**GIRNE AMERICAN UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

**Spring 2015-2016**

**CEN 401 - GRADUATION PROJECT I**

***“GRADE MANAGER SYSTEM”***

**Prepared By,**

Onur ÇINAR (131701011)

Onur Cem EZGİN (121701077)

**Supervised By,**

Asst.Prof.Dr İbrahim ERŞAN

**Submission Date,**23rd May, 2016

**1 INTRODUCTION ………………………………………………….... 3**

**2 SERVER SETUP ………………..……………................................. 4-5-6-7**

**3 INFORMATION ABOUT USABILITY ……………………... 8**

**3.1 Superadmin ................................................................................ 8**

**3.2 Admin ........................................................................................... 8**

**3.3 Superuser ................................................................................... 8**

**3.4 Lecturer ....................................................................................... 8-9**

**4 TECNICAL DETAILS …...............................................………... 10**

**4.1 Languages**  **............................................................................... 10**

**4.2 IDE's ............................................................................................ 10**

**4.3 Design ......................................................................................... 10**

**4.4 JS-Grid ........................................................................................ 10-11**

**4.5 Page Connections ................................................................... 12**

**5 INFORMATION ABOUT DATABASE .............................. 13**

**5.1 Technical Details Of Database ........................................... 13**

**5.1.1 IR Diagram** **............................................................................ 13**

**5.1.2 Entities & Relations** **.............................................................. 14**

**6 REFERENCES ............................................................................. 15**

**1- INTRODUCTION**

**What is our project?**

Our project name is **'Grade Manager'**.Basically this program provide keep notes and grade calculation.Also this software allows you make curve.

**What is our aim in that system ?**

With this system would be transparency in the grading system.All grades can be displayed with a superuser account.This software facilitates the editing and calculating notes.Our another aim is create user-friendly and simple system.

In the universities if the university has no grade calculation system it’s problem for lecturers. Some lecturers calculating in excel some lecturers has different method but its not stable it could occur a lots of errors and mistakes.

Curve and other thinks about grades also big problem without system because it needs a different and hard calculations.

For the university administration its also problem for the transparency and following notes and grades of exams directly.

With the our grade manager system this problems will gone and all of thinks will be simple and easy.

**2 - SERVER SETUP**

**Operating System**

We have linux CentOS 6.4 version in server. Linux server is the best OS for if you are using PHP language. CentOS is the best version of linux for servers.

**Installing MySQL 5**

To install MySQL, we do this:

*yum install mysql mysql-server*

Then we create the system startup links for MySQL (so that MySQL starts automatically whenever the system boots) and start the MySQL server:

*chkconfig --levels 235 mysqld on  
/etc/init.d/mysqld start*

Set passwords for the MySQL root account:

*mysql\_secure\_installation*

### Installing Apache2

Apache2 is available as a CentOS package, therefore we can install it like this:

*yum install httpd*

Now configure your system to start Apache at boot time...

*chkconfig --levels 235 httpd on*

... and start Apache:

*/etc/init.d/httpd start*

### Installing PHP5

We can install PHP5 and the Apache PHP5 module as follows:

*yum install php*

We must restart Apache afterwards:

*/etc/init.d/httpd restart*

### 5Testing PHP5 / Getting Details About Your PHP5 Installation

The document root of the default web site is */var/www/html*. We will now create a small PHP file (*info.php*) in that directory and call it in a browser. The file will display lots of useful details about our PHP installation, such as the installed PHP version.

*vi /var/www/html/info.php*

|  |
| --- |
| <?php  phpinfo();  ?> |

Now we call that file in a browser (e.g. *http://* 51.254.204.78*/info.php*)

### Getting MySQL Support In PHP5

To get MySQL support in PHP, we can install the *php-mysql* package. It's a good idea to install some other PHP5 modules as well as you might need them for your applications. You can search for available PHP5 modules like this:

*yum search php*

Pick the ones you need and install them like this:

*yum install php-mysql php-gd php-imap php-ldap php-mbstring php-odbc php-pear php-xml php-xmlrpc*

APC is a free and open PHP opcode cacher for caching and optimizing PHP intermediate code. It's similar to other PHP opcode cachers, such as eAccelerator and Xcache. It is strongly recommended to have one of these installed to speed up your PHP page.

APC can be installed as follows:

*yum install php-pecl-apc*

Now restart Apache2:

*/etc/init.d/httpd restart*

### phpMyAdmin

[phpMyAdmin](http://www.phpmyadmin.net/) is a web interface through which you can manage your MySQL databases.

First we enable the [RPMforge repository](https://rpmrepo.org/RPMforge/Using) on our CentOS system as phpMyAdmin is not available in the official CentOS 6.4 repositories:

Import the RPMforge GPG key:

*rpm --import http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt*

**On i386 systems:**

*yum install http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.i686.rpm*

phpMyAdmin can now be installed as follows:

*yum install phpmyadmin*

Now we configure phpMyAdmin. We change the Apache configuration so that phpMyAdmin allows connections not just from localhost (by commenting out the *<Directory "/usr/share/phpmyadmin">* stanza):

*vi /etc/httpd/conf.d/phpmyadmin.conf*

Next we change the authentication in phpMyAdmin from *cookie* to *http*:

*vi /usr/share/phpmyadmin/config.inc.php*

Restart Apache:

*/etc/init.d/httpd restart*

Afterwards, you we access phpMyAdmin under *http://* 51.254.204.78*/phpmyadmin/*

After this steps server is ready for working with php language. When the files upload to root file in server page will work.

NOTE: Our servers ip address is :51.254.204.78

### Links

* Apache: <http://httpd.apache.org/>
* PHP: <http://www.php.net/>
* MySQL: <http://www.mysql.com/>
* CentOS: <http://www.centos.org/>
* phpMyAdmin: <http://www.phpmyadmin.net/>

**3 INFORMATION ABOUT USABILITY**

We have four type user in our project.These are;

* Superadmin
* Admin
* Superuser
* Lecturer

**3.1 Superadmin**

This user create and edit all type users included admin type user.We created this user type for the authority clashes.Because an admin type user can edit and delete other admin type user and it can create a big problem in system.For the solving that problem we have one superadmin user just this user can create,edit and delete admin type user.

All type users can be add and edit with this user.We can say this user have permission for user management .

**3.2 Admin**

This type user can manage,create and edit lecturer and Superuser type users. This user is like sub-system manager for user management.This type user have all permissions on lecturer and superuser type user.

**3.3 Superadmin**

Superuser type user is created for inspection to all notes. This type user can see all courses and the all notes and grades of that course.This account can be use by student administration or rectorate. With this user we bring transparency for the grades.

This type user is system manager at the same time because superuser can take back-up of all database in sql format.

**3.4 Lecturer**

Lecturer type user is main part of our project.Lecturer user can create courses and can calculate letter grades of that course. After the creating course lecturer can create clases and can manage the grades of all exams.

Our system gives a chance for curve with the changing scales of letter grades for all courses.System is dynamically calculate letter grade and lecturer can easily take backup of notes and grades in excel format.

**4 TECHNICAL DETAILS**

**4.1 Which languages we used ?**

We used PHP language,Javascript(jquery,json),HTML,CSS

**4.2 Which IDE's we used?**

We used PhpStorm because it is good IDE for PHP language.It shows the errors and it has a FTP client.We can easily upload the files and syncronize the web site easily.

**4.3 Design**

In our project we used semantic-ui library for the designing forms and all objects.

With semantic-ui it gives you good chooses for design with prepared css and jsfiles.

**4.4 Jsgrid**

We used js-grid tables library for the all about tables in our all pages and we create settings(loading-editing-deleting) of this table with json.

More information about js-grid you can see www.js-grid.com

jsgrid is a lightweight client-side data grid control. This is the example script code for connection of jsgrid.

**$("#exampletag").jsGrid({  
 width: "%100",  
 height: "auto",**

This part is for the table properties.

**inserting: true,  
 paging: false,  
 autoload:true,  
  
 controller: {**

This part for the loading data for the table.

**loadData: function(filter) {  
 return $.ajax({  
 type: "GET",  
 url: "/clients/",  
 data: filter  
 });  
 },**

This part for the inserting data for the table

**insertItem: function(item) {  
 return $.ajax({  
 type: "POST",  
 url: "/clients/",  
 data: item  
 });  
 },**

This part for the updating data for the table

**updateItem: function(item) {  
 return $.ajax({  
 type: "PUT",  
 url: "/clients/",  
 data: item  
 });  
 },**

This part for the deleting data for the table

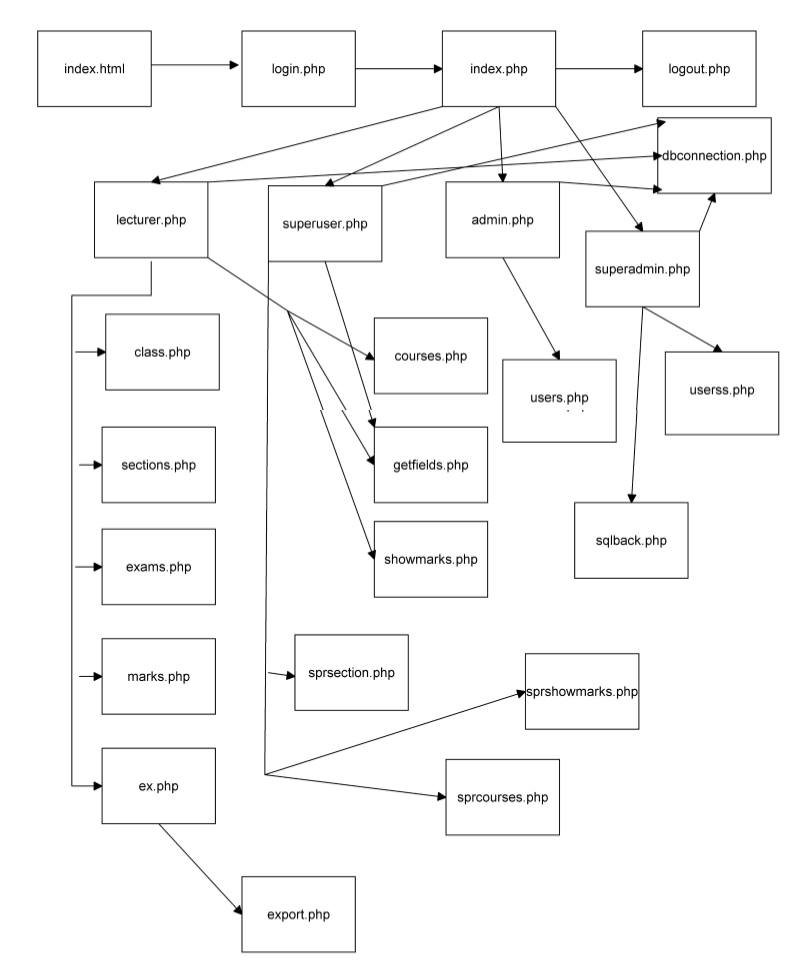
**deleteItem: function(item) {  
 return $.ajax({  
 type: "DELETE",  
 url: "/clients/",  
 data: item  
 });  
 }  
 },**

This part for the columns of table and values of it.

**fields: [  
 { name: "course\_code", title:"Course Code", type: "text", width: 50, validate: "required" },  
 { name: "s\_year", title:"Year", type: "select", items: [{"s\_year":"2016-17"},{"s\_year":"2017-18"},{"s\_year":"2018-19"}], valueField: "s\_year", textField: "s\_year", width: 50 },  
 { name: "semester", title:"Semester", type: "select", items: [{"semester":"Spring"},{"semester":"Fall"},{"semester":"Summer"}], valueField: "semester", textField: "semester", width: 50 },  
 { name: "course\_type", title:"Course Type", type: "select", items: [{"course\_type":"Turorial"},{"course\_type":"Class"},{"course\_type":"Labarotory"}], valueField: "course\_type", textField: "course\_type", width: 50 },  
 { type: "control", editButton:false, deleteButton:false}  
 ]  
});**

**4.5 Page Connections**

In figure 4.1 its possible to see connected and referenced pages to each other.

****Figure 4.1 Page connection table

**5 INFORMATION ABOUT DATABASE**

**5.1 Tecnical Details Of Database**

**What we have used?**

We used MySQL Database System.We managed our database with phpMyAdmin and Navicat.

We used PDO(PHP Data Objects) class connection strings in our project for database connection. In old style connection it was not secture for the SQL injection but in PDO connection its not possible to do SQL injection.

A great benefit of PDO is that it has an exception class to handle any problems that may occur in our database queries. If an exception is thrown within the try{ } block, the script stops executing and flows directly to the first catch(){ } block.

**5.1.1 ER Diagram**

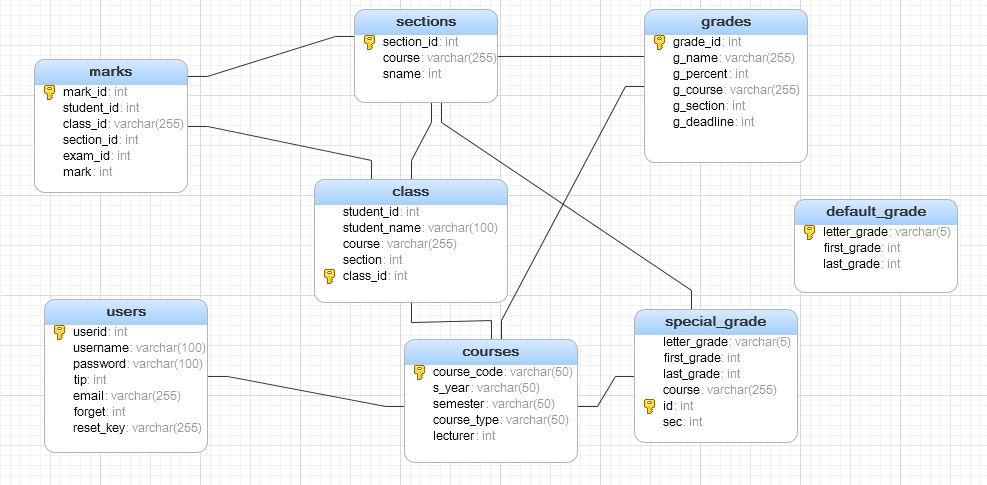
****

Figure 5.1 ER diagram of our database

**5.1.2 Entities&Relationships**

**Entities**

courses

class

users

default\_grade

grades

marks

sections

special\_grade

**Relationships**

marks(section\_id) references sections(section\_id)

marks(class\_id) references courses(course\_code)

class(section) references sections(section\_id)

class(course) references courses(course\_code)

courses(lecturer) references users(userid)

special\_grade(course) references courses(course\_code)

special\_grade(sec) references sections(section\_id)

grades(g\_course) references courses(course\_code)

grades(g\_section) references sections(section\_id)

**REFERENCES**

[1] http://wiki.hashphp.org/PDO\_Tutorial\_for\_MySQL\_Developers

[2] www.youtube.com

[3] www.w3schools.com

[4] www.php.net

[5] www.semantic-ui.com

[6] http://js-grid.com/docs/

[7] https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu

[8] http://www.w3schools.com/php/php\_mysql\_connect.asp