CIS-355: Individual Project

# Implement a website with a database back end

This assignment is worth 40 points (40% of your grade). It is possible to earn 10 points extra credit. To receive full credit you must submit (1) design documentation, (2) a working website, (3) a test plan and verification checklist, and (4) well-commented source code on github.

## **1. Design Documentation (25%)**

Submit the following. I will use the list below as a checklist to grade your work.

* Screen Flow Diagram (see PDF file for example)
* Database Diagram (see PDF file for example)
* Database schema (from phpmyadmin)
* NS-chart, Flowchart, Warnier-Orr Diagram, or some other visual description of the program logic. This should be done at a high level. There is no need to diagram every line of code.
* A narrative explanation of the following questions
  + What is the basic concept of the system? Why is this system useful? What problem does it solve?
  + Who will use the system? What does the system do? What features will the user find useful?
  + When and where will users use the system (home, office, school, golf course)? For extra credit, present one or two Use Case Diagrams. See: <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>
  + What hosting solution did you choose and why? What are the strengths and weaknesses of your choice?
  + What software tools did you choose and why? (You don’t have to do this project in PHP-MySQL.)
  + Why did you choose this as your project? What’s in it for you?
* **Username and password for me to log in**

## **2. Working Website (25%)**

Submit a link to a working website. I will use the list below as a checklist to grade your work.

* CRUD: System allows user to create (insert), retrieve (view), update and delete records in database tables
* Login security and session control: System permissions wok as designed across multiple screens in application
* Idempotency: Post/Redirect/Get problem has been addressed
* Back end data validation: input fields not only perform front-end validation but also populate drop down list boxes (or otherwise demonstrate back end data validation – if you do it in a unique way, please specify your technique so I don’t miss it while I’m grading)

## **3. Test Plan and Verification Checklist (25%)**

Submit the following. I will use the list below as a checklist to grade your work.

* A list of your system’s features
* A list of tests performed to ensured the features are delivered bug-free
* A list of things not tested
* A list of extra credit items, if your system exceeds the specs in section 2 (example: JSON, AJAX)

## **4. Well-Commented Source Code (25%)**

I will look for indications that the content of the course was applied in your code. I will use the list below as a checklist to grade your work.

* Code conforms to “sample well-commented code” section below and code is on github
* Foreign keys: system must implement at least 3 database tables, at least one of which uses foreign keys
* Create table: system should create tables if they don’t exist

## Sample Diagram

This is an example of a high-level diagram, Warnier-Orr style.

* Step 1: Connect to database
* Step 2: Check if any records in table
* Step 3: If records, print name field and add another random record

## Sample Well-Commented Code

This is an example of high-level comments. Note that the comments below match the diagram above. (You can run the code below by visiting <http://cis355.com/lesson01.php>.)

<?php

// filename: lesson01.php, George Corser, cis355, 2014-08-23

// Prints all items in column:name in table:table01 of database:lesson01

// and adds random entry to table:table01

// Step 1: ----- Connect to database -----

$hostname="localhost";

$username="student";

$password="learn";

$dbname="lesson01";

$usertable="table01";

$yourfield = "name";

$con = mysql\_connect($hostname,$username, $password)

or die ("<html><script language='JavaScript'>alert('Cannot connect.'),history.go(-1)</script></html>");

mysql\_select\_db($dbname);

// Step 2: ----- Check if any records in table -----

$query = "SELECT \* FROM $usertable";

$result = mysql\_query($query);

// Step 3: ----- If records, print name field and add another random record

if($result) { // if $result is empty there is no output and no message

while($row = mysql\_fetch\_array($result)){

$name = $row["$yourfield"];

echo "Name: ".$name."<br>"; // generates html code

}

$val1 = "name".rand();

$val2 = rand();

# INSERT INTO `lesson01`.`table01` (`id`, `name`, `desc`) VALUES (NULL, 'delta', 'fourth');

$query = "INSERT INTO `lesson01`.`table01` (`id`, `name`, `desc`) VALUES (NULL, '$val1', '$val2')";

$result2 = mysql\_query($query);

# echo "<br>".$result2;

printf("Last inserted record has id %d\n", mysql\_insert\_id());

echo "<br>Done<br>";

}

?>