**CS6314 Assignment #2- Due 3/09 Monday Noon**

Last Name: Guntakal Patil

First Name: Reetish Chand

NetID (email): RXG190006@utdallas.edu

Submission requirements.

Submit (1) a word document (this file with your answers and listing of any program & its run [log or screenshots]) and (2) a zip file of a folder which will contain all the codes (all the source codes, data and any other items related to this assignment and your work done).

This word document (this file) is your documentation (as a basis to add your answers) to be included here - all your answers, all your program-listings, instructions to compile and run [screen-shots, terminal text, or session log] to show your work done, including how to compile and run for each cases.

The word file should also have (1) in header with the course & section number, your name (Name: last name, first name), your netID (email), and Assignment #, (2) in footer with page number, (3) line number (restart from each page in left margin. Page Layout => Line Numbers). (4) Your program should have some comments (minimal or reasonable), to tell the code-reader what a segment of code is doing, and with the comment in the front/head of the program about you, this course and assignment, etc. (you may copy and paste some out of this document for your comment).

\*\* Your executable codes (that you submit) should run in cs1 without any change or recompilation.

\*\* Bring this cover sheet (this page) to TA for your demo.

Upload (attach) this document (with your answers) and a zip file (containing all the codes [source and data, etc.]).

Score-sheet

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CLO 1-6 |  |  |  |  |  |  | Demo | Doument |
| Task1  Course DB  40% | #1  dom & js | #2 | #3 | #4 | #5 |  |  |  |
| Task2  Course Web  30% | #1  php | #2  js & test |  |  |  |  |  |  |
| Task3  Course style  30% | #1 | #2 | #3 | #4 | #5 | #6 |  |  |

|  |
| --- |
| Deduction - Documentation (this .doc file) and upload  Max -10% if not done or poorly prepared |
| Deduction for Demo (Demo schedule or arrangement will be scheduled by TA, for your demo).  (Note. You do the demo only for the part(s) that you have done and submitted.)  Max -10% if not done or poorly prepared. |

CLO - After successful completion of this course, the student should be able to:

1. Ability to understand web architecture, standards, protocols, tools, and technologies

2. Ability to understand HTML, HTML5 and CSS.

3. Ability to understand JavaScript, JQuery, AJAX, XML, JSON

4. Ability to understand Database Technologies and SQL

5. Ability to understand Server-side programming with PHP

6. Ability to understand Web Services SOAP and RESTful Web Services

7. Ability to understand Web Security Protocols & Standards Semantic Web

[Place Table of Contents (of this document) here]

[Note. For how to insert TOC in a word document. Check

<https://support.microsoft.com/en-us/help/285059/how-to-create-a-table-of-contents-by-marking-text-in-word> ]

**Part1.**

**UTD CS Graduate Course List**

The manager of cs department hired you to be the web developer, to work on UTD CS Course management.

Your first project is to design and implement HTML/CSS with Javascript interacting with a database to do the following tasks.

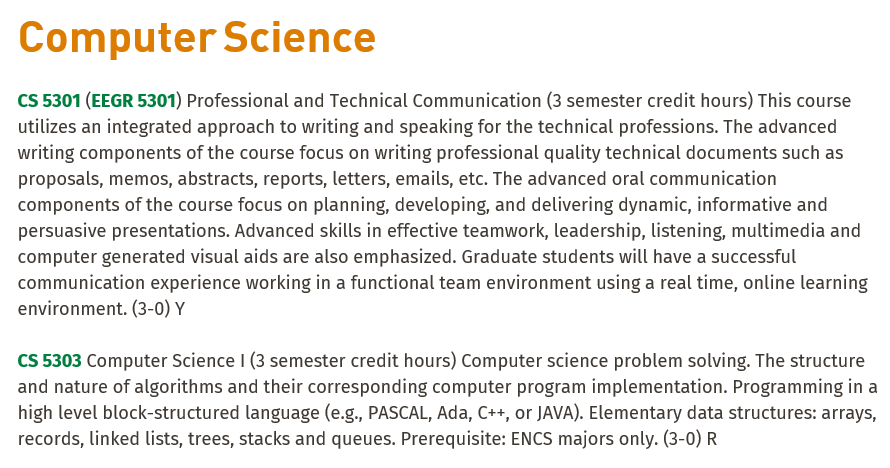
For submission, upload this directory (with all its contents – all the files and all subdirectories, and SQL export file to reconstruct the database and all the tables) in the zip file for this assignment.

**Task0.**

Create a folder (cs6314) for this course, and its subfolder for assignment2 (cs6314/assign2) in Xampp for this assignment. Create a subfolder (course1 in cs6314/assign2/) for this part. You should upload this directory (with all its contents – all the files and all subdirectories) in the zip file for this assignment. You will have a few subfolders as needed.

Consider the UTD CS graduate courses (course description):

<https://course.utdallas.edu/2019/graduate/courses/cs>



|  |
| --- |
| <p id="cs5301"><span class="course\_address">  <a href="https://course.utdallas.edu/2018/graduate/courses/cs5301">CS 5301</a> </span>  <span class="course\_tccns">(<a href="https://course.utdallas.edu/2018/graduate/courses/eegr5301">EEGR 5301</a>)</span>  <span class="course\_title">Professional and Technical Communication</span>  <span class="course\_hours">(3 semester credit hours)</span>  This course utilizes an integrated approach to writing and speaking for the technical professions. The advanced writing components of the course focus on writing professional quality technical documents such as proposals, memos, abstracts, reports, letters, emails, etc. The advanced oral communication components of the course focus on planning, developing, and delivering dynamic, informative and persuasive presentations. Advanced skills in effective teamwork, leadership, listening, multimedia and computer generated visual aids are also emphasized. Graduate students will have a successful communication experience working in a functional team environment using a real time, online learning environment. (3-0) Y  </p>  <p id="cs5303"><span class="course\_address">  <a href="https://course.utdallas.edu/2018/graduate/courses/cs5303">CS 5303</a></span>  <span class="course\_title">Computer Science I</span> <span class="course\_hours">(3 semester credit hours)</span> Computer science problem solving. The structure and nature of algorithms and their corresponding computer program implementation. Programming in a high level block-structured language (e.g., PASCAL, Ada, C++, or JAVA). Elementary data structures: arrays, records, linked lists, trees, stacks and queues. Prerequisite: ENCS majors only. (3-0) R</p> |

Save this web page (in course1 folder) to be updated in the following tasks.

Name the html file: course.html

**Task1**

Create a folder for this task (course1/task1)

With the initial web page provided (of the course catalog listing), your task is to add javascript program(s) to do the following:

(1) to extract all the course information of the catalog web page.

(2) to write each course information of (1) in sql format to a file (course1.sql)

(3) with two new fields for each course: time created and time updated, and using this sql table

(4) then to create sql table (course1) via Xampp database with sql statement (course1.sql)

(5) after the table is done, export the table in various formats (see below)

Name this html file with javascript program: course1.html

Name the table in database: course1

Note. For other files (e.g., style, javascript, etc.) as needed, use course1 as the file name followed by ".cs" or ".js", etc.

<p id="cs5330">

<span class="course\_address">

<a href="https://course.utdallas.edu/2018/graduate/courses/cs5330">CS 5330</a></span>

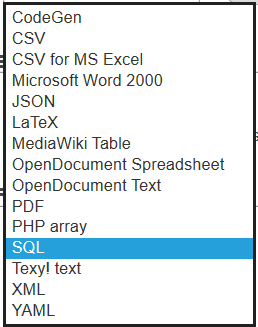
<span class="course\_title">Computer Science II</span>

<span class="course\_hours">(3 semester credit hours)</span>

Basic concepts of computer organization: Numbering systems, two's complement notation, multi-level machine concepts, machine language, assembly programming and optimization, subroutine calls, addressing modes, code generation process, CPU datapath, pipelining, RISC, CISC, and performance calculation. Prerequisite or Corequisite: <a href="https://course.utdallas.edu/2018/graduate/courses/cs5303">CS 5303</a>. (3-0) R</p>

|  |  |
| --- | --- |
| course\_id  course\_prefix  course\_number  course\_address  course\_title  course\_hours  course\_description  course\_prerequisite\_link  course\_prerequisite | <p id="cs5330">  <span class="course\_address">  <a href="https://course.utdallas.edu/2018/graduate/courses/cs5330">CS 5330</a></span>  <span class="course\_title">Computer Science II</span>  <span class="course\_hours">(3 semester credit hours)</span>  Basic concepts of computer organization: Numbering systems, two's complement notation, multi-level machine concepts, machine language, assembly programming and optimization, subroutine calls, addressing modes, code generation process, CPU datapath, pipelining, RISC, CISC, and performance calculation.  Prerequisite or Corequisite:  <a href="https://course.utdallas.edu/2018/graduate/courses/cs5303">CS 5303</a>. (3-0) R  </p> |
| course\_TimeCreated course\_TimeUpdated |  |

Note. To make your design easy, you may define each field to be: varchar (from 100 to 255, or more as needed for a certain field) with "default null" except the course id which will be a primary key of the course table. There could be more than one prerequisite courses for a course. This means that you may design course\_prerequisite to be enumerated (e.g., course\_prerequisite1, …, course\_prerequisite5) or have a second table with two columns (of a course with each prerequisite). You should include all the saved sql files (as you create or get from any tables in this assignment) in a subdirectory (course1/sql) so that the sql files will be also a part of the zip file to be uploaded, and also upload your documentation (this word document).



Save the initial sql file (SQL document) in a file (course0.sql) and after loading the table, the table will be exported and saved in a file (in course1/task1/filename where the filename is specified below):

(a) to save all the course information (SQL document) in a file (course1.sql)

(b) to save all the course information (XML document) in a file (course1.xml)

(c) to save all the course information (PHP array) in a file (coures1.php)

(d) to save all the course information (JASON document) in a file (course1.jason)

In course1/task1, please provide a word document (task1.docx) to show the result of each test case, step-by-step, including the screenshots (initial and final, and test performed). Provide a proper heading and comment for each test cases so that your reader should be able to follow the flow of your work done.

**Task2**

Create a folder (course1/task2) for the work for this task. Create or copy the sql for the course table (in Task1) to be used for this task.

Design and implement a web page (course2.php) with php similar to www10 example. The initial web page contains the content of course1 table (containing all courses for course table). It will provide a few functionalities: (1) to delete an existing course, (2) to update an existing course, and (3) to add a new course.

Name your php program: course2.php

Note. For other files (e.g., style, javascript, etc.) as needed, use course2 as the file name followed by ".cs" or ".js", etc.

You should have a few test cases for this task. Prepare a few sample courses (cs6201, cs6302, cs6303) for test cases to be added, to delete one course (cs6201), and to update a course (cs6302).

Also for testcase setup, provide a few buttons with a few javascript functions, respectively, to add, to delete, or to update a test case (versus manual add, delete, or update).

Name the javascript function: addTestCase1, addTestCase2, …

Name the javascript function: deleteTestCase1, deleteTestCase2, …

Name the javascript function: updateTestCase1, updateTestCase3, …

Once the program runs, it will show all the courses. Save this initial web page (source) in a file (course2.html).

After each test case, save the resulting web pages (in course1/task2/filename where the filename is specified below):

(1) to save each web page in a file after each test case (test1.html for test1, test2.html for test2, …)

In course1/task2, please provide a word document (task2.docx) to show the result of each test case, step-by-step, including the screenshots (initial and final, and test performed) for each test case, sql table before and after each test case, etc. Provide a proper heading and comment for each test cases so that your reader should be able to follow the flow of your work done.

**Task3**

Create a folder (course1/task3) for the work for this task.

Design and implement a web page with php (from Task2 [or Task1 if you cannot complete Task2]) with a style sheet (and javascript as needed).

Name your php program: course3.php

Name your style sheet file: course3.css

Note. For other files (e.g., style, javascript, etc.) as needed, use course3 as the file name followed by ".cs" or ".js", etc.

(1) The web page should have header, footer, and navigation (as discussed in the classes and lab with adv css, layout, multi-columned web page, etc.).

(2) Each course will be listed initially a short form (not a full description) with course number and title. By clicking or hovering (or your choice of design), the full course information will be displayed.

(3) To have a button to add a new course. Once clicked, you should have a block to be shown (or a new web page for add a new course or a pop-up page, etc.).

(4) The web page should provide a search capability to find a course (or courses) in search string, auto-complete or suggestion-list capability given a prefix string. Once selected, the web page will move down to the course (for the current viewport of the web page) as the user likes to navigate to a particular course. You may restrict this option to search course title (or description) only.

(5) Provide responsive web design options using bootstrap

<https://www.w3schools.com/howto/howto_website_bootstrap.asp>

In course1/task3, please provide a word document (task3.docx) to show the result of each case (step-by-step or a code segment) including the screenshots (initial and/or final, and test performed). Provide a proper heading and comment for each test cases so that your reader should be able to follow the flow of your work done.

Hint. A sample code to extract all <p> elements in a web page.

Source: <https://www.w3schools.com/js/js_htmldom_elements.asp>

<https://www.w3schools.com/jsref/met_element_getattribute.asp>

|  |
| --- |
| <html><body>  <h2>Finding HTML Elements by Tag Name</h2>  <div id="main">  <p>The DOM is very useful.</p>  <p>This example demonstrates the <b>getElementsByTagName</b> method.</p>  </div>  <p id="demo"></p>  <script>  var x = document.getElementById("main");  var y = x.getElementsByTagName("p");  document.getElementById("demo").innerHTML =  'The first paragraph (index 0) inside "main" is: ' + y[0].innerHTML;  </script>  </body>  </html> |

|  |
| --- |
| // Source: https://www.w3schools.com/js/js\_htmldom\_collections.asp  var x = document.getElementsByTagName("p");  // The elements in the collection can be accessed by an index number.  // To access the second <p> element you can write:  y = x[1];  // The length property defines the number of elements in an HTMLCollection:  var myCollection = document.getElementsByTagName("p"); var i; for (i = 0; i < myCollection.length; i++) {   myCollection[i].style.backgroundColor = "red"; } |
| // <https://www.w3schools.com/jsref/met_element_getattribute.asp>  var classname = document.getElementsByTagName("H1")[0].getAttribute("class"); |

# Task1 Solution

course1.html

|  |
| --- |
| <html>  <script type="text/javascript">  listItems = [];  for (var index = 0; index < courses.length; index++) {  var ele = {};  if (courses[index].tagName == 'P') {  ele.course\_id = courses[index].id;  ele.course\_number = courses[index].outerHTML;  ele.address = courses[index].innerHTML;  ele.course\_title = courses[index].children[1].innerText;  ele.course\_hours = courses[index].children[2].innerText;  ele.course\_description = ele.course\_description = courses[index].getElementsByClassName("course\_hours")[0].nextSibling.textContent;  try{  ele.course\_prerequisite\_link= document.querySelector("#"+courses[index].id+" > a").outerHTML;  }  catch(err){  ele.course\_prerequisite\_link = null;  }  try{  ele.course\_prerequisite = document.querySelector("#"+courses[index].id+" > a:nth-child(4)").innerText;  }  catch(err){  ele.course\_prerequisite = null;  }  try{  if(document.querySelector("#"+courses[index].id+" > a:nth-child(4)") && document.querySelector("#"+courses[index].id+" > a:nth-child(5)") )  {  ele.course\_prerequisite = document.querySelector("#"+courses[index].id+" > a:nth-child(4)").innerText;  ele.course\_corequisite\_link= document.querySelector("#"+courses[index].id+" > a:nth-child(5)").outerHTML;  ele.course\_corequisite = document.querySelector("#"+courses[index].id+" > a:nth-child(5)").innerText;  }  }  catch(err) {  ele.course\_corequisite=null;  ele.course\_corequisite\_link=null;  }  console.log(" INSERT INTO `courses` (`course\_id`, `course\_number`, `course\_address`, `course\_title`,`course\_hours`,`course\_description`,`course\_prerequisite\_link`,`course\_prerequisite`,`course\_corequisite\_link`,`course\_corequisite`) VALUES (",ele.course\_id,",",ele.course\_number,",",ele.address,",",ele.course\_title,",",ele.course\_hours,",",ele.course\_description,",",ele.course\_prerequisite\_link,",",ele.course\_prerequisite,",",ele.course\_corequisite\_link,",",ele.course\_corequisite);  }  }  </script>  <html> |

Screenshots with proper heading to show that you have done the work for this task. State clearly for each screenshot on what it is about (to show your work for the task is done), including the screenshot(s) of database table.

|  |
| --- |
| Loading the catalog page:    Executing the code in browser console: |
| Output of SQL insert statements after executing the code in browser console:    Create statement in SQL:  CREATE TABLE `courses` (  `course\_id` varchar(255) NOT NULL PRIMARY KEY,  `course\_number` varchar(255) DEFAULT NULL,  `course\_address` varchar(255) DEFAULT NULL,  `course\_title` varchar(255) DEFAULT NULL,  `course\_hours` varchar(255) DEFAULT NULL,  `course\_description` varchar(255) DEFAULT NULL,  `course\_prerequisite\_link` varchar(255) DEFAULT NULL,  `course\_prerequisite` varchar(255) DEFAULT NULL,  `course\_corequisite\_link` varchar(255) DEFAULT NULL,  `course\_corequisite` varchar(255) DEFAULT NULL,  `course\_TimeCreated` timestamp NOT NULL DEFAULT current\_timestamp(),  `course\_TimeUpdated` timestamp NOT NULL DEFAULT current\_timestamp()  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4    After creating table    After executing all the insert commands displayed in browser in SQL – displaying 83 rows    Exporting in SQL format:    Exporting in XML format:    Exporting in PHP array format:    Exporting in JSON format |

Listing of each file or (as needed or if missed)

|  |
| --- |
|  |

# Task2 Solution

course2.php

|  |
| --- |
| <?php // mysqlitest.php  require\_once 'login.php';  $connection = new mysqli($db\_hostname, $db\_username, $db\_password, $db\_database);  if ($connection->connect\_error) die($connection->connect\_error);  /\* This code deletes the record from course1 Table \*/  if (isset($\_POST['delete']) && isset($\_POST['CourseID']))  {  $CourseID = get\_post($connection, 'CourseID');  $query = "DELETE FROM course1 WHERE course\_id='$CourseID'";  $result = $connection->query($query);  if (!$result) echo "DELETE failed: $query<br>" .  $connection->error . "<br><br>";  }  /\* Delete code ended \*/  /\*This code inserts course1 record in the table \*/  if (isset($\_POST['CourseID']) &&  isset($\_POST['CourseNumber']) &&  isset($\_POST['CourseAddress']) &&  isset($\_POST['CourseTitle']) &&  isset($\_POST['CourseHours']) &&  isset($\_POST['CourseDescription']) &&  isset($\_POST['CoursePrerequisiteLink']) &&  isset($\_POST['CoursePrerequisite']) &&  isset($\_POST['CourseCorequisiteLink']) &&  isset($\_POST['CourseCorequisite']) &&  isset($\_POST['CourseTimeCreated']) &&  isset($\_POST['CourseTimeUpdated']) )  {  $CourseID = get\_post($connection, 'CourseID');  $CourseNumber = get\_post($connection, 'CourseNumber');  $CourseAddress = get\_post($connection, 'CourseAddress');  $CourseTitle = get\_post($connection, 'CourseTitle');  $CourseHours = get\_post($connection, 'CourseHours');  $CourseDescription = get\_post($connection, 'CourseDescription');  $CoursePrerequisiteLink = get\_post($connection, 'CoursePrerequisiteLink');  $CoursePrerequisite = get\_post($connection, 'CoursePrerequisite');  $CourseCorequisiteLink = get\_post($connection, 'CourseCorequisiteLink');  $CourseCorequisite = get\_post($connection, 'CourseCorequisite');  $CourseTimeCreated = get\_post($connection, 'CourseTimeCreated');  $CourseTimeUpdated = get\_post($connection, 'CourseTimeUpdated');  $query = "INSERT INTO course1 VALUES" .  "('$CourseID', '$CourseNumber', '$CourseAddress', '$CourseTitle', '$CourseHours', '$CourseDescription', '$CoursePrerequisiteLink','$CoursePrerequisite','$CourseCorequisiteLink','$CourseCorequisite','$CourseTimeCreated','$CourseTimeUpdated')";  $result = $connection->query($query);  if (!$result) echo "INSERT failed: $query<br>" .  $connection->error . "<br><br>";  }  /\*insert code ended \*/  echo <<<\_END  <form action="course2.php" method="post"><pre>  CourseID <input type="text" name="CourseID">  CourseNumber <input type="text" name="CourseNumber">  CourseAddress <input type="text" name="CourseAddress">  CourseTitle <input type="text" name="CourseTitle">  CourseHours <input type="text" name="CourseHours">  CourseDescription <input type="text" name=" CourseDescription">  CoursePrerequisiteLink <input type="text" name="CoursePrerequisiteLink">  CoursePrerequisite <input type="text" name="CoursePrerequisite">  CourseCorequisiteLink <input type="text" name="CourseCorequisiteLink">  CourseCorequisite <input type="text" name="CourseCorequisite">  CourseTimeCreated <input type="datetime-local" name="CourseTimeCreated">  CourseTimeUpdated <input type="datetime-local" name="CourseTimeUpdated">  <input type="submit" value="ADD RECORD"></br>  <button onclick="addTestCase1()">addTestcase</button></br>  </pre></form>  \_END;  /\* this code displays student record from the table \*/  $query = "SELECT \* FROM course1";  $result = $connection->query($query);  if (!$result) die ("Database access failed: " . $connection->error);  $rows = $result->num\_rows;    for ($j = 0 ; $j < $rows ; ++$j)  {  $result->data\_seek($j);  $row = $result->fetch\_array(MYSQLI\_NUM);  echo <<<\_END  <pre>  CourseID: $row[0]  CourseNumber: $row[1]  CourseAddress: $row[2]  CourseTitle: $row[3]  CourseHours: $row[4]  CourseDescription: $row[5]  CoursePrerequisiteLink: $row[6]  CoursePrerequisite: $row[7]  CourseCorequisiteLink: $row[8]  CourseCorequisite: $row[9]  CourseTimeCreated: $row[10]  CourseTimeUpdated: $row[11]  <form action="course2.php" method="post">  <input type="hidden" name="delete" value="yes">  <input type="hidden" name="CourseID" value="$row[0]">  <button><a href='edit.php?edit=$row[0]'>UPDATE Record</a></button></br>  <button onclick="deleteTestCase1()">deleteTestcase</button></br>  <input type="submit" value="DELETE RECORD"></form>  </pre>  \_END;  }    $result->close();  $connection->close();    function get\_post($connection, $var)  {  return $connection->real\_escape\_string($\_POST[$var]);  }  ?>  <script>  function addTestCase1() {  <?php  if (isset($\_POST['CourseID']) &&  isset($\_POST['CourseNumber']) &&  isset($\_POST['CourseAddress']) &&  isset($\_POST['CourseTitle']) &&  isset($\_POST['CourseHours']) &&  isset($\_POST['CourseDescription']) &&  isset($\_POST['CoursePrerequisiteLink']) &&  isset($\_POST['CoursePrerequisite']) &&  isset($\_POST['CourseCorequisiteLink']) &&  isset($\_POST['CourseCorequisite']) &&  isset($\_POST['CourseTimeCreated']) &&  isset($\_POST['CourseTimeUpdated']) )  {  $CourseID = $\_POST['CourseID'];  $CourseNumber = $\_POST['CourseNumber'];  $CourseAddress = $\_POST['CourseAddress'];  $CourseHours = $\_POST['CourseHours'];  $CourseTitle = $\_POST['CourseTitle'];  $CourseDescription = $\_POST['CourseDescription'];  $CoursePrerequisiteLink = $\_POST['CoursePrerequisiteLink'];  $CoursePrerequisite = $\_POST['CoursePrerequisite'];  $CourseCorequisiteLink = $\_POST['CourseCorequisiteLink'];  $CourseCorequisite = $\_POST['CourseCorequisite'];  $CourseTimeCreated = $\_POST['CourseTimeCreated'];  $CourseTimeUpdated = $\_POST['CourseTimeUpdated'];  $sql = "INSERT INTO course1  VALUES ('$CourseID', '$CourseNumber', '$CourseAddress', '$CourseTitle', '$CourseHours','$CourseDescription','$CoursePrerequisiteLink','$CoursePrerequisite','$CourseCorequisiteLink','$CourseCorequisite', '$CourseTimeCreated', '$CourseTimeUpdated')";  if (mysqli\_query($connection, $sql)) {  echo "New record created succeddfully !";  } else {  echo "Error : " . $sql . "<br>" . mysqli\_error($connection);  }  }  mysqli\_close($connection);  ?>  }  </script> |

Screenshots of Test cases (before and after test case)

|  |
| --- |
| Index page    After running course2.php    Adding record CS 7000    New record in DB    Deleting record CS 7000    After deleting record CS 7000    Record deleted in DB    Adding testcase CS 7001    New Record in DB:    Deleting testcase 7001    After deleting testcase CS 7001    Records in DB: |

Screenshots with proper heading to show that you have done the work for this task. State clearly for each screenshot on what it is about (to show your work for the task is done), including the screenshot(s) of database table.

|  |
| --- |
|  |

Listing of each file or (as needed or if missed)

|  |
| --- |
|  |

# Task3 Solution

course3.php

|  |
| --- |
| <html>  <head>  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">  <link rel="stylesheet" type="text/css" href="course3.css">  </head>  <body>  <?php //course3.php  require\_once 'login.php'; /\*connecting database\*/  $connection = new mysqli($db\_hostname, $db\_username, $db\_password, $db\_database);  if ($connection->connect\_error) die($connection->connect\_error); ?>  <nav class="navbar navbar-expand-md navbar-dark bg-success">  <div class="navbar-collapse collapse w-100 order-1 order-md-0 dual-collapse2">    <ul class="navbar-nav mr-auto">  <li class="nav-item">  <a class="nav-link" href="#">Home</a>  </li>  <li class="nav-item">  <a class="nav-link" href="#column2">Courses</a>  </li>  <li class="nav-item">  <a class="nav-link" href="https://cs.utdallas.edu/">Department</a>  </li>  <li class="nav-item">  <a class="nav-link" href="https://cs.utdallas.edu/contact/">Contact</a>  </li>  </ul>    <input type="text" class="search" placeholder="Search.." name="search">  <button type="submit">Go</button>    </div>  </nav>    <div class="insert">  <form action="insert.php" method="post"><pre>  <button><a href='insert.php' class="button1">Add a New Course</a></button>  </pre></form>  </div>        <div class="column2">  <?php  $query = "SELECT \* FROM courses";  $result = $connection->query($query);  if (!$result) die($connection->error);  $rows = $result->num\_rows;  for ($j = 0 ; $j < $rows ; ++$j)  {  $result->data\_seek($j);  $row = $result->fetch\_array(MYSQLI\_ASSOC);    echo '<br> Course Number: <a href='.$row['course\_address']. ' <style=a:hover{display="block"} >' . $row['course\_id'] .'</a>';  echo '<br> Course Title: ' . $row['course\_title'] . '<br><br>';  }  $result->close();  $connection->close();  ?>  </div>  <div>  <footer style=" color: blue; text-align: center;">  <p class="foot"><a href="https://utdallas.edu/"> The University of Texas at Dallas &copy</p>  </footer>  </div>  <script src="search.js" ></script>  </body>    </html>  Search.js  document.getElementById("searchBut").onclick=function(){  var value=document.getElementById("Searchbox").value;  window.location.href = "#"+value; |

course3.css

|  |
| --- |
| body{  background-color: darkorange;  font-style: normal;  }  .foot  {  background-color: White;  font-style: italic;  }  .search{  float:right;  font-style: italic;  }  .column2  {  margin-left:45px;  font:Arial;  }  .insert  {  float:right ;  margin-right: 30px;  margin-top: 105px;  }  .button1  {  width: 75px;  margin-right: 25px;  } |

Screenshots with proper heading to show that you have done the work for this task. State clearly for each screenshot on what it is about (to show your work for the task is done), including the screenshot(s) of database table.

|  |
| --- |
| Task 3 directory    Loading course3.php  webpage with header, navigation bar, a search bar and a button    webpage with footer    Clicking on a course link in a page    Redirection to respective course link    Search bar    Clicking on the Nav Bar links:    Redirection to contact page:    Clicking on department tab:    Redirection to cs department homepage    Searching    Redirecting to respective window position    Adding new course:      Adding course details in form    New record in DB |
|  |

Screenshots of Test cases (before and after test case)

|  |
| --- |
|  |