VSB Plus

Xia Hua - 200368746
Xinyu Liu - 200362878
Priscilla Chua - 200363504
Jingkang Yang - 200362586

Introduction

What is it?

- A web-based simulator designed to help students choose courses wisely.

Why doing this?

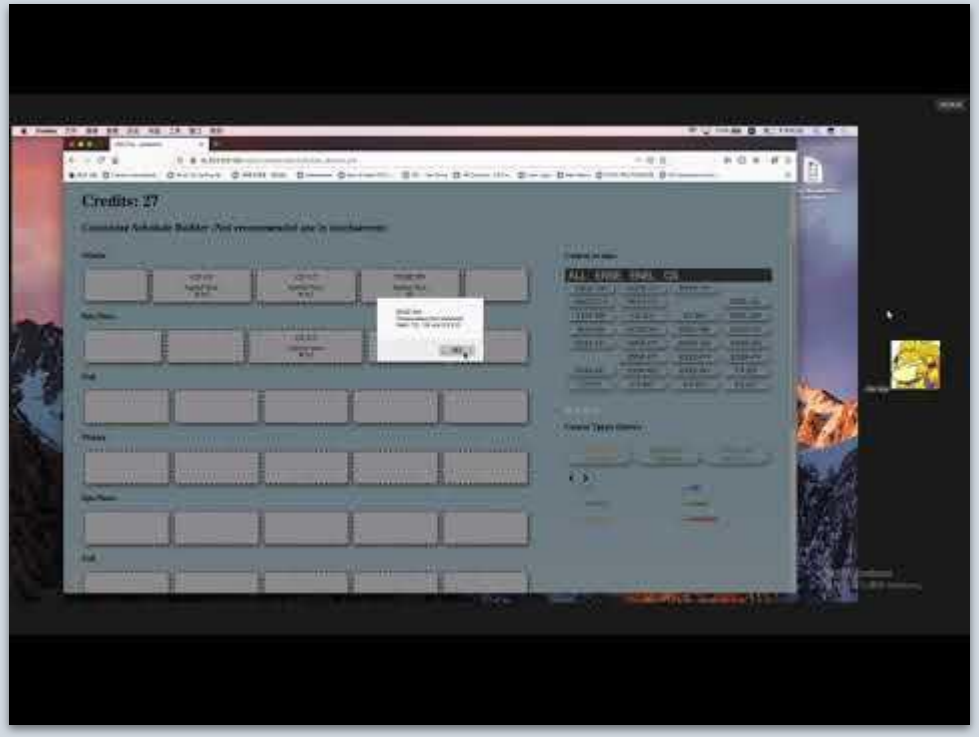
- Help students avoid unnecessary courses.

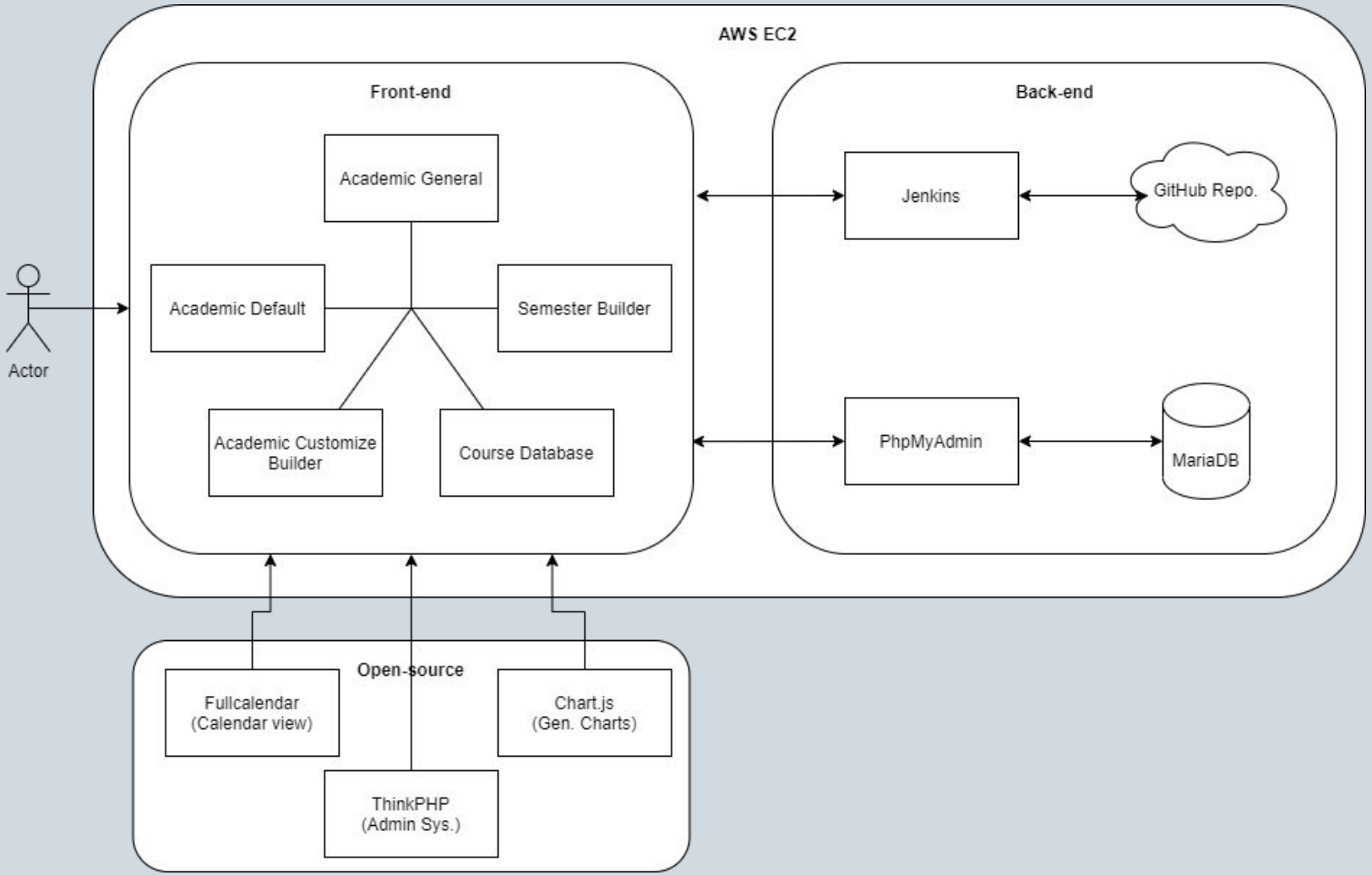
Who is the user?

- Students who do not understand the course registration requirements.
- Course advisors who want to explain problems to students visually.

Demo!

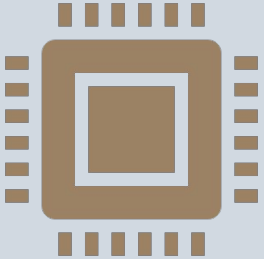
Demo





Architectural Design

AWS EC2



What is AWS EC2?

Amazon Elastic Compute Cloud (Amazon EC2)

Why AWS EC2?

Expandable

Experienced

Economical

Front-end

Academic General Status

- Graphically display

Academic Default Schedule

- Official schedule
- Progress status

Academic Customize Schedule Builder

- Create own schedule

Semester Customize Schedule Builder

- Current and Upcoming semester
- Lecture information

Course List Database

- Filter and Search
- Course details

Open Sources

Fullcalendar

- Weekly/monthly schedule

Chart.js

- Graphical display

ThinkPHP

- Admin system

Fullcalendar

What is fullcalendar?

- Open source JavaScript API using a very small embedded virtual DOM library.

Functionality

- Generate the calendar with multiple settings.

Why?

- Lightweight
- Easy setup
- Powerful

Website:

https://fullcalendar.io/

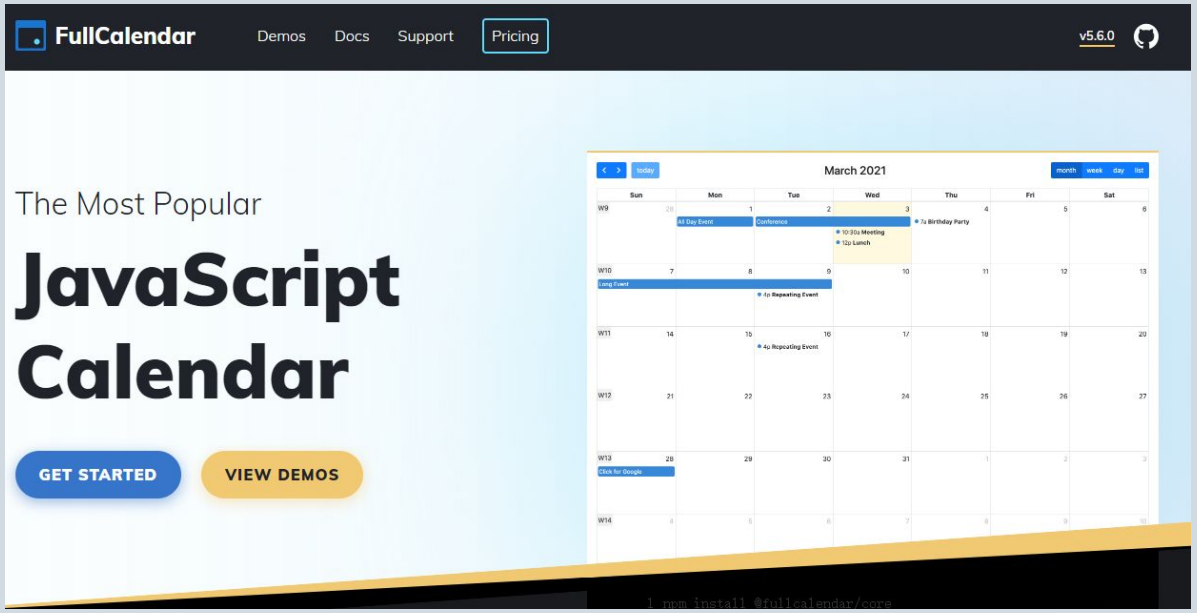


Chart.js

What is Chart?

- Open source JavaScript chart library based on HTML5.

Functionality

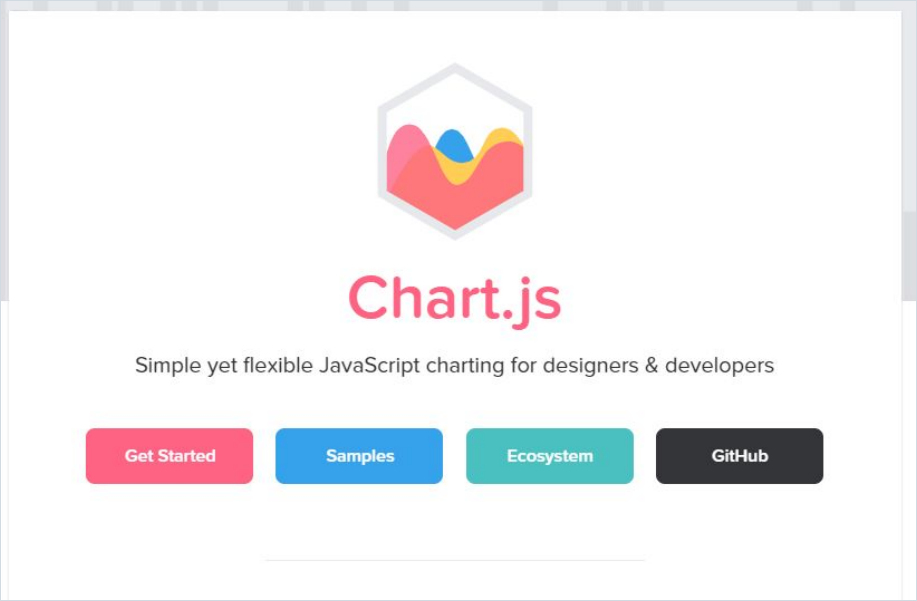
- Generate the graph or chart to visualize the course data.

Why?

- Free to use
- Based on HTML5 (Canvas Element)
- Easy to use and lightweight

Website:

https://www.chartjs.org/



ThinkPHP

What is ThinkPHP?

- Free and open source, fast and simple PHP development web framework

Functionality

- Quickly developing the admin web system

Why?

- Used Experience
- MVC pattern
- Lightweight

GitHub:

https://github.com/top-think

Back-end

Version control
- GitHub

Pipeline/server automation
- Jenkins

Data parsing
- Python (Beautiful Soup)

Data storage
- MariaDB (MySQL)

Database management
- PhpMyAdmin

Version Control (GitHub)

Why using version control?

- Tracking and managing changes to the source code.

Why GitHub?

- Most popular open source community.

Pipeline/server automation (Jenkins)

Why using pipeline automation?

- Faster development of new products and easier maintenance of existing deployments.

Why Jenkins?

- Easier for developers to integrate changes to the project, and easier for users to obtain a fresh build.

Data parsing (Python - Beautiful Soup)

Why data parsing?

- To get text from websites and store in JSON.

Why using Python?

- A large number of third-party libraries.

Data storage (MariaDB)

Why using database?

- No database API support

Why MariaDB?

- More flexible than local JSON files.
- Complex queries needed.

Database management (PhpMyAdmin)

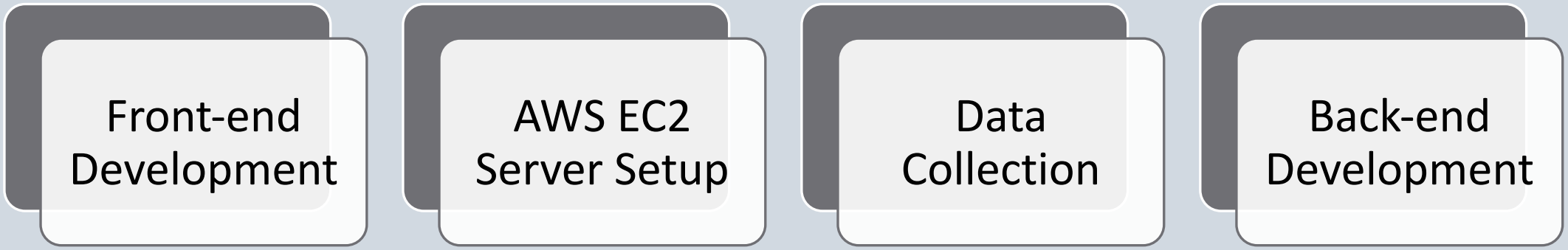
Why using database management?

- Faster and visible than query commands.

Why PhpMyAdmin?

- A free software tool in PHP to handle MySQL or MariaDB database server.

Project Roadmap



Front-end Development

Implementing structural design

- HTML
- CSS
- JS

Research and configure settings of open source software.

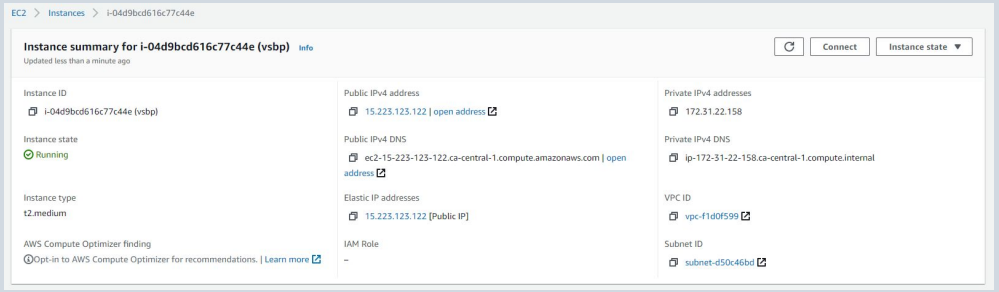
- Fullcalendar
- Char.js

AWS EC2 Server Setup

Select and start server instance.

Install MariaDB and PhpMyAdmin.

Install and configure Jenkins. Start Jenkins pipeline job.



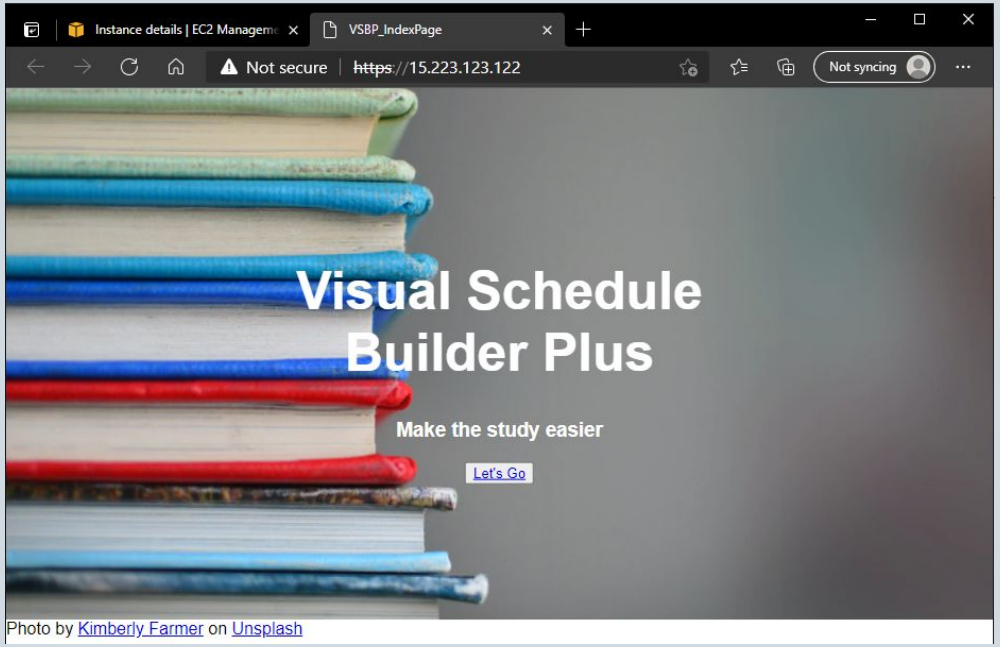


Photo by Kimberly Farmer on Unsplash

EC2 Instance Startup

Select and start t2.medium instance type.

Installed and configured

- emacs text editor - emacs-nox
- Apache HTTP server - httpd
- php7.2
- php connector - lamp-mariadb10.2-php7.2
- Turn on PHP errors on browser

v) Install the latest version

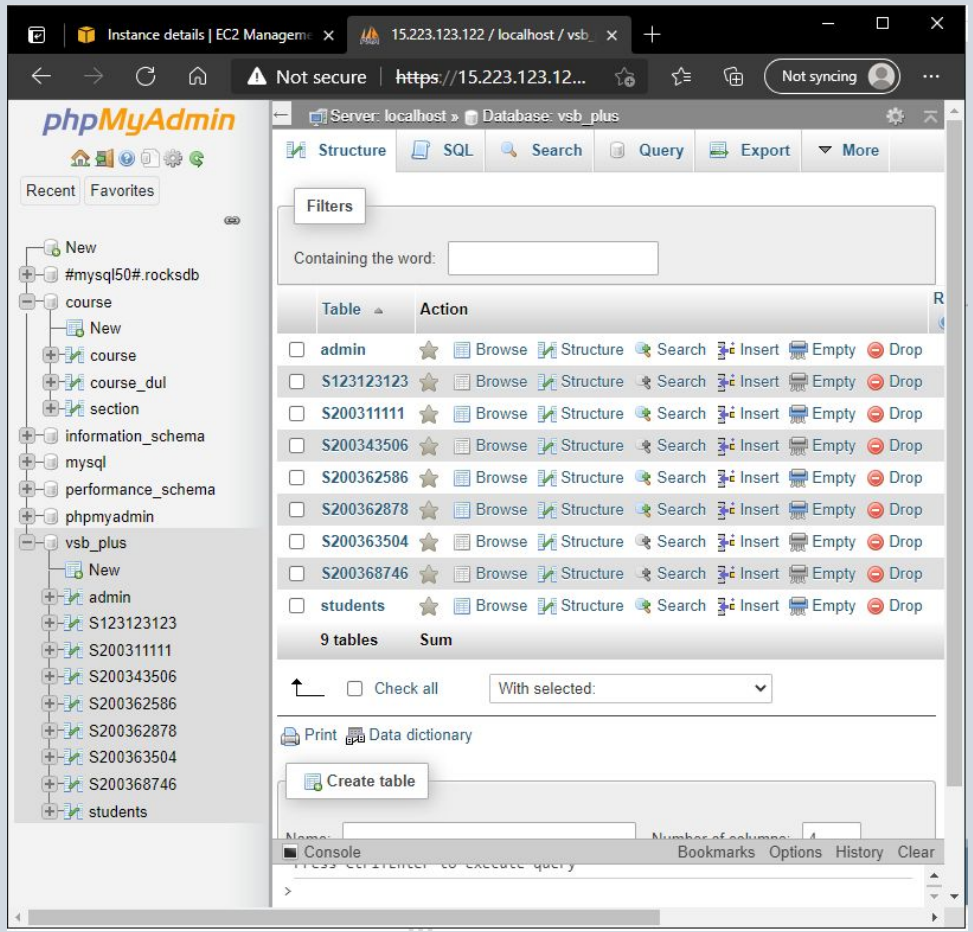
yum install amazon-linux-extras Install amazon-linux-extras

amazon-linux-extras Check latest version avaliable

amazon-linux-extras install lamp-mariadb10.2-php7.2 php7.2

lamp-mariadb10.2-php7.2: This is a php conector that allows php to connect to mariadb which is a mysql database.

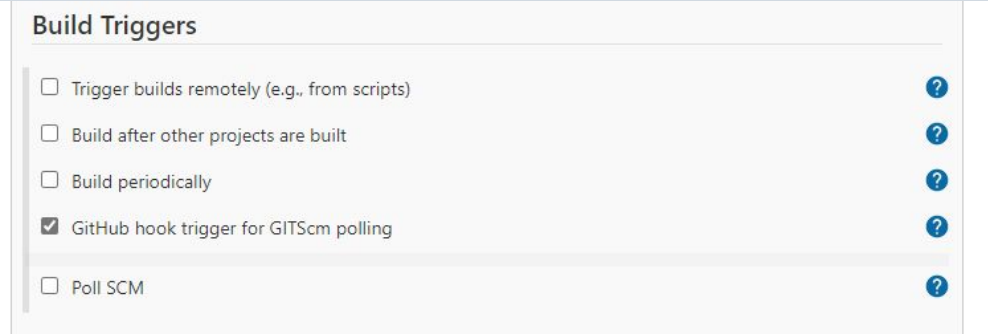
php7.2: The latest version of php

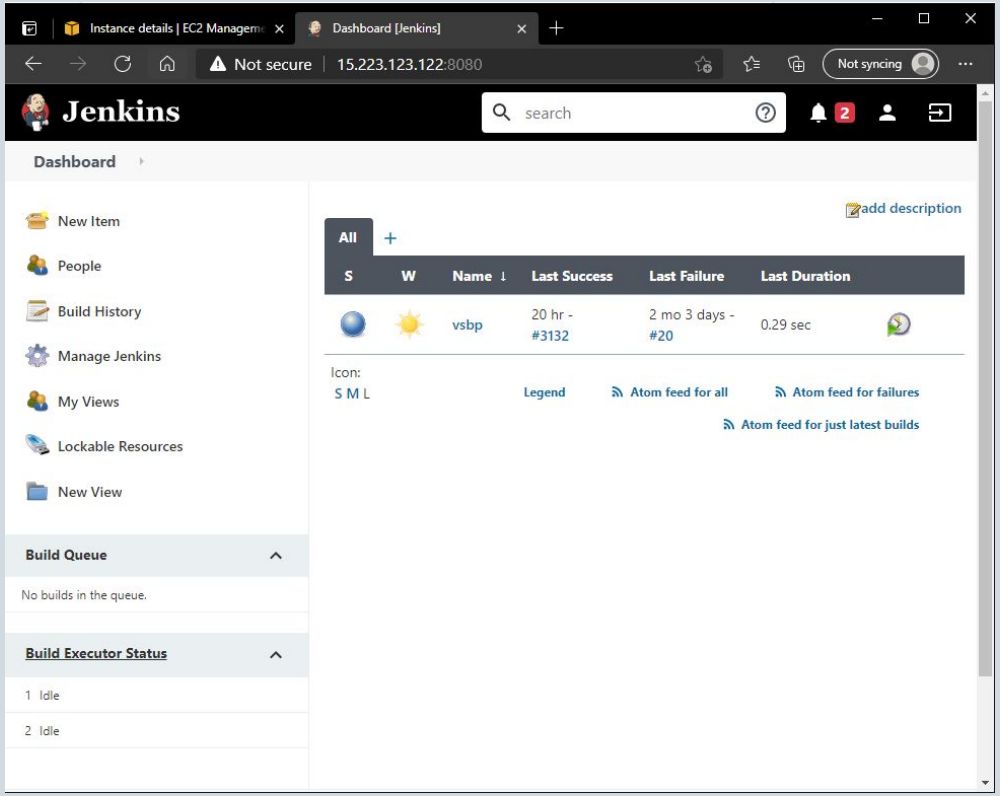


MariaDB & PhpMyAdmin Setup

Installed and configured

- MariaDB - mariadb-server
- phpMyAdmin dependency - php-mbstring
- phpMyAdmin - phpMyAdmin-5.0.2-english.tar.gz





Jenkins pipeline job

Installed and configured

- java 1.8 Oracle's JDK - java-1.8.0-openjdk-devel
- Jenkins redhat stable version - jenkins.repo

Created and configured Jenkins job

- GitHub Project
- Source Code Management - Git
- Build Triggers - GitHub hook trigger

Course Data Collection

Python

- Request library - Get the specific web page
- Beautifulsoup library - Selected the HTML elements
- Regular Expression - Match course info and store as list
- Export to json file

```python
try:
    kv = {'user-agent': 'Mozilla/5.0'}
    r = requests.get(url, timeout=30, headers=kv, params = params)
    r.raise_for_status()
    r.encoding = r.apparent_encoding
    return r.text
except:
    return "Connection error"
```


```
"term": "2020 Fall",
"short_name": "CS 110",
"title": "Programming and Problem Solving",
"faculty": "CS",
"credit": "3",
"description": "An introduction to problem-solving techniques, the fundamental concepts of programming, and th
"prerequisite": "Precalculus 30, Calculus 30, or Math 102",
"section": [{
    "course_ID": "30585",
    "section_num": "001",
    "instructors": "Xue-Dong  Yang (P)",
    "days": "MWF",
    "time": "10:30 am - 11:20 am",
    "loc": "Classroom Building 126",
    "course_type": "Lecture",
    "exam_days": "S",
    "exam_date": "Dec 12, 2020 - Dec 12, 2020",
    "exam_time": "2:00 pm - 5:00 pm"
```

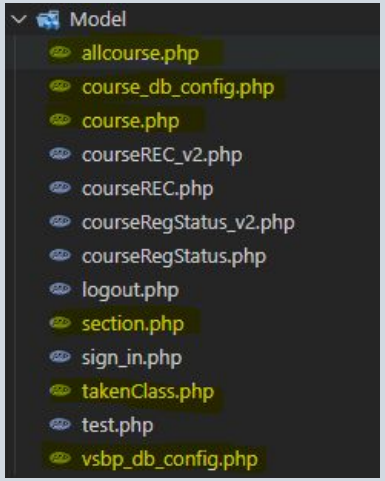
Back-end Development

Course/Student DB configuration

Course Prerequisites Pending

Course Recommendation

```php
<?php
/**
 * Connect to server database "course"
 *
 * Requirments:
 *  1) If connection is goes false, print the error
 *
 * php Steps:
 *  1) define server
 *  2) make connection
 *  3) If connection is false, print the error
 *
 * @version     1.0
 * @link        http://15.223.123.122/vsbp/Code/Model/section.php
 * @link        http://15.223.123.122/vsbp/Code/Model/course.php
 * @link        http://15.223.123.122/vsbp/Code/Model/allcourse.php
 * @author      Jingkang Yang (sid: 200362586) <yang242j@uregina.ca>
 */

// define('DB_SERVER', 'localhost');
define('DB_SERVER', '127.0.0.1');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'vsbp');
define('DB_NAME', 'course');

/* Attempt to connect to MySQL database */
$conn = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

```



Course/Student DB config

Database configuration files

- course_db_config
- vsbp_db_config

Fetch default JSON data

- all the course names - allcourse.php
- one course general info - course.php
- one course section info - section.php
- all course history of one student - takenClass.php

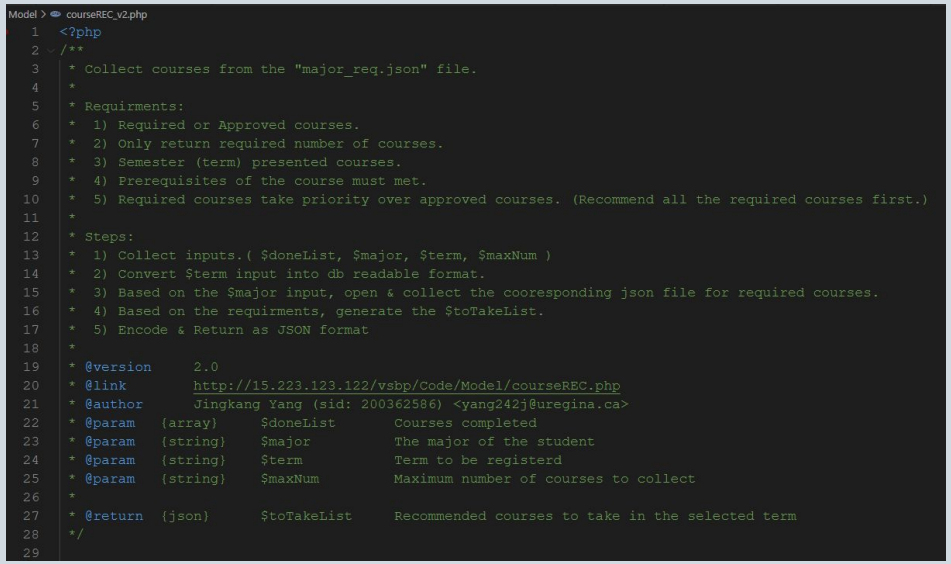
CS	115	3	CS 110 with a minimum grade of 65% and one of MATH...	CS 110 && (MATH 110 || MATH 103)	CS 110 [>= 65] && (MATH 110 || MATH 103 [>= 80])

```php
/**
 * @param   {string}    $expStr       The preg expression of the course prerequisites.
 * @param   {array}     $doneList     A list contains the short name of all completed courses.
 * @return  {boolean}                 True for matched, False for not.
 */
function exp_matched($expStr, $totalCredit, $doneList)
{
    $expStr = trim($expStr);

    // Basic: if expStr == null, return true
    if ($expStr == null) {
        return true;
    }

    // Base case: Pending credit requirements
    if (preg_match_all("/^(Credit\s\[(.*?)\])$/i", $expStr) == 1) {
        $creditStr = preg_split("/(\s\[)/i", $expStr);
        $creditExp = rtrim($creditStr[1], ']');
        return eval('return ' . $totalCredit . $creditExp . ';') ? true : false;
    }

    // Basic: exact one course name "ENSE 400"
    if (preg_match_all("/([a-z]+\s[0-9]+)/i", $expStr) == 1) {
        // Check if has condition
        if (preg_match_all("/([a-z]+\s[0-9]+\s\[(.*?)\])/i", $expStr) == 1) {
            $splitedStr = preg_split("/(\s\[)/i", $expStr);
            if (array_key_exists($splitedStr[0], $doneList)) {
                $gradeExp = rtrim($splitedStr[1], ']');
                return eval('return ' . $doneList[$splitedStr[0]] . $gradeExp . ';') ? true : false;
            } else { return false; }
        }
        return array_key_exists($expStr, $doneList) ? true : false;
    }

    // &&: split "ENSE 400 && ENEL 400"
    $andComp = preg_split("/(&{2})/", $expStr);
    if (sizeof($andComp) > 1) {
```

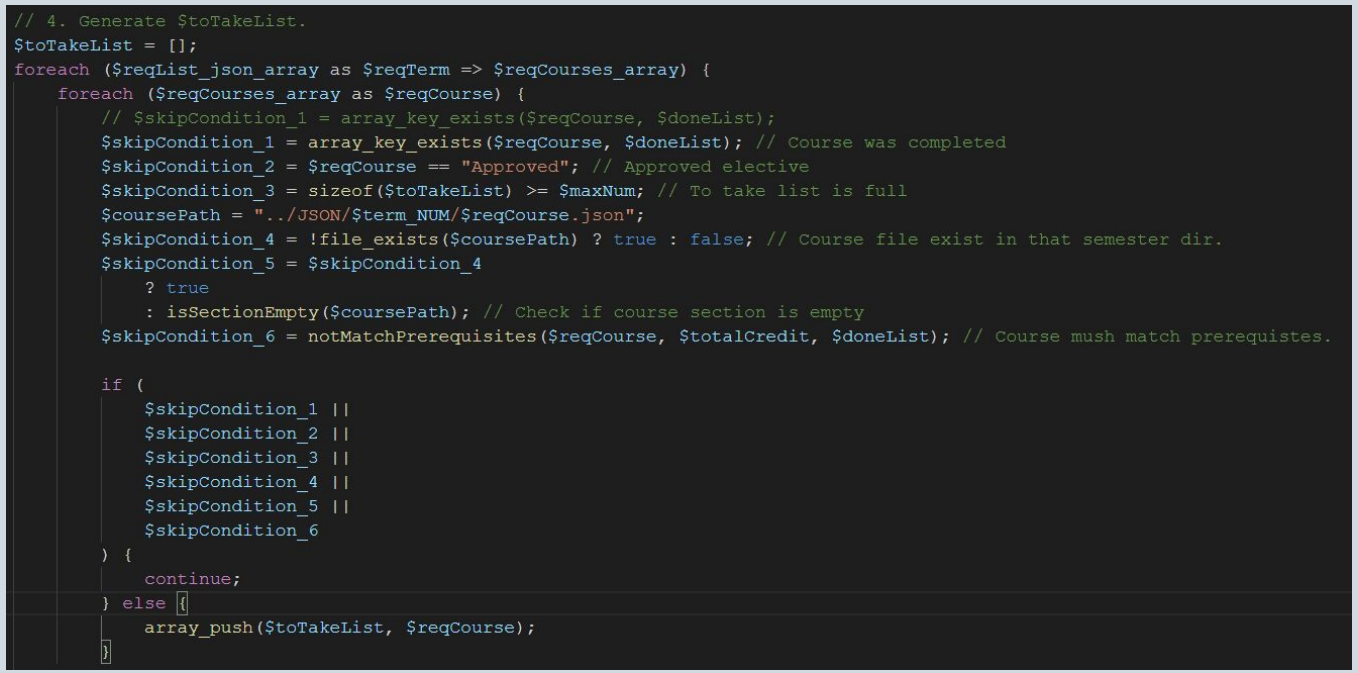
Course Prerequisites Pending

Pending if the student's course history matches the prerequisites of the selected course.

- Course prerequisites sentences => RegEx string
- Compare course history with the RegEx string.
- Return TRUE/FALSE



```
Model > courseREC_v2.php
<?php
/**
 * Collect courses from the "major_req.json" file.
 *
 * Requirments:
 *  1) Required or Approved courses.
 *  2) Only return required number of courses.
 *  3) Semester (term) presented courses.
 *  4) Prerequisites of the course must met.
 *  5) Required courses take priority over approved courses. (Recommend all the required courses first.)
 *
 * Steps:
 *  1) Collect inputs.( $doneList, $major, $term, $maxNum )
 *  2) Convert $term input into db readable format.
 *  3) Based on the $major input, open & collect the cooresponding json file for required courses.
 *  4) Based on the requirments, generate the $toTakeList.
 *  5) Encode & Return as JSON format
 *
 * @version     2.0
 * @link        http://15.223.123.122/vsbp/Code/Model/courseREC.php
 * @author      Jingkang Yang (sid: 200362586) <yang242j@uregina.ca>
 * @param   {array}     $doneList       Courses completed
 * @param   {string}    $major          The major of the student
 * @param   {string}    $term           Term to be registerd
 * @param   {string}    $maxNum         Maximum number of courses to collect
 *
 * @return  {json}      $toTakeList     Recommended courses to take in the selected term
 */

```



```
// 4. Generate $toTakeList.
$toTakeList = [];
foreach ($reqList_json_array as $reqTerm => $reqCourses_array) {
    foreach ($reqCourses_array as $reqCourse) {
        // $skipCondition_1 = array_key_exists($reqCourse, $doneList);
        $skipCondition_1 = array_key_exists($reqCourse, $doneList); // Course was completed
        $skipCondition_2 = $reqCourse == "Approved"; // Approved elective
        $skipCondition_3 = sizeof($toTakeList) >= $maxNum; // To take list is full
        $coursePath = "../JSON/$term_NUM/$reqCourse.json";
        $skipCondition_4 = !file_exists($coursePath) ? true : false; // Course file exist in that semester dir.
        $skipCondition_5 = $skipCondition_4
            ? true
            : isSectionEmpty($coursePath); // Check if course section is empty
        $skipCondition_6 = notMatchPrerequisites($reqCourse, $totalCredit, $doneList); // Course mush match prerequistes.

        if (
            $skipCondition_1 ||
            $skipCondition_2 ||
            $skipCondition_3 ||
            $skipCondition_4 ||
            $skipCondition_5 ||
            $skipCondition_6
        ) {
            continue;
        } else {
            array_push($toTakeList, $reqCourse);
        }
```

Course Recommendation

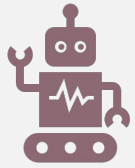
Recommend course standards

- Course NOT completed
- Required course takes priority
- Recommend top 10 in the list
- Course available on selected semester
- Prerequisites must match.
- Return courses list.

Overall Experience

Software Development Cycle

Compromises



Technologies