

Chapter 1**INTRODUCTION****1.1 Purpose:**

This module aims at being a tool for the course outcomes in each of the subjects that the students of RVCE are registered for, thereby aiding the faculty of the organization. This helps in easy consolidation of course outcomes of each of the students of all semesters in all departments in the subjects that they are registered for. The module then generates dynamic charts, reports and graphs by processing this data. Faculty can view all this data at their convenience as this data is stored permanently by the module. For the ease of use, the faculty needn't enter the data of all the students at once. The module supports the entry in a staggered format so that the faculty can enter this data at their convenience. Also the integrity of data is of utmost importance. Strong validation is done so that incorrect data is not entered by the user.

1.2 Scope:

This module is of utmost importance to educational institutions as for the real education of the students, it is essential to frequently tweak the learning process based on continuous feedback. This Course Outcome module is a major step in this direction as based on the performance of the students in tests, we can evaluate their strengths and weaknesses. More emphasis is paid to these weak spots during the learning process where as more challenging questions and assignments must be provided in the areas that the students excel. It also reduces the load on the faculty as this process would otherwise have to be done manually by each of the faculty.

1.3 Motivation:

There are several reasons why we choose to select this project.

- First and foremost since we have this subject as part of a curriculum, taking up a project will help us gain a better understanding about the subject theoretically as well as practically.
- In the digital age, the demand for automation is incredibly high. No one wants to do work that can be automated by a computer. At the same time exceptionally high quality results are expected from these systems. Thus the need for a state-of-the-art software such as the Course Outcome Mapping System is incredibly high in educational institutions that are

competing to be among the best in the nation. For the faculty to manually perform the actions that this software does would require incredibly high amount of time to be invested by them. Not to mention this would make the process very error prone. This software provides all these functions in a very user friendly environment.

1.4 Literature Survey:

A Course Outcome Mapping System is a software that is designed to manage the course outcome requirements of all students in all the subjects that they have registered for in that semester. The Course Outcome Mapping System module is a platform designed to aid the faculty of R V College of Engineering, Bangalore. It allows the faculty to first create the template of the question paper for the test that is being evaluated, i.e. the number of questions, the course outcome that it evaluates and the marks that are allotted to it. The faculty can then enter the marks that each student has scored for those questions. This module will then process this data. Detailed reports, dynamic charts and graphs are then generated based on this. Consolidated reports for each student, test, semester and subject may also be generated as required by the faculty. This data is permanently stored by the system and then accessible for viewing even at a later date. Faculty also have the option of downloading these graphs and charts for use elsewhere. These features provide a clear and succinct summary of the performance of all the students. This gives the faculty adequate feedback allowing them to modify their teaching to accommodate the particular needs of each of the students, thereby enriching the entire learning ecosystem. Hence, the Course Outcome Mapping System.

Chapter 2**SOFTWARE REQUIREMENT SPECIFICATIONS****2.1 Overall Description**

This system has been designed to aid the faculty in viewing course outcomes of each of their students in the subjects that are taught by them. The graphs and charts that are dynamically generated by the system means that the faculty have succinct summaries at their fingertips. Also, it provides consolidated data for each student, semester, department or subject thereby providing both a broad overview and detailed summaries based on what is required. The system ensures enforces strict validation on the data. It provides these features in a safe and user friendly environment.

2.2 Specific Requirements

The requirements of the Course Outcome Mapping System are to develop:

- A web based portal to upload template of the question papers
- An interface to authenticate each member of faculty
- A method to provide and store the list of students that are eligible to write a particular test
- A web interface to enter the score of each student in each test of all subjects that they have registered for
- Dynamic graphs and charts summarizing the course outcome attainment
- Consolidated reports for each student, subject, semester and department.

2.2.1 Functionality

The main user group and access level is:

1. Subject faculty

2.2.1.1 Functional Requirement –Faculty

- Web interface to enter template of the question paper of each test
- An interface to authenticate that member of faculty
- A web interface to enter the score of each student in each test of all subjects that they have registered for
- Dynamic graphs and charts summarizing the data
- Consolidated reports for each student, subject, semester and department

2.2.1.2 Security Requirements

It is of utmost importance to ensure that there is protection against unauthorized access to the course outcome details of the students. Faculty must be provided with a login ID and password which grant access only to the subjects that being taught by them. Similarly, the HOD is provided a privileged login which enables them to view and manage course outcome data that has been entered by any member of their department.

2.3 Non-functional Requirements

The PCs used must be at least Pentium 4 machines so that they can give optimum performance of the product. In addition to these requirements, the system should also embrace the following requirements:-

Reliability: The system should have little or no downtime and be able to handle multiple concurrent users.

Ease of Use: The general and administrative views should be easy to use and intuitive. Online help and documentation should be provided.

System and Browser Compatibility Testing: The system should be accessible on the following browsers - Microsoft Internet Explorer 7.0+, Netscape Navigator 6.0+ and Mozilla 1.3+, Google Chrome.

2.3.1 Design Constraints

The designers must design the database in such a way that any change in the information of a client should be updated and saved effectively in the database despite the fact that multiple clients access the database.

2.3.2 Operating environment

The required specifications/requirements needed for the running of the above developed application consists of the operating environment factors.

2.3.3 Hardware Requirement

1. Operating System: Windows XP ,Windows Vista, Windows 7, Windows 8
2. Processor: Pentium 3.0 or higher
3. RAM: 256 Mb or more
4. Hard Drive: 10 GB or more

2.3.4. Software Requirement

1. Front End : HTML 5.0, CSS3, Bootstrap
2. Server side scripting: PHP and JavaScript (using PHP editor)
3. Back End : LAMP
4. Operating System : Ubuntu 12.04

2.3.5 User Interface

The interface design must be user friendly and intuitive.

1. One faculty cannot edit the course outcome information of another.
2. A faculty can manipulate only his/her respective course outcome information.

Chapter 3**PROJECT DESIGN****3.1 Design and Implementation Constraints**

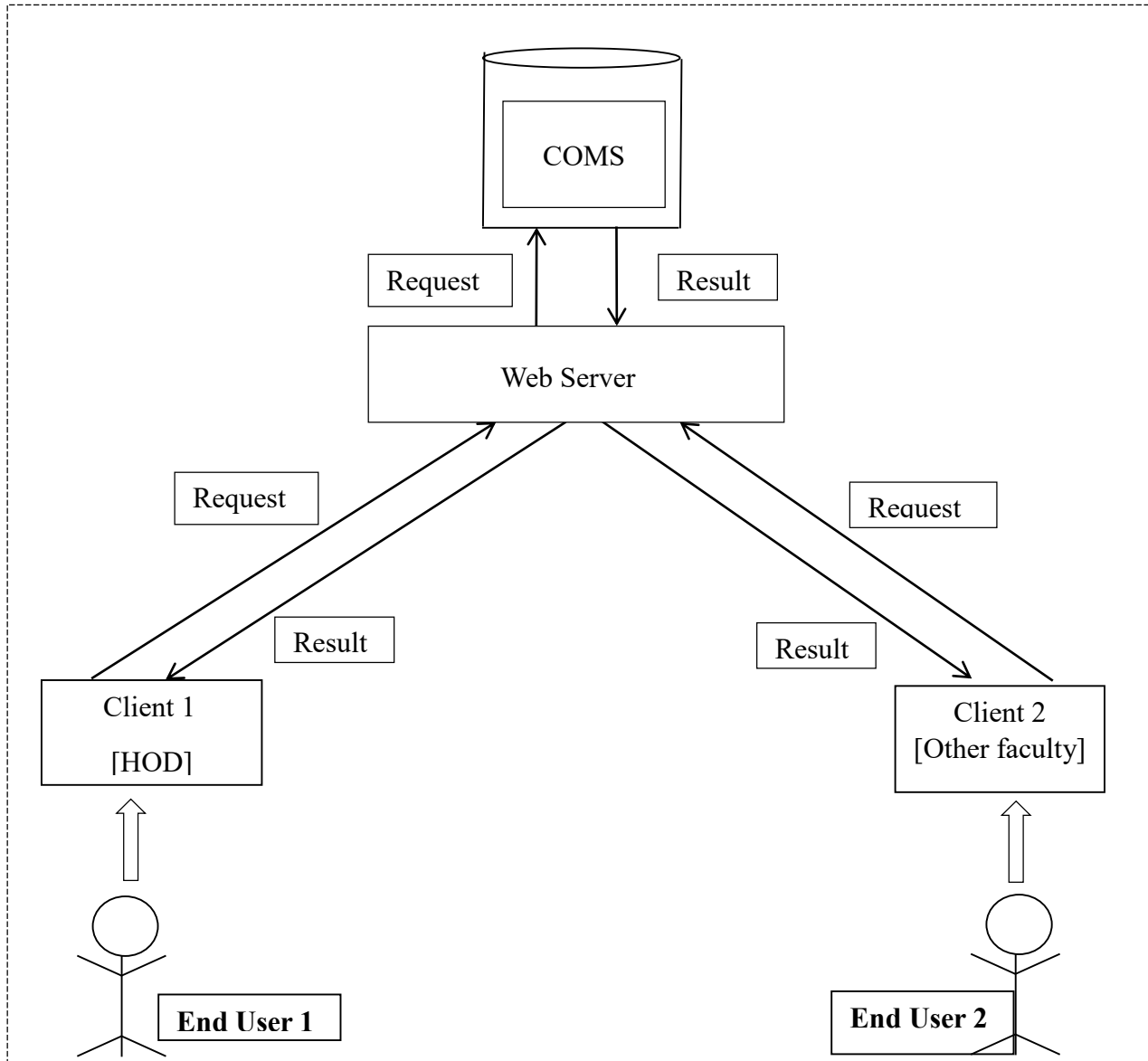
- Client side program is written using HTML and Javascript whereas server side programs are written in only PHP.
- The presentation layer/web site is designed to work efficiently on the particular browser.
- The Apache web server being used is not connected to the external network.

3.2 Assumptions and dependencies

- It is assumed that the test format complies with the standard test format.
- There are a fixed number of course objectives
- The test consists of a quiz for 15 marks and a theory section for 50 marks.
- Every question in the test satisfies exactly one course objective.

3.3 System Architecture

Figure 3.3 System Block Diagram



3.4 Data Flow Diagrams

3.4.1 Level 0 Diagram

Diagram: Data Flow Diagram

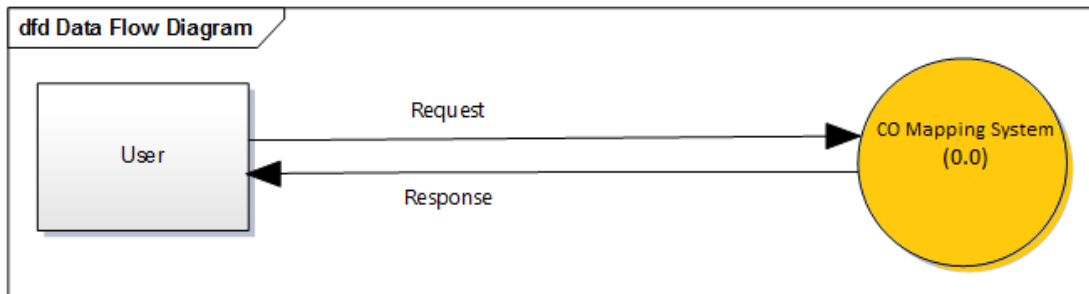


Figure 3.4.1 Dataflow Diagram Level 0

- The user sends a request which is to be authenticated, which if allowed for that type of user and if that kind of request exist is allowed and converted into a query.
- If that requests is not authenticated a response to the GUI or user end is generated for example, validations on fields.
- Query hence is executed on the server side and a output is generated and displayed to the user.

3.4.2 Level 1 Diagram

Diagram: Data Flow Model

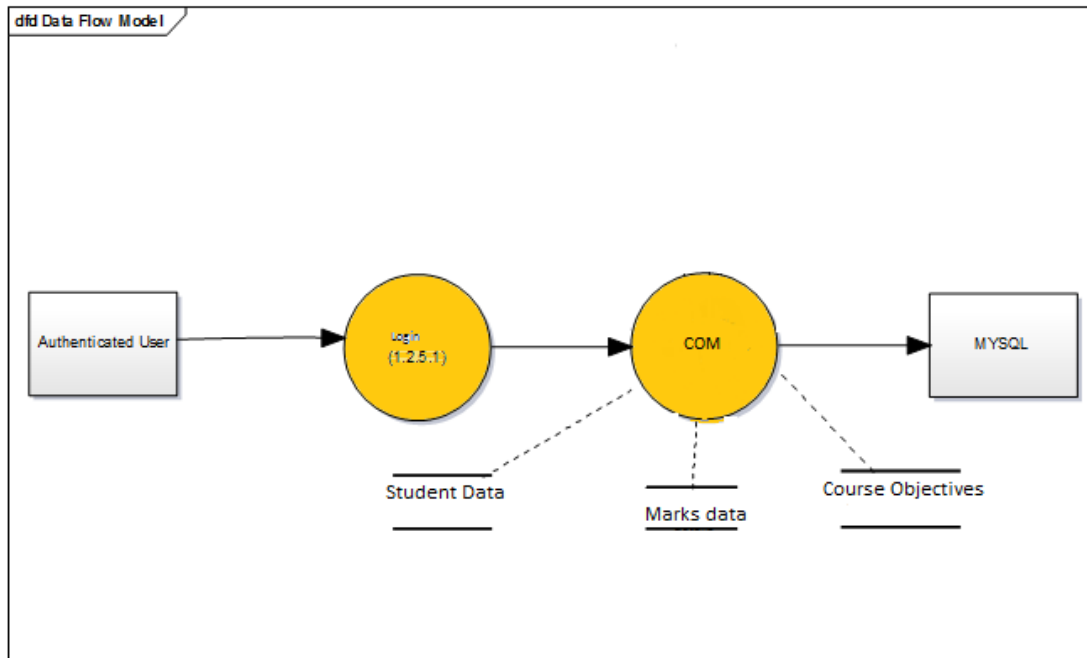


Figure 3.4.2 Dataflow Diagram Level 1

[Login details] - It contains username and password for login (faculty).

[Enrollment] - Contains enrollment details of the students.

[Student Data] - Contains student basic information.

[Marks Data] - Contains details about the marks of each student and each test.

[Course Objectives] - Contains the course outcomes of each course.

3.4.3 Level 2 Diagram

Diagram: Data Flow Model

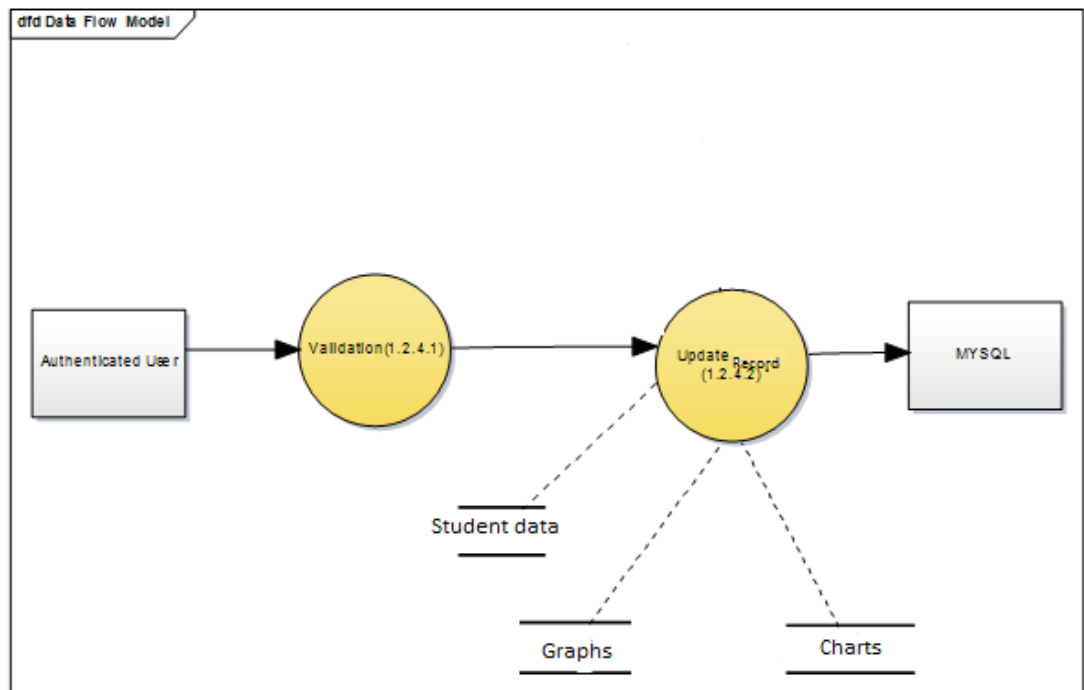


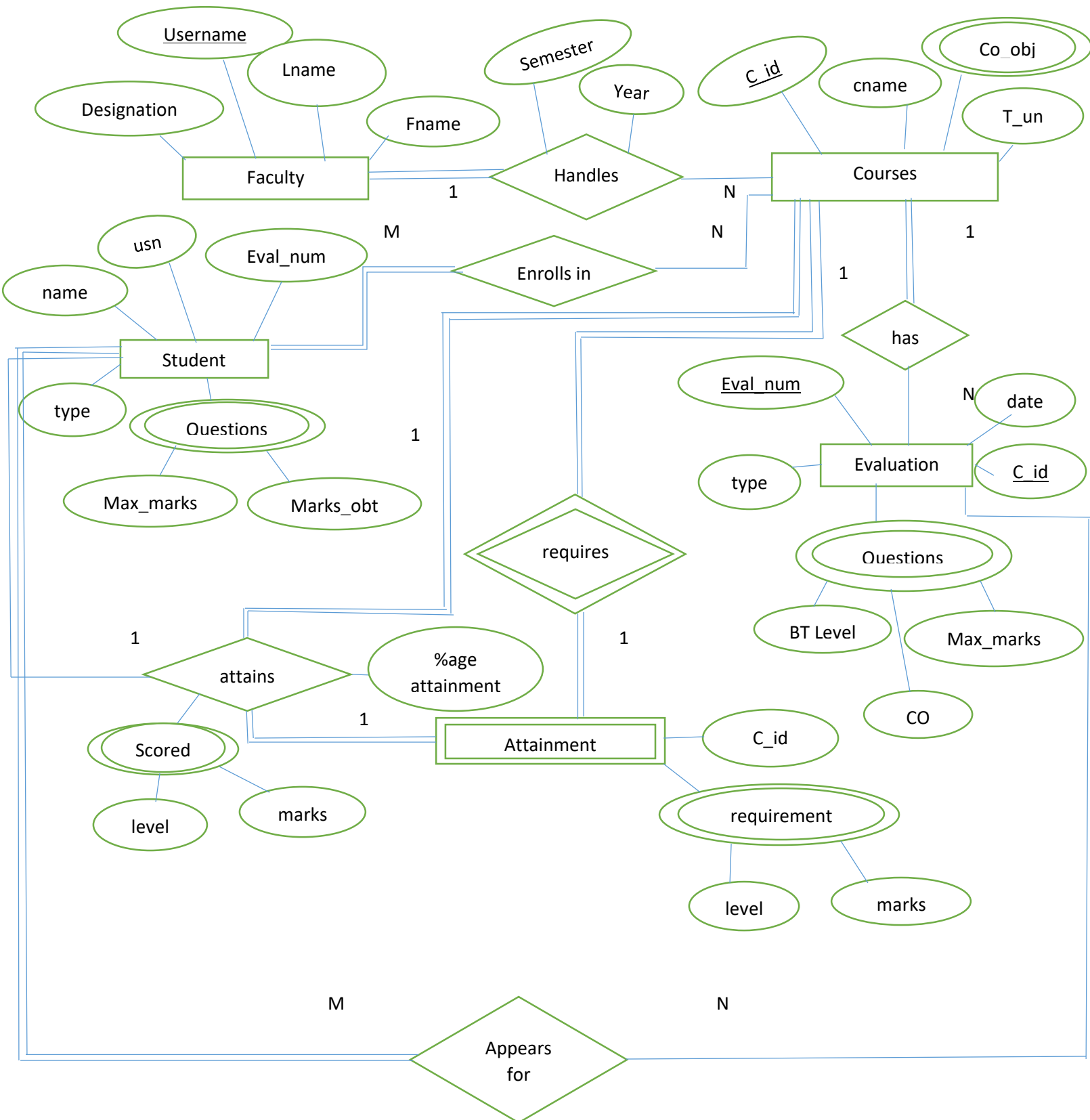
Figure 3.4.3 Dataflow Diagram Level 2

[Student Marks] - Contains student obtained marks information.

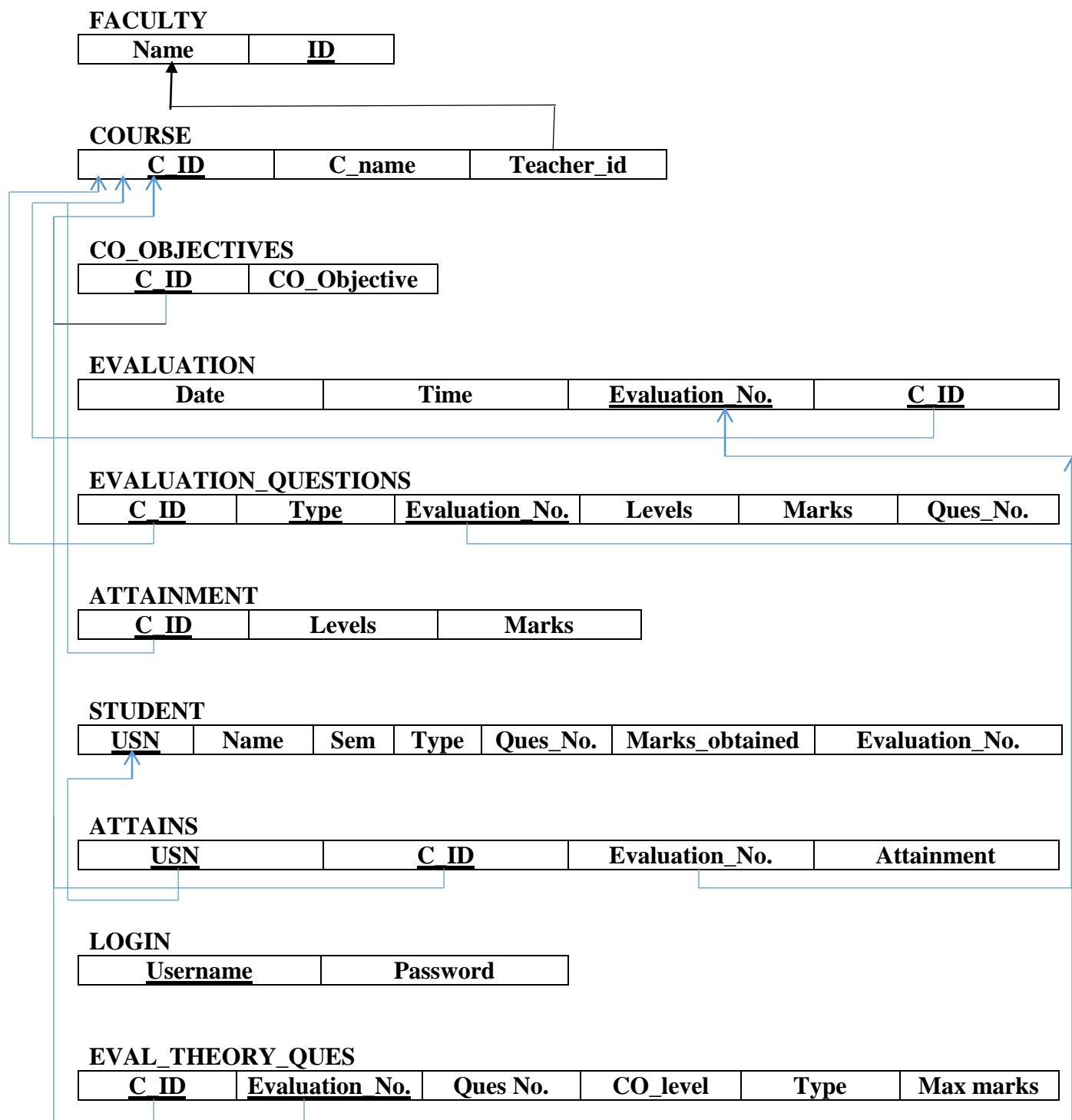
[Graphs] - Contains the comparison entities information for the generation of appropriate graphs.

[Charts] - Contains the entities information.

3.5 ER DIAGRAM



3.6 SCHEMA DIAGRAM



3.7 Normalized Schema

FACULTY

<u>username</u>	fname	lname	designation
-----------------	-------	-------	-------------

Diagram showing a line connecting the primary key username to the foreign keys fname, lname, and designation in the FACULTY table.

COURSE

<u>c_code</u>	c_name	t_un
---------------	--------	------

Diagram showing a line connecting the primary key c_code to the foreign key c_name in the COURSE table.

EVALUATION

<u>c_code</u>	<u>eval_num</u>	date	type	noofquiz	time
---------------	-----------------	------	------	----------	------

Diagram showing a line connecting the primary key c_code to the foreign keys eval_num, date, type, and time in the EVALUATION table.

EVAL_QUESTIONS

<u>c_code evq</u>	ques_num	co_level	max_marks	type	eval_num
-------------------	----------	----------	-----------	------	----------

Diagram showing a line connecting the primary key c_code evq to the foreign keys ques_num, co_level, max_marks, and eval_num in the EVAL_QUESTIONS table.

ATTAINMENT

<u>c_code attain</u>	co_level	req_marks
----------------------	----------	-----------

Diagram showing a line connecting the primary key c_code attain to the foreign keys co_level and req_marks in the ATTAINMENT table.

STUDENT

<u>usn</u>	sname	c_code_stu
------------	-------	------------

Diagram showing a line connecting the primary key usn to the foreign keys sname and c_code_stu in the STUDENT table.

STU_ATTAINMENT

<u>c_code attain</u>	<u>usn</u>	eval_num_appear	type_appear	ques_num_appear	marks_scored
----------------------	------------	-----------------	-------------	-----------------	--------------

Diagram showing a line connecting the primary key c_code attain to the foreign keys usn, eval_num_appear, type_appear, ques_num_appear, and marks_scored in the STU_ATTAINMENT table.

LOG_IN

<u>Username</u>	Password
-----------------	----------

Diagram showing a line connecting the primary key Username to the foreign key Password in the LOG_IN table.

EVAL_THEORY_QUES

<u>c_code</u>	ques_num.	co_level	type	max_marks	eval_num
---------------	-----------	----------	------	-----------	----------

Diagram showing a line connecting the primary key c_code to the foreign keys ques_num., co_level, type, max_marks, and eval_num in the EVAL_THEORY_QUES table.

3.8 Definition of tables

LOGIN TABLE

```
CREATE TABLE `LOG_IN` (  
  `username` VARCHAR( 20 ) NOT NULL ,  
  `password` VARCHAR( 150 ) NOT NULL ,  
  PRIMARY KEY ( `username` ));
```

FACULTY TABLE

```
CREATE TABLE `FACULTY` (  
  `username` VARCHAR( 20 ) NOT NULL ,  
  `fname` VARCHAR( 15 ) NOT NULL ,  
  `lname` VARCHAR( 15 ) NOT NULL ,  
  `designation` VARCHAR( 30 ) NOT NULL ,  
  PRIMARY KEY ( `username` ),  
  FOREIGN KEY ( username ) REFERENCES log_in( username ));
```

COURSE TABLE

```
CREATE TABLE `CO_OBJECTIVES` (  
  `c_code` VARCHAR( 10 ) NOT NULL ,  
  `cname` VARCHAR(40) NOT NULL ,  
  `t_un` VARCHAR( 20 ) NOT NULL ,  
  PRIMARY KEY ( `c_code`,`t_un` ),  
  FOREIGN KEY ( t_un ) REFERENCES log_in( username ));
```

STUDENT TABLE

```
CREATE TABLE `STUDENT` (  
  `usn` VARCHAR( 10 ) NOT NULL ,  
  `sname` VARCHAR( 30 ) NOT NULL ,  
  `c_code_stu` VARCHAR( 10 ) NOT NULL ,
```

PRIMARY KEY (`usn`),
FOREIGN KEY (c_code_stu) REFERENCES course(c_code));

STUDENT ATTAINMENT TABLE

```
CREATE TABLE `STU_ATTAINMENT` (  
  `c_code_attain` VARCHAR( 10 ) NOT NULL ,  
  `usn` VARCHAR( 10 ) NOT NULL ,  
  `eval_num_appear` INT(5) NOT NULL ,  
  `type_appear` VARCHAR( 10 ) NOT NULL ,  
  `ques_num_appear` INT(5) NOT NULL ,  
  `marks_scored` INT(5) NOT NULL ,  
  PRIMARY KEY ( `usn`,`c_code_attain`,`ques_num_appear` ),  
  FOREIGN KEY ( `c_code_attain`,`usn` ) REFERENCES student( `c_code_stu`,`usn` ));
```

THEORY EVALUATION TABLE

```
CREATE TABLE `EVAL_THEORY_QUES` (  
  `c_code` VARCHAR( 10 ) NOT NULL ,  
  `ques_num` VARCHAR( 5 ) NOT NULL ,  
  `co_level` VARCHAR( 5 ) NOT NULL ,  
  `type` VARCHAR( 10 ) NOT NULL ,  
  `max_marks` INT(5) NOT NULL ,  
  `eval_num` INT(11) NOT NULL ,  
  PRIMARY KEY ( `c_code`,`ques_num`,`eval_num` ),  
  FOREIGN KEY ( c_code ) REFERENCES course( c_code ));
```

EVALUATION QUESTION TABLE

```
CREATE TABLE `EVAL_QUESTIONS` (  
  `c_code_evq` VARCHAR( 10 ) NOT NULL ,  
  `ques_num` VARCHAR( 5 ) NOT NULL ,  
  `co_level` VARCHAR( 5 ) NOT NULL ,
```

```
`max_marks` INT( 10 ) NOT NULL ,  
`type` VARCHAR( 10 ) NOT NULL ,  
`eval_num` INT( 11 ) NOT NULL ,  
PRIMARY KEY ( `c_code_evq`,`ques_num`,`max_marks` ),  
FOREIGN KEY ( c_code_evq ) REFERENCES course( c_code ));
```

EVALUATION TABLE

```
CREATE TABLE `EVALUATION` (  
`c_code_eval` VARCHAR( 10 ) NOT NULL ,  
`date` DATE NOT NULL ,  
`eval_num` INT(5) NOT NULL ,  
`type` VARCHAR( 20 ) NOT NULL ,  
`noofquiz` INT(11) NOT NULL ,  
PRIMARY KEY ( `c_code_eval`,`date` ),  
FOREIGN KEY ( c_code_eval ) REFERENCES course( c_code ));
```

ATTAINMENT TABLE

```
CREATE TABLE `ATTAINMENT` (  
`c_code_attain` VARCHAR( 10 ) NOT NULL ,  
`co_level` INT(5) NOT NULL ,  
`req_marks` INT(10) NOT NULL ,  
PRIMARY KEY ( `c_code_attain` ),  
FOREIGN KEY ( c_code_attain ) REFERENCES course( c_code ));
```


3.9 Definition of relations and the cardinality ratios of the participating entities

Sl. No.	Participating Entity 1	Relation	Participating Entity 2	Cardinality Ratio
1.	Faculty	Handles	Courses	1:N
2.	Student	Enrolls in	Courses	M:N
3.	Student	Appears	Evaluation	M:N
4.	Courses	Requires	Attainment	1:1
5.	Courses	Has	Evaluation	1:N
6.	Student	Attains	Attainment	1:1
7.	Courses	Attains	Attainment	1:1

Table 3.9 Cardinality Ratios

Chapter 4**IMPLEMENTATION****4.1 Client-Server Specification**

The software runs on Client Server Architecture where the server side consists of php scripts running on the server. The server can be run on windows as well as linux based systems. It works best with Apache server 2.4, php 5.5.9 and Mysql server 5.5.xx installed on the system.

The clients can use any AJAX compatible browser. Further specifications for the browsers are listed below:

Mozilla Firefox – Version 31

Google Chrome – Version 44

Having the above mentioned versions of the browsers at the client side enables the use of AJAX and JQuery which is used for the development of the software.

The software developed has fat server architecture, which implies that the clients only need to have minimal configuration for their systems.

4.2 Technologies

The various technologies used for development purposes are:

(I) HTML5

(II) Javascript

(III) JQuery

(IV) AJAX

(V) XML

(VI) CSS

(VII) JSON

(VIII) PHP

(IX) MySQLi

4.2.1 HTML5

- HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It was finalized, and published, on 28 October 2014 by the World Wide Web Consortium.
- Its core aims are to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices.
- Various new APIs have been included :- Geolocation, Local Storage, etc.

4.2.2 JavaScript

- JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications.
- JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.
- Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.
- JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.
- We can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.

4.2.3 JQuery

- JQuery is a fast and concise JavaScript library created by John Resig in 2006. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for Rapid Web Development.
 - The JQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
 - The JQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.
 - JQuery is very lightweight library - about 19KB in size.
-

4.2.4 AJAX

- Ajax is a set of web development techniques utilizing many web technologies used on the client-side to create asynchronous Web applications.
- With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.
- Ajax is not a technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display – and allow the user to interact with – the information presented.

4.2.5 XML

- XML stands for EXtensible Markup Language.
- XML is designed to store and transport data.
- XML is designed to be self-descriptive.
- XML is designed to be both human- and machine-readable.
- XML is a W3C Recommendation.

4.2.6 CSS

- Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
- CSS saves time.
- Pages load faster.
- Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Offline Browsing - CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
- Platform Independence - The Script offer consistent platform independence and can support latest browsers as well.

4.2.7 JSON

- JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange.
- It is used while writing JavaScript based applications that includes browser extensions and websites.
- It is primarily used to transmit data between a server and web applications.
- Web services and APIs use JSON format to provide public data.
- It can be used with modern programming languages.

4.2.8 PHP

The programming language used for the development work is PHP. The reason for selection of this language includes among many others the following few.

- Open Source, PHP is completely free.
- PHP can be easily embedded directly into HTML.
- Platform independent can run on Windows Linux or Mac servers.
- Run faster on the internet and easily integrate AJAX, Callback etc.
- Interfaces very easily with Apache/MySQL
- Lots of good books and on-line help.
- It's available with documentation in many languages.
- Easy to learn compared to many other scripting languages. It has a syntax that is easy to parse and is actually rather human-friendly.
- Lots of hosting services have it ready to use, no special configuration. Pretty easy to access other web-based tools through PHP i.e. google maps, etc

4.2.9 MySQL

MySQLi is a relational data base management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

- MySQL is an open source tool.
- MySQL is a popular choice of database for use in web applications, and is a central component of the widely-used LAMP web application software stack — LAMP is an acronym for "Linux, Apache, MySQL, PHP".
- MySQL is primarily an RDBMS and therefore ships with no GUI tools to administer MySQL databases or manage data contained within.
- Microsoft SQL server Express Management studio provides an ease in creating tables by a graphical as well as query based interface.

Chapter 5**TESTING**

Software Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer software. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors.

5.1 Unit Testing

Unit testing is done to verify and validate whether individual units of source code are fit for use. A unit is the smallest testable part of an application. The smallest unit of the Course Outcome Mapping System may be a login, change password, or validation integrity check operation.

5.1.1 Unit Test Case 1**Table 5.1** Unit Test Case 1 for Course Outcome Mapping System

Sl.no of test case:	1
Name of test:	Change password test
Item / Feature being tested:	Change password form
Sample input:	Old Password = “hello”, New Password = “hi”, Retype = “hi”,
Expected output:	Message: Password Changed Successfully
Actual output	Message: Password Changed Successfully
Remarks:	Test succeeded

5.1.2 Unit Test Case 2**Table 5.2** Unit Test Case 2 for Course Outcome Mapping System

Sl.no of test case:	2
Name of test:	Change password test
Item / Feature being tested:	Change password form
Sample input:	Old Password = “hello”, New Password = “hi”, Retype = “hell”,
Expected output:	Message: Could not change password

Actual output	Message: Could not change password
Remarks:	Test succeeded

5.1.3 Unit Test Case 3

Table 5.3 Unit Test Case 3 for Course Outcome Management System

Sl.no of test case:	3
Name of test:	Validation test for user integrity
Item / Feature being tested:	Redirection if user not logged in
Sample input:	Type the URL 'add_marks.php'
Expected output:	Redirect to 'login.php'
Actual output:	Redirect to 'login.php'
Remarks:	Test succeeded

5.2 Integration Testing

Integration testing (sometimes called Integration and Testing, abbreviated "I&T" is the activity of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Individual forms which had been unit tested are integrated and then tested to check whether navigation is possible from one form to the other in a seamless manner. Thus, the working of connectivity can be tested.

5.2.1 Integration Test Case 1

Table 5.4 Integration Test Case1 for Course Outcome Management System

Sl.no of test case:	1
Name of test:	Form navigation
Item / Feature being tested:	Login button
Sample input:	Username= "abc" ,password= "mypwrd"
Expected output:	Move to next page and other pages on successive clicks
Actual output:	Move to next page and other pages on successive clicks

Remarks:	Test succeeded
----------	----------------

5.2.2 Integration Test Case 2

Table 5.5 Integration Test Case2 for Course Outcome Management System

Sl.no of test case:	2
Name of test:	Data retrieval
Item / Feature being tested:	View Course Outcome
Sample input:	Select Course code for CO Graph
Expected output:	Display CO Graph of that paper
Actual output:	Display CO Graph of that paper
Remarks:	Test succeeded

5.3 System Testing

5.3.1 System Test Case 1

Table 5.6 System Test Case1 for Course Outcome Management System

Sl.no of test case:	1
Name of test:	Add marks
Item / feature being tested:	Add marks
Sample input:	Add marks
Expected output:	Particular internal marks get added
Actual output:	Particular internal marks get added
Remarks:	Test succeeded

Chapter 6

RESULTS

6.1 Snapshots



Figure 6.1.1 Snapshot of Main Page

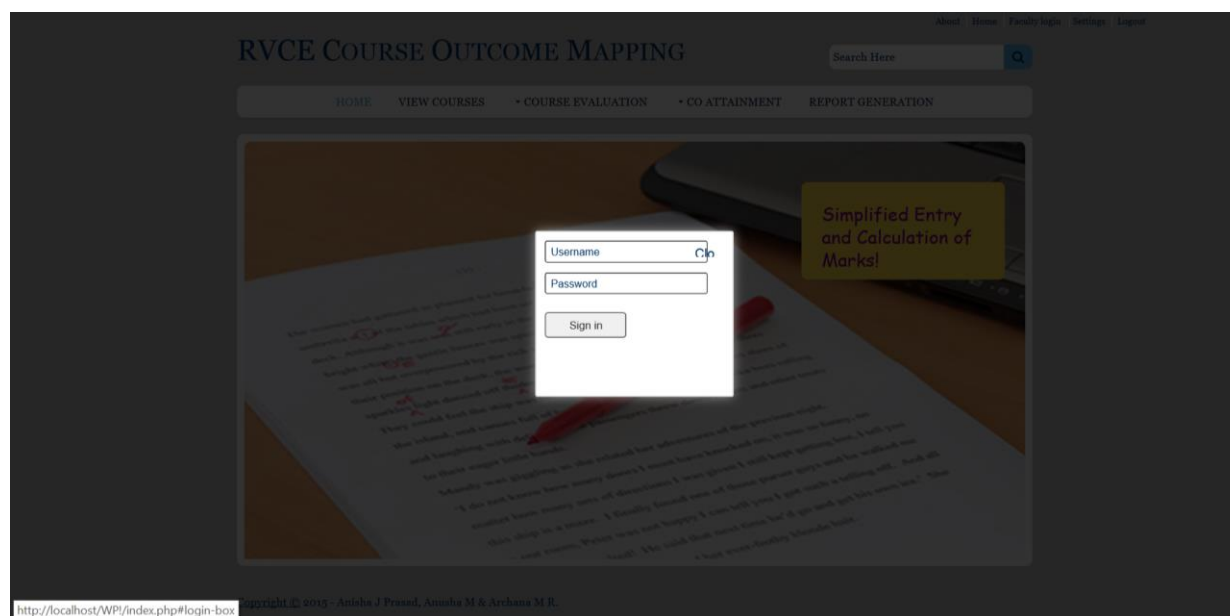


Figure 6.1.2 Snapshot of Login Page

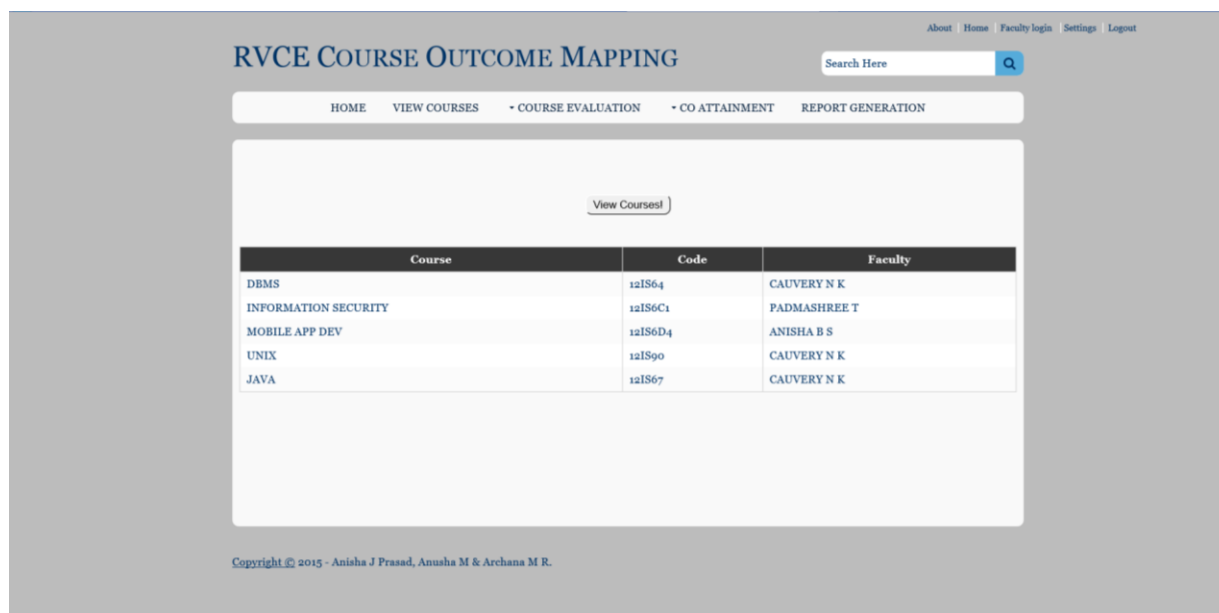


Figure 6.1.3 Snapshot of View Courses Page

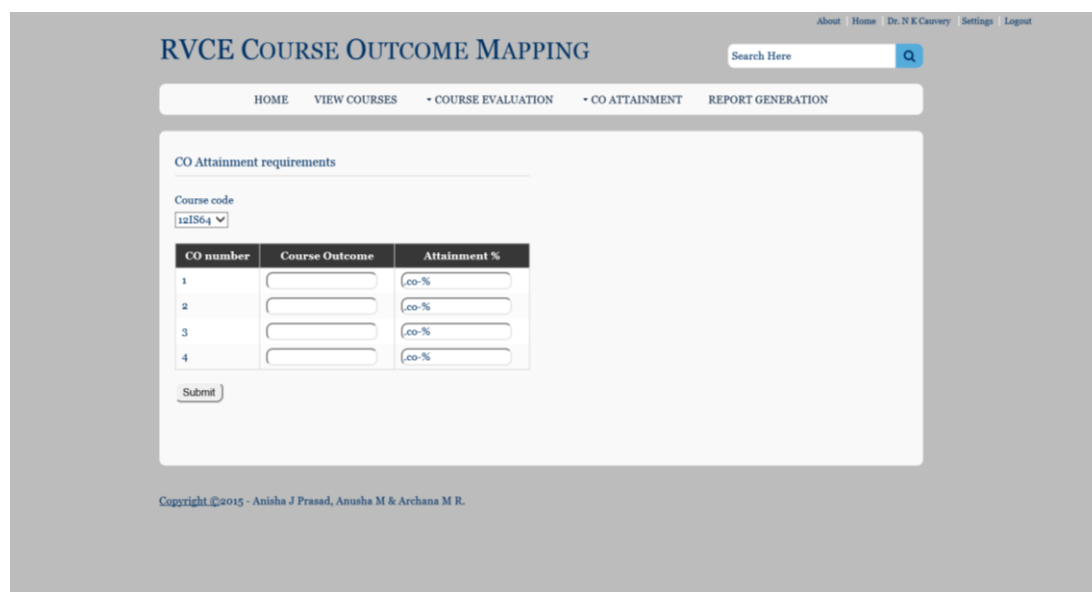


Figure 6.1.4 Snapshot of View Course attainment requirements Page

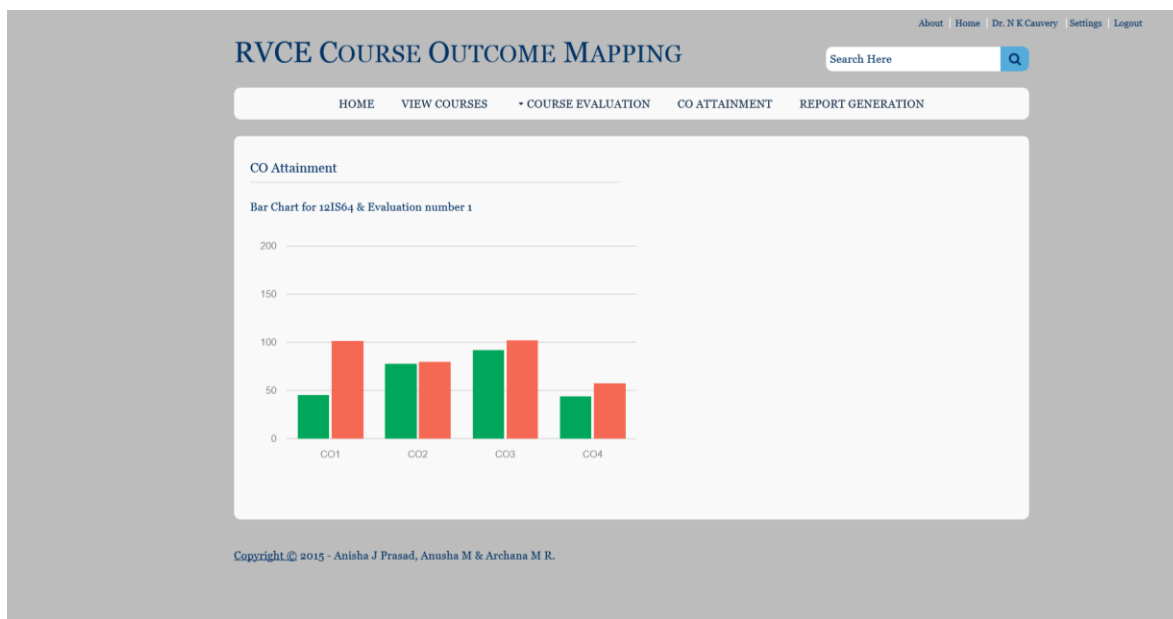


Figure 6.1.5 Snapshot of graph of test wise marks Page



Figure 6.1.6 Snapshot of graph of average of tests Page

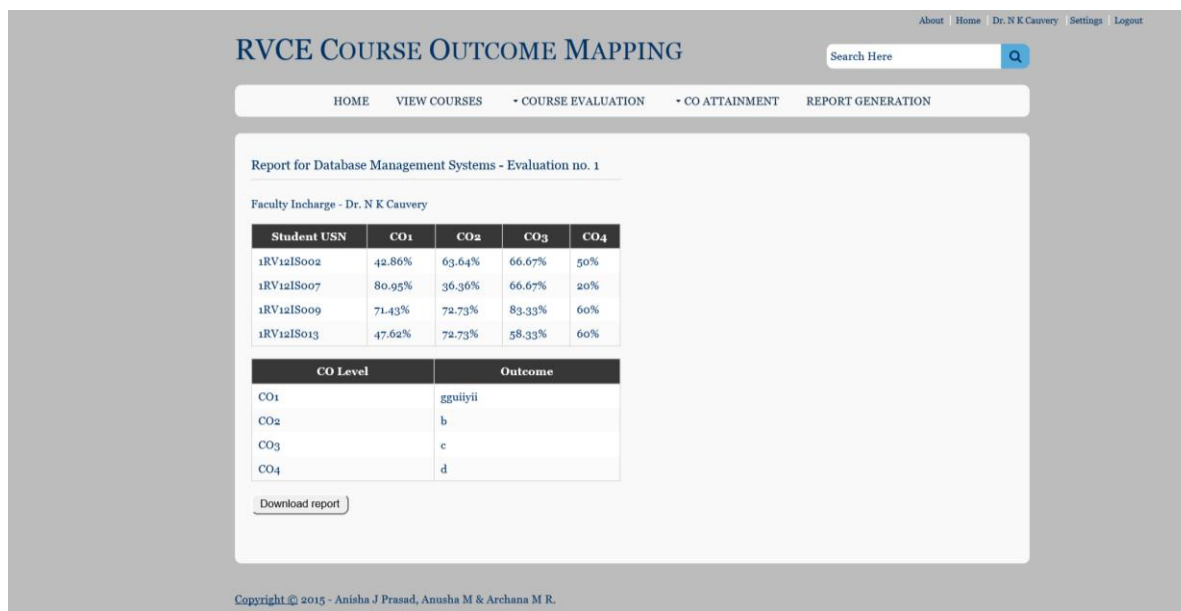


Figure 6.1.7 Snapshot of report generation Page

6.2 Advantages

- A common portal in which staff can view CO.
- Updates can be made quickly and easily.
- All kinds of records are found at one stop.
- Users have no limit on the amount of content they can upload.

6.3 Limitations

- The project runs on the assumption of fixed scoring rules and CO rules.
- User has to make all the entries.
- Cross browser compatibility (does not work on some older versions of IE).

Chapter 7**CONCLUSION**

The Course Outcome Mapping System is a project designed for departments in autonomous institutions where students of each year are segregated into batches and assigned a member of staff as the counsellor for that batch. The counsellor takes charge of the portfolio of each student belonging to his/her respective batch. The student portfolio includes marks, attendance, certificates, tuition, grade point averages and other relevant information. The Course Outcome Mapping System performs this batch segregation and assists the counsellors in tracking the performance of students with respect to marks, attendance and extra-curricular activities. The Course Outcome Mapping System also enables the parents to keep track of the performance of their wards and interact with counselors personally using the appointment scheduler included in the portal.

7.1 Future Enhancements

- Dynamically update graphs based on subject selection.
- Automatic suggestions on the grades and CO of the certificates available to the counsellor.

REFERENCES

We have borrowed heavily from multiple sources, but would like to specifically mention and thank the following resources.

- 1) Mala Dutta, Kamal K Sethi and Ajay Khatri, “*Web Based Integrated Development Environment*”, IJITEE, ISSN: 2278-3075, Volume-3, Issue-10, March 2014.
- 2) PJ Deitel, HM Deital, TR Nieto, “*Internet & World Wide Web How to Program*”, 1st Edition, Pearson Education Limited Publications, 2012, ISBN 9788131701126.
- 3) Jennifer Niederst Robbins, “*Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*”, 2nd edition, 2014.
- 4) David A. Botwel and Joseph G. Davis, “*A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software*”, IJCSMC, Vol. 4, Issue. 2, February 2015, pg.154 – 165.
- 5) Elmasri and Navathe, “*Fundamentals of Database Systems*”, 5th edition, Addison-Wesley, 2007.
- 6) An Oracle White Paper in Enterprise Architecture December 2012, “*An Oracle White Paper in Enterprise Architecture - Oracle Enterprise Architecture Framework: Information Architecture Domain, Version 2.0*”
- 7) Hans Admiraal, *Pitfalls using UML in RUP*
- 8) Sparx Systems 2015, *Sparx Systems Enterprise Architect 12 Reviewer's Guide*
- 9) www.apachefriends.org for lampp installation and user guidelines.
- 10) www.php-mysql-tutorial.com for php and mysql examples.
- 11) www.w3schools.com.
- 12) David Sklar-Learning PHP 5,1st edition, O'Reilly Media, Inc., 2007.

APPENDIX A - LIST OF ACRONYMS

1. CO-Course Outcome
2. SRS: Software Requirement Specifications
3. ER: Entity Relationship
4. OS: Operating System
5. RAM: Random Access Memory
6. SQL: Structured Query Language
7. PHP: Hypertext Preprocessor(Personal Home Page)
8. RDBMS: Relational Database management systems

APPENDIX B – CODING

```

<!DOCTYPE html>

<html>

<head>

<title>RVCE CO Mapping</title>

<meta charset="utf-8">

<meta name="view port" content="width=device-width, initial-scale=1.0, maximum-scale=1.0,
user-scalable=no">

<link href="layout/styles/layout.css" rel="stylesheet" type="text/css" media="all">

<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>
type="text/javascript"

</head>

<body id="top">

<!--<div class="wrapper row0">-->

    <div id="topbar" class="clear">

        <nav>

            <ul>

                <li><a href="about.php">About</a></li>

                <li><a href="index.php">Home</a></li>

                <li><a href="#login-box" class="login-window">Faculty
Login</a></li>

                <li><a href="settings.php">Settings</a></li>

            </ul>

        </nav>

    </div>

<!--</div>-->

<br>

<div class="wrapper row1">

    <header id="header" class="clear">

        <div id="logo" class="fl_left">

```

```
<h1><a href="index.html">RVCE Course Outcome Mapping</a></h1>
<p></p>
</div>
<div class="fl_right">
  <form class="clear" method="post" action="#">
    <fieldset>
      <legend>Search:</legend>
      <input type="text" value="" placeholder="Search Here">
      <button class="fa fa-search" type="submit" title="Search"><em>Search</em></button>
    </fieldset>
  </form>
</div>
</header>
</div>
<div class="wrapper row2">
  <div class="rounded">
    <nav id="mainnav" class="clear">
      <ul class="clear">
        <li class="active"><a href="index.html">Home</a></li>
        <li><a href="view_courses.html">View Courses</a></li>
        <li><a class="drop" href="#">Course Evaluation</a>
          <ul>
            <li><a href="co_attain.php">CO attainment requisite</a></li>
            <li><a href="eval_pattern.php">Create evaluation pattern</a></li>
            <li><a class="drop" href="#">Add student marks</a>
              <ul>
                <li><a href="manual_entry.php">Manual entry</a></li>
                <li><a href="upload_marks.php">Upload Excel sheet</a></li>
              </ul>
            </li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</div>
```

```
</ul>
</li>
<li><a href="#">CO Attainment</a></li>
<li><a href="report_gen.php">Report Generation</a></li>
</ul>
</nav>
</div>
</div>
<div id="login-box" class="login-popup">
  <a href="index.html" class="close"><img src="" class="btn_close" title="Close Window"
alt="Close" /></a>
  <form method="post" class="signin" action="#">
    <fieldset class="textbox">
      <label class=
        <span></span>
      <input id="username" name="username" value="" type="text" autocomplete="on"
placeholder="Username">
      </label>
      <label class=
        <span></span>
      <input id="password" name="password" value="" type="password"
placeholder="Password">
      </label>
      <button class="submit button" type="button">Sign in</button>
    </fieldset>
  </form>
</div>
<div class="wrapper row3">
  <div class="rounded">
    <main class="container clear">
      <!-- main body -->
```

```

    <br><br><br>
    <center><button type="button" onclick="loadDoc()">View Courses!</button></center>
        <br><br>
</table id="demo"></table>
    <div class="clear"></div>
</main>
</div>
</div>
<div class="wrapper row4">
    <!--<div class="rounded">-->
    <p class="fl_left">Copyright &copy; 2015 - Anisha J Prasad, Anusha M & Archana M
R.</p></div>
<!-- JAVASCRIPTS -->
<script>
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (xhttp.readyState == 4 && xhttp.status == 200) {
            myFunction(xhttp);
        }
    }
    xhttp.open("GET", "courses.xml", true);
    xhttp.send();
}
function myFunction(xml) {
    var i;
    var xmlDoc = xml.responseXML;
    var table="<tr><th>Course</th><th>Code</th><th>Faculty</th></tr>";
    var x = xmlDoc.getElementsByTagName("COURSEINFO");
    for (i = 0; i <x.length; i++) {

```

```
table += "<tr><td>" +  
x[i].getElementsByTagName("COURSE")[0].childNodes[0].nodeValue +  
"</td><td>" +  
x[i].getElementsByTagName("CODE")[0].childNodes[0].nodeValue +  
"</td><td>" +  
x[i].getElementsByTagName("FACULTY")[0].childNodes[0].nodeValue +  
"</td></tr>";  
}  
document.getElementById("demo").innerHTML = table;  
}  
</script>  
<script src="layout/scripts/jquery.min.js"></script>  
<script src="layout/scripts/jquery.fitvids.min.js"></script>  
<script src="layout/scripts/jquery.mobilemenu.js"></script>  
<script src="layout/scripts/tablet/jquery.tablet.min.js"></script>  
</body>  
</html>
```