|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name** | |  | | | | **Student Number** | |  |
| **Unit Code/s & Name/s** | | ICTWEB502 Create dynamic web pages | | | | | | |
| **Assessment Type** | | Case Study  Assignment  Project  Other *(specify)* | | | | | | |
| **Assessment Name** | | Server Side Scripting | | | | | | |
| **Assessment Due Date** | | **19/4/18** | | | **Assessment Received Date** | | | 22/3/18 |
| **Student Declaration:** I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature. | | | | | | | | |
| **Student Signature** | |  | | | | | **Date** |  |
| **Assessor Feedback:**  **Student provided with feedback** | | | | | | | | |
| **Attempt 1** | | | **Satisfactory** | **Not Satisfactory** | | | **Date** | / / |
| **Attempt 2** | | | **Satisfactory** | **Not Satisfactory** | | | **Date** | / / |
| **Supplementary Assessment**  *(Apprentices/Trainees only)* | | | **Satisfactory** | **Not Satisfactory** | | | **Date** | / / |
| **Assessor Name** | David Hunt | | | **Assessor Signature** | | |  | |
| **Note to assessor: Please record any reasonable adjustment below that has occurred during this assessment e.g. written assessment given orally.** | | | | | | | | |
|  | | | | | | | | |
| **PRIVACY DISCLAIMER:** TAFE Queensland is collecting your personal information for assessment purposes. The information will only be accessed by authorised employees of TAFE Queensland. Some of this information may be given to the Australian Standards Quality Authority (ASQA) or its successor and/or TAFE Queensland for audit and/or reporting purposes. Your information will not be given to any other person or agency unless you have given us written permission or we are required by law. | | | | | | | | |

|  |  |
| --- | --- |
| **Instructions to Student** | **General Instructions:**  For you to complete this assessment successfully your program must be structured following the Model View Control method. You must use Netbeans as your web editing tool, not a web authoring application i.e Dreamweaver.  *This an individual assessment as such student must not share code, or edit other students as stated as per TAFE Queensland Student Rules assessment requirements.*  You must take ownership of your code. For all webpages that you complete, there must be at the author’s block as shown below.  **Author: Your Name**  **Student Number:**  **Purpose: What the webpage is design to do**  **Known bugs:**  **Date:**  Also your code must be fully commended where you have edited code.  **Number of Attempts:**  You will receive two (2) attempts for this assessment. Should your 1st attempt be not satisfactory, your teacher will discuss the relevant questions with you and will arrange a 2nd attempt to be scheduled. Should your 2nd attempt not be successful, or you fail to undertake the 2nd attempt, you will be deemed “not satisfactory” for this assessment item. Only one re-assessment attempt may be granted for each assessment item, with the exception of Apprentices or Trainees who are permitted an additional supplementary assessment.  **Assessment Criteria:**  To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate the following key skills/tasks/knowledge to industry standard:   * Identify the client and server-side dynamic content * Create the dynamic content * Test the dynamic pages |
| **Evidence Required to be Submitted and Method of Submission** | Once your project is completed, you must zip your project files into an archive file with the following naming convention **As5\_InternetProgramming1\_your\_name** |
| **Instructions to Assessor** | **Work, Health and Safety:** A work health and safety check of the assessment environment is to be conducted prior to the assessment and any hazards addressed appropriately.  **Examination Range and Conditions:**  *This an individual assessment as such student must not share code, or edit other students as stated as per TAFE Queensland Student Rules assessment requirements.*  **Materials to be supplied:**  Project starter files   * *a development environment - Netbeans* * *a server Apache* * *a database server MySql* * *browsers. IE and Firefox.*   ***Please Note****: that Coomera classrooms A03.010 and A03.015 have the appropriate software installed.* |
| **Assessment Context and Description** *(if required)* | Individual Exercise - To demonstrate your knowledge of HTML,CSS, PHP and MYSQL. |
| **Note to Student** | An Assessment Mapping Matrix is available from your teacher upon request. The mapping matrix shows how the knowledge and skills that you are being asked to demonstrate align to the requirements of each Unit of Competency. |

|  |
| --- |
| *An introduction to the projects**The design of the SportsPro Technical Support application* The SportsPro Technical Support application consists of web pages that provide functions for three types of users. First, it lets administrators perform functions such as maintaining the Products, Customers, and Technicians tables of the tech\_support database. Second, it lets technicians perform functions such as updating incidents. And third, it lets customers perform functions such as registering products.  Most projects have you add one or more new pages to the SportsPro application. For example, project 6-1 has you add two pages that let an administrator manage the products in the database. If you complete all of the projects for this book, you’ll have a relatively realistic web application.  **Note that the first number in a project refers to the chapter in the book. So, for example, you can complete projects 6-1 through 6-5 after reading chapter 6. Similarly, you can complete project 7-1 after reading chapter 7. However, some projects don’t require all of the preceding chapters. That’s why each project lists the required chapters.** *The design of the tech\_support database.* The tech\_support database is used to track technical support incidents. It consists of the seven tables shown in the diagram that follows. The incidents table contains one row for each technical support incident. Each row in the incidents table is related to one row in the customers table, which contains information about the company’s customers; one row in the products table, which contains information about the company’s products; and one row in the technicians table, which contains information about the company’s technical support staff.  In addition, a table named registrations keeps track of the products that are registered to each customer, a table named countries stores the countries of the world, and a table named administrators stores the usernames and passwords for the administrators. Note that the administrators table is not related to any of the other tables. |
| *In addition to the column data types shown above, you should know that the customerID, incidentID, and techID columns in the customers, incidents, and technicians tables are AUTO\_INCREMENT columns. So, the values of these columns are set automatically when new rows are added to these tables. For more details about this database, you can use WorkBench to view the structure and data that’s stored in the database.* *How to install the database* *To install the tech\_support database, you can start phpMyAdmin and run the tech\_support.sql file that’s provided by your instructor.* *How to restore the database* *As you test some of the projects that you develop, you’ll need to add, modify, and delete rows in the database. Then, at some point, you may want to restore the original data. To do that, you can use phpMyAdmin to run the tech\_support.sql file again. This deletes both the structure and the data of the current tech\_support database and restores the original database.* *How to structure your directories* *As you develop the web pages needed for each project, you will need to decide where to store the files needed to implement each project. To keep each project independent of other projects, make sure to store each project in its own directory. For example, store the Manage Projects project in its own directory. This directory should have a descriptive name such as manage\_projects or project\_manager.* *How to format the web pages* *As you develop the web pages needed for each project, you will need to apply some formatting to them. To make that easier, you can use the* ***main.css*** *file that’s provided by your instructor. If necessary, you can modify this file, but it contains all of the tags needed to format the pages as shown in this document* *A starting point for the projects* *To make it easy to get started, your instructor may give you the tech\_support directory. This directory includes some of the files for a web site that can help you get started with the projects. These files include the* ***tech\_support.sql*** *file that you can use to create the tech\_support database, and a* ***main.css*** *file that you can use to format the web pages. If you run the web site, it displays a menu like the one shown here:*    *Most projects correspond to one of the links on this menu. However, at this point, if you click on any of these links, they display a message that indicates that the page is under construction. That’s because you still need to write the code that implements these projects.* *The projects* he description of each project includes an image of how the pages should appear in a browser, a description of how the pages operate, and specifications for how the project should be coded. This information is detailed enough for you to complete each project. However, you’ll need to use your best judgment on how to code many of the details. To do that, write the code in the way that you think is best, based on the skills that were presented in the book.  *Unless you’re instructed otherwise, you can implement each project using any programming techniques you wish. In some cases, however, the project’s specifications will direct you to use a specific programming technique. For example, a project may direct you to use sessions. In that case, you should implement the project as directed.*  *Project 6-1: Manage products*  For this project, you’ll create an application that lets an admin user view and delete existing products. In addition, this application lets the user add new products by entering the product information into text boxes. (*Required reading: chapters 1-6*)  The Product List page    Operation   * When the user clicks the Delete button for a product, the product is deleted from the database. * When the user clicks the Add Product link, the Add Product page is displayed. * *When the user clicks the Home link, the main menu is displayed.*   The Add Product page    Operation   * When the user enters the data for a new product into the text boxes and clicks the Add Product button, the product is added to the database and the Product List page is displayed again, so the user can view the newly added product. * When the user clicks the View Product List link, the Product List page is displayed.   Specifications   * *Validate the data the user enters on the Add Product page to be sure that the user enters a product code, name, version, and release date. If this data isn’t provided, display an Error page that indicates that a required field was not entered.* |
| 2: Manage technicians  For this project, you’ll create an application that lets an admin user view and delete existing technicians. In addition, this application lets the user add a new technician. (*Required reading: chapters 1-6*)  The Technician List page    Operation   * When the user clicks the Delete button for a technician, the technician is deleted from the database. * When the user clicks the Add Technician link, the Add Technician page is displayed. |
| The Add Technician page    Operation   * When the user enters the data for a new technician into the text boxes and clicks the Add Technician button, the technician is added to the database.   Specifications   * Validate the data the user enters on the Add Technician page to be sure that the user enters data in every text box. If this data isn’t provided, display an Error page that indicates that a required field was not entered. |
| 3: Manage customers  For this project, you’ll create an application that lets an admin user maintain customer data. To start, this application lets the user select an existing customer. Then, the user can view or update the customer’s data. (*Required reading: chapters 1-6*)  The Select Customer page    Operation  When the user enters a last name and clicks the Search button, the application displays a table of customers with the specified last name.  When the user clicks the Select button for a customer, the data for that customer is displayed on the View/Update Customer page. |
| The View/Update Customer page    Operation   * When the user clicks the Update Customer button for a customer, the application updates the database. The user can also click the Back button or the Search Customers link to return to the Search Customers page without modifying the database.   Specifications   * **US is the country code for the United States.** |