

# e- DAILY CLASS WORK REPORT

A Project Report Submitted to

**Jawaharlal Nehru Technological University, Hyderabad**

In partial fulfilment of the requirements  
for the award of the degree of

**BACHELOR OF TECHNOLOGY**  
**IN**  
**COMPUTER SCIENCE AND ENGINEERING**

by

<b>Aniruddha Ramgir</b>	<b>13E11A05D4</b>
<b>Nangunuri Rahul</b>	<b>13E11A05D0</b>
<b>M Nishanth Kumar</b>	<b>13E11A05C0</b>
<b>M Yugandher Reddy</b>	<b>13E11A05C1</b>
<b>R Vinay Reddy</b>	<b>13E11A05C5</b>

Under the guidance of

**V. Satyanarayana**  
**Associate Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
(Affiliated to JNTUH Hyderabad, Approved by AICTE and Accredited by NBA)  
Ibrahimpatnam - 501 510, Hyderabad  
**2016 – 2017**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
(Affiliated to JNTUH Hyderabad, Approved by AICTE and Accredited by NBA)  
Ibrahimpatnam - 501 510, Hyderabad

**Certificate**

This is to certify that the project work entitled “**e- Daily Class Work Report (eDCWR)**” is the bonafide work done

by

<b>Aniruddha Ramgir</b>	<b>13E11A05D4</b>
<b>Nangunuri Rahul</b>	<b>13E11A05D0</b>
<b>M Nishanth Kumar</b>	<b>13E11A05C0</b>
<b>M Yugandher Reddy</b>	<b>13E11A05C1</b>
<b>R Vinay Reddy</b>	<b>13E11A05C5</b>

in the Department of Computer Science and Engineering, **BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY**, Ibrahimpatnam is submitted to **Jawaharlal Nehru Technological University, Hyderabad** in partial fulfilment of the requirements for the award of **B.Tech** degree in **Computer Science and Engineering** during **2013-2017**.

**Guide:**

**V. Satyanarayana**

Associate Professor,  
Dept of CSE,  
Bharat Institute of Engineering and Technology,  
Ibrahimpatnam – 501 510, Hyderabad.

**Department Academic Incharge:**

**Mr. N. Sainath**

Associate Professor,  
Dept of CSE,  
Bharat Institute of Engineering and Technology,  
Ibrahimpatnam – 501 510, Hyderabad.

---

Viva-Voce held on.....

---

**Internal Examiner**

---

**External Examiner**

## **ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to **Sri CH. Venugopal Reddy**, Secretary & Correspondent of BIET, for providing congenial atmosphere and encouragement.

We would like to thank **Mr. N. Sainath, Associate Professor & Academic I/C** and **DR. J.R.V Jenny, Associate Professor & Admin I/C, Computer Science and Engineering** for their expert guidance and encouragement at various levels of our Project.

We are thankful to our guide **V. Satyanarayana, Associate Professor, Computer Science and Engineering** for his sustained inspiring Guidance and cooperation throughout the process of this project. His wise counsel and suggestions were invaluable.

We express our deep sense of gratitude and thanks to all the **teaching and non-teaching staff** of our college who stood with us during the project and helped us to make it a successful venture.

We place highest regards to our **parents**, our **friends** and **well wishers** who helped a lot in making the report of this project.

<b>Aniruddha Ramgir</b>	<b>13E11A05D4</b>
<b>Nangunuri Rahul</b>	<b>13E11A05D0</b>
<b>M Nishanth Kumar</b>	<b>13E11A05C0</b>
<b>M Yugandher Reddy</b>	<b>13E11A05C1</b>
<b>R Vinay Reddy</b>	<b>13E11A05C5</b>

## **DECLARATION**

We hereby declare that this project report is titled “**e-Daily Class Work Report (eDCWR)**” is a genuine project work carried out by us, in **B.Tech (Computer Science and Engineering)** degree course of **Jawaharlal Nehru Technology University Hyderabad, Hyderabad** and has not been submitted to any other course or university for the award of my degree by me.

**Signature of the Student**

1.

2.

3.

4.

5.

## Contents Chapter Number

<b>1. Introduction</b>	1
<b>2. Analysis</b>	
2.1 Feasibility Study.....	4
2.2 Requirement Analysis .....	5
2.3 Current Conditions.....	6
2.4 Current Solutions .....	7
2.5 Proposed Solution.....	8
2.6 Modules.....	9
<b>3. Design</b>	
3.1 Introduction.....	10
3.2 UML Diagrams .....	11
3.3 E-R Diagram .....	15
<b>4. Technologies used</b>	
4.1 HTML.....	17
4.2 PHP.....	19
4.2 mySQL.....	13
4.2 JavaScript.....	26
4.2 CSS.....	27
<b>5. Implementation and Coding</b>	
5.0 Introduction.....	29
5.1 Login Page .....	29
5.2 Admin Page.....	30
5.3 System Setup Page.....	36
5.4 Report Entry Page.....	40
5.5 DCWR Page.....	45
5.6 Schedule Page .....	48
5.7 Lesson-plan Page .....	50
<b>6. Testing and Validation</b>	
6.1 Introduction.....	53
6.2 Types of Testing .....	54
6.3 Test Cases.....	55
<b>7. Output Screens.....</b>	56
<b>8. Conclusion .....</b>	64
<b>9. Bibliography</b>	
9.1 Book References .....	66
9.2 Websites.....	66

# **Chapter 1**

# **INTRODUCTION**

# 1. INTRODUCTION

## 1.1 Problem Definition

In order to maintain classes throughout the semester, Academic Institutions follow a course/lesson plan that informs about the order in which classes shall be held. Daily reports are generated and are periodically cross-validated with the lesson plan, such that any inconsistencies can be identified and managed; doing so aids the institutions in reaching course deadlines effectively and in completing curricula without any hurry or stress. This report is known as *Daily Class Work Report*. A typical Daily Class Work Report requires three candidates, a Report book, and a Lesson-plan document.

A Lesson-plan is created by Academic In-charges, after considering various factors such as potential holidays, with the aim that it must be followed for a stress-free administration.

Each page of a Daily Class Work Report book has a table with rows about the hours in a day and the columns about the days in a week. Each table cell has two boxes, one for the subject details and another for the topic discussed in that subject; and between table cells, there is an additional space provided for remarks related to the above cell. An additional column is provided for signature and details of the Class Representative.

These candidates consist of a *Class Representative* (a student), a *Class In-charge* (or a mentor) and an *Academic In-charge*. The CR (Class Representative) notes down the details regarding the classes conducted and the topics discussed on the DCWR (Daily Class Work Report) and forwards it to the CI (Class In-charge). The CI verifies the report for errors and in turn forwards it to the Academic In-Charge. They cross-check the report with the lesson-plan and make relevant remarks wherever necessary.

## 1.2 Objective of the project

e-Daily Class Work Report, as a project, has a set of defined deliveries and objectives, and they are as follows.

- Provide a clean and intuitive User-Interface for the participants.
- Provide hassle-free data entry support for Class Representatives.
- Use ‘colour-flags’ on the report, such that understanding the report becomes easier for the In-charges.
- Design a non-redundant database model.
- Develop a set of well-coupled web pages.
- Implement secure authentication functionalities.

## **Chapter 2**

# **ANALYSIS**

## 2. ANALYSIS

### 2.1 Feasibility Study

Below, we discuss the feasibilities of the proposed solution.

#### 2.1.1 Technical Feasibility

We believe that it is technically feasible, since there will not be much difficulty in gathering resources required for the development of this system and its maintenance. We will be utilising the resources which are already available in the organisation.

#### 2.1.2 Economic Feasibility

Development of this application is economically feasible, as the organisation does not need to spend much money to the develop of this system. The development will only require an environment with an effective supervision. By doing so, we can attain maximum usability of the current resources, and the organisation will not need to invest more to maintain it after its development.

### 2.2 Requirement Analysis

The development of e-DCWR has been done by using various open-source softwares, and they are listed below.

- **XAMPP:**

It is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

- **Notepad++:**

Notepad++ is a text editor and source code editor for use with Microsoft Windows. Unlike Notepad, the built-in Windows text editor, it supports tabbed editing, which allows working with multiple open files in a single window. Notepad++ is one of the most popular source code editors in the world, and supports syntax highlighting and code folding for over 50 programming, scripting, and mark-up languages.

- **Google Chrome:**  
Google Chrome is a freeware web browser developed by Google. It was first released in 2008, for Microsoft Windows, and was later ported to Linux, OS X, iOS and Android. Google Chrome is also the main component of Chrome OS, where it serves a platform for running web apps. It was mainly used for testing purposes.
- **Github:**  
GitHub is a web-based Git repository hosting service. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.  
GitHub offers both plans for private repositories, and free accounts which are commonly used to host open-source software projects.

### 2.2.1 System requirements

To efficiently run e-DCWR on a client computer, it must satisfy a few requirements which are stated below.

- **Google Chrome:**  
Google Chrome is a freeware web browser developed by Google. Any other equivalent browser works fine as well.
- **Internet access:**  
512 Kbps or above is required for a smooth experience.

## **2.3 Existing Conditions**

### **2.3.1 Overview**

As mentioned earlier, the present system requires a physical book that must be maintained for a proper execution. But, there are physical limitations to this system and are addressed in the following section.

### **2.3.2 Limitations**

Like every other system, Daily Class Work Report model also has its problems, and most of them seem to be physical limitations.

- The book maintained by the institutions requires a lot of data to be crammed into tiny table cells. Interpreting the information from those tables can be just as difficult as entering data into it.
- The Report must be carried to the Class In-charge every time a clarification is needed or it must be submitted for approval. This consumes a lot of valuable time of a student and of the faculty.
- Academic In-charges must put aside their work, in order to cross-check the reports of each class in their departments with the lesson plan, every time they are submitted.
- If any error gets detected by the Academic In-charge, the whole process must be restarted, wasting a lot of time and energy.
- Data can be tampered with, by anybody and discovering the tampered data will be difficult for the management.
- All previous records must be traversed, in order to access past- information.
- Every class requires a new DCWR book every semester. In a single academic semester of 5 departments, with 3 sections each, at least 100 books will be needed; this results in an unnecessary use of paper and natural resources.

The overall expense in time and resources does not make the existing system suitable for use, thus requiring an overhaul.

## **2.4 Current Solutions**

### **2.4.1 Overview**

A pre-existing alternative to the above mentioned system is using software such as Microsoft Excel. Using such software has the following benefits and cons.

### **2.4.2 Advantages**

- Does not require any kind of paper or related product, as everything will be stored virtually.
- The data can be entered without as much difficulty, and reading through the reports will be significantly easier.
- Instead of carrying the DCWR to appropriate In-charges, they can be sent online, conveniently.

### **2.4.3 Disadvantages**

- Requires using third-party cloud vendors for storing and accessing reports.
- Although data entry and retrieval is improved, maintaining documents will be a hassle, as every year, at least 100 documents must be maintained, assuming there are 5 departments in total, with 3 sections each.
- Microsoft Excel must be owned by the student, class In-charges and Academic In-charges for this to work.
- Redundancy is an issue, as every Excel sheet will have similar rows such as department name, batch, year, semester and section.

## **2.5 Proposed Solution**

### **2.5.1 Overview**

The proposed solution for the problems, mentioned in the previous systems, is using a separate website for the DCWR needs. This system is called *e-Daily Class Work Report*. The website is designed and developed solely for this purpose and it also allows for future improvements without much difficulty. The system is robust, cost-efficient and secure, making it a much suitable than others.

### **2.5.2 Advantages**

The advantages provided by e- DCWR are similar to the alternative and improves, exponentially, on it.

- Does not require any kind of paper or related product, as everything will be stored on the Institutional database.
- The specifically designed User- Interface allows for a smooth data entry and storing.
- Reading through the data is also improved, as the system uses different ‘colour-flags’ such as red, green and orange to isolate inconsistencies.
- As everything is stored and maintained online, in the database, a single click of a button is enough to forward, submit or approve the report. A single click can also help In-charges in sending remarks to the CRs regarding errors.
- RDBMS allows for storing data with very less redundancy.
- All previous records can be accessed in this system.

## 2.6 Modules

e- DCWR uses various modules to improve readability and the maintenance (debugging) of the project. Each module, described below, can be made up of multiple web pages or can be used in multiple web pages. The prime modules are described below.

- **Login Module:**  
Authenticates the user details and based on the account type, it redirects to an appropriate module
- **Academic In-charge Module:**  
Presents a unique page for the Academic In-charge, wherein they can select a class in their respective departments. During the beginning of a New Year or semester, it allows for Academic In-charges to create and upload new schedules, Lesson-plans and other such data.
- **Class Representative Module:**  
Allows Class Representatives to add or modify details relating to classes that have occurred, during the semester.
- **Reporting Module:**  
The data, entered by the CR, will be stored in the database and a report will be generated. This report will be displayed to the respective CRs, Class In-charges and Academic In-charges. The report will have an option for approval and also an option that allows for In-charges to notify lower level members to make respective changes, in case of any inconsistencies.  
This module is incorporated in the ‘Home page’, where Class In-charges and Academic In-charges can access other data such as Lesson-plan, Schedule details.

# **Chapter 3**

# **DESIGN**

## 3. DESIGN

### 3.1 Introduction

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organised in a way that supports reasoning about the structures and behaviours of the system. A system architecture can comprise system components, the expand systems developed, that will work together to implement the overall system.

### 3.2 UML Diagrams

#### 3.2.1 Deployment Diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artefacts to deployment targets. Artefacts represent concrete elements in the physical world that are the result of a development process. Examples of artefacts are executable files, libraries, archives, database schemas, configuration files, etc. Deployment target is usually represented by a node which is either hardware device or some software execution environment. Nodes could be connected through communication paths to create networked systems of arbitrary complexity.

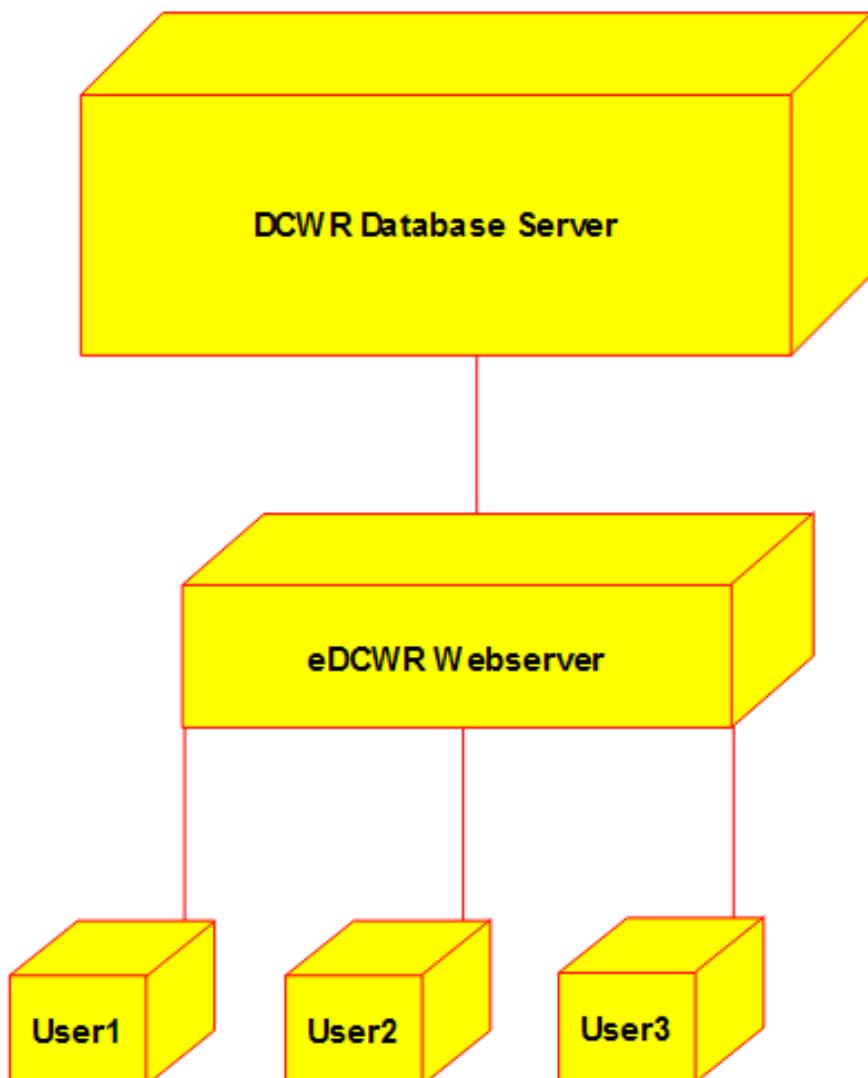


Figure 3.2.1: Deployment Diagram

### 3.2.2 Use Case Diagram #1

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

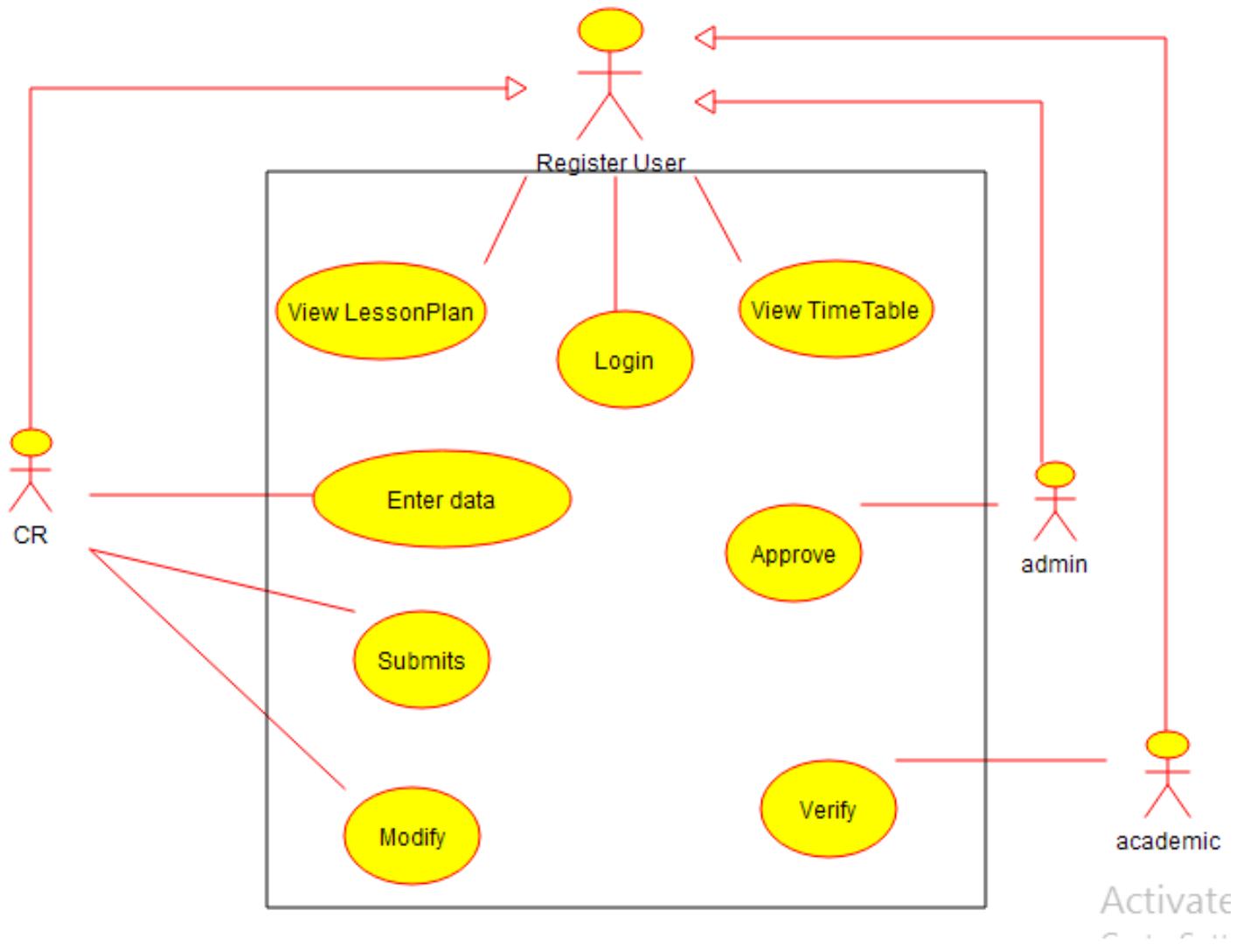


Figure 3.2.2 : Use Case Diagram

### 3.2.3 Use Case Diagram #2

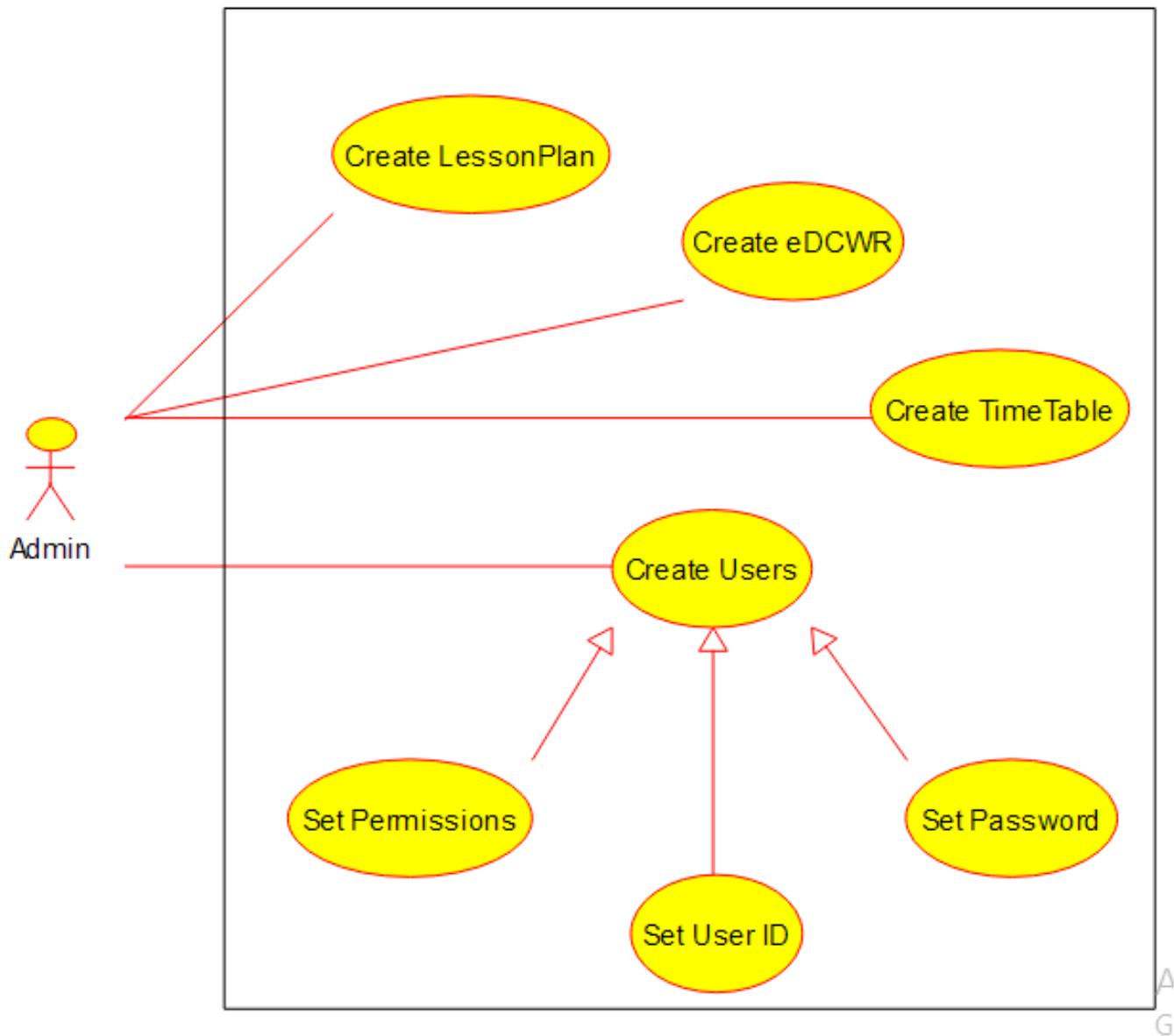


Figure 3.2.3 : Use Case Diagram

### 3.2.4 Activity Diagram

Activity diagram is UML behaviour diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow. The actions coordinated by activity models can be initiated because other actions finish executing, because objects and data become available, or because some events external to the flow occurs.

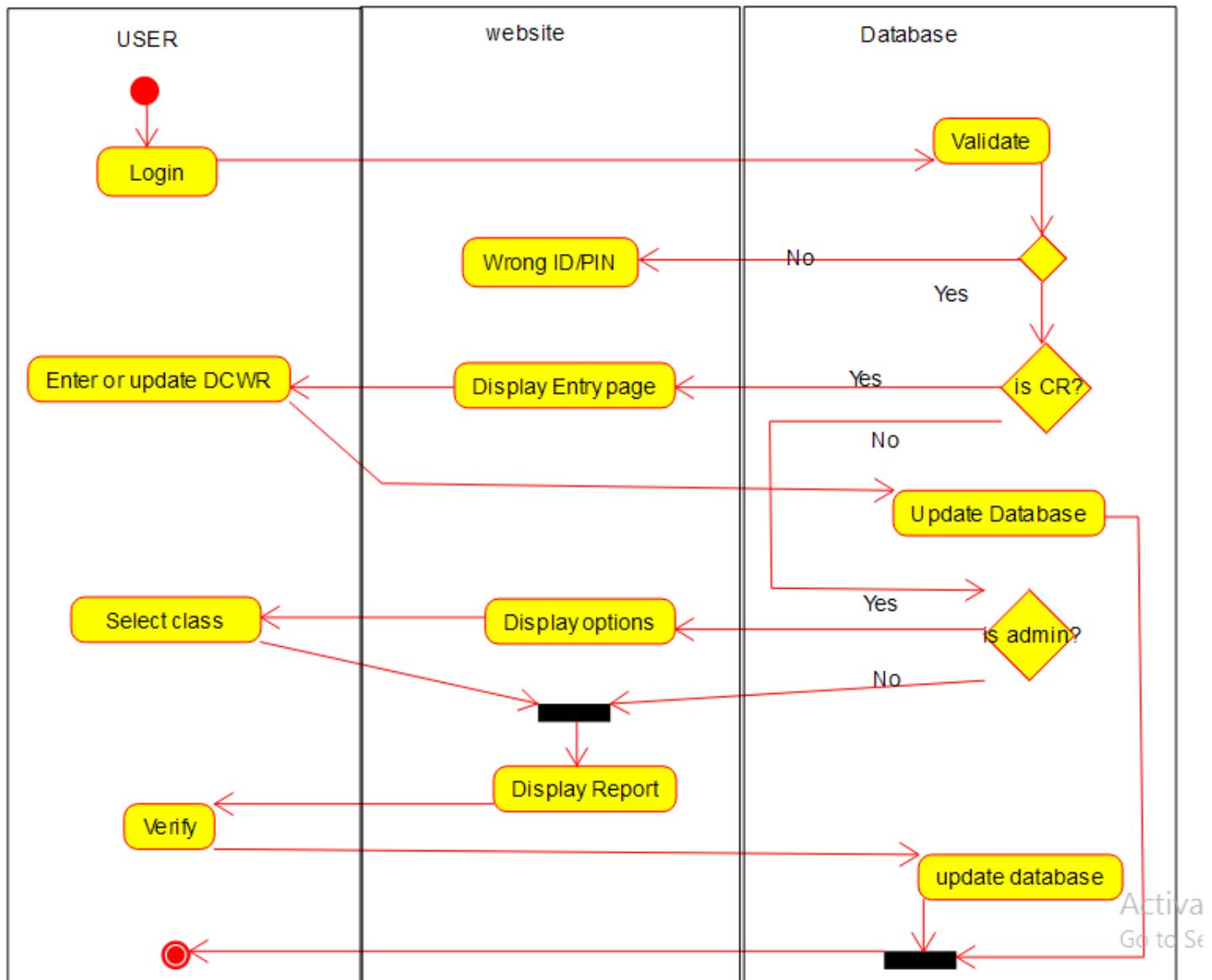


Figure 3.2.4 : Activity Diagram

### 3.3 Database Architecture / ER Diagram

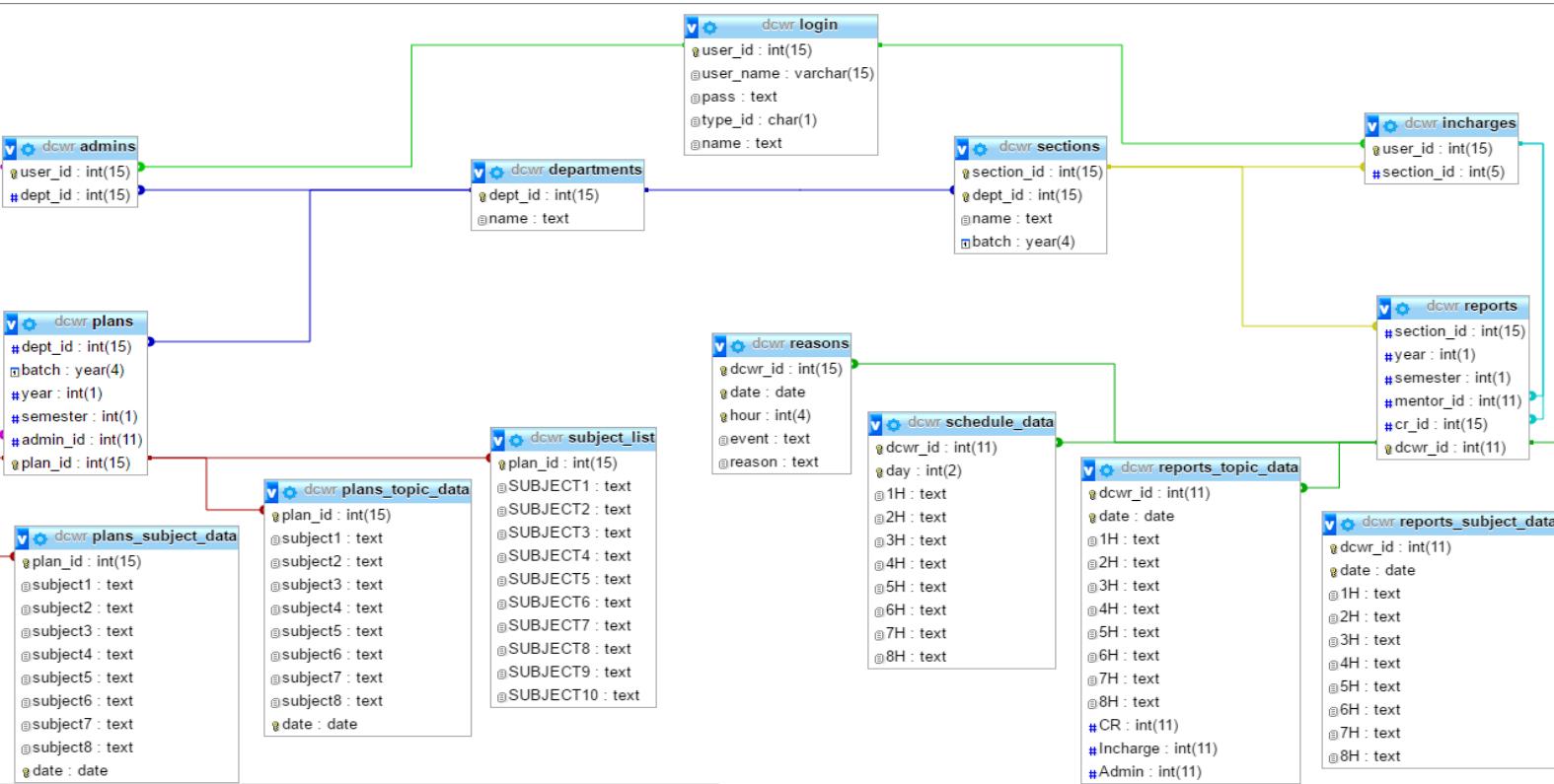


Figure 3.3 : E-R Diagram

## **Chapter 4**

# **TECHNOLOGIES USED**

## 4. TECHNOLOGIES USED

### 4.1 HTML

Html is a language which is used to create web pages with html marking up a page to indicate its format, telling the web browser where you want a new line to begin or how you want the text or images aligned and more are possible. We have used the following tags in our project.

#### 4.1.1 Table

Tables are so popular with web page authors as they let you arrange the elements of a web page in such a way that the browser won't rearrange them. Web page authors frequently use tables to structure web pages.

#### TR

TR is used to create a row in a table encloses <TH> and <TD> elements. <TR> contains many attributes. Some of them are

- ALIGN: specifies the horizontal alignment of the text in the table row.
- BGCOLOR: Specifies the background colour for the row.
- BORDERCOLOR: Sets the external border colour for the row.
- VALIGN: Sets the vertical alignment of the data in this row.

#### TH

TH is used to create table heading.

- ALIGN: Sets the horizontal alignment of the content of the table cell. Sets LEFT, RIGHT.
- BACKGROUND: Specifies the background image for the table cell.
- BGCOLOR: Specifies the background colour of the table cell
- VALIGN: Sets the vertical alignment of the data. Sets to TOP, MIDDLE, BOTTOM or BASELINE.
- WIDTH: Specifies the width of the cell. Set to a pixel width or a percentage of the display area.

#### TD

TD is used to create table data that appear in the cells of a table.

- ALIGN: Specifies the horizontal alignment of content in the table cell. Sets to LEFT, CENTER, RIGHT.
- BGCOLOR: Specifies the background image for the table cell.
- BGCOLOR: sets the background colour of the table cells.
- WIDTH: Specifies the width of the cell

#### **4.1.2 Frames**

Frames are used to either run off the page or display only small slices of what are supposed to be shown and to configure the frame we can use <FRAMESET>. There are two important points to consider when working with <FRAMESET>.

- <FRAMESET> element actually takes the place of the <BODY> element in a document.

#### **4.1.3 Form**

The purpose of FORM is to create an HTML form; used to enclose HTML controls, like buttons and text fields.

##### **Attributes:**

- ACTION: Gives the URL that will handle the form data.
- NAME: Gives the name to the form so you can reference it in code set to an alphanumeric string.
- METHOD: method or protocol is used to send data to the target action URL. The GET method is the default, it is used to send all form name/value pair information in an URL. Using the POST method, the contents of the form are encoded as with the GET method, but are sent in environment variables.

#### **4.1.4 Controls in HTML**

##### **<INPUT TYPE =BUTTON>:**

Creates an html button in a form.

##### **Attributes:**

- NAME: gives the element a name. Set to alphanumeric characters.
- SIZE: sets the size.
- VALUE: sets the caption of the element.

##### **<INPUT TYPE = PASSWORD>:**

Creates a password text field, which makes typed input.

##### **Attributes:**

- NAME: gives the element a name, set to alphanumeric characters.
- VALUE: sets the default content of the element.

##### **<INPUT TYPE=SUBMIT>:**

Creates a submit button that the user can click to send data in the form back to the web server.

##### **Attributes:**

- NAME: Gives the element a name. Set to alphanumeric characters.
- VALUE: Gives this button another label besides the default, Submit Query. Set to alphanumeric characters.

##### **<INPUT TYPE=TEXT>:**

Creates a text field that the user can enter or edit text.

##### **Attributes:**

- NAME: Gives the element a name. Set to alphanumeric characters.
- VALUE: Holds the initial text in the text field. Set to alphanumeric characters.

## 4.2 PHP

PHP is a server-side scripting language designed primarily for web development, but is also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP*: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification.

### 4.2.1 Syntax

```
<!DOCTYPE html>
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo'<p>Hello World</p>';?>
</body>
</html>
```

An example of PHP ‘for loop’:

```
<?php
for($x=0;$x<=100;$x++){
echo"The number is: $x<br>";
}?>
```

## 4.2.2 Data types

PHP stores integers in a platform-dependent range, either a 64-bit or 32-integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from that of other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, hexadecimal, and binary notations.

Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++.

Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources.

Arrays can contain elements of any type that PHP can handle, including resources, objects, and other arrays. Order is preserved in lists of values and in hashes with both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, nowdoc or heredoc syntax.

## 4.2.3 Functions

PHP defines a large array of functions in the core language and many are also available in various extensions; these functions are well documented in the online PHP documentation. However, the built-in library has a wide variety of naming conventions and associated inconsistencies, as described under history above.

Custom functions may be defined by the developer, e.g.:

```
function myAge($birthYear){// defines a function, this one is named "myAge"
$yearsOld=date('Y')-$birthYear;// calculates the age
return $yearsOld.' year' . ($yearsOld!=1?'s':");// returns the age in a descriptive form
}

echo'I am currently '.myAge(1981).' old.';// outputs the text concatenated// with the return
value of myAge()
```

#### **4.2.4 Object-Oriented Programming**

Basic object-oriented programming functionality was added in PHP 3 and improved in PHP 4. This allowed for PHP to gain further abstraction, making creative tasks easier for programmers using the language. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like value types. The drawback of this method was that code had to make heavy use of PHP's "reference" variables if it wanted to modify an object it was passed rather than creating a copy of it. In the new approach, objects are referenced by handle, and not by value.

PHP 5 introduced private and protected member variables and methods, along with abstract classes, final classes, abstract methods, and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model.

If the developer creates a copy of an object using the reserved word `clone`, the Zend engine will check whether a `__clone()` method has been defined. If not, it will call a default `__clone()` which will copy the object's properties. If a `__clone()` method is defined, then it will be responsible for setting the necessary properties in the created object. For convenience, the engine will supply a function that imports the properties of the source object, so the programmer can start with a by-value replica of the source object and only override properties that need to be changed.

The following is a basic example of object-oriented programming in PHP:

```
class Person
{
    public $firstName;
    public $lastName;

    public function __construct($firstName, $lastName = "") {
        $this->firstName = $firstName;
        $this->lastName = $lastName;
    }

    public function greet() {
        return 'Hello, my name is ' . $this->firstName .
            (($this->lastName != "") ? (' ' . $this->lastName) : '');
    }

    public static function staticGreet($firstName, $lastName) {
        return 'Hello, my name is ' . $firstName . ' ' . $lastName . '';
    }
}

$he = new Person('John', 'Smith');
$she = new Person('Sally', 'Davis');
$other = new Person('iAmine');

echo $he->greet(); echo '<br />';
echo $she->greet(); echo '<br />';
echo $other->greet();
echo '<br />';

echo Person::staticGreet('Jane', 'Doe');
```

## 4.3 MySQL

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications.

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

**A Relational DataBase Management System (RDBMS)** is a software that:

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

### 4.3.1 RDBMS Terminology:

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

#### **Database:**

A database is a collection of tables, with related data.

#### **Table:**

A table is a matrix with data. A table in a database looks like a simple spreadsheet.

#### **Column:**

One column (data element) contains data of one and the same kind, for example the column postcode.

#### **Row:**

A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.

#### **Redundancy:**

Avoids storing data more than once, causing the system to faster.

#### **Primary Key:**

A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.

**Foreign Key:**

A foreign key is the linking pin between two tables.

**Compound Key:**

A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.

**Index:**

An index in a database resembles an index at the back of a book.

**Referential Integrity:**

Referential Integrity makes sure that a foreign key value always points to an existing row.

**4.3.2 Key points as to why MySQL is used:**

1. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:
2. MySQL is released under an open-source license. So you have nothing to pay to use it.
3. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
4. MySQL uses a standard form of the well-known SQL data language.
5. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
6. MySQL works very quickly and works well even with large data sets.
7. MySQL is very friendly to PHP, the most appreciated language for web development.
8. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
9. MySQL is customisable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

```
mysql> help
```

#### 4.3.3 List of all MySQL commands:

Note that all text commands must be first on line and end with ;'

?	(?) Synonym for 'help'.
clear	(\c) Clear the current input statement.
connect	(\r) Reconnect to the server. Optional arguments are db and host.
delimiter	(\d) Set statement delimiter.
edit	(\e) Edit command with \$EDITOR.
ego	(\G) Send command to mysql server, display result vertically.
exit	(\q) Exit mysql. Same as quit.
go	(\g) Send command to mysql server.
help	(\h) Display this help.
nopager	(\n) Disable pager, print to stdout.
notee	(\t) Don't write into outfile.
pager	(\P) Set PAGER [to_pager]. Print the query results via PAGER.
print	(\p) Print current command.
prompt	(\R) Change your mysql prompt.
quit	(\q) Quit mysql.
rehash	(\#) Rebuild completion hash.
source	(\.) Execute an SQL script file. Takes a file name as an argument.
status	(\s) Get status information from the server.
system	(\!) Execute a system shell command.
tee	(\T) Set outfile [to_outfile]. Append everything into given out file.
use	(\u) Use another database. Takes database name as argument.
charset	(\C) Switch to another charset. Might be needed for processing

## 4.4 Java Script

Java script originally supported by Netscape navigator is the most popular web scripting language today. Java script lets you embed programs right in your web pages and run these programs using the web browser. You place these programs in a <SCRIPT> element, usually within the <HEAD> element. If you want the script to write directly to the web page, place it in the <BODY> element. JavaScript is the programming language of HTML and the Web. Programming makes computers do what you want them to do. JavaScript is easy to learn.

### 4.4.1 JavaScript Methods:

#### **WriteLn:**

Document.writeLn() is a method, which is used to write some text to the current web page.

#### **onClick:**

Occurs when an element is clicked.

#### **onLoad:**

Occurs when the page loads.

#### **onMouseDown:**

Occurs when a mouse button goes down.

#### **onMouseMove:**

Occurs when the mouse moves.

#### **onUnload:**

Occurs when a page is unloaded.

### 4.4.2 JavaScript Can Change HTML Content

One of many JavaScript HTML methods is getElementById(). This example uses the method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript":

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

### JavaScript Can Change HTML Styles (CSS)

Changing the style of an HTML element, is a variant of changing an HTML attribute:

```
document.getElementById("demo").style.fontSize = "25px";
```

### JavaScript Can Hide/Show HTML Elements

Hiding HTML elements can be done by changing the display style:

```
document.getElementById("demo").style.display = "none";  
document.getElementById("demo").style.display = "block";
```

## 4.5 CSS

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files

### 4.5.1 Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes. CSS solved a Big Problem. HTML was never intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

When tags like `<font>`, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

The style definitions are normally saved in external .css files.

With an external stylesheet file, you can change the look of an entire website by changing just one file!

### 4.5.2 CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

## **Chapter 5**

# **IMPLEMENTATION AND CODING**

## 5. IMPLEMENTATION AND CODING

### 5.1 Introduction

All of the major web pages and their scripts are described below. The languages that were used to implement e-DCWR are HTML, PHP, JavaScript and CSS; quite a lot of MySQL queries are also included in the PHP scripts.

### 5.2 Login Page

```
<?php
if(isset($_POST['submit'])){
    validate();
}
function validate(){
include("connection.php");
$result=$conn->query(" SELECT * FROM login where user_name = '".$_POST[user]' AND
pass = '".$_POST[pass]"');
if(!empty($row=$result->fetch_assoc())){
$_SESSION['logged_in']=1;
$_SESSION['user_id']=$row['user_id'];
if($row[type_id]=='A'){
unset($_GET['loginFailed']);
header('Location: admin_info.php');
exit();
}
elseif($row['type_id']=='C'){
unset($_GET['loginFailed']);
header('Location: incharge_process.php');
exit();
}
else{
unset($_GET['loginFailed']);
header('Location: incharge_process.php');
exit();
}
}
else{
session_unset();
session_destroy();
$conn->close();
header('Location: login.php?loginFailed=incorrect');//Incorrect password message is not being displayed if this line executes
die();
}
}
if(isset($_GET['loginFailed'])){
if($_GET["loginFailed"]=="incorrect"){


```

```

echo"<script type='text/javascript'>alert('Incorrect Username and/or password')</script>";
}
if($_GET["loginFailed"]=="AuthFailed"){
echo"<script type='text/javascript'>alert('Authorization Failed. You are not allowed to access
the current DCWR.')</script>"; } }?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Login to eDCWR</title>
<link rel="stylesheet" href="css/reset.css">
<link rel='stylesheet prefetch' href='http://fonts.googleapis.com/css?family=Roboto:
400,100,300,500,700,900|RobotoDraft:400,100,300,500,700,900'>
<link rel='stylesheet prefetch' href='http://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/
css/font-awesome.min.css'>
<link rel="stylesheet" href="css/login.css">
</head>
<body>
<!-- Pen Title-->
<div class="pen-title">
<h1>eDCWR</h1>
</div>
<div class="container">
<div class="card"></div>
<div class="card">
<h1 class="title">Login</h1>
<form role = "form" action="login.php" method="post">
<div class="input-container">
<input type="text" name="user" required="required"/>
<label for="Username">Username</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="password" name="pass" required="required"/>
<label for="Password">Password</label>
<div class="bar"></div>
</div>
<div class="button-container">
<button type="submit" name="submit"><span>Submit</span></button>
</div>
<!-- <div class="footer"><a href="#">Forgot your password?</a></div> -->
</form>
</div>
</div>
</body>
</html>

```

## 5.3 Admin Page

```

$_SESSION['semester']=$_POST['semester'];
$_SESSION['mentor']=$_POST['mentor'];
$result=$conn->query('SELECT * FROM admins WHERE user_id = '.$_SESSION['user_id'].'');//get dept_id
if(!empty($row=$result->fetch_assoc())){
$_SESSION['dept_id']=$row['dept_id'];
//echo $_SESSION['dept_id'];
}
$result=$conn->query('SELECT * FROM sections WHERE batch= "'.$_POST['batch'].'" AND ( dept_id= '.$_SESSION['dept_id']. ' AND name= "'.$_POST['section'].'")');//get section_id
if(!empty($row=$result->fetch_assoc())){
$_SESSION['section_id']=$row['section_id'];
//echo $_SESSION['section_id'];
}
else{
echo "Entered Section does not exist in your department";
}
$result=$conn->query('SELECT * FROM `incharges` WHERE user_id = '.$_SESSION['mentor'].'');//get Mentor details.
if(!empty($row=$result->fetch_assoc())){
if($_SESSION['mentor']==$row['user_id']){//check if entered Mentor exists and only then, insert new dcwr row.
$result=$conn->query('SELECT COUNT(*) AS SUM FROM `reports` WHERE section_id='.$_SESSION['section_id']. ' AND ( mentor_id = '.$_SESSION['mentor']. ' AND (year='.$_SESSION['year']. ' AND semester='.$_SESSION['semester'].'))');
if(!empty($row=$result->fetch_assoc())){//Add to reports table only if the record does not exist //does not work. It will not enter the if statement, if the record does not exist
if($row['SUM']>0){
echo "<script type='text/javascript'>alert('Record already exists!')</script>";
}
else{
if($conn->query('INSERT INTO `reports`(`section_id`, `year`, `semester`, `mentor_id`) VALUES ('.$_SESSION['section_id'].','.$_SESSION['year'].','.$_SESSION['semester'].','.$_SESSION['mentor'].')')===TRUE){
header('Location: admin_info.php');//Redirect to itself after a successful Insertion
exit();
}
else{
echo "<script type='text/javascript'>alert('Could not assign DCWR_ID for the entered details.')</script>";
}
return;
}//empty Record code---INSERT HERE
}
}
}
else{

```

```

echo'Mentor ID does not exist';
} } }
?>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Admin Page</title>
<link rel="stylesheet" href="css/hamburger.css">
<link rel="stylesheet" href="css/reset.css">
<link rel='stylesheet prefetch' href='http://fonts.googleapis.com/css?family=Roboto:400,100,300,500,700,900|RobotoDraft:400,100,300,500,700,900'>
<link rel='stylesheet prefetch' href='http://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css'>
<link rel="stylesheet" href="css/login.css">
</head>
<body>
<header>
<button class="hamburger">&#9776;</button>
<button class="cross">&#735;</button>
</header>

<div class="menu">
<ul>
<a href="setup.php"><li>SETUP</li></a>
<a href="logout.php"><li>LOG OUT</li></a>
</ul>
</div>
<!-- Mixins-->
<!-- Pen Title-->
<div class="pen-title">
<h1>Additional Information Required</h1>
</div>
<div class="container">
<div class="card"></div>
<div class="card">
<h1 class="title">DCWR Details</h1>
<form role = "formaction"="admin_info.php" method="post">
<div class="input-container">
<input type="text" name="batch" required="required"/>
<label for="Username">Batch</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="year" required="required"/>
<label for="Username">Year</label>

```

```

<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="semester" required="required"/>
<label for="Username">Semester</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="section" required="required"/>
<label for="Username">Section</label>
<div class="bar"></div>
</div>
<div class="button-container">
<button type="submit" name="view"><span>Submit</span></button>
</div>
</form>
</div>
<div class="card alt">
<div class="toggle"></div>
<h1 class="title">Register</h1>
<div class="close"></div>
<form role="form" action="admin_info.php" method="post">
<div class="input-container">
<input type="text" name="batch" required="required"/>
<label for="Username">Batch</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="year" required="required"/>
<label for="Username">Year</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="semester" required="required"/>
<label for="Username">Semester</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="section" required="required"/>
<label for="Username">Section</label>
<div class="bar"></div>
</div>
<div class="input-container">
<input type="text" name="mentor" required="required"/>
<label for="Username">Mentor ID</label>
<div class="bar"></div>

```

```
</div>
<div class="button-container">
<button type="submit" name="create"><span>Submit</span></button>
</div>
</form>
</div>
</div>
<script src='js/button.js'></script>
<script src="js/index.js"></script>
<script src='js/navbar_animate.js'></script>
</body>
</html>
```

## 5.4 System Setup Page

```
<?php
include("connection.php");
include("logged_in.php");
if(isset($_GET['result'])){
switch($_GET["result"]){
case0:
echo "<script type='text/javascript'>alert('Upload success.');//</script>";
break;
case1:
"<script type='text/javascript'>alert('1 upload failed.');//</script>";
break;
case2:
"<script type='text/javascript'>alert('2 uploads failed.');//</script>";
break;
}
}
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>System Setup</title>
<link rel="stylesheet" href="css/home.css">
</head>
<body>
<!-- / eDCWR -->
<form role = formmethod="post" action="upload.php" enctype="multipart/form-data">
<input type="submit" id="submit-form" name="plan" style="display:none;">
<div class="tab">
<table align="center" border='0' cellpadding='0' cellspacing='0'>
<tr class='time'>
<th>Batch</th>
<th>Year</th>
<th>Semester</th>
</tr>
<tr>
<td colspan="3"><!-- batch -->
<?php //Extract batch years from Sections table
$sql=$conn->query("SELECT DISTINCT `batch` FROM `sections` ORDER BY `batch` DESC ");
//GET ALL Batches in the database
$rows=$sql->fetch_all(MYSQLI_ASSOC); //I dont get this part either. http://stackoverflow.com/questions/8849201/how-to-load-mysqli-result-set-into-two-dimensional-array
$result=array_column($rows,'batch');// http://php.net/manual/en/function.array-column.php
//Both above lines are used to convert a single column into a 2D array with index.
```

```

echo "<td class='text' >";
echo '<select class="select-style" name="batch">';
for($count=0;$count<4;$count++) { //Go through 4 most recent batches (descending order)
if(($result[$count])!=NULL) { //display only if its not NULL. USELESS
echo '<option value="'. $result[$count].'">' . ($result[$count]).'</option>'; //echo each row in the
column using index.
}
}
echo "</select>";
echo "</td>";
?>
<td class='text'><!-- Year -->
<select class="select-style" name="year">
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
</select>
</td>
<td class='text'><!-- Semester -->
<select class="select-style" name="semester">
<option value="1">1</option>
<option value="2">2</option>
</select>
</td>
</tr>
<tr>
<th class='time'>Section</th>
<th class='time'>Assign CR</th>
<th class='time'>Assign Incharge</th>
</tr>
<tr>
<!-- section name -->
<?php //Extract section names from Sections table
$sql=$conn->query("SELECT DISTINCT `name` FROM `sections` ");//GET ALL section
names in the database
$rows=$sql->fetch_all(MYSQLI_ASSOC);
$result=array_column($rows,'name');
//Both above lines are used to convert a single column into a 2D array with index.
echo "<td class='text' >";
echo '<select class="select-style" name="section_name">';
echo '<option disabled selected value>Select</option>';
foreach($result as $res) { //Go through 4 most recent batches (descending order)
echo '<option value="'. $res.'">' . ($res).'</option>'; //echo each row in the column using index.
}
echo "</select>";
echo "</td>";

```

```

?>
<?php//Extract CR IDs from Sections table
$sql=$conn->query("SELECT `user_id` FROM `login` WHERE `type_id`='C'");//GET ALL
section names in the database
$rows=$sql->fetch_all(MYSQLI_ASSOC);
$result=array_column($rows,'user_id');
//Both above lines are used to convert a single column into a 2D array with index.
echo "<td class='text' >";
echo'<select class="select-style" name="cr_id">';
echo'<option disabled selected value>select</option>';
foreach($resultas$res){//Go through 4 most recent batches (descending order)
echo'<option value='".$res."'>'.'$res'. '</option>';//echo each row in the column using index.
}
echo"\"</select>";
echo"\"</td>";
?>
<?php//Extract Incharge IDs from Sections table
$sql=$conn->query("SELECT `user_id` FROM `login` WHERE `type_id`='F'");//GET ALL
section names in the database
$rows=$sql->fetch_all(MYSQLI_ASSOC);
$result=array_column($rows,'user_id');
//Both above lines are used to convert a single column into a 2D array with index.
echo "<td class='text' >";
echo'<select class="select-style" name="incharge_id">';
echo'<option disabled selected value>select</option>';
foreach($resultas$res){//Go through 4 most recent batches (descending order)
echo'<option value='".$res."'>'.'$res'. '</option>';//echo each row in the column using index.
}
echo"\"</select>";
echo"\"</td>";
?>
</tr>
<tr>
<thclass='time'>Lesson Plan</th>
<thclass='time'>Subject list</th>
<thclass='time'>Schedule</th>
</tr>
<tr>
<tdclass='text purple'>
<divclass="fileUploadbtnbtn-primary">
< s p a n > U p l o a d < /


```

```

<tdclass='text purple'><divclass="fileUploadbtnbtn-primary">
<s p a n > U p l o a d </
span><inputclass="upload" name="schedule_data" id="schedule_data" type="file"/></
div>
</td>
</tr>
</table>
</div>
</form>
<form role = formmethod="get"action="setup.php">!-- check if role and action are nec-
essary -->
<div>
<tablealign=centerclass='options'border='0'cellpadding='5'cellspacing='0'>
<trclass='time'>
<th>SUBMIT</th>
<th>LOG OUT</th>
</tr>
<tr>
<!--
class="text green" ><input name="submit" value="SUBMIT" type="submit" class="button2
text green "></td> -->
<tdclass="text green"><labelfor="submit-form" class="button2 text green ">SUBMIT</
label></td>
<tdclass="text red"><a href="logout.php">LOG OUT</a></td>
</tr>
</table></div></form></body></html>

```

## 5.5 Report Entry Page

```
<?php
include("connection.php");//Remember to add REASONS part.
include("logged_in.php");
$id=$_SESSION['user_id'];
$login='SELECT * FROM login where user_id = '.$_SESSION['user_id'].' ';
$result=$conn->query($login);
if(!empty($row=$result->fetch_assoc())){
$name=$row['name'];
$section='SELECT * FROM sections WHERE section_id='.$_SESSION['section_id'].' ';
$dept='SELECT * FROM departments WHERE dept_id='.$_SESSION['dept_id'].' ';
$reports=' SELECT * FROM login WHERE user_id = (SELECT mentor_id FROM
reports WHERE dcwr_id='.$_SESSION['dcwr_id'].')';
$plans=' SELECT * FROM login WHERE user_id = (SELECT admin_id FROM
plans WHERE plan_id='.$_SESSION['plan_id'].')';
$result=$conn->query($section);
if(!empty($row=$result->fetch_assoc())){
$section=$row['name'];
}
$result=$conn->query($dept);
if(!empty($row=$result->fetch_assoc())){
$dept=$row['name'];
}
$result=$conn->query($reports);
if(!empty($row=$result->fetch_assoc())){
$reports=$row['name'];
}
$result=$conn->query($plans);
if(!empty($row=$result->fetch_assoc())){
$plans=$row['name'];
}
}
}

function enterData()
{
include("connection.php");
//echo "<script type='text/javascript'>alert(\"". $_POST["date"] ." ".$_POST["hour"]. "
".$_POST["event"]. ".$_POST["subject"]. "\")</script>";
$sql1=$conn->query(" SELECT * FROM `reports_topic_data` WHERE date = \"".
$_POST["date"]." and dcwr_id = ".$_SESSION['dcwr_id']." ");
$sql2=$conn->query(" SELECT * FROM `reports_subject_data` WHERE date = \"".
$_POST["date"]." and dcwr_id = ".$_SESSION['dcwr_id']." ");
```

```

if(!empty($row1=$sql1->fetch_assoc())||!empty($row2=$sql2->fetch_assoc())){//
check if record already exists
if(!($row1['Incharge']==0)){//check if Class Incharge has already signed
echo"<script type='text/javascript'>alert('Record has already been approved by the
Class Incharge. Cannot update now.')</script>;
}
else{
$conn->query('UPDATE `reports_subject_data` SET `'.$_POST["hour"].'` = "'.$_
$_POST["subject"].'" WHERE `reports_subject_data`.`dcwr_id` = '.$_
$_SESSION['dcwr_id'].' AND `reports_subject_data`.`date` = "'.$_POST["date"].'"');//
update subject

$conn->query('UPDATE `reports_topic_data` SET `'.$_POST["hour"].'` = "'.$_
$_POST["topic"].'" WHERE `reports_topic_data`.`dcwr_id` = '.$_
$_SESSION['dcwr_id'].' AND `reports_topic_data`.`date` = "'.$_POST["date"].'"');//
update topic
}
}
else{//create new record if it does not exist.
$conn->query('INSERT INTO `reports_subject_data` (`dcwr_id`, `date`, '$_
$_POST["hour"].') VALUES ('.$_SESSION['dcwr_id'].','.$_POST["date"].','.$_
$_POST["subject"].')');//Adding subjects
$conn->query('INSERT INTO `reports_topic_data` (`dcwr_id`, `date`, '$_
$_POST["hour"].') VALUES ('.$_SESSION['dcwr_id'].','.$_POST["date"].','.$_
$_POST["topic"].')');//Adding topics
}
$sql1=$conn->query(" SELECT * FROM `reasons` WHERE date = '$_
$_POST["date"].'" and (dcwr_id = '.$_SESSION['dcwr_id'].') and hour = '$_
$_POST["hour"].'");
if(!empty($row1=$sql1->fetch_assoc())){//check if record already exists
$conn->query('UPDATE `reasons` SET `event` = "'.$_POST["event"].'" WHERE
`reasons`.`dcwr_id` = '.$_SESSION['dcwr_id'].' AND (`reasons`.`date` = "'.$_
$_POST["date"].'" AND `reasons`.`hour` ='.$_POST["hour"].'));
$conn->query('UPDATE `reasons` SET `reason` = "'.$_POST["reason"].'" WHERE
`reasons`.`dcwr_id` = '.$_SESSION['dcwr_id'].' AND (`reasons`.`date` = "'.$_
$_POST["date"].'" AND `reasons`.`hour` ='.$_POST["hour"].'));
}
else{//create new record for REASON and EVENT if it does not exist.
$conn->query('INSERT INTO `reasons` (`dcwr_id`, `date`, `hour`, `event`, `reason`)
VALUES ('.$_SESSION['dcwr_id'].','.$_POST["date"].','.$_POST["hour"].','.$_
$_POST["event"].','.$_POST["reason"].')');
}
header('Location: dcwr_entry.php');
exit();

```

```

}

if(isset($_POST['submit']))
{
enterData();
}

?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>DCWR Entry</title>
<link rel="stylesheet" href="css/home.css">
</head>
<body>
<!-- / eDCWR -->
<div>
<table border='0' cellpadding='0' cellspacing='0'>
<tr class='time'>
<th></th>
<th>Name</th>
<th>Unique ID</th>
<th>Department</th>
<th>Section</th>
<th>Class Incharge</th>
<th>Academic Incharge</th>
<th></th>
</tr>
<tr>
<td class='days'>Profile</td>
<td class='info' data-tooltip='your name'><?php echo $name; ?></td>
<td class='info' data-tooltip='youruniqueID'><?php echo $id; ?></td>
<td class='info' data-tooltip='your department name'><?php echo $dept; ?></td>
<td class='info' data-tooltip='your section name'><?php echo $section; ?></td>
<td class='info' data-tooltip='your section name'><?php echo $reports; ?></td>
<td class='info' data-tooltip='your section name'><?php echo $plans; ?></td>
<td class="text red"><a href="logout.php">LOG OUT</a></td>
</tr>
</table>
</div>
<form role = formmethod="post" action="dcwr_entry.php">
<div class='tab'>
<table align="center" class='options' border='0' cellpadding='0' cellspacing='0'>
<tr class='time'>
<th>Date</th>

```

```

<th>Hour</th>
<th>Event</th>
<th>Cause</th>
<th>Subject</th>
<th>Topic</th>
<th>SUBMIT</th>
</tr>
<!-- Display rows for 8 hours. -->
<tr>
<tdclass='text'><inputname="date" class="select-style" type="date" style=" font-size: 1.3rem"></td>
<th>
<selectclass="select-style" name="hour">
<optionvalue="1">Hour 1</option>
<optionvalue="2">Hour 2</option>
<optionvalue="3">Hour 3</option>
<optionvalue="4">Hour 4</option>
<optionvalue="5">Hour 5</option>
<optionvalue="6">Hour 6</option>
<optionvalue="7">Hour 7</option>
<optionvalue="8">Hour 8</option>
</select>
</th>
<thclass='text'><!-- Event. -->
<selectclass="select-style" name="event">
<optionvalue="On Track">On track</option>
<optionvalue="Substituted">Subsิตuted</option>
<optionvalue="Delayed">Delayed</option>
<optionvalue="Cancelled">Cancelled</option>
</select>
</th>
<thclass='text'><inputtype="text" name="reason" placeholder="enter cause" class='textBox'></th>
<?php//Extract subject list from Subject list table
$sql=$conn->query(" SELECT * FROM `subject_list` WHERE plan_id = ".$SESSION['plan_id']." ");//GET ALL
if(!empty($row1=$sql->fetch_assoc())){
echo'<th class='text'>';
echo'<select class="select-style" name="subject">';
for($count=1;$count<10;$count++){//Go through all subjects
if($sub=$row1['SUBJECT'].$count.")!=NULL){//display only if its not NULL
echo'<option value="'.$sub.'">'.'($sub).'.'</option>';//$row1['SUBJECT'. $count . "]
}
}
}

```

```

echo'<option value="NULL">NULL</option>';//additional option to denote that no
classes were conducted
echo"</select>";
echo"</th>";
}
?>
<thclass='text'><inputtype="text" name="topic" placeholder="enter
topic" class='textBox' required="required"></th>
<td class='text green'><inputname="submit" value="SUBMIT" type="submit" class="button1
text green "></td>
</tr>
</table>
</div>
</form>
<div>
<tablealign="center" class='options' border='0' cellpadding='5' cellspacing='0'>
<trclass='time'>
<th>DCWR</th>
<th>LESSON PLAN</th>
<th>SCHEDULE</th>
</tr>
<tr>
<tdclass="text purple"><a href="dcwr.php">VIEW</a></td>
<tdclass="text orange"><a href="plan.php">VIEW</a></td>
<tdclass="text purple"><a href="schedule.php">VIEW</a></td>
</tr>
</table>
</div>

```

## 5.6 DCWR Page

```
<?php
include("connection.php");
include("logged_in.php");
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>DCWR</title>
<link rel="stylesheet" href="css/home.css">
</head>
<body>
<!-- / plan -->
<div class='tab'>
<table border='0' cellpadding='0' cellspacing='0'>
<tr class='time'>
<th></th>
<th>SUBJECT 1</th>
<th>SUBJECT 2</th>
<th>SUBJECT 3</th>
<th>SUBJECT 4</th>
<th>SUBJECT 5</th>
<th>SUBJECT 6</th>
<th>SUBJECT 7</th>
<th>SUBJECT 8</th>
</tr>
<?php //try to GET date from URL, else do this. SUBMIT button will append date to url and
redirect
if(isset($_GET['date'])){
$startDate=new DateTime($_GET['date']);//WORKS.
}
else{
$startDate=new DateTime('-7 days');//begins at -5 days
}
$count=0;
$ErrorCount=0;
while($count<6){
$dateString=$startDate->format('Y-m-d');
$sql1=$conn->query(" SELECT * FROM `reports_topic_data` WHERE date = '".$date-
String."' and dcwr_id = ".$SESSION['dcwr_id']. " ");
$sql2=$conn->query(" SELECT * FROM `reports_subject_data` WHERE date = '".$date-
String."' and dcwr_id = ".$SESSION['dcwr_id']. " ");
if(!empty($row1=$sql1->fetch_assoc())&&!empty($row2=$sql2->fetch_assoc())){
echo "<tr>";
echo "<td class='days'>".$startDate->format('Y-m-d')."</td>";
```

```

for($i=1;$i<9;$i++){
    $j="$i";
    $j.= "H";
    $sql3=$conn->query(" SELECT * FROM `reasons` WHERE date = '".$dateString."' and
    (dcwr_id = ".$SESSION['dcwr_id']."' and hour =".$i.") ");
    if(!empty($row3=$sql3->fetch_assoc())){
        switch($row3['event']){
            case"On Track":
                echo"<td class='info' data-tooltip='".$row2[$j]."'>".$row1[$j]."</td>";
                break;
            case"Substituted":
                echo"<td class='info yellow' data-tooltip='".$row2[$j]."'>".$row1[$j]."</td>";
                break;
            case"Delayed":
                echo"<td class='info orange' data-tooltip='".$row2[$j]."'>".$row1[$j]."</td>";
                break;
            case"Cancelled":
                echo"<td class='info red' data-tooltip='".$row2[$j]."'>".$row1[$j]."</td>";
                break;
            default//useless but satefy feature.
                echo"<td class='info' data-tooltip='".$row2[$j]."'>".$row1[$j]."</td>";
        }
    }
    else//in case reason does not exist
        echo"<td class='info' data-tooltip='".$row2[$j]."'>".$row1[$j]."</td>";
    }
}
echo"</tr>";
}

elseif($ErrorCount<2){//Tries to fetch twice. Enters "IF" only if the number of failures is less
that 2 (0 and 1)
$startDate=$startDate->add(newDateInterval('P1D'));
$ErrorCount++;
continue;
}
else{
    echo"<script type='text/javascript'>alert('Unable to fetch complete report. (Holidays/Missing
data')</script>";
    break;
}
$count++;
$startDate=$startDate->add(newDateInterval('P1D'));
}
?>
</table>
</div>
<form role = formmethod="get"action="dcwr.php">

```

```
<div>
<table class='options' border='0' cellpadding='5' cellspacing='0'>
<tr class='time'>
<th>Initial Date</th>
<th>Plan</th>
<th>e-DCWR</th>
</tr>
<tr>
<td class='text'><input name="date" type="date" style=" font-size: 1.3rem;"></td>
<td class="text green"><input type="submit" value="UPDATE" class="button2 text green" name="update"></td>
<td class="text red"><a href="javascript:history.go(-1)">Go back</a></td><!-- Goes back to actual previous page -->
</tr>
</table>
</div>
</form>
</body>
```

## 5.7 Schedule Page

```
<?php
include("connection.php");
include("logged_in.php");
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Schedule</title>
<link rel="stylesheet" href="css/home.css">
</head>
<body>
<!-- / plan -->
<div class='tab'>
<table border='0' cellpadding='0' cellspacing='0'>
<tr class='time'>
<th></th>
<th>SUBJECT 1</th>
<th>SUBJECT 2</th>
<th>SUBJECT 3</th>
<th>SUBJECT 4</th>
<th>SUBJECT 5</th>
<th>SUBJECT 6</th>
<th>SUBJECT 7</th>
<th>SUBJECT 8</th>
</tr>
<?php
$count=1;
while($count<7){
$sql=$conn->query(" SELECT * FROM `schedule_data` WHERE day = '".$count."' and
dcwr_id = ".$_SESSION['plan_id']." ");
if(!empty($row1=$sql->fetch_assoc())){
echo "<tr>";
echo "<td class='days'>".$count."</td>";
echo "<td>".($row1['1H'])."</td>";
echo "<td>".($row1['2H'])."</td>";
echo "<td>".($row1['3H'])."</td>";
echo "<td>".($row1['4H'])."</td>";
echo "<td>".($row1['5H'])."</td>";
echo "<td>".($row1['6H'])."</td>";
echo "<td>".($row1['7H'])."</td>";
echo "<td>".($row1['8H'])."</td>";
echo "</tr>";
}
}
```

```

else{
echo"<script type='text/javascript'>alert('Does not exist in the database.')</script>";
break;
}
$count++;

}
?>
</table>
</div>
<div>
<tableclass='options'border='0'cellpadding='5'cellspacing='0'>
<trclass='time'>
<th>e-DCWR</th>
</tr>
<tr>
<tdclass="text green"><a href="javascript:history.go(-1)">Go back</a></td><!-- should
go back to actual previous page -->
</tr>
</table>
</div>
</body>
</html>

```

## 5.8 Lesson-plan Page

```
<?php
include("connection.php");
include("logged_in.php");
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Lesson Plan</title>
<link rel="stylesheet" href="css/home.css">
</head>
<body>
<!-- / plan -->
<div class='tab'>
<table border='0' cellpadding='0' cellspacing='0'>
<tr class='time'>
<th></th>
<th>SUBJECT 1</th>
<th>SUBJECT 2</th>
<th>SUBJECT 3</th>
<th>SUBJECT 4</th>
<th>SUBJECT 5</th>
<th>SUBJECT 6</th>
<th>SUBJECT 7</th>
<th>SUBJECT 8</th>
</tr>
<?php //try to GET date from URL, else do this. SUBMIT button will append date to url and
redirect
if(isset($_GET['date'])){
$startDate=new DateTime($_GET['date']); //WORKS.
}
else{
$startDate=new DateTime('-7 days'); //begins at -5 days
}
$count=0;
$ErrorCount=0;
while($count<6){
$dateString=$startDate->format('Y-m-d'); //Dont forget to change reports to plans.
$sql1=$conn->query(" SELECT * FROM `plans_topic_data` WHERE date = '".$dateString."' "
and plan_id = ".$_SESSION['plan_id']."' ");
$sql2=$conn->query(" SELECT * FROM `plans_subject_data` WHERE date = '".$date-
String."' and plan_id = ".$_SESSION['plan_id']."' ");
if(!empty($row1=$sql1->fetch_assoc())&&!empty($row2=$sql2->fetch_assoc())){
echo "<tr>";
echo "<td class='days'>".$startDate->format('Y-m-d')."</td>"; //add tooltips
```

```

echo "<td class='info' data-tooltip='".($row2['subject1'])." '>".($row1['subject1'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject2'])." '>".($row1['subject2'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject3'])." '>".($row1['subject3'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject4'])." '>".($row1['subject4'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject5'])." '>".($row1['subject5'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject6'])." '>".($row1['subject6'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject7'])." '>".($row1['subject7'])."</td>";
echo "<td class='info' data-tooltip='".($row2['subject8'])." '>".($row1['subject8'])."</td>";
echo "</tr>";
}
elseif($ErrorCount<1){
$startDate=$startDate->add(new DateInterval('P1D'));
$ErrorCount++;
continue;
}
else{
echo "<script type='text/javascript'>alert('Some dates are unavailable.(Holidays/Missing data')</script>";
break;
}
$count++;
$startDate=$startDate->add(new DateInterval('P1D'));
}
?>
</table>
</div>
<form role = formmethod="get"action="plan.php">
<div>
<tableclass='options'border='0'cellpadding='5'cellspacing='0'>
<trclass='time'>
<th>Initial Date</th>
<th>Plan</th>
<th>e-DCWR</th>
</tr>
<tr>
<tdclass='text'><inputname="date" type="date" style=" font-size: 1.3rem;"></td>
<tdclass="text green"><inputtype="submit" class="button2 text green " name="UPDATE"></td>
<tdclass="text red"><a href="javascript:history.go(-1)">Go back</a></td><!-- Goes back to actual previous page -->
</tr>
</table>
</div>
</form>
</body>
</html>

```

# **Chapter 6**

# **TESTING TOOLS**

## 6. TESTING TOOLS

### 6.1 Introduction

Although testing can determine the correctness of software under the assumption of some specific hypotheses (see hierarchy of testing difficulty below, testing cannot identify all the defects within the software. Instead, it furnishes a *criticism* or *comparison* that compares the state and behaviour of the product against oracles—principles or mechanisms by which someone might recognise a problem. These oracles may include (but are not limited to) specifications, contracts, comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria.

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions, but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of the code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organisation may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed

Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organisation develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers and other stakeholders. Software testing is the process of attempting to make this assessment.

## **6.2 Types of Testing**

### **6.2.1 White box testing**

White box testing is a testing case design method that uses the control structure of the procedural design to derive test cases. All independent paths in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

### **6.2.2 Black Box Testing**

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialisation and termination error. Here all the input data must match the data type to become a valid entry. The following are the different tests at various levels:

- Unit Testing**

Unit testing is essential for the verification of the code produced during the coding phase and the goal is to test the internal logic of the module/program. In the Generic code project, the unit testing is done during the coding phase of data entry forms, whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

- Integration Testing**

All the tested modules are combined into subsystems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

- Validation Testing**

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subject to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specifications; they are listed out and are corrected.

- System Testing**

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves: Implementing the system in a simulated production environment and testing it. Introducing errors and testing for error handling.

## 6.3 Test Cases

### Test Case 1 - Incorrect Login

Whenever an incorrect id or an incorrect password is entered, the login page displays a message saying “Incorrect Username and/or password”.

### Test Case 2 - Unauthorised User

If the user, who is trying to log in, is an ex-CR or In-charge i.e., belonging to the previous semester and not now, they will be denied access and a message displaying “Authorisation Failed. You are not allowed to access the current DCWR”.

### Test Case 3–Data Unavailable

If the CR has not entered daily class work data into the database (especially during holidays), a message will be displayed saying “Unable to fetch complete report(Holidays/Missing data).”

Test Case	Expectation	Actual Result	Report P/F
Incorrect Login	Display Message “In- correct Username and/ or Password”	Valid Message Dis- played	Pass (Working Properly)
Unauthorized User	Display Message “Au- thorization Failed.You are not allowed to ac- cess the current DCWR”	Valid Message Dis- played	Pass (Working Properly)
Data Unavailable	Display Message “Un- able to fetch complete report(Holidays/Miss- ing data).”	Valid Message Dis- played	Pass (Working Properly)

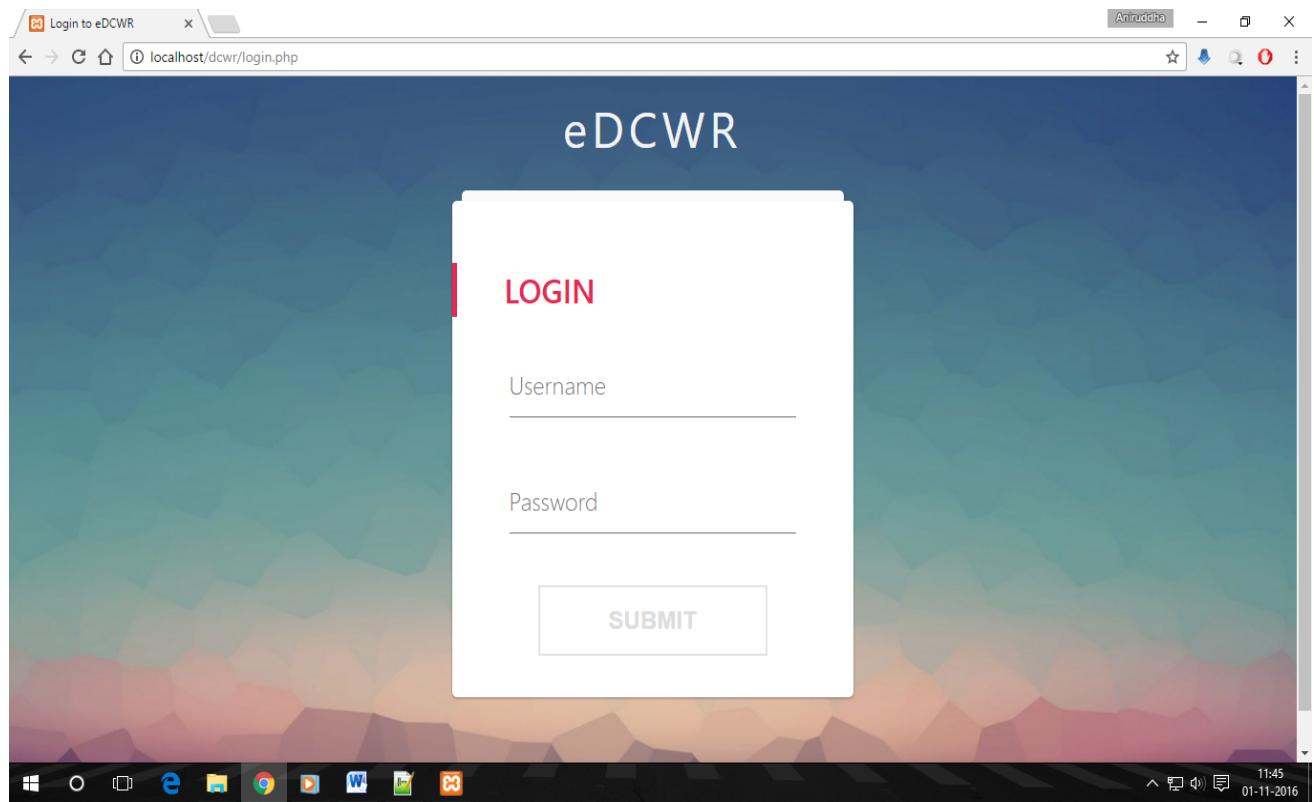
**Table 6.3 : Test Cases for Modules**

# **Chapter 7**

# **OUTPUT SCREENS**

## 7. OUTPUT SCREENS

### 7.1 Login Page



**Figure 7.1: Login Page**

## 7.2 Admin Page

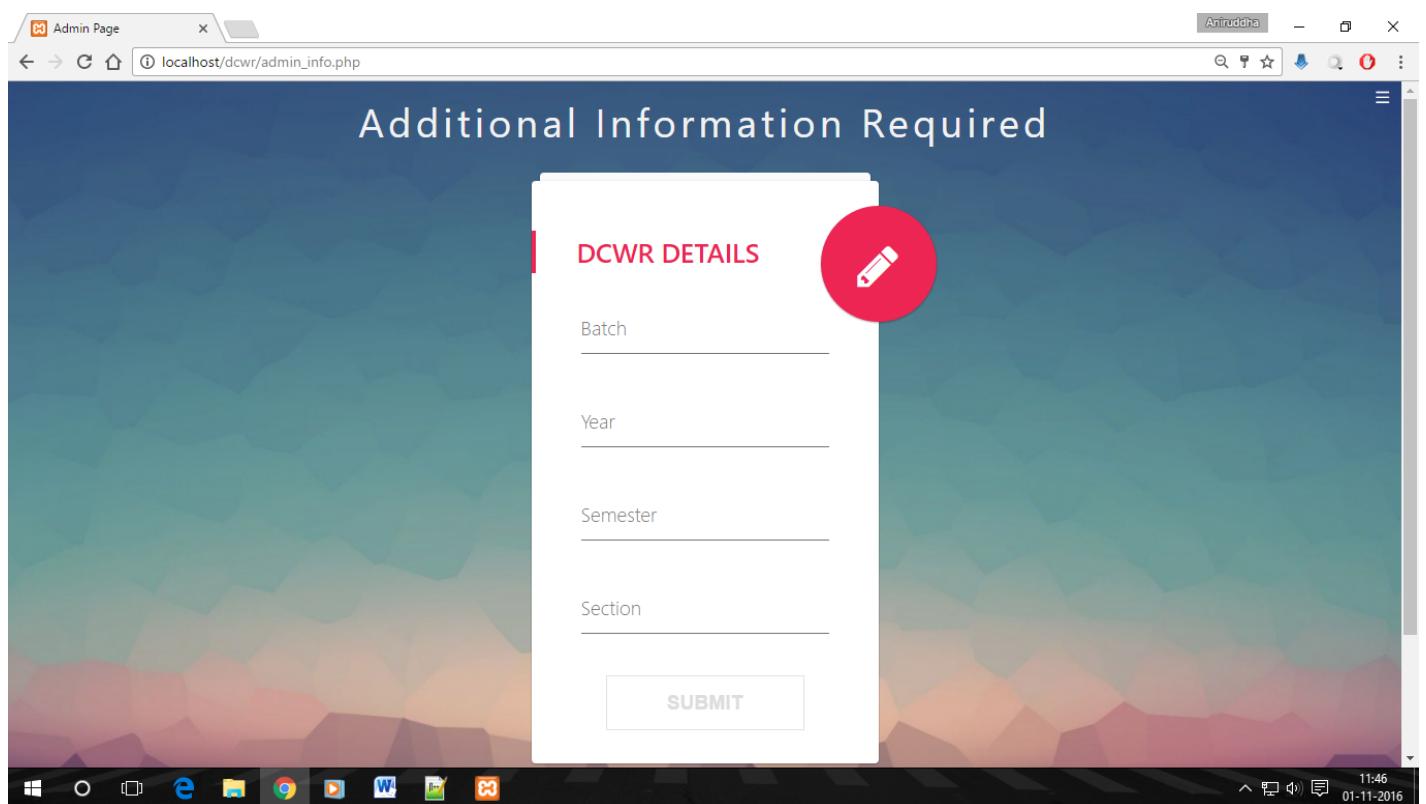


Figure 7.2.1: Admin Page to access current records

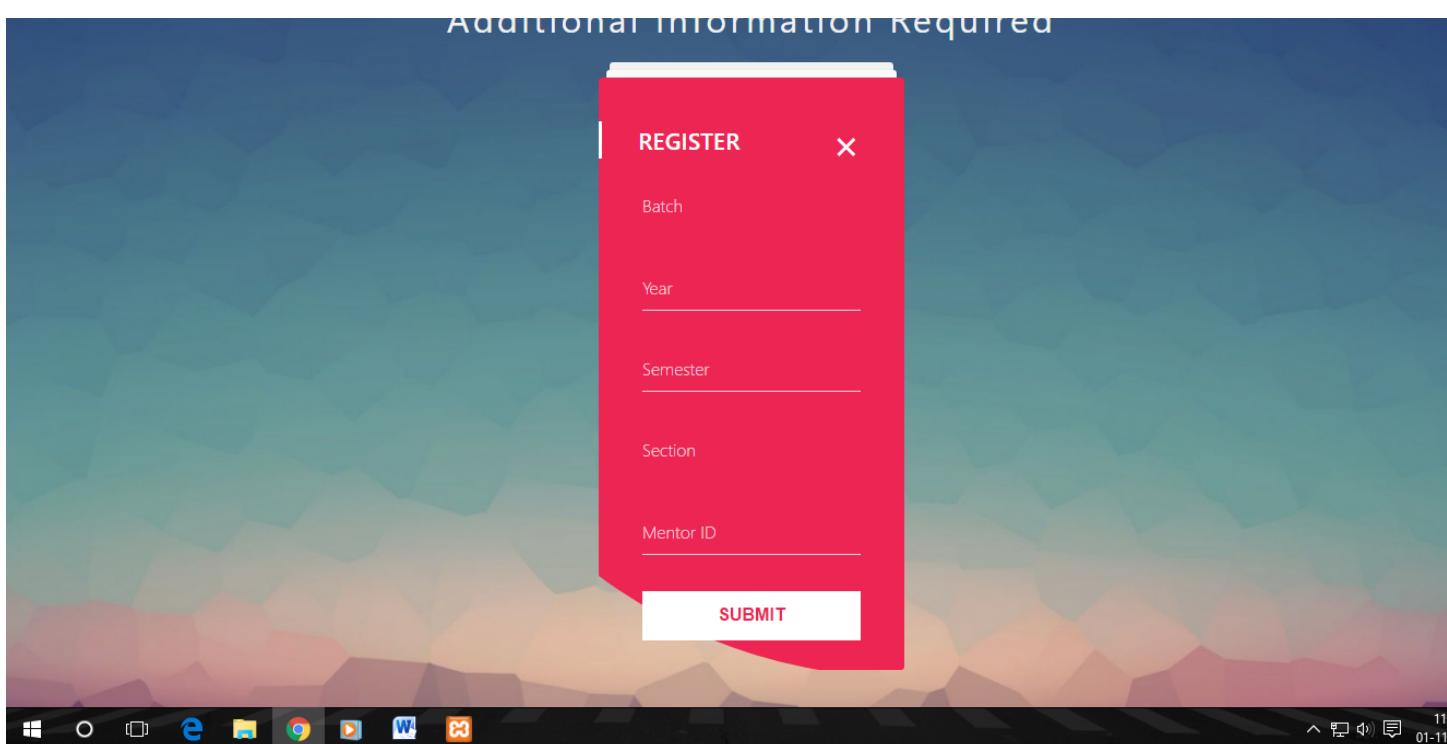


Figure 7.2.2: Admin Page to create new records

### 7.3 System Setup page

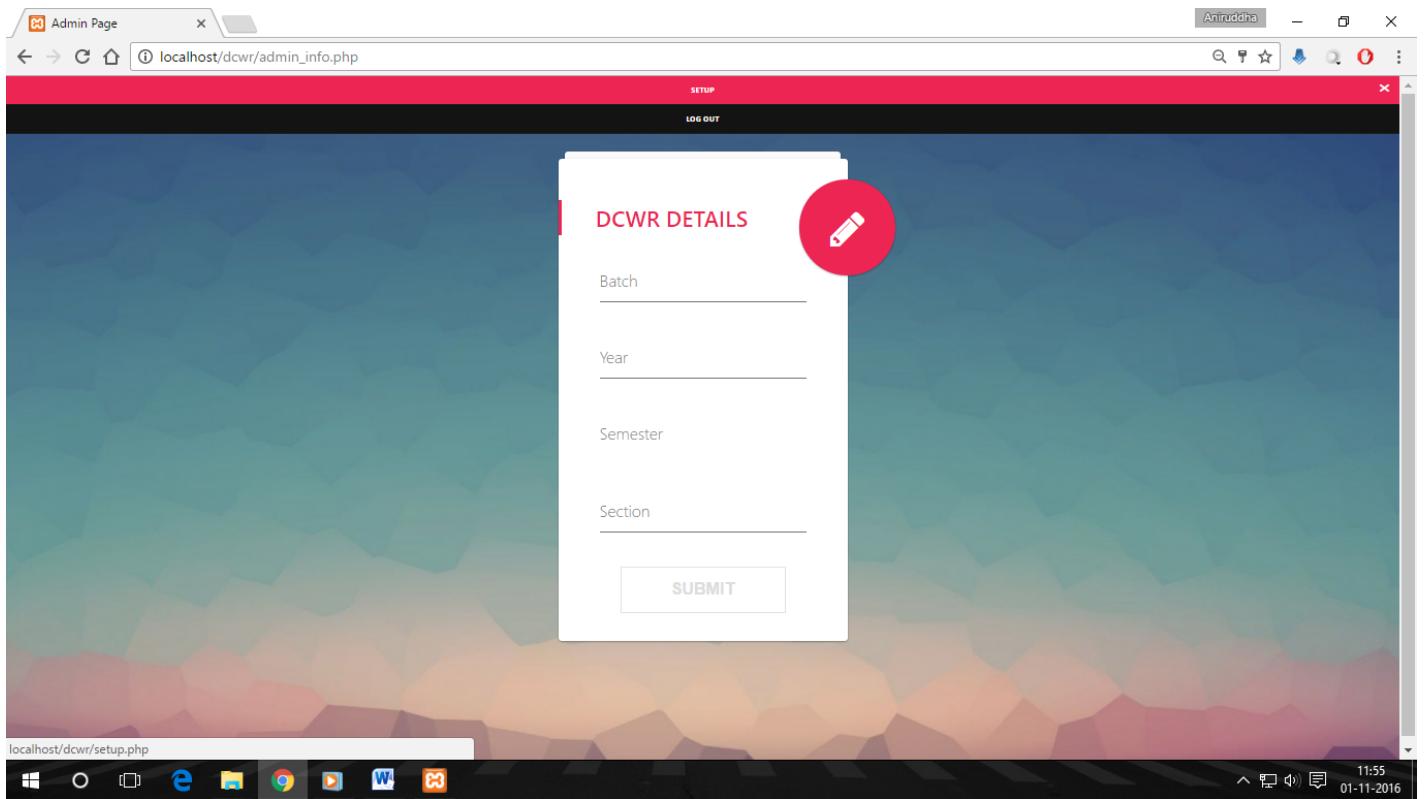


Figure 7.3.1: Admin Page displaying setup options

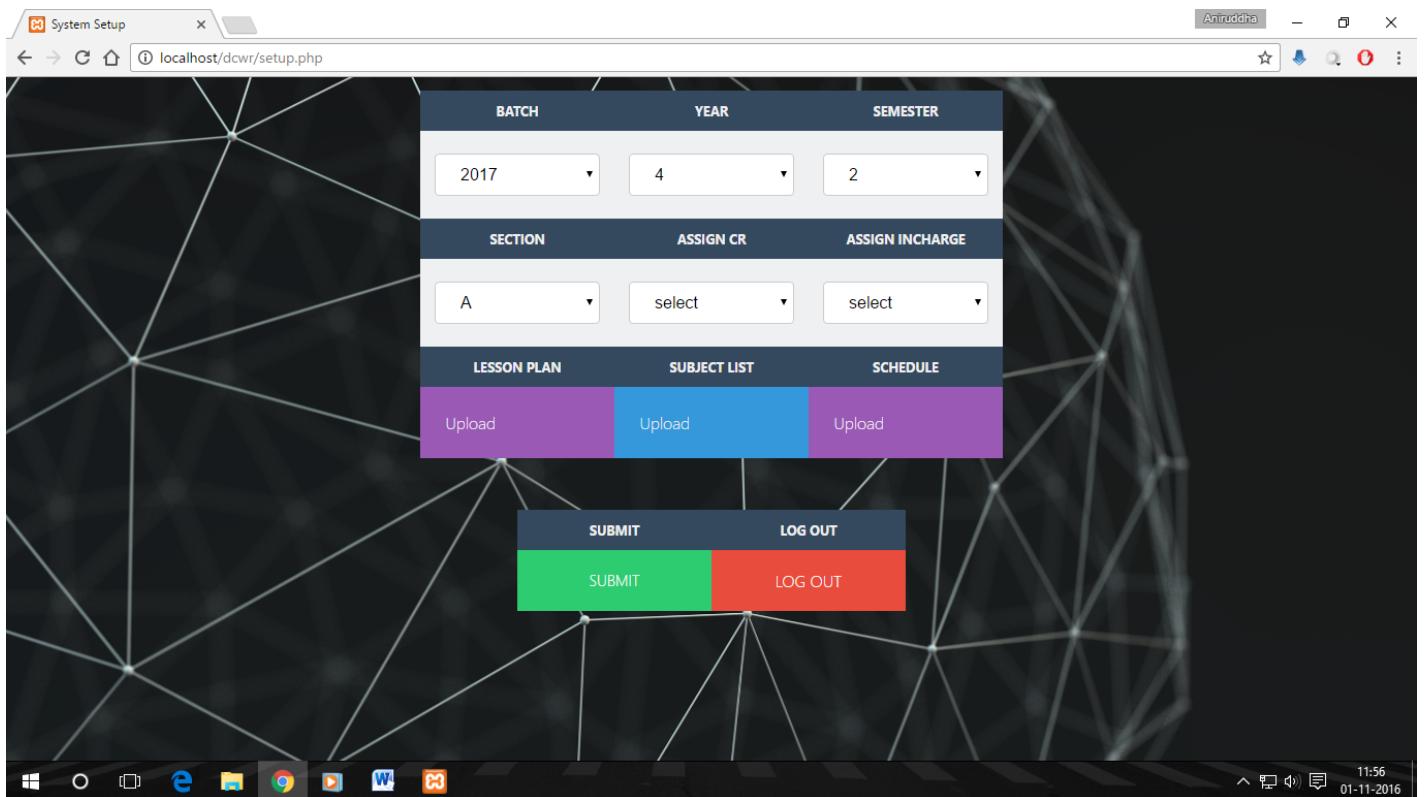


Figure 7.3.2: Page to upload relevant data

## 7.4 DCWR Entry Page

The screenshot shows the DCWR-Entry page. At the top, there is a header with fields for NAME (Chetan), UNIQUE ID (3), DEPARTMENT (CSE), SECTION (A), CLASS INCHARGE (bala), and ACADEMIC INCHARGE (aniruddha). Below the header is a red "LOG OUT" button. The main area contains a form with fields for DATE (01-11-2016), HOUR (Hour 1), EVENT (On track), CAUSE (enter cause), SUBJECT (DWDM), and TOPIC (Introduction). There is also a green "SUBMIT" button. To the left of the form is a calendar for November 2016, showing the days from 31 to 4. The background features a network-like geometric pattern.

Figure 7.4: DCWR-Entry page

## 7.5 DCWR Page for CR

The screenshot shows the DCWR-Display for CR page. At the top, there is a header with a date field (2016-11-01) and buttons for DWDM, SUBJECT 2 (highlighted in yellow), SUBJECT 3, SUBJECT 4, SUBJECT 5, SUBJECT 6, SUBJECT 7, and SUBJECT 8. Below the header is a timeline with colored boxes for each subject: Introduction (white), Facade Patterns (yellow), File commands (orange), Seminar (red), LUNCH (light blue), topicX (purple), topicY (teal), and DWDM (light blue). The timeline is set to the date 2016-11-01. Below the timeline is a control panel with buttons for INITIAL DATE (dd-mm-yyyy), PLAN (UPDATE), and E-DCWR (GO BACK). The background features a network-like geometric pattern.

Figure 7.5: DCWR-Display for CR page

## 7.6 Lesson Plan Page

	SUBJECT 1	SUBJECT 2	SUBJECT 3	SUBJECT 4	SUBJECT 5	SUBJECT 6	SUBJECT 7	SUBJECT 8
2016-08-29	Cloud Computing	DWDM	DWDM	Big Data	lunch	Cloud Computing	Design Patterns	club Activity
2016-08-30	DWDM	Information Retrieval Systems	Cloud Computing	Design Patterns	lunch	Big Data	Cloud Computing	club Activity
2016-08-31	DWDM LAB	DWDM LAB	DWDM LAB	Linux Programming	lunch	Cloud Computing	DWDM	club Activity
2016-09-01	Information Retrieval Systems	Cloud Computing	Design Patterns	Design Patterns	lunch	Big Data	Information Retrieval Systems	club Activity
2016-09-02	Big Data	Design Patterns	Cloud Computing	Cloud Computing	lunch	Design Patterns	dbms	club Activity
2016-09-03	Design Patterns	Big Data	Information Retrieval Systems	Cloud Computing	lunch	Linux Programming	Linux Programming	club Activity

Figure 7.6: Common Lesson Plan page

## 7.7 Schedule Page

DAY	SUBJECT 1	SUBJECT 2	SUBJECT 3	SUBJECT 4	SUBJECT 5	SUBJECT 6	SUBJECT 7	SUBJECT 8
1	DWDM	LP	CC	IRS	Lunch	DP	Big Data	LP
2	LP	CC	IRS	DWDM	Lunch	Big Data	LP	DWDM
3	DWDM	LP	CC	IRS	Lunch	DP	Big Data	LP
4	LP	CC	IRS	DWDM	Lunch	Big Data	LP	DWDM
5	LAB	LAB	LAB	IRS	Lunch	DP	CC	Big Data

Figure 7.6: Common Schedule Plan page

## 7.8 DCWR - Home Page

The screenshot shows the DCWR Home Page for a faculty member named bala. The page includes a header with profile information (Name: bala, Unique ID: 2, Department: CSE, Section: A, Class Incharge: bala, Academic Incharge: aniruddha) and a 'LOG OUT' button. Below this is a table showing the class schedule for the week starting 2016-08-29:

	NAME	UNIQUE ID	DEPARTMENT	SECTION	CLASS INCHARGE	ACADEMIC INCHARGE		
PROFILE	bala	2	CSE	A	bala	aniruddha	<b>LOG OUT</b>	
	1ST HOUR	2ND HOUR	3RD HOUR	4TH HOUR	5TH HOUR	6TH HOUR	7TH HOUR	8TH HOUR
2016-08-29	Cloud Computing	Cloud Computing	Information Retrieval Systems	Design Patterns	lunch	Linux Programming	Big Data	club Activity
2016-08-30	Cloud Computing	DWDM	DWDM	Big Data	lunch	Cloud Computing	Design Patterns	club Activity
2016-08-31	DWDM	Information Retrieval Systems	<b>Cloud Computing SUBJECT NAME</b>	Design Patterns	lunch	Big Data	Cloud Computing	club Activity
2016-09-01	First	<b>Second</b>	TOPIC NAME	fourth				
2016-09-02	Design Patterns	ELECTIVE2	ELECTIVE2	Design Patterns	lunch	Big Data	Information Retrieval Systems	club Activity
2016-09-03	August, 2016 ▾		Third					

Below the schedule is a calendar for August 2016. At the bottom of the page are buttons for 'DCWR', 'LESSON PLAN', 'SCHEDULE', 'APPROVE', and 'REJECT'. The status bar at the bottom right shows the date as 01-11-2016 and the time as 12:24.

Figure 7.6: DCWR for faculty page

### 7.8.1 “REJECT” Option

The screenshot shows a Google Mail message with the subject 'Inconsistencies in DCWR'. The message body contains the following text:

```

Inconsistencies in DCWR
someone@example.com
Inconsistencies in DCWR
BODY

```

At the bottom of the message, there is a rich text editor toolbar and a status bar indicating 'Waiting for plus.google.com...' and the date/time '01-11-2016 12:17'.

Figure 7.6: Page to notify about errors in DCWR

# **Chapter 8**

# **CONCLUSION**

## 8. CONCLUSION

The major problems educational institutions face with the usage of the traditional DCWR system are, seemingly, fixed in the proposed solution of using an online system called *e-Daily Class Work Report*. It successfully manages to save costs, time and natural resources, provide better data-managing abilities.

However, similar to the traditional model, the proposed system also requires an administrator, a Class In-Charge and a Class Representative, meaning that the load on each person is reduced, but it still requires all of them to function correctly. There are a few such possible and achievable extensions which can be made, and they will be added in the next versions, hopefully.

### 8.1 Future Enhancements

e-DCWR, currently, works as intended and achieves the specified goal effectively. But, there are possible improvement which can be made in the future builds.

- **Analysis module:**

Adding such additional module can analyse all past data relating to the inconsistencies and various reasons behind them and make possible predictions. The analysis can also be used to visualise and draw graphs, which will aid In-charges in making appropriate decisions during the semester.

- **Reducing data usage:**

The CSS and scripting required by the web pages could be reduced, so that older computers can run it as well, subsequently increasing the range of hardware that can run it.

- **Single Website for all:**

The current system requires different universities/colleges to maintain their own individual e-DCWR website. A single website can be used instead, where all colleges can log in and store their details at one place; this will aid in saving storage space and also in fixing bugs quickly, as more support will be available to one project.

## **Chapter 9**

# **BIBLIOGRAPHY**

## 9. BIBLIOGRAPHY

### 9.1 Book References

1. Scott Guelich, Shishir Gundavaram, and Gunther Birznieks. CGI Programming with Perl. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1996, pp. 265.
2. Randy Jay Yarger, George Reese, and Tim King. MySQL and mSQL. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1999, pp. 7.
3. Randy Jay Yarger, George Reese, and Tim King. MySQL and mSQL. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1999, pp. 154-155.
4. Scott Guelich, Shishir Gundavaram, and Gunther Birznieks. CGI Programming with Perl. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1996, pp. 84.
5. Randal L. Schwartz and Tom Christiansen. Learning Perl, 2<sup>nd</sup> Edition. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1997, pp. 76.
6. Randy Jay Yarger, George Reese, and Tim King. MySQL and mSQL. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1999, pp. 51.
7. "Web Programming" by *Chris Bates*.
8. "Database Management Systems" by *Ramakrishnan and Gehrke*.
9. "Software Engineering" by *Roger S. Pressman*.

### 9.2 Websites

1. <http://php.net/manual/en/>
2. [www.w3schools.com](http://www.w3schools.com)
3. [www.tutorialspoint.com](http://www.tutorialspoint.com)
4. [www.stackoverflow.com](http://www.stackoverflow.com)
5. [www.quora.com](http://www.quora.com)
6. <http://www.perl.com/pub/a/2002/02/20/css.html>