faculty must be logged onto the system before this use

case begins.

**POST-CONDITION:** If the use case is successful, the lecture/lesson plan is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **MAINTAIN ATTENDANCE RECORD**

This use case allows the actor with role ‘Faculty’ to maintain attendance record.

This includes adding, changing and deleting attendance related information from

the system.

**ACTORS:** The following actor interact and participant in this use case:

Faculty.

**BASIC FLOW:** This use case starts when the faculty wishes to add, change, or

delete attendance record of student from the system.

1. Login.
2. Once the faculty provides the requested information, one of the sub-flows is executed.
3. Add student attendance.
4. Update attendance.
5. Delete attendance.

**ALTERNATIVE FLOWS:**

1. Attendance not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The faculty must be logged onto the system before this use

case begins.

**POST-CONDITION:** If the use case is successful, the student attendance is

added, updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **MAINTAIN INTERNAL TEST/ CLASS TEST**

This use case allows the actor with role ‘Faculty’ to maintain internal test and

class test. This includes adding, changing and deleting test related information

from the system.

**ACTORS:** The following actor interacts and participant in this use case:

Faculty.

**BASIC FLOW:** This use case starts when the faculty wishes to add, change,

or delete attendance record of student from the system.

1. Login.
2. Once the faculty provides the requested information, one of the sub-flows is executed.
3. Take internal test and class test.
4. Update test information.
5. Delete test information.

**ALTERNATIVE FLOWS:**

1. Test information not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The faculty must be logged onto the system before this use

case begins.

**POST-CONDITION:** If the use case is successful, the internal test is added,

updated, or deleted from the system. Otherwise, the system state is unchanged.

1. **GENERATE WEEKLY REPORT**

This use case allows the actor with role ‘Faculty’ to generate weekly report. This

includes adding, changing and deleting weekly report information from the

system.

**ACTORS:** The following actors interact and participant in this use case:

Faculty.

**BASIC FLOW:** This use case starts when the faculty wishes to add, change, or

delete report from the system.

1. Login.
2. Once the faculty provides the requested information, one of the sub-flows is executed.
   * + 1. Generate weekly report.
       2. Update weekly report.
       3. Delete weekly report.

**ALTERNATIVE FLOWS:**

1. Weekly report not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The Faculty must be logged onto the system before this use

case begins.

**POST-CONDITION:** If the use case is successful, the weekly report will be

generated, updated, or deleted from the system. Otherwise, the system state is

unchanged.

**12. VIEW REPORT**

This use case allows the actor with role ‘Faculty and student’ to generate report and

view report. This includes adding, changing and deleting report information from the

system.

**ACTORS:** The following actors interact and participant in this use case:

Faculty and Student.

**BASIC FLOW:** This use case starts when the faculty wishes to add, change, or

delete report and student wishes to view report from the system.

1. Login.
2. Once the faculty provides the requested information, one of the sub-flows is executed.
   * + 1. Generate weekly report.
       2. Update weekly report.
       3. Delete weekly report.
       4. View report.

**ALTERNATIVE FLOWS:**

1. Report not found.
2. Update cancelled.
3. Delete cancelled.

**PRE-CONDITION:** The Faculty and student must be logged onto the system before

this use case begins.

**POST-CONDITION:** If the use case is successful, the report will be generated,

updated, or deleted and view from the system. Otherwise, the system state is unchanged.

**Lesson plan**

A lesson plan is a step-by-step guide that provides a structure for an essential

learning. Before planning a lesson, it is essential to classify the learning outcomes for

the class. It is important because it helps the teacher in maintaining a standard teaching

pattern. Pre-planning helps the teacher to be better equipped in answering questions

asked by the students during the lecture.

* Input and Output Screen Design

DASHBOARD

It is the screen of main dashboard. here only authorized can enter threw login.

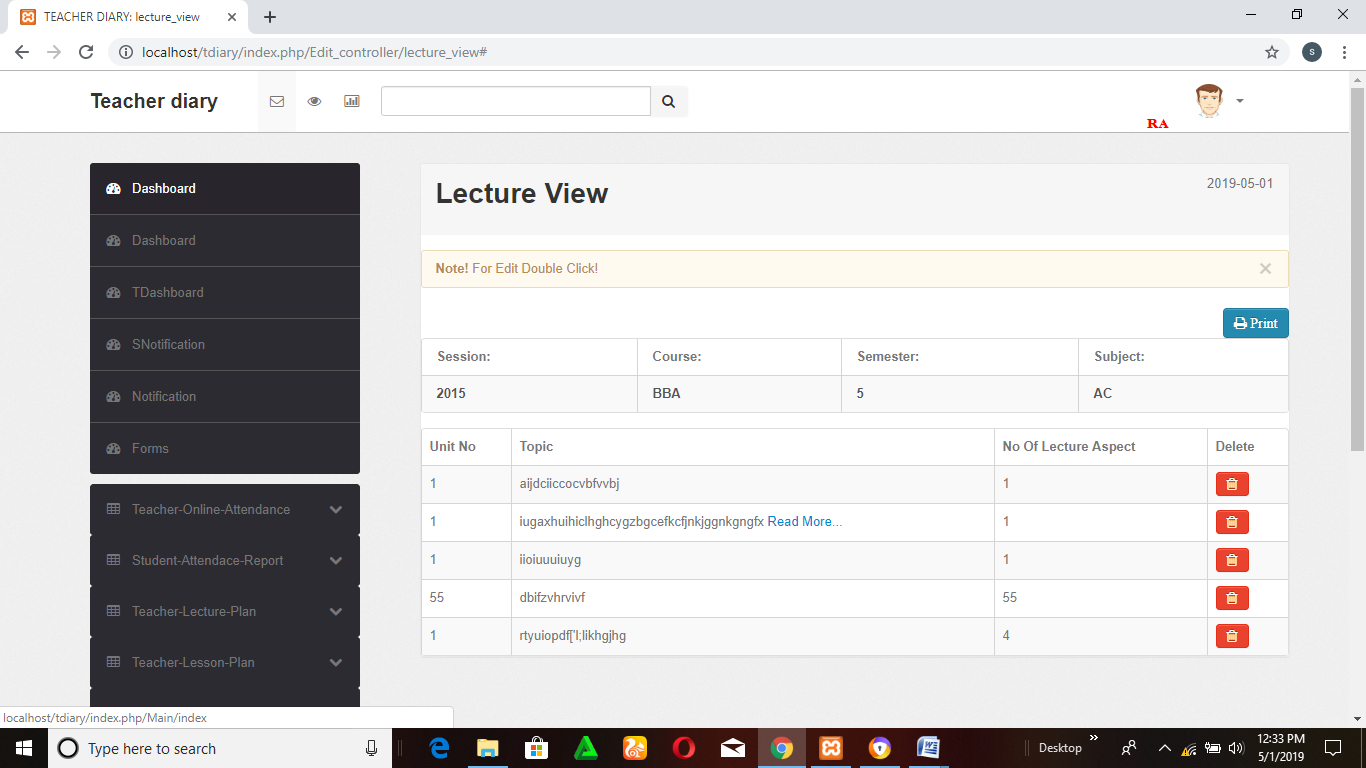


Figure : Dashboard

Forms

This is the form page where new user can register by admin.

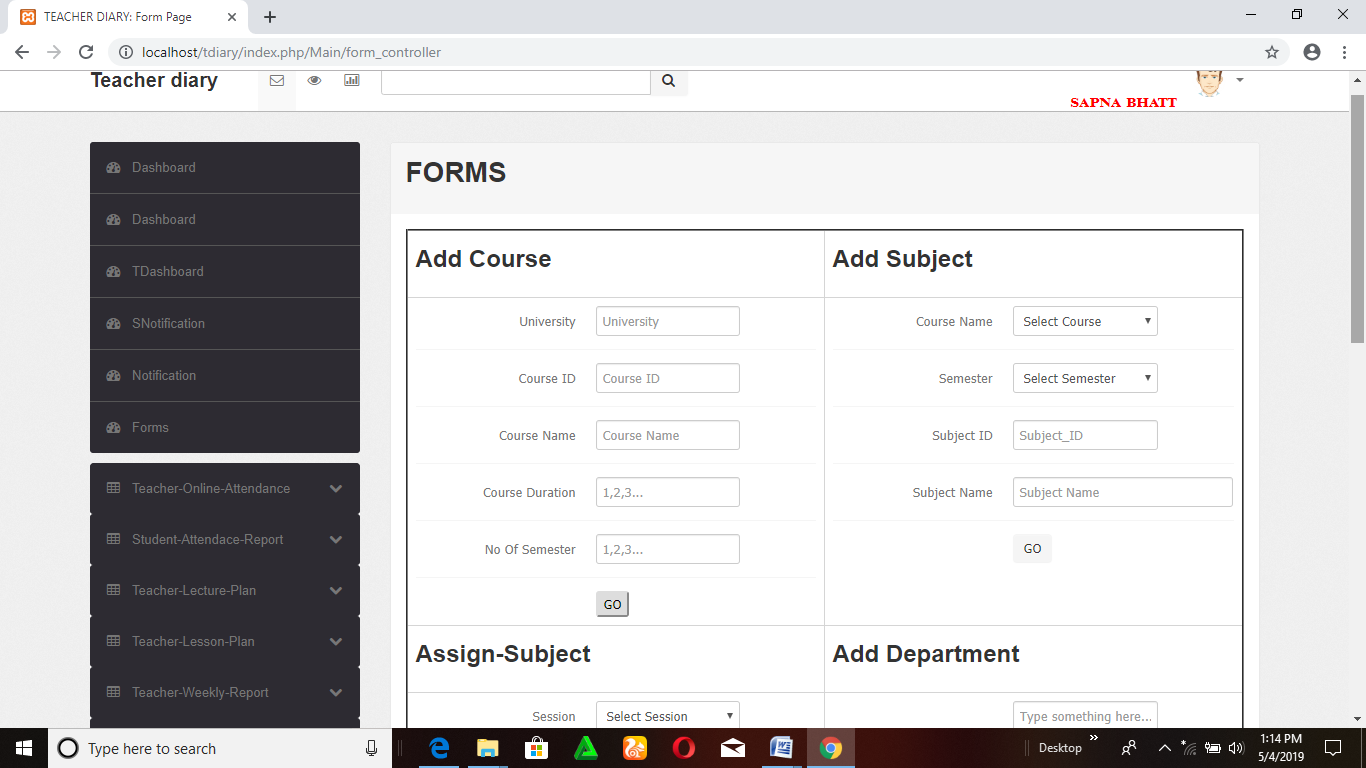


Figure : Master entities

ONLINE\_ATTENDANCE

In this page here faculty has the facility to take online attendance which is directly viewed by student in student console.

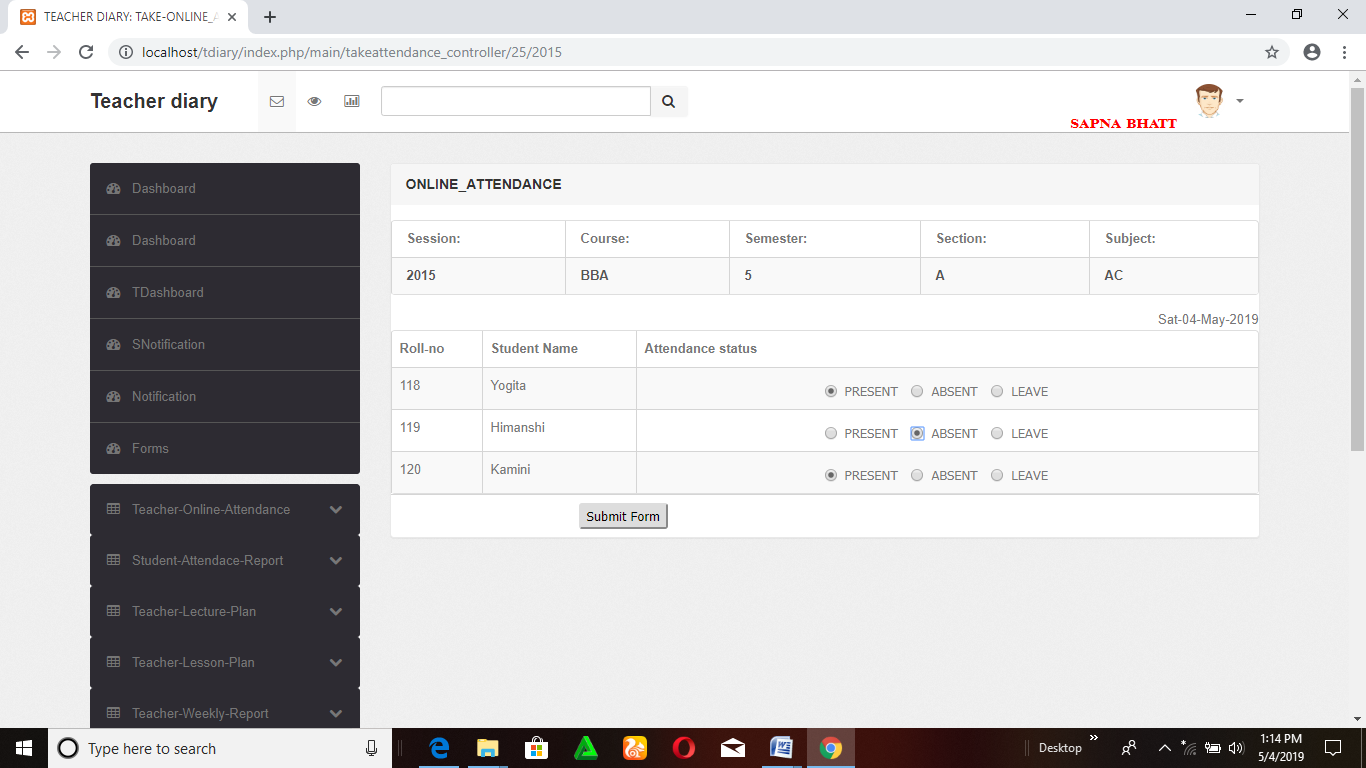


Figure : Attendance page

LESSON PLAN

It is the page which is used by faculty as well as students to vied the lecture details

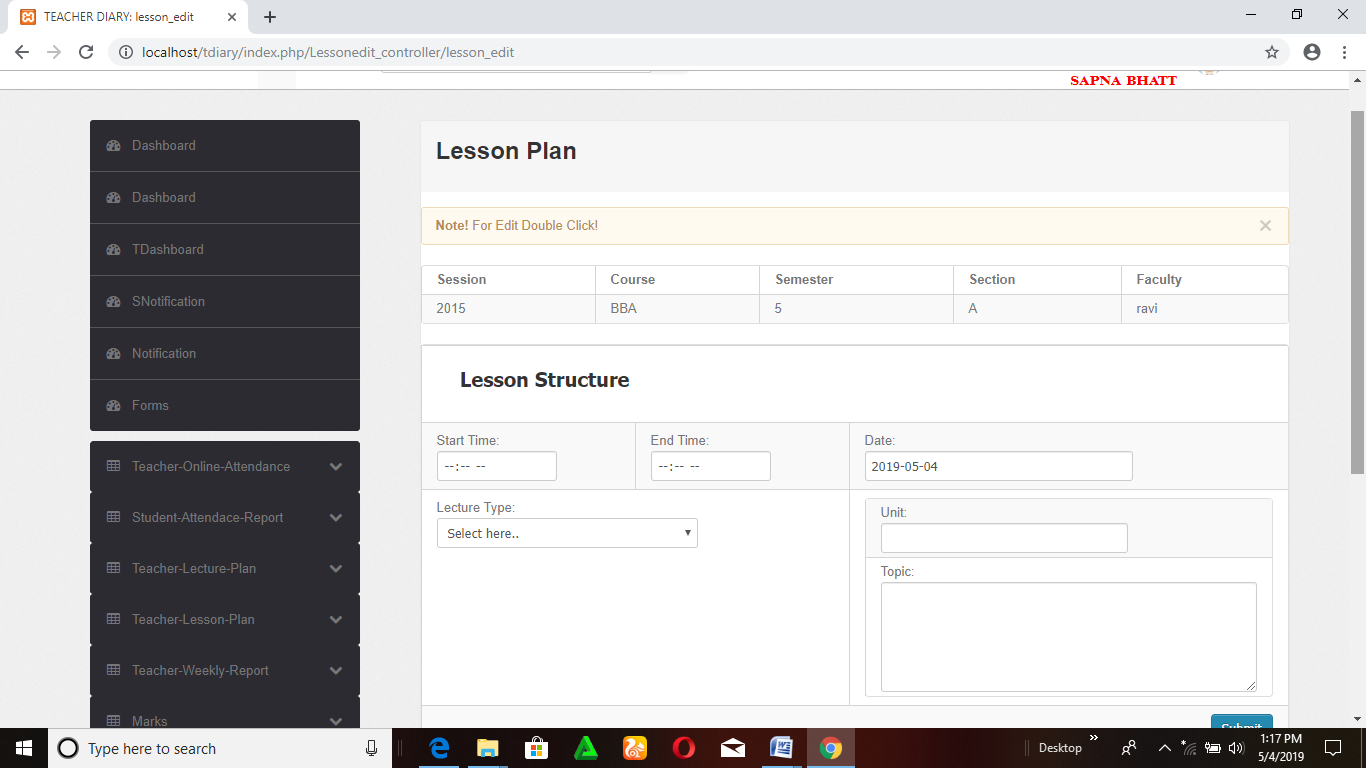


Figure : Lesson Plan

WEEKLY PLAN

In is the page where faculty can plan about their weekly schedule.

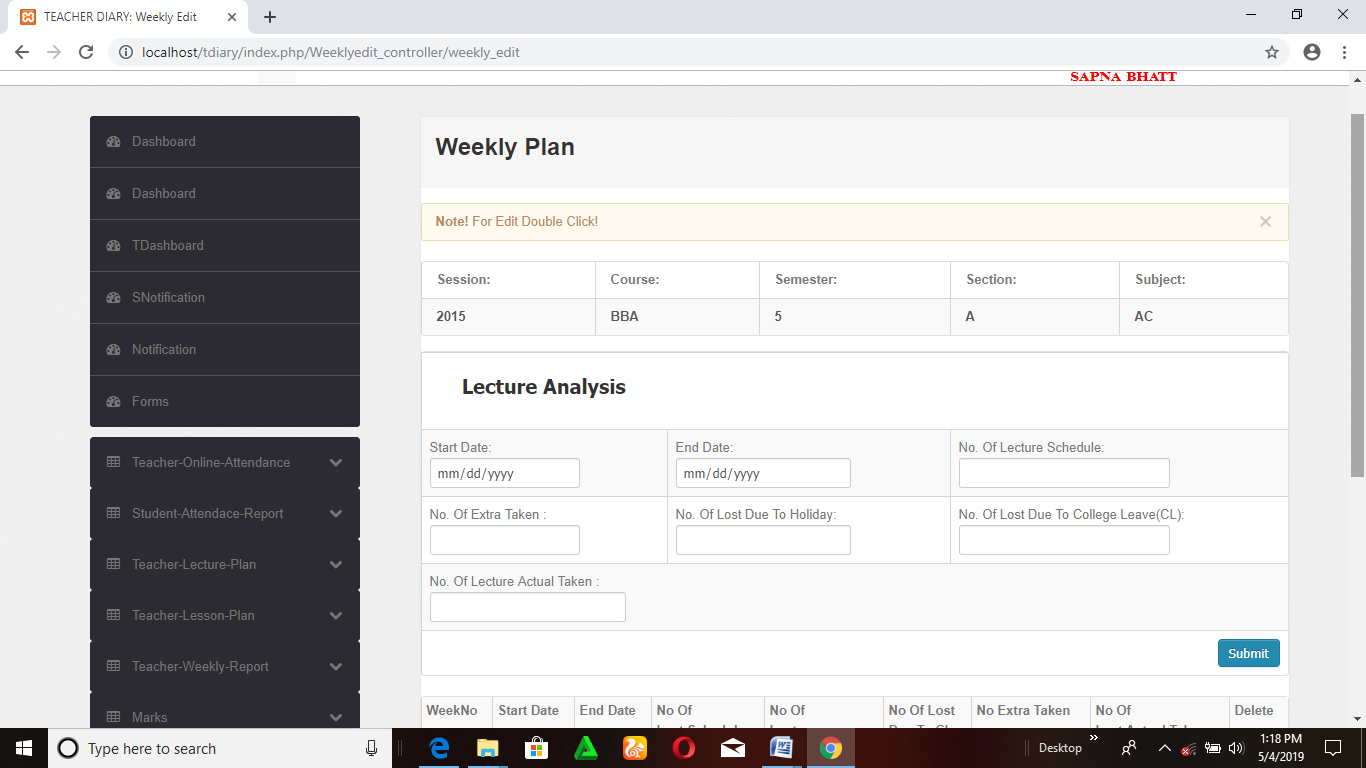


Figure : Weekly Plan

EDIT\_MARKS

Here only faculty can edit the marks and viewed by both faculty and students.

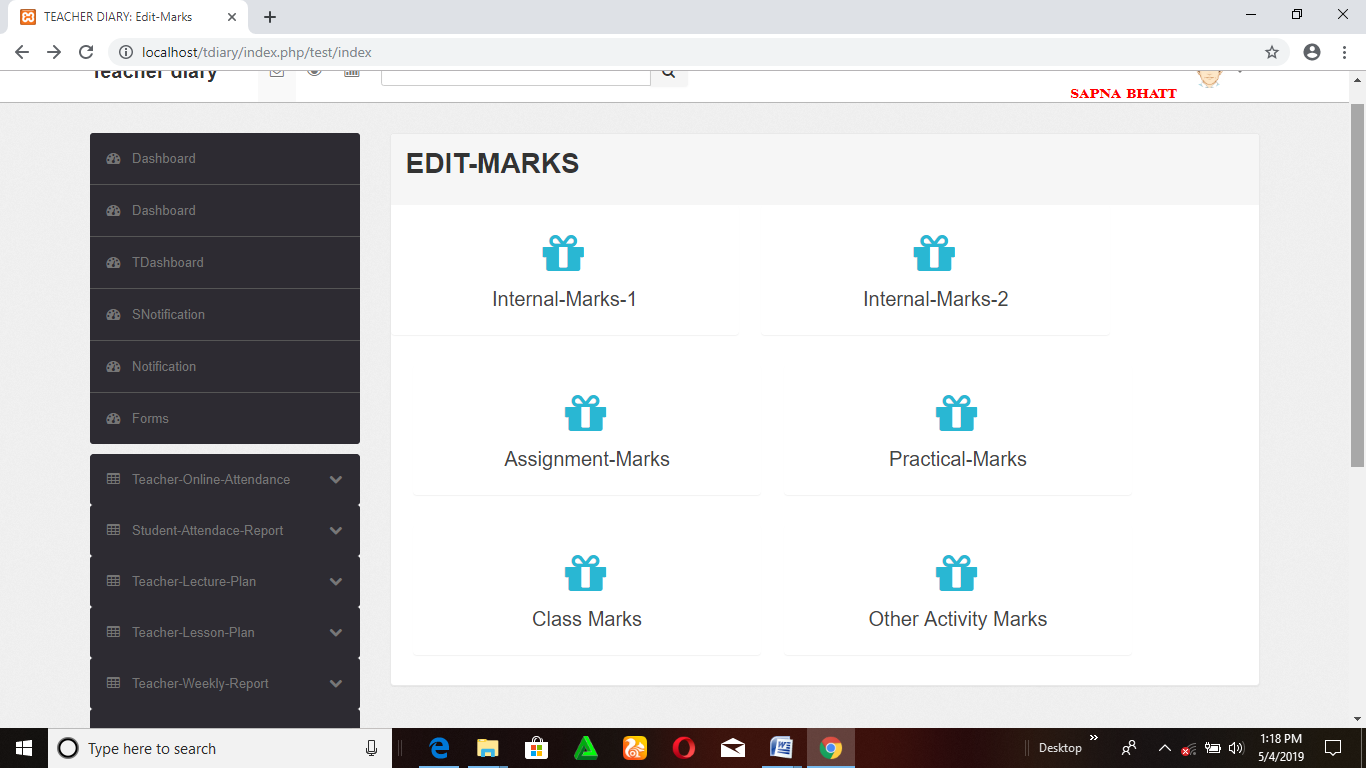


Figure : Edit Marks

GIVEN ASSIGNMENT

Here faculty will give the assignment along with given date and submission date. This page is only used by faculty



Figure : Given Assignment

ASSIGNMENT

This page is used by both faculty as well as students. Student view this page to take assignment details.

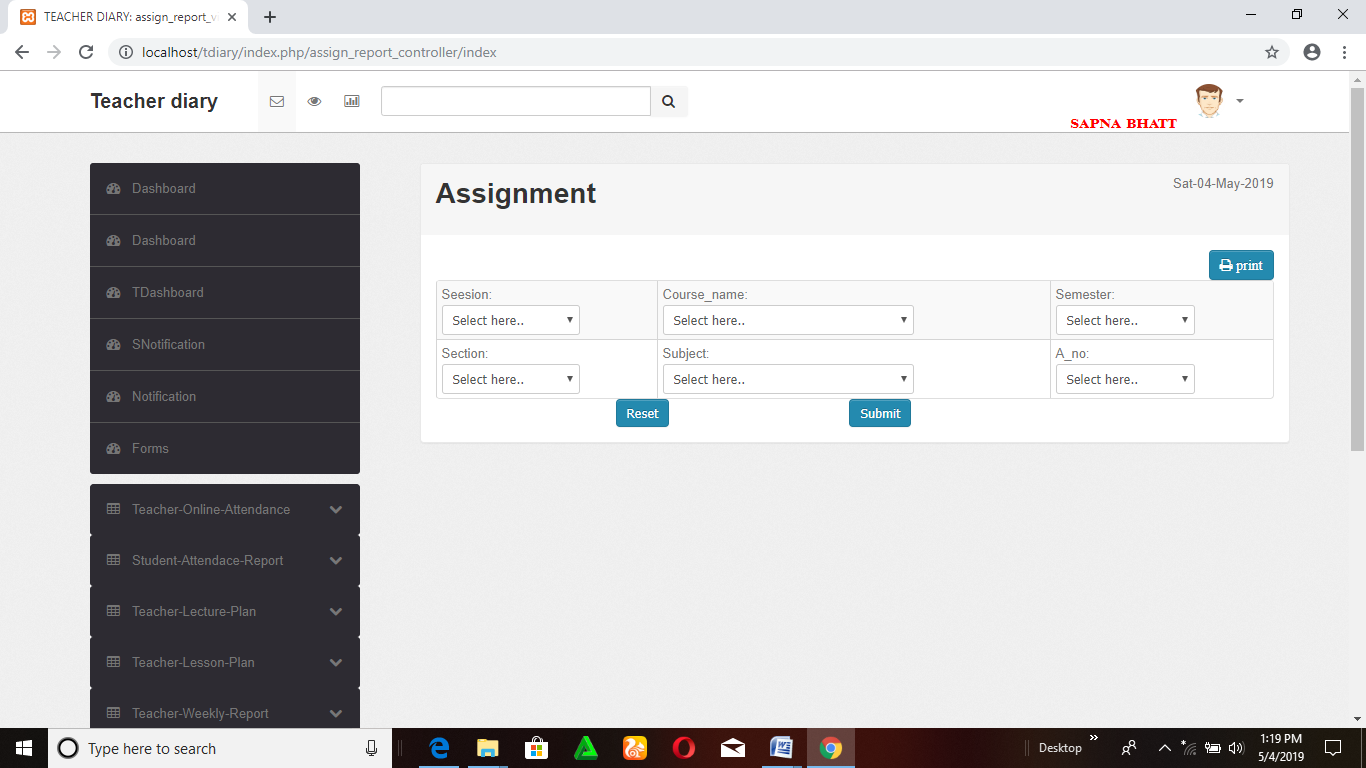


Figure : Assignment

UPLOAD TIME TABLE

This page is used by faculty to update a time table.

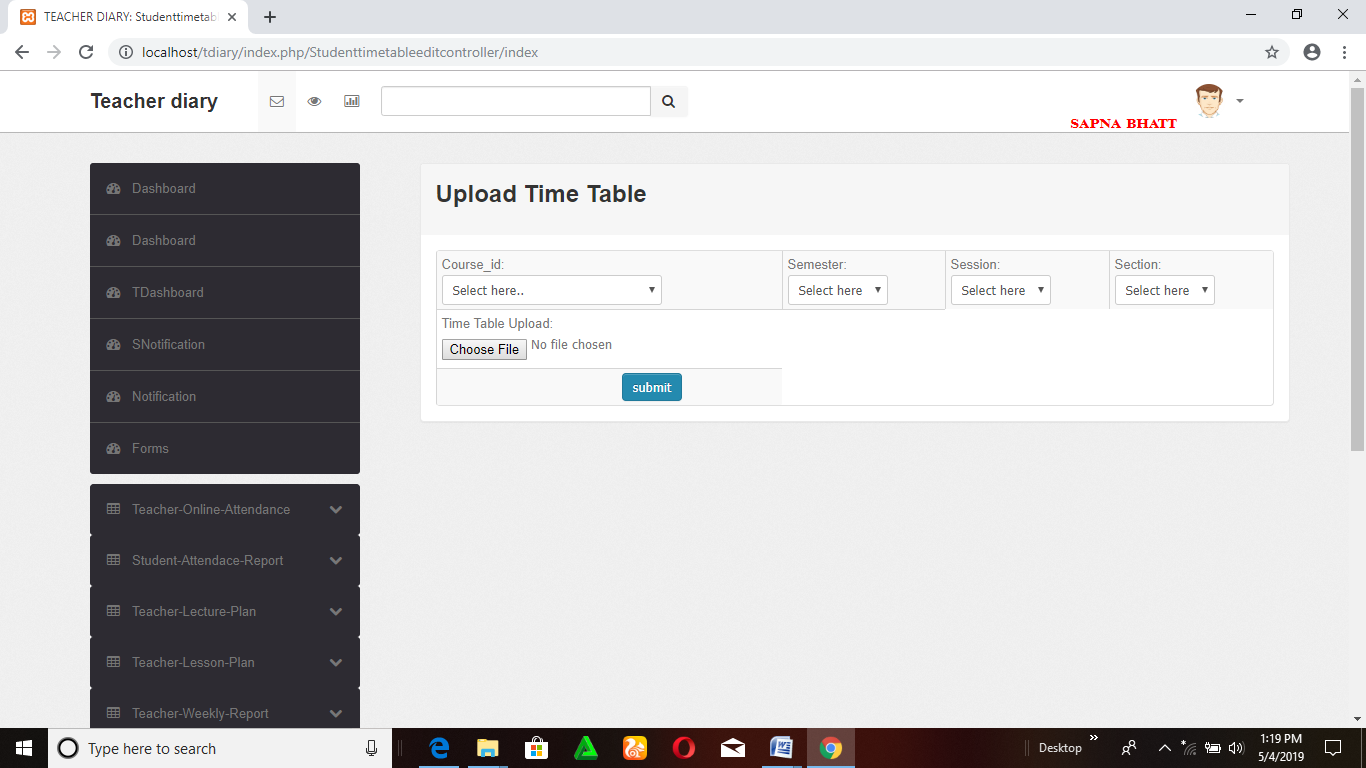


Figure : Upload Time Table

EDIT TIME TABLE

Here the faculty can update the timetable which is viewed by both faculty as well as students

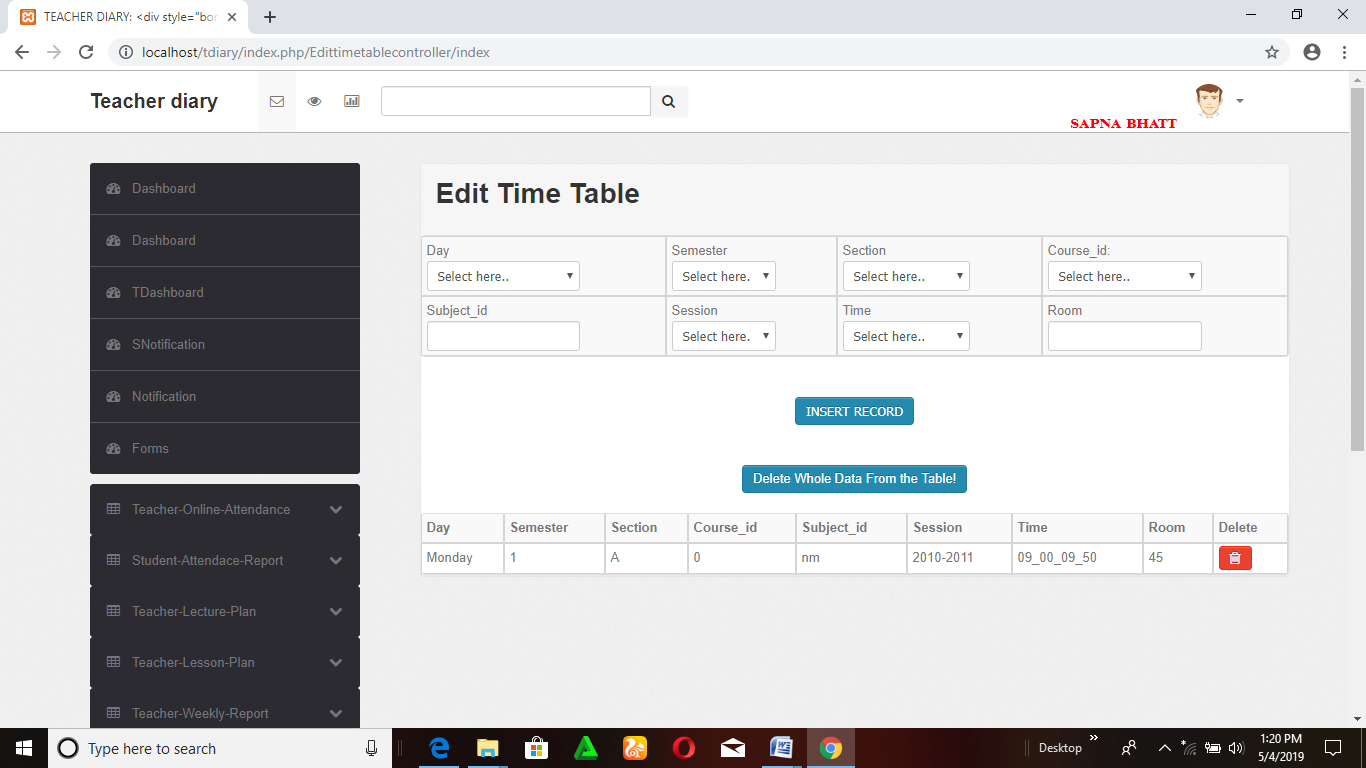


Figure : Edit Time Table

VIEW TIME TABLE

In this submenu student and faculty can view the time table

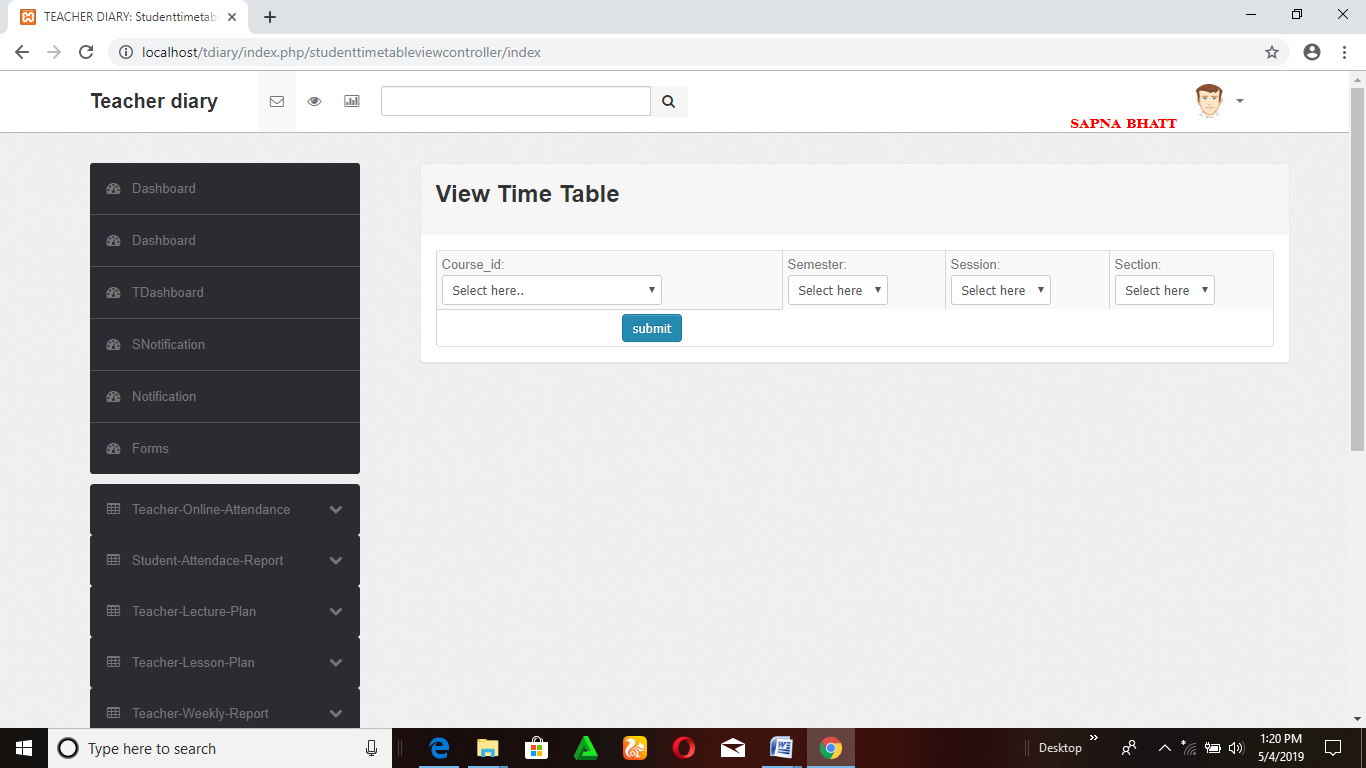


Figure : View Time Table

LECTURE VIEW

This submenu is used by both faculty and student to view the lecture details.

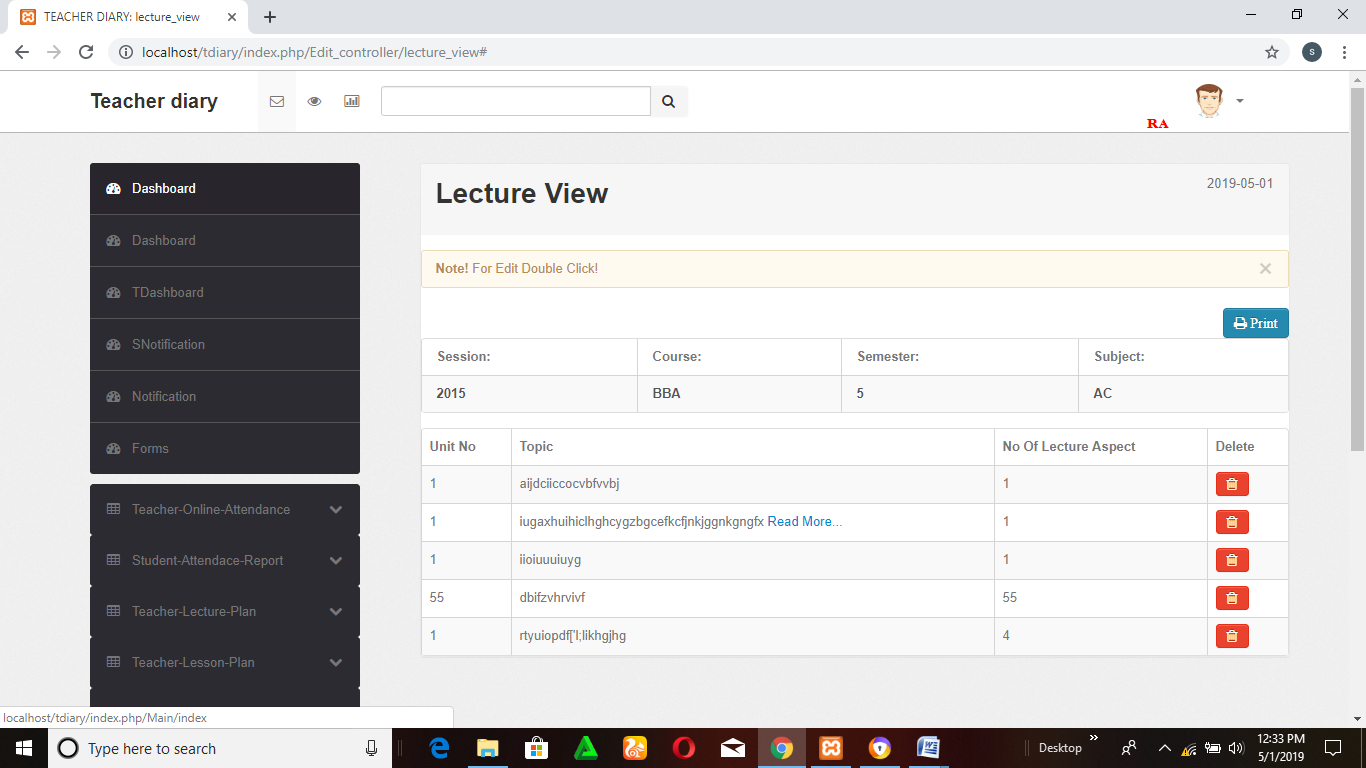


Figure : Lecture View

Output:-

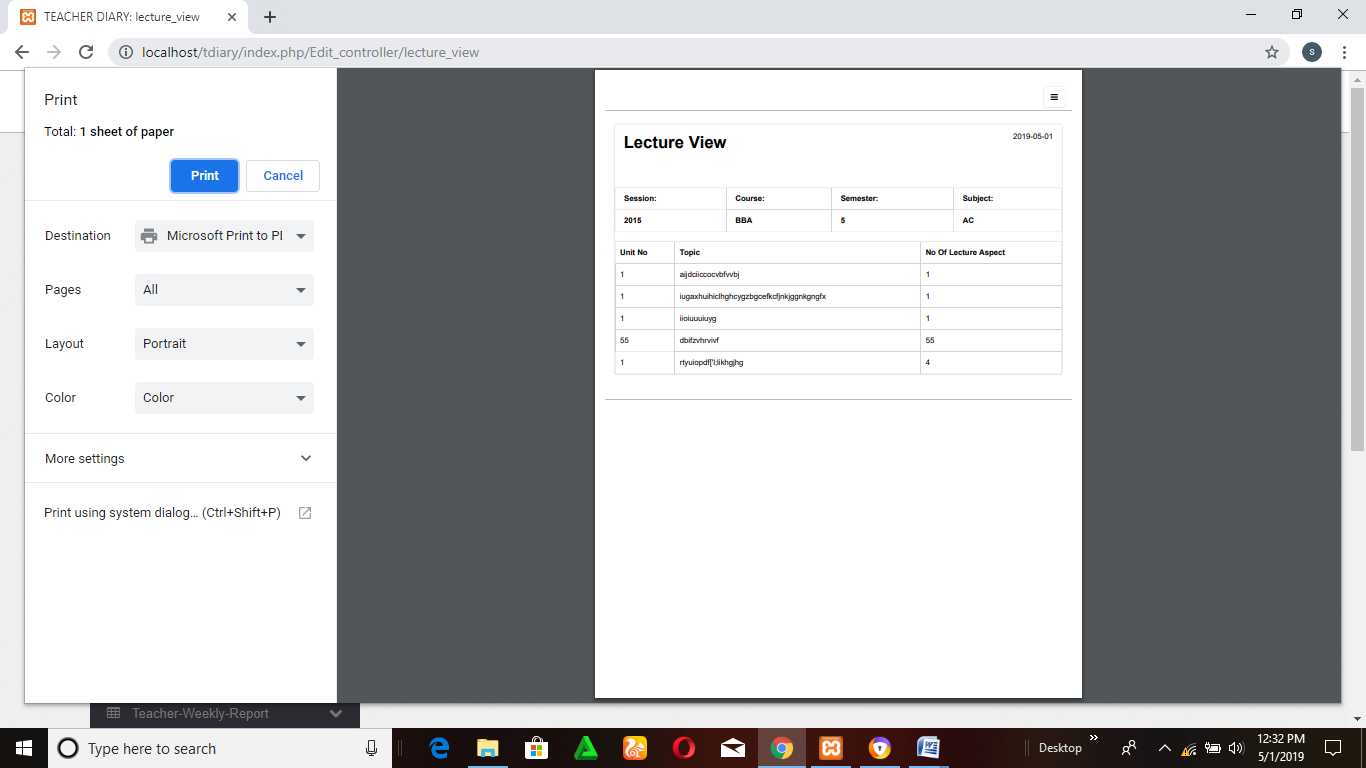


Figure : output of lecture view

* Process involved

Basically there are four types of users who deal with the system. The users are listed below:

1. Admin
2. Faculty
3. Parent
4. Students

The characteristics of each user are explained below:

(1) Admin

* Admin manages the faculty details.
* Admin manages the new user login.
* Admin manages the events with name, date and description.
* Admin manages the time-table by standard and division wise.
* Admin manage the circular with description.
* Admin manage the profile.
* Admin manage the holiday list with date.
* Admin manage the achievements of the school with image.
* Admin manage the standards.
* Admin display the exam-schedule by standard wise.

(2) Faculty

* Faculty can post student complain to their parents.
* Faculty can view the parent feedback.
* Faculty can upload the assignment and other study materials.
* Faculty manages the result declaration.
* Faculty manages the student details.
* Faculty manages the attendance and performance details.
* Faculty can view the leave-application.
* Faculty can view photo gallery.
* Faculty can view the holiday list.
* Faculty can view the circulars.
* Faculty can view the achievements.
* Faculty can view the student details.

.

(3) Students

* Students can view photo gallery.
* Students can view the holiday list.
* Students can view the circulars.
* Students can view the achievements.
* Students can view the student details.
* Students can view the all assignments and study materials.
* Students can view attendance and performance details.
* Students can view events.
* Students can view the time-table.
* Students can view the faculty details.
* Students can view the results and exam schedule.

Modules description

There are 5 modules in our system:-

* Student management & faculty management
* Attendance management system
* Internal marks & assignment
* Lesson plan lecture plan weekly report
* Time table management

STUDENT MANAGEMENT & FACULTY MANAGEMENT:

In student management & faculty management module here admin gives the facility to students & teachers to register themselves in the dashboard only registered students and faculty can use this dashboard.

Only authorized user can use this dashboard, authorized user means that the date of birth and email id must be matched otherwise user cannot be able to use this dashboard.

At the time of registration faculty and student fill the form, in the form some basic information are asked.

**Teacher’s module**

Teacher

**Student module**

Syllabus upload

Attendance

Assignment details

Verified email

Registration

Arrangement

Student details

Report

Lecture plan

Time table

Internal test

Lesson plan

Sign up

Login

Student

Sign up

View attend

Assignment

Result

Time table

Syllabus

Student details

Login

ATTENDANCE MANAGEMENT SYSTEM:

In attendance management module here admin gives many facilities to the faculty as well as to students.

There are two consoles

1. Teacher online attendance

2. Student attendance report

**Teacher online attendance: -** There are 4 sub menus

*Take-attendance*: - Here faculties have the facility to take online attendance. In this faculty add the new class which they teach and can also deletes the class.

*View-consolidate: -* In this, faculty can check the report of attendance, which they take previously.

*View-day-wise:* - As the name represent here faculty can view attendance day wise.

*View-total:* - Here faculty can view the attendance daily, weekly, monthly or any particular day by searching the date, day or month with year.

**Student attendance report:** - Here student have the facility to view his attendance report by giving some basic details which the system asked. There are three sub menu mentioned below

*View-consolidate: -* In this, student can check the report of attendance which they take previously.

*View-day-wise: -* As the name represent here student can view the attendance day wise.

*View-total:* - Here student can view the attendance daily, weekly, monthly or any particular day by searching the date, day or month with year

INTERNAL MARKS & ASSIGNMENT

In internal marks & assignment module here admin has given 4 sub menus which are

mentioned below:

*Marks:* - In marks console here faculty can give marks online and can also deletes the marks. In marks console here two submenu are given

**Edit:** - In this faculty can edit the marks table, which is directly view by student timely.

**View: -** In view sub menu here faculty can check the marks detail of whole class and can also evaluate student performance. In this faculty can also edit the mentioned detail by double clicking in the portal.

*Assignment:* - In assignment console here admin give four sub menus which are used to facilitate the faculty as well as students.

**Given-Assignment:** -Here faculty gave the assignment to the whole class simultaneously by mentioning the last submission date and provided date as well.

**View given assignment:** - In this only faculty can view the given assignment and can also edit and delete.

**Check-Assignment:** - Here faculties check the assignment and also give the grade simultaneously which will view by students to check their performance

**View-Check-Assignment:** - In this console only faculty can view and update.

*Student-Marks:* - In student-marks console here only one sub menu given which is used by students.

**View:** - In view here only student can view their marks individually and cannot be permitted to edit any detail.

*Student-Assignment:*- In this console there are two sub menu which are followed below:

**View-**Assignment: this submenu is only used by student to view the assignment and related information

**View-Checked-Assignment:** - Here student view that which of the assignments are checked.

LECTURE PLAN, LESSON PLAN AND WEEKLY REPORT

In Lecture plan, Lesson plan and Weekly report module, here admin has given 3 sub menus which are mentioned below:

*Teacher-Lecture-plan:* - In lecture plan faculty will make their lecture plan before starting his class. She/he can also edit, delete and view their plan. In this there are two submenu given below:

**Edit-** In this faculty has the facility to edit, view, & also delete the lecture plan schedule.

**View-** In this faculty can only view the lecture plan

*Teacher-Lesson-Plan* – In teacher-lesson-plan faculty previously made their plan lesson wise and teaches in the class accordingly. Here two sub modules are mentioned below :

**Edit: -** In this faculty can edit their lesson plan according the time given and this is only edited by faculty and viewed by both faculty and students.

**View:** -In this faculty can view the daily lesson plan and teach accordingly.

*Teacher-Weekly-Report-* In this console the teacher weekly report is generated by teacher and viewed by director only to give the feedback to the teacher.

**Edit:** Here teacher has to permission of edit the report.

**View:** In this faculty can view his weekly report to maintain his/her report.

TIME TABLE MANAGEMENT

*Admin-Timetable:* - It is timetable which is used by admin and also generated by admin

**Edit-timetable:** -Here admin can edit the timetable according

**View-timetable:** - In this console here admin timetable is vied by admin and cannot

Be edited.

*Teacher-Timetable:* -This timetable is used by faculty of theorganization.

**Edit-timetable-** Here faculty have the permission to edit the timetable accordingly

**View-timetable**-Here faculty can view the time table according to which He/she taken the

class or lecture.

*Student-Timetable*-This console is used by student.

**View-timetable**- By this submenu student can able to see his timetable according

to which they attend class.

TEST CASES

* 1. LOGIN

TEST CASE ID: T1

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online Teacher diary management system will be entered into the system. |
| INPUTS: Username, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system .  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. Sign up

TEST CASE ID: T2

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter in dashboard. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Username, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get enter to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. RESET PASSWORD

TEST CASE ID: T3

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to Change user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Username, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. INTERNAL MARKS

TEST CASE ID: T4

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id, password & other inputs. | EXECUTION HISTORY: |
| PRECONITION: User must visit before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Subject. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. GIVEN ASSIGNMENT

TEST CASE ID: T5

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to give assignment to the students. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Subject. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. CHECKS ASSIGNMENT

TEST CASE ID: T6

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to enter in checks assignment submenu to mark the checked assignment. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Subject. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. LECTURE PLAN

TEST CASE ID: T7

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Subject, Unit No., No. of Lecture Aspect. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. LESSON PLAN

TEST CASE ID: T8

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Faculty, Start Time, End Time, Date, Lecture Type, Unit, Topic. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. WEEKLY PLAN

TEST CASE ID: T9

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Subject, Start Date, End Date, No. of lecture schedule, No. of Extra Taken, No. of lost due to holiday, No. of lost due to college Leave, No. of lecture actual taken. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. ONLINE ATTENDANCE

TEST CASE ID: T10

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to mark the attendance. | EXECUTION HISTORY: |
| PRECONITION: User must visit before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Subject. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. VIEW ATTENDANCE REPORT

TEST CASE ID: T11

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to view the report of the class. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course, Semester, Section, Roll no., Date From, Date up to. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. UPLOAD TIME TABLE

TEST CASE ID: T12

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to upload the time table | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course id, Semester, Section, Time Table upload. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. VIEW TIMETABLE

TEST CASE ID: T13

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to the time table by student as well as faculty. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Session, Course id, Semester, Section. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. EDIT TIME TABLE

TEST CASE ID: T14

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to edit the time table. | EXECUTION HISTORY: |
| PRECONITION: User must visit before this use case begin | RESULT: After log in the online teacher diary management system will be entered into the system. |
| INPUTS: Day, Session, Course id, Semester, Section, Subject id, Time, Room. | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of ONLINE TEACHERS DIARY MANAGEMENT SYSTEM WEBSITE.  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of ONLINE TEACHER DIARY MANAGEMENT SYSTEM.  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* Test Report, Printout of the Report &Coke Sheet :

MARKS

* + 1. Practical Marks

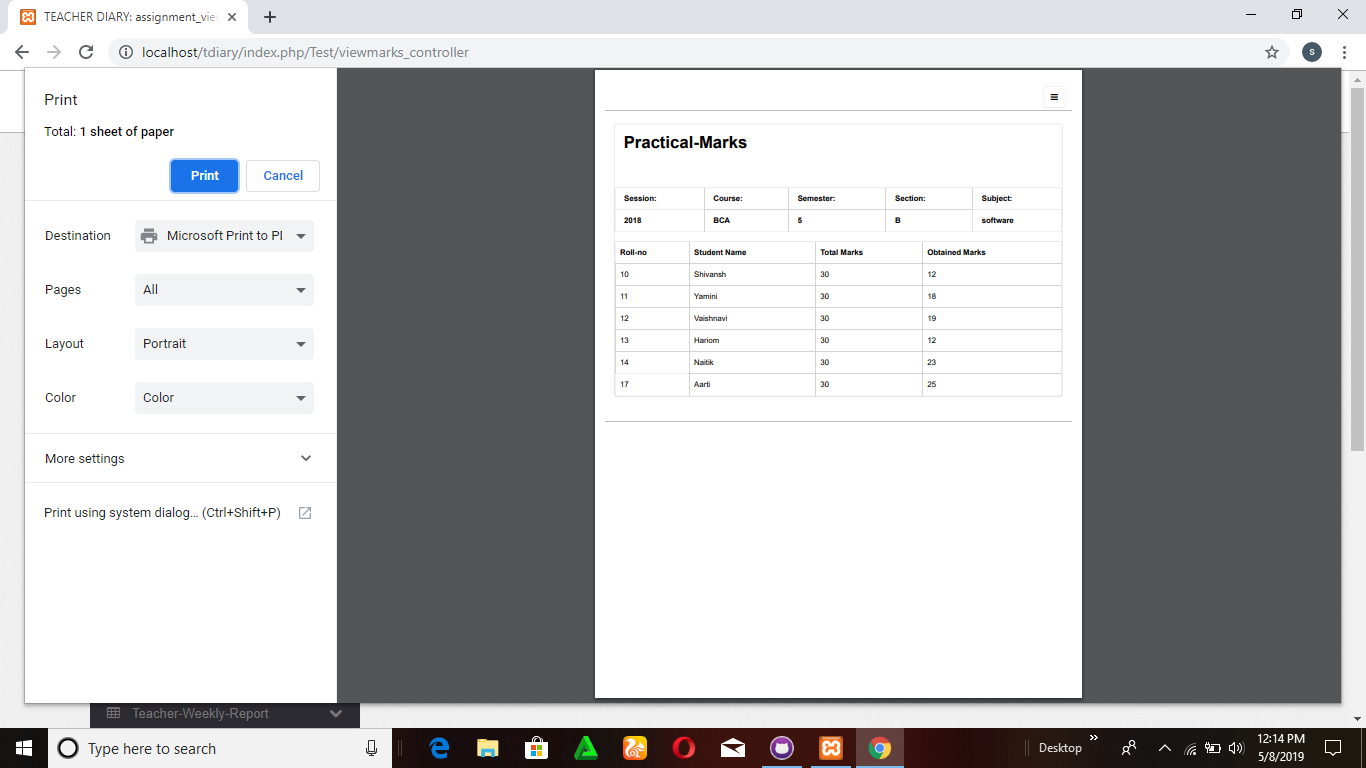


Figure : Practical marks

* + - 1. INTERNAL MARKS 2

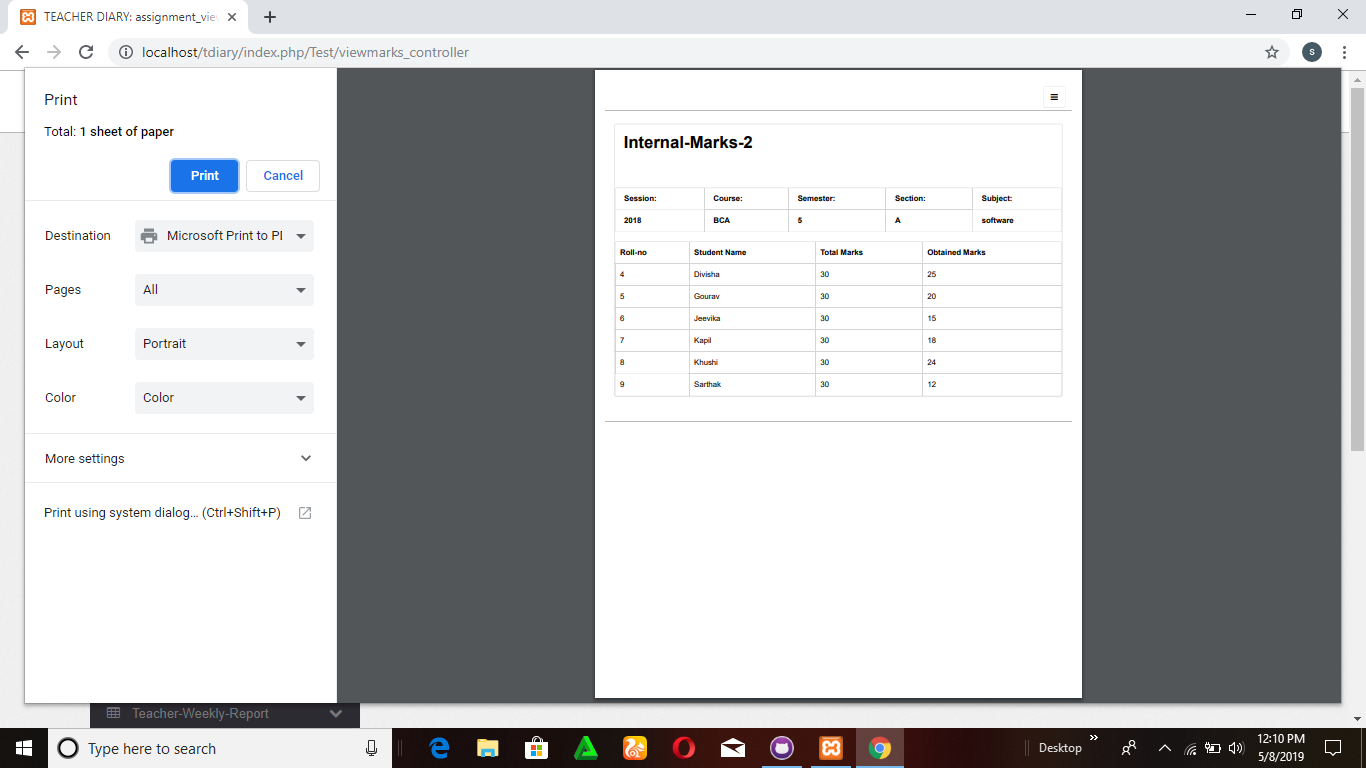


Figure : Internal Marks

* + - 1. ASSIGNMENT MARKS

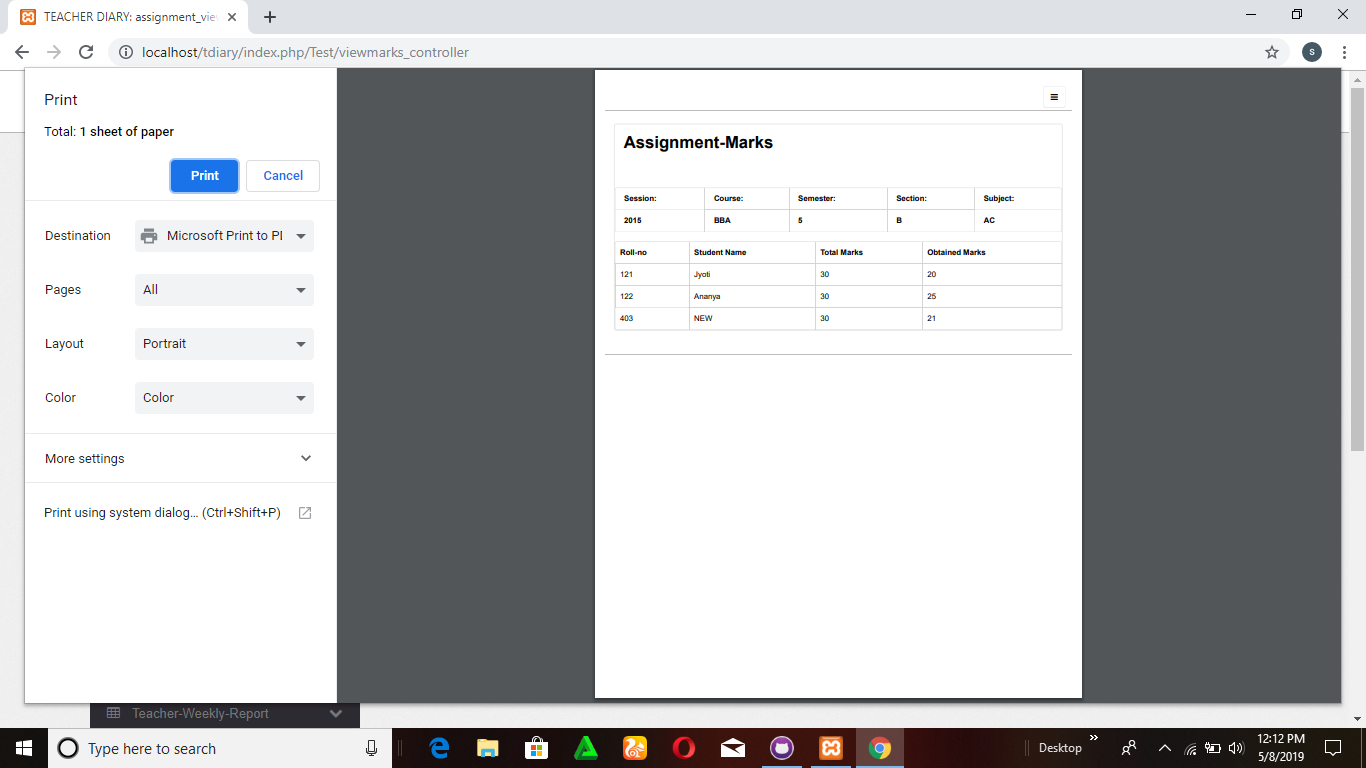


Figure : Assignment Marks

* + - 1. LECTURE VIEW

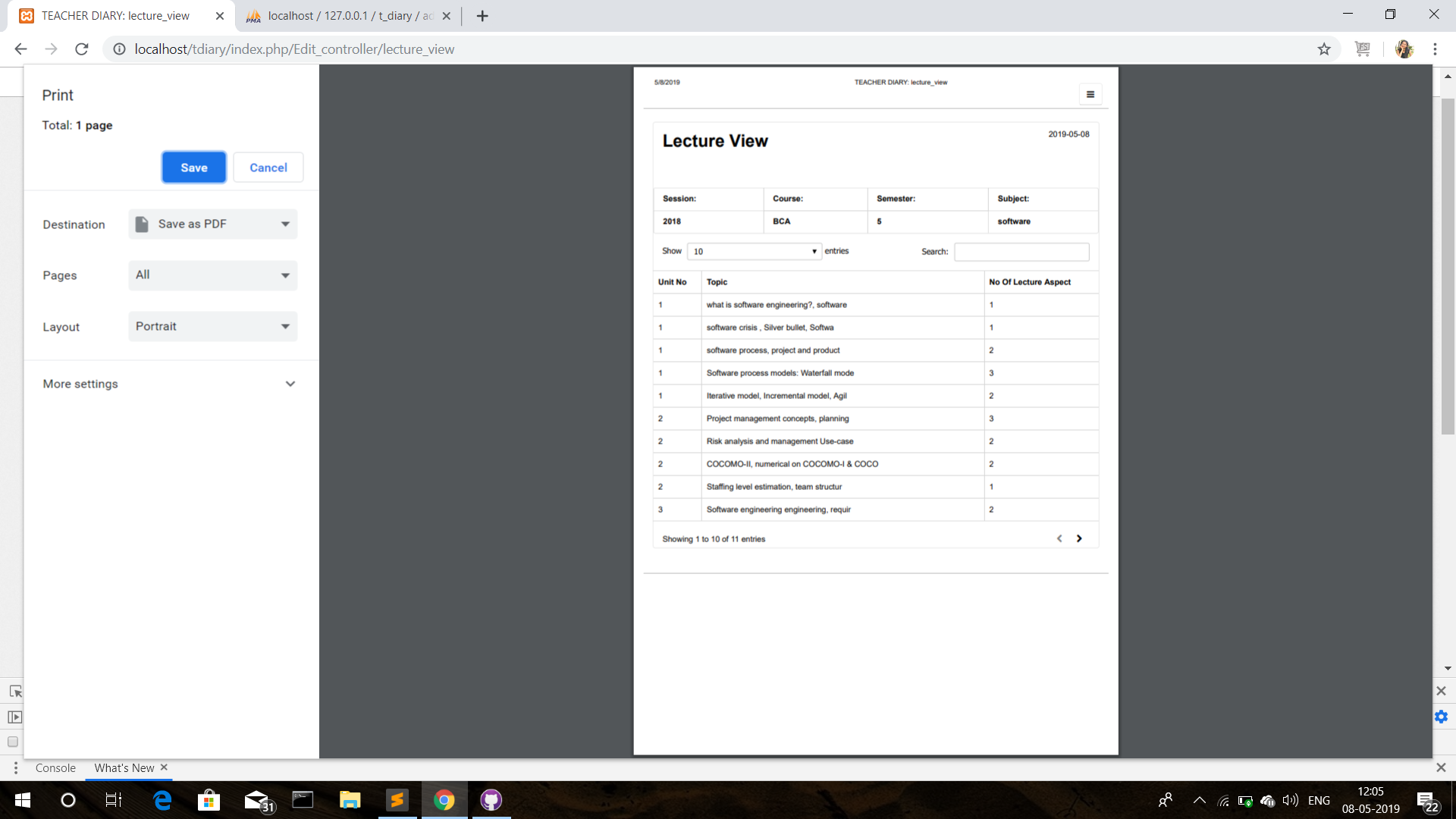


Figure : Lecture View

* + - 1. WEEKLY DETAILS

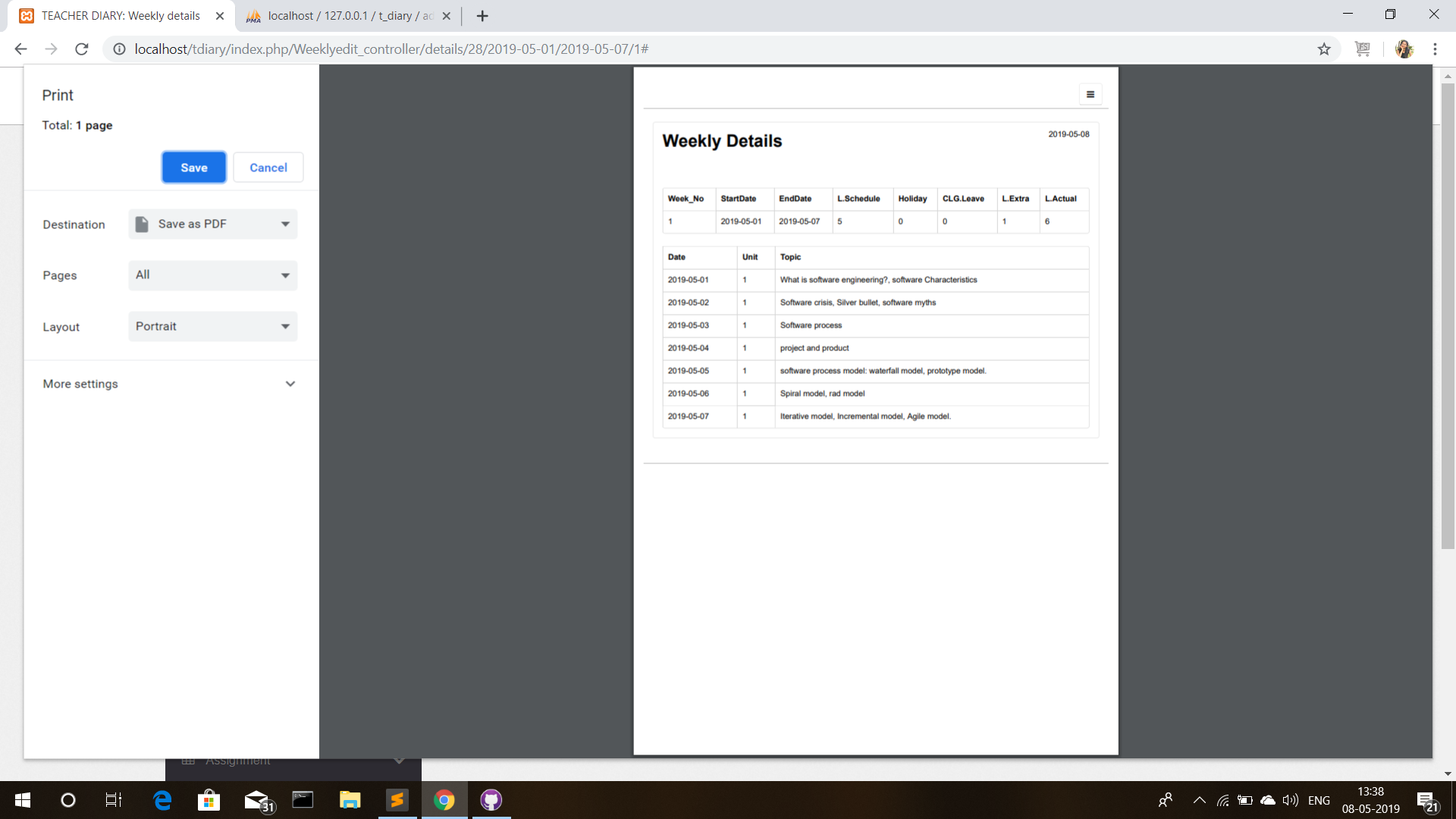


Figure : Weekly Detail