**CS6314 Project Part1**

|  |  |  |  |
| --- | --- | --- | --- |
| Member of Team | NetID | First Name | Last Name |
| 1. | mxb240023 | Mahima | Bhushan |

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 (please specify what you have done by each team member has done below).

|  |  |  |  |
| --- | --- | --- | --- |
| Member Name | 1. Mahima Bhushan | Demo | Documentation |
| Task1  course DB  40% | Done 100% of the Task1 |  | Done 100% |
| Task2  course Web  40% | Done 100% of the Task2 |  | Done 100% |
| Task3  course style  20% | Done 100% of the Task3 |  | Done 100% |
| Task4  Documentation & Demo  (Deduction if not done) | Done 100% of the Task4 |  | Done 100% |

|  |
| --- |
| Deduction - Documentation (this .doc file) and upload  Max -70% 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 -50% if not done or poorly prepared. |
| Deduction (To use of xampp - all work/project to be done using xampp frame)  Do all your work/project done using/via xampp frame. If not, there will be severe penalty (-70%). |

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

# Table of Contents

## [1.Preparation](#_Preparation)

## [2.Task1](#_Task1)

## [3.Task2](#_Task2)

## [4.Task3](#_Task3)

## [5.Task4](#_Task4)

## [6.Task1 Solution](#_6.Task1_Solution)

## [7.Task2 Solution](#_7.Task2_Solution)

## [8.Task3 Solution](#_8.Task3_Solution)

## [9.Task4 Solution](#_Task4_Solution)

## 1.Preparation

**CS Graduate Course Listing - Offering S25**

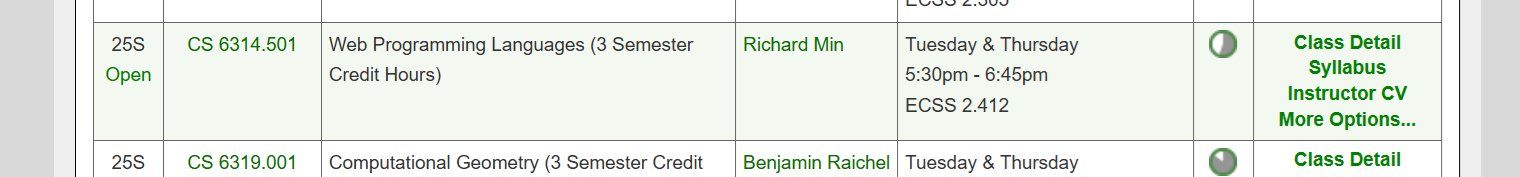
The manager of cs department hired you to be the web developer, to work on UTD CS Course management.

Your first project is to design and implement HTML/CSS with Javascript and PHP with SQL 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.

Create a folder (cs6314) for this course, and its subfolder for each part (e.g., cs6314/task1) in Xampp for this assignment (task). For your submission, upload the zip file of this directory (with all its contents – all the files and all subdirectories) and this document (as your project report). You will also have a few more subfolders as needed. Note that this folder should contain also a folder for all sql statements (so that all tables can be created using these sql for the project, by exporting the mysql database of all the tables for this project in sql format.

Consider the UTD CS course offering (see the attachment for the web page containing all cs course offering):



Save this web page (in cs6314 folder) to be used in the following tasks.

Name the html file: course.html

Your tasks for Project Part1 is to do:

#1. Extract for each course offering: (1) year-semester (S25), (2) Course Number & Section, (3) Course Title, (4) Instructor, (5) Instructor NetID, (6) Date & Time of the course/section, (7) Classroom Location. This is to be done by Javascript program(s) interacting with HTML document of course offering.

#2. Generate SQL statements to create table and insert the information to be run by PHP program(s)

#3. Have a PHP program(s) to generate a web page of course offerings of S25 (reading MySQL database table created and populated in #2), with functionality to list, add new entry, or to update or delete an entry, as the system is interacting with user to work with the table in the database.

#4. Do all your work/project done using/via xampp frame. If not, there will be severe penalty (-70%).

In cs6314/task1, please provide a word document (task1.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.

## 2.Task1

Create a folder for this task (cs6314/task1)

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

(0) design database & table for course (offering) to be used for the following subtasks.

(1) to extract all the course information using javascript (task1.html and task1.js)

(2) to write each course information of (1) in sql format to a file (task1.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 (task1.sql)

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

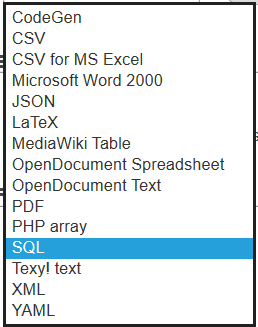
Name this html file with javascript program and sql: task1.html, task1.js, task1.sql

Name the table in database: course1

Hint. For coding hint, check www10 examples.

Note. For a file (e.g., style, javascript, etc.) as needed, use task1 prefix of each file for the file name (e.g., task1misc.html).

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. You should include all the saved sql files (as you create or get from any tables in this assignment) in a subdirectory (cs6314/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 statements in a file (task1.sql) and after loading the table, the table will be exported and saved in a file (in cs6314/task1/filename where the filename is specified below):

(a) to save all the course information (SQL document) in a file (task1out.sql)

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

(c) to save all the course information (PHP array) in a file (task1out.php)

(d) to save all the course information (JASON document) in a file (task1out.jason)

In cs6314/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.

## 3.Task2

Create a folder (cs6314/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 (task2.php) with php similar to www10 example. The initial web page contains the content of cs6314 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: task2.php

You should have a few test cases for this task. Prepare a few sample courses (5 courses which you have taken or you are taking in this semester) for test cases to be added, to be updated, and to be deleted.

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 (from sql table via php).

Save this initial web page (source) in a file (task2out.html).

After each test case, save the resulting web pages (in cs6314/task2/filename where the filename is specified below):

(1) to save each web page in a file after each test case (test1out.html for test1, test2out.html for test2, …)

In cs6314/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.

## 4.Task3

Create a folder (cs6314/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). You may also use UTD branding style for this task.

Name your php program: task3.php

Name your style sheet file: task3.css

(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) Initially each course will be listed in a short form with course/section number and title. By clicking or hovering (or your choice of design), the full course information will be displayed (as available).

(3) To have a button to add a new course for your choice of course(s). 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.). The selected course(s) will be sent to the server to be added to the database table.

(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 (using course number, title, instructor name, day or time of class). 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 cs6314/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.

## 5.Task4

Create a folder (cs6314/task4) for the work for this task.

Deliver your team documentation of this project part1 tasks that you have done:

(1) project technical documents (including code listing, usability assessment, …) – this word document

~~(2) project management plan~~

~~(3) use case and user-interface (of web pages with screenshots)~~

Note. If you have screenshots in the document, make sure that it is readable (font size & no need to zoom).

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"); |

|  |
| --- |
| // 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"; } |

## 6.Task1 Solution

-Have renamed coursebook2025.html to course.html

-Added the below code in **course.html** to provide a button to extract the course information from the website using the task1.js and download a task1.sql file which can be uploaded to xampp->phpMyAdmin->database

|  |
| --- |
| <!-- Button to trigger the JavaScript extraction -->  <button onclick="extractCourses()">Extract & Download SQL</button>  <!-- Link to your JavaScript file -->  <script src="task1/task1.js"></script> |

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.

|  |
| --- |
| - The button is added as follows to extract and download the course info in sql format.  - Code for creating the table course1 with the column headers:  A screenshot of a computer program  Description automatically generated  -task1.sql file is generated in the downloads folder  A blue and white stripe  Description automatically generated  -Create a university\_courses database, import the task1.sql table (course1) via Xampp database  A screenshot of a computer  Description automatically generated  A screenshot of a computer  Description automatically generated  -course1 table snippet in phpmyadmin:  A screenshot of a computer  Description automatically generated  - Under format there are many options available choose SQL, XML, JASON and PHP and save it as task1out.xml, task1out.sql, task1out.php, task1out.json,  A screenshot of a computer  Description automatically generated  -Final output of file Structure |

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

|  |
| --- |
| task1.js  task1.sql  task1out.sql  task1out.php  task1out.json  task1out.xml  task1.docx |

## 7.Task2 Solution

task2.php

|  |
| --- |
| <?php  $conn = new mysqli("localhost", "root", "", "university\_courses");  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  // Add course  if (isset($\_POST['add'])) {  $stmt = $conn->prepare("INSERT INTO course1 (course\_number, course\_title, instructor\_name, date\_time, location) VALUES (?, ?, ?, ?, ?)");  $stmt->bind\_param("sssss", $\_POST['course\_number'], $\_POST['course\_title'], $\_POST['instructor\_name'], $\_POST['date\_time'], $\_POST['location']);  $stmt->execute();  }  // Update course  if (isset($\_POST['update'])) {  $stmt = $conn->prepare("UPDATE course1 SET course\_title=?, instructor\_name=?, date\_time=?, location=? WHERE course\_number=?");  $stmt->bind\_param("sssss", $\_POST['course\_title'], $\_POST['instructor\_name'], $\_POST['date\_time'], $\_POST['location'], $\_POST['course\_number']);  $stmt->execute();  }  // Delete course  if (isset($\_POST['delete'])) {  $stmt = $conn->prepare("DELETE FROM course1 WHERE course\_number=?");  $stmt->bind\_param("s", $\_POST['course\_number']);  $stmt->execute();  }  // Display courses  $result = $conn->query("SELECT \* FROM course1");  ?>  <!DOCTYPE html>  <html>  <head>  <title>Task 2 - Courses Management</title>  <script src="task2.js"></script>  </head>  <body>  <h2>Courses:</h2>  <table border="1">  <tr>  <th>Course Number</th>  <th>Title</th>  <th>Instructor</th>  <th>Date/Time</th>  <th>Location</th>  </tr>  <?php while($row = $result->fetch\_assoc()): ?>  <tr>  <td><?= htmlspecialchars($row['course\_number']) ?></td>  <td><?= htmlspecialchars($row['course\_title']) ?></td>  <td><?= htmlspecialchars($row['instructor\_name']) ?></td>  <td><?= htmlspecialchars($row['date\_time']) ?></td>  <td><?= htmlspecialchars($row['location']) ?></td>  </tr>  <?php endwhile; ?>  </table>  <h2>Add / Update / Delete Course</h2>  <form method="POST">  Course Number: <input type="text" name="course\_number" required><br>  Title: <input type="text" name="course\_title"><br>  Instructor: <input type="text" name="instructor\_name"><br>  Date/Time: <input type="text" name="date\_time"><br>  Location: <input type="text" name="location"><br><br>  <button name="add">Add</button>  <button name="update">Update</button>  <button name="delete">Delete</button>  </form>  <h2>Test Cases (via JavaScript)</h2>  <button onclick="addTestCase1()">Add Test Case 1</button>  <button onclick="addTestCase2()">Add Test Case 2</button>  <button onclick="addTestCase3()">Add Test Case 3</button>  <button onclick="updateTestCase1()">Update Test Case 1</button>  <button onclick="updateTestCase2()">Update Test Case 2</button>  <button onclick="updateTestCase3()">Update Test Case 3</button>  <button onclick="deleteTestCase1()">Delete Test Case 1</button>  <button onclick="deleteTestCase2()">Delete Test Case 2</button>  <button onclick="deleteTestCase3()">Delete Test Case 3</button>  </body>  </html>  <?php  $conn->close();  ?> |

Screenshots of Test cases (before and after test case)

|  |
| --- |
| testcase 1:  Adding a new course CS6314:  Before:    After:    testcase 2:  Adding a new course with ID CS6363  Before:    After:    testcase 3:  Adding a new course with ID CS6360  Before:    After:  A screenshot of a computer  Description automatically generated  testcase 4:  Update the course title to 'Web Programming Language' from 'Web Programming'  Before:  A screenshot of a computer  Description automatically generated  After:    testcase 5:  change the course title to 'Design and Analysis of Algorithms' from 'Algorithms'  Before:  A screenshot of a computer  Description automatically generated  After:    testcase 6:  Update the location to 2.416 from 2.412  Before:    After:  A screenshot of a computer  Description automatically generated  testcase 7:  Deleting the course CS6314:  Before:  A screenshot of a computer  Description automatically generated  After:    testcase 8:  Deleting the course CS6314:  Before:    After:    testcase 9:  Deleting the course id CS6360  Before:    After: |

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.

|  |
| --- |
| -Manually add, update and delete course on the web page:  A screenshot of a computer  Description automatically generated  -Testcase buttons as seen on the web page:    -Snippet of the web page on how the course info is displayed  A table of a number of events  Description automatically generated with medium confidence  -task2 folder content: |

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

|  |
| --- |
| task2.php  task2.js  task2.sql  task2out.html  test1out.html  test2out.html  test3out.html  test4out.html  test5out.html  test6out.html  test7out.html  test8out.html  test9out.html  task2.docx |

## 8.Task3 Solution

task3.php

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Course Management - UT Dallas</title>  <link rel="stylesheet" href="https://code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">  <link rel="stylesheet" type="text/css" href="task3.css">  <!-- First load jQuery -->  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <!-- Then load jQuery UI -->  <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>  <!-- Bootstrap JS -->  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script>  <!-- Custom JS -->  <script src="task3.js"></script>  </head>  <body>  <!-- Header -->  <header class="utd-header text-center py-3">  <img src="images/utd-logo.png" alt="UT Dallas Logo" height="140">  <h2 class="mt-2">Course Management System</h2>  </header>  <!-- Navigation -->  <nav class="navbar navbar-expand-md navbar-dark bg-utd-green">  <a class="navbar-brand" href="#">UTD CMS</a>  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar">  <span class="navbar-toggler-icon"></span>  </button>  <div class="collapse navbar-collapse" id="navbar">  <ul class="navbar-nav ml-auto">  <li class="nav-item"><a class="nav-link" href="#">Home</a></li>  <li class="nav-item"><a class="nav-link" href="#">Courses</a></li>  <li class="nav-item"><a class="nav-link" href="#addCourseModal" data-toggle="modal">Add Course</a></li>  </ul>  </div>  </nav>  <!-- Main Content -->  <main class="container my-4">  <!-- Search Bar -->  <input type="text" id="courseSearch" class="form-control" placeholder="Search courses...">  <!-- Course Listings -->  <div id="coursesAccordion" class="mt-3">  <?php  $conn = new mysqli("localhost", "root", "", "university\_courses");  $result = $conn->query("SELECT \* FROM course1");  while($row = $result->fetch\_assoc()):  ?>  <div class="card mb-1">  <div class="card-header bg-utd-orange">  <a class="card-link text-white" data-toggle="collapse" href="#course<?= preg\_replace('/[^a-zA-Z0-9]/', '', $row['course\_number']) ?>">  <?= htmlspecialchars($row['course\_number']) ?> - <?= htmlspecialchars($row['course\_title']) ?>  </a>  </div>  <div id="course<?= preg\_replace('/[^a-zA-Z0-9]/', '', $row['course\_number']) ?>" class="collapse">  <div class="card-body">  <strong>Instructor:</strong> <?= htmlspecialchars($row['instructor\_name']) ?><br>  <strong>Time:</strong> <?= htmlspecialchars($row['date\_time']) ?><br>  <strong>Location:</strong> <?= htmlspecialchars($row['location']) ?>  </div>  </div>  </div>  <?php endwhile; ?>  <?php $conn->close(); ?>  </div>  </main>  <!-- Add Course Modal -->  <div class="modal fade" id="addCourseModal">  <div class="modal-dialog">  <div class="modal-content">  <div class="modal-header bg-utd-orange text-white">  <h4 class="modal-title">Add New Course</h4>  <button type="button" class="close" data-dismiss="modal">&times;</button>  </div>  <div class="modal-body">  <!-- Update the form element here -->  <form id="addCourseForm" method="POST" action="task3\_addcourse.php">  <input type="text" name="course\_number" placeholder="Course Number" required class="form-control mb-2">  <input type="text" name="course\_title" placeholder="Course Title" required class="form-control mb-2">  <input type="text" name="instructor\_name" placeholder="Instructor" required class="form-control mb-2">  <input type="text" name="date\_time" placeholder="Date/Time" required class="form-control mb-2">  <input type="text" name="location" placeholder="Location" required class="form-control mb-2">  <button type="submit" class="btn bg-utd-green text-white">Add Course</button>  </form>  </div>  </div>  </div>  </div>  <!-- Footer -->  <footer class="text-center py-3 bg-dark text-white">  © 2025 The University of Texas at Dallas  </footer>  </body>  </html> |

task3.css

|  |
| --- |
| .bg-utd-green { background-color: #008542; }  .bg-utd-orange { background-color: #C75B12; }  .utd-header {  background-color: #124734;;  border-bottom: 5px solid #fbf7f4;  }  .utd-header h2 {  color: white; /\* Set the text color to white \*/  }  body {  font-family: Arial, sans-serif;  background-color: #f8f8f8;  }  footer {  font-size: 14px;  } |

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.

|  |
| --- |
| - Create a folder (cs6314/task3) for the work for this task.  A blue folders with black text  Description automatically generated  - Snippet to show the folder task3 residing in XAMPP->htdocs folder  A screenshot of a computer  Description automatically generated  -Header and Navigation as seen on Web page:  A screen shot of a computer  Description automatically generated  -Footer code snippet:    -Look of the webpage  A screen shot of a course management system  Description automatically generated  - On Click the courses: the full course information will be displayed    -Add\_Course button as seen on web page Navigation bar:    -On Click of add course: pop-up page to fill in the details  A screenshot of a computer  Description automatically generated  -Search string gives a suggestion list based on the given prefix string:  A screenshot of a computer  Description automatically generated  A screen shot of a computer  Description automatically generated  -On selecting the option it points to the respective course details  A screenshot of a credit card  Description automatically generated  -On minimizing the window the design is responsive, navigation bar ->add course is hidden in the list  A screenshot of a computer  Description automatically generated |

|  |
| --- |
| -On Click of add course: pop-up page to fill in the details      -Before Adding:    -After Adding:    -task3 folder file content:  Images -> folder contains the ut dallas logo used in the header |

## 9.Task4 Solution

Listing of the programs below (html, js, css, php, sql, … as you have done for Task1,2,3).

task4.html

|  |
| --- |
| None |

task4.css

|  |
| --- |
| None |

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.

|  |
| --- |
| -Create a folder (cs6314/task4) for the work for this task.    - project technical documents (including code listing, usability assessment, …) |

Screenshots of Test cases (before and after test case)

|  |
| --- |
| None |