|  |
| --- |
| Software Engineering Principles |
| ABET Accreditation Web App |
| Final Deliverables |

|  |
| --- |
| Patrick Brannen and Ryan Tanner  12-5-2014 |

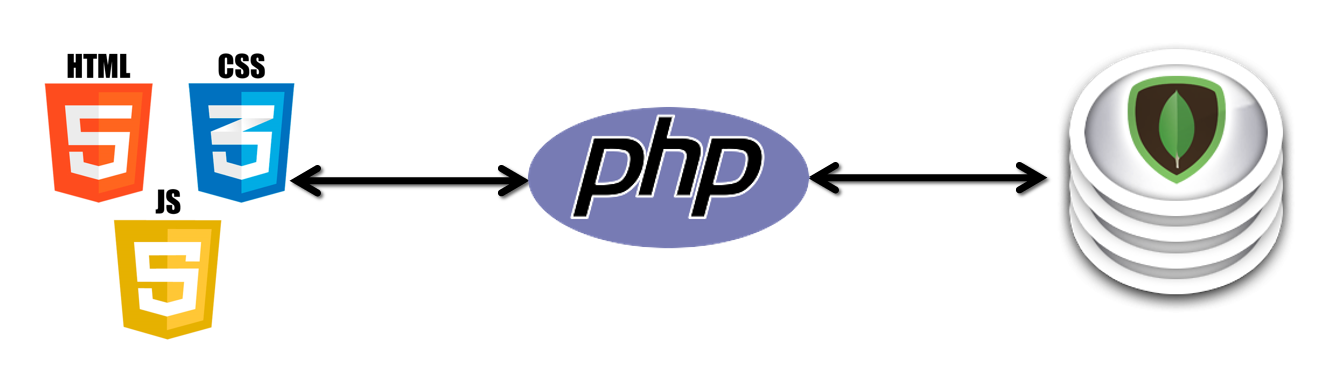
Configuration: Requires MongoDB with database named ABET and PHP with mongodb driver (<http://php.net/manual/en/mongo.installation.php>)

To load database: run mongo, use ABET, load(“ABETdatabase.js”)

URL: patrickbrannen.com/ABET

Usernames and passwords: test, test; admin, password

Repo location: https://github.com/brannp66/SEP-ABET



Requirements:

1. The software shall use PHP as its server side language.

Rationale: PHP has good libraries for supporting MongoDB.

1. The software shall use MongoDB as the database.
   1. Rationale: This was a requirement by the customer.
2. The software shall use HTML, CSS, and Javascript for the front end user interface.
   1. Rationale: These are the standard Web interface design languages.
3. The software shall use JSON for data transfer.
   1. Rationale: JSON is good to use with Javascript and dynamically generated pages.
4. The software shall communicate with the server via HTTP connections.
   1. Rationale: No sensitive data is being exchanged so security is not an issue.
5. The project shall be backed up with git as the source control.
   1. Rationale: Git is a very good source control system and works well with this project.
6. The software (database) shall be non relational.
   1. Rationale: The product will possibly need to be altered later and using a non-

relational database will work better to accommodate the alterations.

1. The software will support both CAC  and EAC outcomes.
   1. Rationale: The customer has requested both be supported.
2. Instructors will have accounts that they must log in to use the system
   1. Rationale: Users will need to keep track of their info and instructors will need to save

and reload their evaluations.

1. The software shall support multiple outcomes for a single course.
   1. Rationale: Courses monitor more than one outcome so the software will too.
2. The software shall allow access by instructors via the web.
   1. Rationale: A web access is the simplest way for instructors to access the forms and

data supplied by the software and will make using the software easy.

1. The software shall allow instructors to specify their basis for evaluation
   1. Rationale: The instructors should be able to change the basis for evaluation as their

criteria changes.

1. The software shall allow instructors to input the number of students that meet a specified criteria
   1. Rationale: The statistics generated by the software are based on this input so it is

necessary for the software.

1. The software will allow instructors to fill out ABET forms
   1. Rationale: This is the basis for the software and is the core functionality required.
2. The software will take student evaluations and create statistics based off of the input.
   1. Rationale: This is another core component to the software and supplies the

information required by the instructors.

1. The software will generate user input sheets based off of ABET outcomes.
   1. Rationale: These generated sheets are what will be used by instructors to track their

class statistics.

1. The software will display students enrolled in the course.
   1. Rationale: The instructor should be able to see all the students enrolled in their

course.

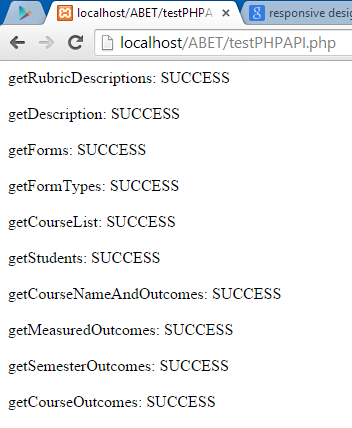
1. The software will calculate percentages of students meeting the requirements based on instructor input
   1. Rationale: These are a main part of the statistics needed by the instructors from the

software.

1. The software will separate students by major and associate students with the appropriate CAC and EAC outcomes.
   1. Rationale: Computer Engineering Students use EAC Outcomes while computer

science student use CAC, however they may be in the same classes

Testing: Our tests were run using the testPHPAPI.php file. These are the results:



chooseSemester.php

<?php

session\_start();

include 'phpAPI.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title>ABET Accredidation</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/foundation.css" />

<script src="js/vendor/modernizr.js"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">

</script>

<script>

function loadCourses(str) {

var instructor = <?php echo(json\_encode($\_SESSION['instructor'])); ?>;

//leave options blank if nothing selected

if(str=="") {

document.getElementById("courses").innerHTML = "";

return;

}

else

{

//default xmlhttp request

if (window.XMLHttpRequest)

{

xmlhttp = new XMLHttpRequest();

}

//http request for old ie

else

{

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("courses").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET","getCourses.php?s="+str+"&i="+instructor,true);

xmlhttp.send();

}

}

function toggleButton(str) {

if(str==""){

document.getElementById('btn').disabled=true;

}

else{

document.getElementById('btn').disabled=false;

}

}

</script>

</head>

<body>

<div class='choose'>

<h3>Select Semester and Course</h3>

<form action='forms.php' method="POST">

<div class='dropdown'>

<select name = 'semester' onchange="loadCourses(this.value)">

<option value="" disabled selected style='display:none;'>

Select Semester

</option>

<option value="Fall2014">Fall 2014</option>

<option value="Spring2015">Spring 2015</option>

<option value="Fall2015">Fall 2015</option>

<option value="Spring2016">Spring 2016</option>

<option value="Fall2016">Fall 2016</option>

<option value="Spring2017">Spring 2017</option>

<option value="Fall2017">Fall 2017</option>

</select>

</div>

<div class='dropdown' id='courses'>

</div>

<input type='submit' value='View Forms' class='button tiny' id='btn' disabled>

</form>

</div>

</body>

</html>

form.php

<?php

session\_start();

include 'phpAPI.php';

$form = json\_decode($\_POST['form']);

$\_SESSION['semester'] = $form->{'semester'};

$\_SESSION['courseId'] = $form->{'courseId'};

$\_SESSION['instructor'] = $form->{'instructor'};

$\_SESSION['CACOutcome'] = $form->{'CACOutcome'};

$\_SESSION['EACOutcome'] = $form->{'EACOutcome'};

$\_SESSION['numRubrics'] = count($form->{'rubrics'});

?>

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Outcome Sheet</title>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js">

</script>

<script src="js/generateForm.js"></script>

<link rel="stylesheet" href="css/foundation.css" />

<link rel="stylesheet" href="css/style.css" />

<script src="js/vendor/modernizr.js"></script>

</head>

<body>

<div class='form' id='form'>

</div>

<script type='text/javascript'>

generateForm(<?php echo $\_POST['form'] ?>);

</script>

</body>

</html>

forms.php

<?php

session\_start();

error\_reporting(0);

?>

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Outcome Sheet</title>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js">

</script>

<script src="js/generateForm.js"></script>

<link rel="stylesheet" href="css/foundation.css" />

<link rel="stylesheet" href="css/style.css" />

<script src="js/vendor/modernizr.js"></script>

</head>

<body>

<div class='choose'>

<h3>Choose Outcome</h3>

<?php

include 'phpAPI.php';

$semester = $\_POST['semester'];

$courseId = $\_POST['course'];

$instructor = $\_SESSION['instructor'];

$forms = getForms($instructor, $semester, $courseId);

foreach($forms as $form) {

$jsonForm = json\_encode($form);

echo "<form method='POST' action='form.php'>";

echo "<input type='hidden' name='form' value='".$jsonForm."'>";

if($form->getCACOutcome()) {

echo " CAC-" . $form->getCACOutcome() . " ";

}

if($form->getEACOutcome()) {

echo " EAC-" . $form->getEACOutcome() . " ";

}

echo "<input type='submit' value='Generate' class='button tiny'>";

echo "</form>";

}

?>

</div>

</body>

</html>

admin.php

<?php

session\_start();

include 'phpAPI.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title>ABET Accredidation</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/foundation.css" />

<script src="js/vendor/modernizr.js"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">

</script>

<script>

function getOutcomes(str) {

//leave options blank if nothing selected

if(str=="") {

document.getElementById("OutcomeForms").innerHTML = "";

return;

}

else

{

//default xmlhttp request

if (window.XMLHttpRequest)

{

xmlhttp = new XMLHttpRequest();

}

//http request for old ie

else

{

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("OutcomeForms").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET","getCycle.php?s="+str,true);

xmlhttp.send();

}

}

</script>

</head>

<body>

<form action='stats.php' method = 'POST' id='form'>

<div class='choose'>

<h1>Choose Outcome</h1>

<div class='dropdown'>

<select name = 'semester' onchange="getOutcomes(this.value)">

<option value="" disabled selected style='display:none;'>

Select Semester

</option>

<option value="Fall2014">Fall 2014</option>

<option value="Spring2015">Spring 2015</option>

<option value="Fall2015">Fall 2015</option>

<option value="Spring2016">Spring 2016</option>

<option value="Fall2016">Fall 2016</option>

<option value="Spring2017">Spring 2017</option>

</select>

</div>

<div id = "OutcomeForms">

</div>

</div>

</form>

</body>

</html>

generateForm.js

function generateForm(form) {

var formDiv = document.getElementById("form");

var headerDiv = getHeaderDiv(form);

formDiv.appendChild(headerDiv);

var formTag = document.createElement("form");

formTag.setAttribute("action", "saveForm.php");

formTag.setAttribute("method", "POST");

if(form['studentsCAC']) {

var tableDiv = getCACTableDiv(form);

formTag.appendChild(tableDiv);

}

if(form['studentsEAC']) {

var tableDiv = getEACTableDiv(form);

formTag.appendChild(tableDiv);

}

var basisDiv = getBasisDiv();

formTag.appendChild(basisDiv);

var submitButton = document.createElement("INPUT");

submitButton.setAttribute("type", "submit");

submitButton.setAttribute("value", "Submit");

submitButton.setAttribute("class", "button tiny")

formTag.appendChild(submitButton);

formDiv.appendChild(formTag);

}

var getHeaderDiv = function(form) {

var header = document.createElement("div");

var courseElement = document.createElement("h3");

var semesterElement = document.createElement("h3");

var instructorElement = document.createElement("h5");

var outcomeElement = document.createElement("h5");

courseElement.innerHTML = form['courseId'].toUpperCase();

courseElement.setAttribute("id","course");

semesterElement.innerHTML = form['semester'];

semesterElement.setAttribute("id","semester");

instructorElement.innerHTML = "Instructor: " + form['instructor'];

instructorElement.setAttribute("id","instructor");

instructorElement.setAttribute("style","text-transform:capitalize");

outcomeElement.innerHTML = "Outcome: <strong>" + form['description'] +

"</strong>";

outcomeElement.setAttribute("id", "outcome");

header.appendChild(courseElement);

header.appendChild(semesterElement);

header.appendChild(instructorElement);

header.appendChild(outcomeElement);

return header;

}

var getCACTableDiv = function(form) {

var tableDiv = document.createElement("div");

var major = document.createElement("h4");

var table = getTable(form['rubrics'],"CAC");

var studentList = getStudentListElement(form['studentsCAC']);

major.innerHTML = "Computer Science Undergrads";

tableDiv.appendChild(major);

tableDiv.appendChild(table);

tableDiv.appendChild(studentList);

return tableDiv;

}

var getEACTableDiv = function(form) {

var tableDiv = document.createElement("div");

var major = document.createElement("h4");

var table = getTable(form['rubrics'], "EAC");

var studentList = getStudentListElement(form['studentsEAC']);

major.innerHTML = "Computer Engineering Undergrads";

tableDiv.appendChild(major);

tableDiv.appendChild(table);

tableDiv.appendChild(studentList);

return tableDiv;

}

var getTable = function(rubrics, type) {

var table = document.createElement("table");

table.appendChild(getTableHeader());

for(var i = 0; i < rubrics.length; i++) {

var row = document.createElement("tr");

var cell = document.createElement("td");

cell.innerHTML = rubrics[i];

row.appendChild(cell);

for(var j = 0; j < 4; j++) {

cell = document.createElement("td");

cell.innerHTML = "<input type=\"number\" min=\"0\" max=\"100\" value=\"0\"

name=\"" + type + "-" + i + "-" + j + "\">";

row.appendChild(cell);

}

table.appendChild(row);

}

return table;

}

var getTableHeader = function() {

var row = document.createElement("tr");

var data = document.createElement("th");

row.appendChild(data);

var data = document.createElement("th");

data.appendChild(document.createTextNode('1 (weak)'));

row.appendChild(data);

var data = document.createElement("th");

data.appendChild(document.createTextNode('2 (poor)'));

row.appendChild(data);

var data = document.createElement("th");

data.appendChild(document.createTextNode('3 (good)'));

row.appendChild(data);

var data = document.createElement("th");

data.appendChild(document.createTextNode('4 (excellent)'));

row.appendChild(data);

return row;

}

var getStudentListElement = function(studentList) {

var listElement = document.createElement("h5");

listElement.innerHTML =" Students: ";

for(var i = 0; i < studentList.length; i++) {

listElement.innerHTML += studentList[i] + ", ";

}

//slice off trailing comma

listElement.innerHTML = listElement.innerHTML.slice(0,-2);

return listElement;

}

var getBasisDiv = function () {

var basis = document.createElement("div");

var label = document.createElement("h4");

var input = document.createElement("textarea");

input.setAttribute("name", "basedOn");

input.setAttribute("id", "basis");

var note = document.createElement("h5");

label.innerHTML = "The above evaluation is based on:";

note.innerHTML = "This can be a project, an assignment or homework." +

"Please list above - or provide reference to the basis of" +

"evaluation (e.g. Assignment 2, Exam - question 5, etc.)";

basis.appendChild(label);

basis.appendChild(input);

basis.appendChild(note);

return basis;

}

getCourses.php

<?php

include 'phpAPI.php';

$semester = $\_GET['s'];

$instructor = $\_GET['i'];

$courseList = getCourseList($instructor, $semester);

echo "<select name = 'course'onchange='toggleButton(this.value)'>

<option value='' disabled selected style='display:none;'>

Select Course</option>";

foreach($courseList as $course){

echo sprintf('<option value="%s">%s</option>', $course['course'],

$course['course']);

}

echo "</select>";

?>

login.css

.login {

position: relative;

margin: 0 auto;

padding: 20px 20px 20px;

width: 310px;

background: white;

border-radius:3px;

box-shadow: 0 0 200px rgba(255,255,255,0.5), 0 1px 2px rgba(0,0,0,0.3);

}

.choose {

position: relative;

margin: 0 auto;

padding: 20px 20px 20px;

width: 310px;

background: white;

border-radius:3px;

box-shadow: 0 0 200px rgba(255,255,255,0.5), 0 1px 2px rgba(0,0,0,0.3);

}

style.css

html {

background-color:#fff;

}

#upArrow {

float: left;

width:20px;

height:15px;

}

#downArrow {

float: left;

width:20px;

height:15px;

}

.login {

position: relative;

margin: 0 auto;

padding: 20px 20px 20px;

width: 310px;

background: white;

border-radius:3px;

box-shadow: 0 0 200px rgba(255,255,255,0.5), 0 1px 2px rgba(0,0,0,0.3);

}

.choose {

position: relative;

margin: 0 auto;

padding: 20px 20px 20px;

width: 310px;

background: white;

border-radius:3px;

box-shadow: 0 0 200px rgba(255,255,255,0.5), 0 1px 2px rgba(0,0,0,0.3);

}

.form {

position: relative;

margin: 0 auto;

padding: 20px 20px 20px;

width: 750px;

background: white;

border-radius:3px;

box-shadow: 0 0 200px rgba(255,255,255,0.5), 0 1px 2px rgba(0,0,0,0.3);

}

.hidden {

visibility:hidden;

}

#course {

float:left;

}

#semester {

float:right;

}

#instructor {

clear:left;

}

#outcome {

clear:both;

}

#basis {

}

h3 {

font-size: 18px;

}

h4 {

font-weight: bold;

font-size: 16px;

}

h5 {

font-size: 14px;

}

h6 {

font-weight: bold;

font-size: 14px;

}

getCycle.php

<?php

include 'phpAPI.php';

$semester = $\_GET['s'];

$outcomeList = getSemesterOutcomes($semester);

//print\_r($outcomeList['CACOutcomes']);

echo "<h3>CAC Outcomes</h3>";

foreach($outcomeList['CACOutcomes'] as $outcome) {

echo "<button type='submit' value='CAC-" . $outcome . "' name='form'

class='btn'>CAC-" . $outcome . "</button>";

}

echo "<h3>EAC Outcomes</h3>";

foreach($outcomeList['EACOutcomes'] as $outcome) {

echo "<button type='submit' value='EAC-" . $outcome . "' name='form'

class='btn'>EAC-" . $outcome . "</button>";

}

?>

index.php

<?php

session\_start();

if (isset($\_POST['submit'])) {

if (!$\_POST['username'] | !$\_POST['password']) {

echo "<script type='text/javascript'>alert('You did not complete all the

required fields');</script>";

}

$m = new MongoClient(); //connect

$db = $m->selectDB("ABET");

$users = new MongoCollection($db, 'Users');

$conditions = array('$and' => array(array('name' => $\_POST['username']),

array('password' => $\_POST['password'])));

if($conditions.count() === 1) {

if($\_POST['username'] == "admin") {

header("Location: admin.php");

}

else {

header("Location: chooseSemester.php");

}

}

else{

echo "<script type='text/javascript'>alert('Incorect Credentials');

</script>";

}

if($\_POST['username'] == "admin") {

header("Location: admin.php");

}

else {

$\_SESSION['instructor'] = $\_POST['username'];

header("Location: chooseSemester.php");

}

}

?>

<!doctype html>

<html>

<head>

<title>ABET Accredidation</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/login.css">

<link rel="stylesheet" href="css/foundation.css" />

<script src="js/vendor/modernizr.js"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">

</script>

</head>

<body>

<div class='login'>

<h1>Login</h1>

<form method="POST" action="index.php">

<p>

<input type="text" name="username" value placeholder="Username">

</p>

<p>

<input type="password" name="password" value placeholder="Password">

</p>

<input type="submit" value="Login" name='submit' class='button'>

</form>

</div>

</body>

</html>

phpAPI.php

<?php

class Form {

public function Form($instructor, $semester, $courseId, $courseName,

$studentsCAC, $studentsEAC, $rubrics, $description,

$CACOutcome, $EACOutcome) {

$this->instructor = $instructor;

$this->semester = $semester;

$this->courseId = $courseId;

$this->courseName = $courseName;

$this->studentsCAC = $studentsCAC;

$this->studentsEAC = $studentsEAC;

$this->rubrics = $rubrics;

$this->description = $description;

$this->CACOutcome = $CACOutcome;

$this->EACOutcome = $EACOutcome;

}

public function setInstructor($instructor) {

$this->instructor = $instuctor;

}

public function setSemester($semester) {

$this->semester = $semester;

}

public function setCourseId($courseId) {

$this->courseId = $courseId;

}

public function setCourseName($courseName) {

$this->courseName = $courseName;

}

public function setStudentsCAC($studentsCAC) {

$this->studentsCAC = $studentsCAC;

}

public function setStudentsEAC($studentsEAC) {

$this->studentsEAC = $studentsEAC;

}

public function setRubrics($rubrics) {

$this->rubrics = $rubrics;

}

public function setDescription($description) {

$this->description = $description;

}

public function getInstructor() {

return $this->instructor;

}

public function getSemester() {

return $this->semester;

}

public function getCourseID() {

return $this->courseId;

}

public function getCourseName() {

return $this->courseName;

}

public function getStudentsCAC() {

return $this->studentsCAC;

}

public function getStudentsEAC() {

return $this->studentsEAC;

}

public function getRubrics() {

return $this->rubrics;

}

public function getDescription() {

return $this->description;

}

public function getCACOutcome() {

return $this->CACOutcome;

}

public function getEACOutcome() {

return $this->EACOutcome;

}

}

/\*

\* gets statistics based on EAC or CAC and the letter

\*

\* $letter = letter of form

\* $type = CAC or EAC

\* $semester = semester the class was in

\*/

function getStats($letter, $type, $semester){

$m = new MongoClient(); //connect to mongo

$db = $m->selectDB("ABET"); //select the database

$allResults = new MongoCollection($db, 'Results');

$type = strtoupper($type);

$letter = strtoupper($letter);

$type = $type . 'Outcome'; //set the outcome type

$conditions = array('$and' => array(array($type => $letter),

array('semester' => $semester)));

//set the conditions for finding the statistics for the outcome

$results = $allResults->find($conditions);

$outcomes = array();

foreach ($results as $result){

array\_push($outcomes, $result);

}

return $outcomes;

}

/\*

\* get the description of a specific rubric

\*

\*

\* $letter = letter of form

\* $type = CAC or EAC

\*

\*/

function getRubricDescriptions($letter, $type){

$m = new MongoClient(); //connect

$db = $m->selectDB("ABET");

$outcomes = new MongoCollection($db, 'EACandCACMatchups');

$conditions = array($type => $letter);

$result = $outcomes->findOne($conditions);

return $result['rubrics'];

}

/\*

\* gets the description of the outcome

\*

\* $letter = letter of form

\* $type = CAC or EAC

\*

\*

\*/

function getDescription($letter, $type) {

$m = new MongoClient(); //connect

$db = $m->selectDB("ABET");

$outcomes = new MongoCollection($db, 'Outcomes');

$conditions = array('$and' => array(array('letter' => $letter),

array('type' => $type)));

$result = $outcomes->findOne($conditions);

return $result['description'];

}

/\*

\* Returns an array with an entry for each form that can be generated fpr a class

\* Each entry contains the data for one form

\*

\* $instructor = the instructor for the class

\* $semester = the semester the class is being taught ex:Fall2014

\* $courseId = the id for a course ex:cse3342

\*

\*/

function getForms($instructor, $semester, $courseId) {

$forms = array();

// Get the course Name and Outcomes that it measures

$courseInfo = getCourseNameAndOutcomes($courseId);

$courseName = $courseInfo[0];

//Reduce the lists of outcomess to only the outcomes looked at in $semester

$outcomes = getMeasuredOutcomes($semester, $courseId);

$CACOutcomes = $outcomes[0];

$EACOutcomes = $outcomes[1];

//Get the students for the class

$students = getStudents($courseId, $instructor, $semester);

//Get the types of form

$formTypes = getFormTypes($CACOutcomes, $EACOutcomes);

//Create form objects

foreach($formTypes as $formType) {

$name = "";

//Get CAC data if needed

if($formType['CAC']) {

$studentsCAC = $students[0];

$description = getDescription($formType['CAC'], "CAC");

$CACOutcome = $formType['CAC'];

}

//If CAC is not needed EAC is

else {

$studentsCAC = NULL;

$CACOutcome = NULL;

$description = getDescription($formType['EAC'], "EAC");

}

//Get EAC data if needed

if($formType['EAC']) {

$studentsEAC = $students[1];

$EACOutcome = $formType['EAC'];

}

else {

$studentsEAC = NULL;

$EACOutcome = NULL;

}

//get the rubrics

$rubrics = $formType["rubrics"];

//create form object and add to arry

$form = new Form($instructor, $semester, $courseId, $courseName,

$studentsCAC, $studentsEAC, $rubrics, $description,

$CACOutcome, $EACOutcome);

array\_push($forms, $form);

}

return $forms;

}

/\*

\* Gets the types of forms that can be generated for a form

\* Matches CAC and EAC Outcomes to combine forms where possible

\*

\* $CACOutcomes = CAC outcomes that are being measured

\* $EACOutcomes = EAC ourttcomes that are being measured

\*

\*/

function getFormTypes($CACOutcomes, $EACOutcomes) {

$forms = array();

$m = new MongoClient(); //connect

$db = $m->selectDB("ABET");

$matches = new MongoCollection($db, 'EACandCACMatchups');

$results = $matches->find();

//Match EAC and CAC outcomes

foreach($results as $result) {

//If there is a match add both outcomes

if(in\_array($result['EAC'], $EACOutcomes) &&

in\_array($result['CAC'], $CACOutcomes)) {

$tempArray = array("CAC" => $result['CAC'], "EAC" => $result['EAC'],

"rubrics" => $result['rubrics']);

array\_push($forms, $tempArray);

}

//If there is no match add the CAC Outcome by it self

else if(in\_array($result['CAC'], $CACOutcomes)) {

$tempArray = array("CAC" => $result['CAC'], "rubrics" => $result['rubrics']);

array\_push($forms, $tempArray);

}

//If ther is no match add EAC outcome by it self

else if(in\_array($result['EAC'], $EACOutcomes)) {

$tempArray = array("EAC" => $result['EAC'], "rubrics" => $result['rubrics']);

array\_push($forms, $tempArray);

}

}

return $forms;

}

/\*

\*

\* gets the list of courses taught by a professor for a semester

\*

\* $instructor = the teacher of a class

\* $semester = the semester taught

\*/

function getCourseList($instructor, $semester){

$client = new MongoClient();

$db = $client->selectDB('ABET');

$allCourses = new MongoCollection($db, 'ClassRoster');

$conditions = array('$and' => array(array('semester' => $semester),

array('instructor' => $instructor)));

$results = $allCourses->find($conditions);

$courseList = array();

foreach($results as $result){

array\_push($courseList, $result);

}

return $courseList;

}

/\*

\* gets a list of all students in a class

\*

\* $courseID = the ID number of a course

\* $instructor = the teacher of a class

\* $semester = semester class taught

\*/

function getStudents($courseID, $instructor, $semester){

$client = new MongoClient();

$db = $client->selectDB('ABET');

$allCourses = new MongoCollection($db, 'ClassRoster');

$conditions = array('$and' => array(array('semester' => $semester),

array('instructor' => $instructor), array('course' => $courseID)));

$result = $allCourses->findOne($conditions);

$studentsCAC = $result['studentsCAC'];

$studentsEAC = $result['studentsEAC'];

$studentList = array($studentsCAC, $studentsEAC);

return $studentList;

}

/\*

\* get the name of a course and the outcomes for it

\*

\* $courseID = id of the course

\*

\*

\*

\*/

function getCourseNameAndOutcomes($courseID){

$client = new MongoClient();

$db = $client->selectDB('ABET');

$allCourseOutcomes = new MongoCollection($db, 'CourseOutcomes');

$conditions = array('courseId' => $courseID);

$result = $allCourseOutcomes->findOne($conditions);

$courseName = $result['courseName'];

if($result['CACOutcomes'][0] != 'NULL')

$CACOutcomes = $result['CACOutcomes'];

else

$CACOutcomes = NULL;

if($result['EACOutcomes'][0] != 'NULL')

$EACOutcomes = $result['EACOutcomes'];

else

$EACOutcomes = NULL;

$courseInfo = array($courseName, $CACOutcomes, $EACOutcomes);

return $courseInfo;

}

/\*

\*

\* get a list of outcomes measured for a semester

\*

\* $semester = semester for outcomes

\* $courseId = id of a course

\*

\*/

function getMeasuredOutcomes($semester, $courseId){

$courseOutcomes = getCourseOutcomes($courseId);

$cycleOutcomes = getSemesterOutcomes($semester);

$cac = array();

$eac = array();

if($courseOutcomes['CACOutcomes'][0] != NULL){

foreach($courseOutcomes['CACOutcomes'] as $course){

foreach($cycleOutcomes['CACOutcomes'] as $cycle){

if($course == $cycle)

array\_push($cac, $course);

}

}

}

if($courseOutcomes['EACOutcomes'][0] != NULL){

foreach($courseOutcomes['EACOutcomes'] as $course){

foreach($cycleOutcomes['EACOutcomes'] as $cycle){

if($course == $cycle)

array\_push($eac, $course);

}

}

}

$commonOutcomes = array($cac, $eac);

return $commonOutcomes;

}

/\*

\* gets the outcomes for a smester

\*

\* $semester = semester for outcomes

\*

\*

\*\*/

function getSemesterOutcomes($semester){

$client = new MongoClient();

$db = $client->selectDB('ABET');

$allCycles = new MongoCollection($db, 'CycleOfOutcomes');

$cycleConditions = array('semester' => $semester);

$cycleOutcomes = $allCycles->findOne($cycleConditions);

return $cycleOutcomes;

}

/\*

\* get outcomes for a course

\*

\* $courseId = id of class

\*

\*\*/

function getCourseOutcomes($courseId) {

$client = new MongoClient();

$db = $client->selectDB('ABET');

$allCourseOutcomes = new MongoCollection($db, 'CourseOutcomes');

$courseConditions = array('courseId' => $courseId);

$courseOutcomes = $allCourseOutcomes->findOne($courseConditions);

return $courseOutcomes;

}

?>

stats.php

<?php

session\_start();

include 'phpAPI.php';

$form = $\_POST['form'];

$semester = $\_POST['semester'];

error\_reporting(0);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title>ABET Accredidation</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/foundation.css" />

<script src="js/vendor/modernizr.js"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

</head>

<body>

<div class='form'>

<?php

$letter = substr($form, -1);

$type = substr($form, -5,3);

$letter = strtoupper($letter);

$type = strtoupper($type);

$results = getStats($letter, $type, $semester);

$stats = array(array());

for($i=0; $i< count($results); $i++){

for($j=0; $j< count($results[$i][$type . 'Results']); $j++){

$stats[$letter . strval($j+1)]['weak'] += $results[$i][$type .

'Results'][$letter . strval($j+1)][0];

$stats[$letter . strval($j+1)]['poor'] += $results[$i][$type .

'Results'][$letter . strval($j+1)][1];

$stats[$letter . strval($j+1)]['good'] += $results[$i][$type .

'Results'][$letter . strval($j+1)][2];

$stats[$letter . strval($j+1)]['excellent'] += $results[$i][$type .

'Results'][$letter . strval($j+1)][3];

$stats[$letter . strval($j+1)]['rowTotal'] = $stats[$letter .

strval($j+1)]['excellent'] +

$stats[$letter . strval($j+1)]['good'] + $stats[$letter .

strval($j+1)]['poor'] +

$stats[$letter . strval($j+1)]['weak'];

}

}

$descriptions = getRubricDescriptions($letter, $type);

echo '<h3>' . $type . '-' . $letter . '</h3>';

echo '<h5>Outcome: <strong>' . getDescription($letter, $type) . '</strong>

</h5>';

?>

<table>

<tr>

<th></td>

<th>1(weak)</td>

<th>2(poor)</td>

<th>3(good)</td>

<th>4(excellent)</td>

<th>%S+E</td>

</tr>

<?php

for($i=1; $i< count($stats); $i++){

echo "<tr>";

echo "<td>" . $descriptions[$i-1] . "</td>";

echo "<td>" . ($stats[$letter . strval($i)]['weak']/

$stats[$letter . strval($i)]['rowTotal']) \*100 . "%</td>";

echo "<td>" . ($stats[$letter . strval($i)]['poor']/

$stats[$letter . strval($i)]['rowTotal'])\*100 . "%</td>";

echo "<td>" . ($stats[$letter . strval($i)]['good']/

$stats[$letter . strval($i)]['rowTotal'])\*100 . "%</td>";

echo "<td>" . ($stats[$letter . strval($i)]['excellent']/

$stats[$letter . strval($i)]['rowTotal'])\*100 . "%</td>";

$excellent = ($stats[$letter . strval($i)]['excellent']/

$stats[$letter . strval($i)]['rowTotal'])\*100;

$good = ($stats[$letter . strval($i)]['good']/

$stats[$letter . strval($i)]['rowTotal'])\*100;

echo "<td>" . ($good+$excellent) . "%</td>";

echo "</tr>";

}

?>

</table>

<button id="btn">Go Back</button>

<script type="text/javascript">

document.getElementById("btn").onclick = function () {

location.href = "admin.php";

};

</script>

</div>

</body>

</html>

testPHPAPI.php

<?php

include 'phpAPI.php';

error\_reporting(0);

function testGetStats() {

$letter = "A";

$semester = "Spring2014";

$type = "CAC";

$expectedResult = "Is variable and is shown in the final

results on the website";

$result = getStats($letter, $semester, $type);

echo $expectedResult . ' <br />';

echo $result . ' <br />';

// if($result == $expectedResult) {

// return "SUCCESS <br />";

// }

// else{

// return "FAILURE <br />";

// }

}

function testGetRubricDescriptions() {

$letter = "A";

$type = "CAC";

$expectedResult="Able to apply math concepts to problems";

$result = getRubricDescriptions($letter, $type);

if($result[0] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetDescription() {

$letter = "A";

$type = "CAC";

$expectedResult="An ability to apply knowledge of computing and

mathematics appropriate to the discipline";

$result = getDescriptions($letter, $type);

if($result == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetForms() {

$instructor = "test";

$semester = "Spring2015";

$courseId = "testCourse";

$expectedResult='Test Course';

$result = getForms($instructor, $semester, $courseId);

if($result[0]->getCourseName() == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetFormTypes() {

$CACOutcomes = array("A", "H", "G");

$EACOutcomes = array("A", "D", "G");

$expectedResult="G";

$result = getFormTypes($CACOutcomes, $EACOutcomes);

if($result[0]['EAC'] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetCourseList() {

$instructor = "test";

$semester = "Spring2015";

$expectedResult='testCourse';

$result = getCourseList($instructor, $semester);

if($result[0]['course'] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetStudents() {

$instructor = "test";

$semester = "Spring2015";

$courseId = "testCourse";

$expectedResult="roolo berger";

$result = getStudents($courseId, $instructor, $semester);

if($result[0][0] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetCourseNameAndOutcomes() {

$courseId = "testCourse";

$expectedResult="Test Course";

$result = getCourseNameAndOutcomes($courseId);

if($result[0] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetMeasuredOutcomes() {

$semester = "Spring2015";

$courseId = "testCourse";

$expectedResult="A";

$result = getMeasuredOutcomes($semester, $courseId);

if($result[0][0] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetSemesterOutcomes() {

$semester = "Spring2015";

$expectedResult="A";

$result = getSemesterOutcomes($semester);

if($result['CACOutcomes'][0] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

function testGetCourseOutcomes() {

$courseId = "testCourse";

$expectedResult="A";

$result = getCourseOutcomes($courseId);

if($result['CACOutcomes'][0] == $expectedResult) {

return "SUCCESS <br />";

}

else{

return "FAILURE <br />";

}

}

//echo "getStats: " . testGetStats() . "TEST <br/>" . "<br/>";

echo "getRubricDescriptions: " . testGetRubricDescriptions() . "<br/>";

echo "getDescription: " . testGetRubricDescriptions() . "<br/>";

echo "getForms: " . testGetForms() . "<br/>";

echo "getFormTypes: " . testGetFormTypes() . "<br/>";

echo "getCourseList: " . testGetCourseList() . "<br/>";

echo "getStudents: " . testGetStudents() . "<br/>";

echo "getCourseNameAndOutcomes: " . testGetCourseNameAndOutcomes() . "<br/>";

echo "getMeasuredOutcomes: " . testGetMeasuredOutcomes() . "<br/>";

echo "getSemesterOutcomes: " . testGetSemesterOutcomes() . "<br/>";

echo "getCourseOutcomes: " . testGetCourseOutcomes() . "<br/>";

?>

empty.php

<?php

session\_start();

include 'phpAPI.php';

error\_reporting(0);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title>ABET Accredidation</title>

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/foundation.css" />

<script src="js/vendor/modernizr.js"></script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

</head>

<body>

<h1>There doesn't seem to be any data for that outcome + semester</h1>

<button id="btn">Go Back</button>

<script type="text/javascript">

document.getElementById("btn").onclick = function () {

location.href = "admin.php";

};

</script>

</body>

</html>