

ADD/DROP ONLINE PORTAL

(MANUAL/README)

Brief Description of the Project

In this project, we have made a portal for adding and/or dropping courses online. All the processes required in this academic procedure are done online. Students are now just required to log on to the portal and fill the form which would be further given approval by the required teachers/academics. Teachers and academic staff can also approve reject the courses simply by logging in.

System Requirements

Server Side:

- Operating Systems: Windows, Ubuntu or any other operating system.
- Apache web server version 2.4.4 or later.
- MySql 5.6.12 or later for database management.
- PHP Server 5.4.12 or later .
- phpMyAdmin MySql management interface(optional).
- Composer for Google oauth 2.0 authentication.
- Curl.

Client Side:

- Any browser with JavaScript and cookies enabled.

How to use the application

To setup the project, just follow these instructions.

1. Download the project either from SVN or from google drive.
2. If you are using Windows, install WAMP server by following this guide.
<http://www.wampserver.com/en/>

If using Ubuntu install LAMP server by following the link given below-

<https://help.ubuntu.com/community/ApacheMySQLPHP>

3. Install curl if using Ubuntu.
4. Install composer (needed for the Oauth 2.0 authentication)
You can install these using the link given below-
<https://developers.google.com/+/api/>
5. If you would like to run the files on local server (localhost), all the PHP files have to be put in a folder named as www which is present in /var/www in Ubuntu and in the WAMP/www directory in Windows.
6. If you are hosting it somewhere else, you need to put it in the appropriate directory. Talk to your ISP regarding the same.
7. The SQL file of the database named “add_drop”(located in the folder MyProject/dist) should be imported in phpMyadmin. To do the same, go to the ‘import’ option in phpmyadmin and select the file and click on OK.
8. For using the project first change the admin id by following these instructions-
 1. Open the file class.admin.php located in MyProject/src.
 2. And change the value of \$adminid to the email you want to serve as admin/academic staff.
9. After changing admin id you can add instructors using admin id and update the list. For your convenience we have added you both as a teacher and a advisor already.
10. You also need to change the username and password of the database.
Follow the following steps-
 - Open the file class.database.php located in MyProject/src.
 - Modify the lines 4,5,6,7 like this-

1. `hostIp="your_ip_address"`
2. `username="your_username"`
3. `password="your_password"`
4. `databaseName="your_database_name"`

11. And now, we just need to run the `signin.php` file. To do that, open any browser and type '`hostname/MyProject/src/signin.php`'.

Eg. If running on localhost, type `localhost/MyProject/src/signin.php`.

A sample running example

- The student first fills the form.
- After that the instructor performs its action. If the instructor approves the request then the value is updated in the database. If the instructor disapproves the request then the values is updated in the database and the process stops. The student can later check this manually by logging in the portal or by checking the list which will be produced by the academic section at the end of the add/drop date.
- If the request is approved by the instructor, the corresponding course advisor can then log in and approve/reject those request that were approved by instructor. If the advisor disapproves the request then the values is updated in the database and the process stops. The student can later check this manually by logging in the portal or by checking the list which will be produced by the academic section at the end of the add/drop date.
- After getting approved by the course advisor, the administration can approve/reject the course and the corresponding value is updated in the database.
- After the add/drop period ends, the academic staff can generate the final list of all the requests.
- The student can anytime check the status of his form.

Description of various files

These are the different files that are included in our project-

- **class.database.php**-This file is used to change the hostIp, username ,password and database name of the host.
- **class.admin.php**-This file is used to set the email id of the academic person.
- **signin.php**-This is the file provided by the google API.When a user logs in this file creates session and cookies for the user.After that all other files are rendered on this file.
- **index.php**-This is the file with the entire layout of the page.Dependent on who logs in the corresponding content is shown.
- **write.php**-This is used to store the add/drop request of a student in the database.
- **instructor.php**-This is used to show all the add/drop requests made by the student corresponding to a particular instructor so that he can approve /disapprove student's request.
- **instructorresponse.php**-This is used to store the information whether instructor has approved the request or not in the database.
- **advisor.php**- This is used to show all the add/drop requests made by the students corresponding to a particular advisor so that he can approve /disapprove student's request
- **advisorresponse.php**- This is used to store the information whether advisor has approved the request or not in the database.
- **academic.php**- This is used to show all the add/drop requests made by the all the students so that the administration can approve /disapprove student's request
- **academicresponse.php**-- This is used to store the information whether administration has approved the request or not in the database
- **status.php**-This file is used to check the status of the form.
- **checkmail.php**-This file is used to check if the user is a student/instructor/advisor/administration.

