# Appendix A

***Assignment***

# Dalton Consulting

### *Case Study*

Dalton Consulting is an Information Technology consulting company that provides software and networking consulting services. Dalton IT designs and develops new software applications and networks and updates existing programs, databases and network infrastructure for their clients.

Dalton wants to develop a web based application to help manage its projects and staffing and has chosen your company to develop the web-site.

You must select an appropriate data-access Application Programming Interface (API) and code the data-access layer for this web application.
You will also be required to create a thorough test plan with adequate test cases/data and then test this application and document the results.

The Dalton database holds information about clients, projects and skills required for the projects as well as the staff (consultants) working on the projects and the skills they possess. It also holds information about the allocation of staff to the various projects and details of the project managers. (an SQL script of the Dalton database is provided for you)

There will be a second database with a table that holds the username and password for staff that are allowed to log into the websites private access area to view the Dalton database data. (you will need to create a database called “Login” and a table called “tblUsers”).

**Tasks:**
You are required to develop a range of web based scripts to view, add, edit and delete records in the database. The application must

- follow good web design principles
- use professional programming techniques including the use of code libraries, sound software re-use, strong data validation and sound error handling.
- be a multi-layer application that implements a Presentation Layer, Business Logic Layer and a Data access layer.

You are required to consult with the teacher (who will act as your client) to establish the requirements of the system and ensure the completed web-site meets those requirements.

You may be required at any time to show progress of this assignment to your teacher.

Milestones will include:

- Creating a set of HTML/CSS web pages as the front-end of this application.
- Create the DAL functions to connect to and interact with the Dalton database
- Create a database/table to hold information of authorised users
- Testing of your application to meet customer requirements.

**There are three parts to this assignment:**

1. **Creating a HTML/CSS front-end.**
2. **Coding the data-access layer in PHP to access the back-end (database) of the web application**
3. **Testing the completed application**

## Part 1

This part of the project requires you to create a prototype Website that you can demonstrate to the client (your teacher) and thus allow the client to provide feedback and request any changes he/she requires.

**The basic content of the prototype website should include: (3 individual Web pages)**

- A separate banner that you have designed yourself to be displayed at the top of the page. The banner should include a logo Dalton IT Consulting (which you will design yourself) and will appear at the top of each web page.
- A separate CSS file that includes all styles used in the web pages
- The 3 webpages should include as a minimum:
  1. A home Page – which shows basic information about the organisation.
  2. A content page that displays information including
     - Some details about the company. (You make up this information)
     - The types of project work the company is capable of doing. (This should include pictures and text – look at the data in the dalton database to give you an idea of what to write here).
  3. A "Contact Us" page that includes a form that will collect the personal details and contact details of potential clients. The page should include a wide range of input types (text fields, text areas, radio buttons, check boxes, Lists etc.) that should all be styled via the CSS stylesheet. This page should also include general contact details of the company.

*(**NOTE**: more webpages will be added in Part 2 of this project)*

It is recommended that you read the article "9 Essential Principles for Good Web Design" on the PSD Tutsplus website on the following link:
http://psd.tutsplus.com/tutorials/designing-tutorials/9-essential-principles-for-good-web-design/
and make sure you apply the good design principles to your prototype. You should also use your knowledge gained during the "Web Design" subject you completed last semester.

You will be required to demonstrate your design to the client (your teacher) to receive feedback.

**NOTE: There are no PHP or MySQL requirements for this part of the assignment and it will not be officially assessed.**

**YOU SHOULD CREATE A TEST PLAN FOR YOUR APPLICATION BEFORE MOVING ON TO PART 2 *(see part 3 at the end of this document for more information)*.**

## Part 2

**Part 2 of the project requires:**

**A set of database driven web pages that view, add, edit, and delete records from individual tables in the database,**

**General Items**

1 Use the prototype you developed for part 1 as the starting point of part 2 of the system.

2. You should already have a separate CSS file that includes all styles used in the website so far. Styles should be included in this CSS file for the tables and other form elements that you include as part 2.

3. All data entry or changes must be validated in a professional manner. All validation must be carried out using PHP and not JavaScript. A separate library should be set up for these functions.

4. You may require researching the internet regarding validation and determine what other aspects you should be providing protection against (there are many available and free PHP scripts that provide basic functionality). Incorporate these into your validation. If you use Public domain functions, provide a link to the online content in a readme.txt file or as an embedded comment in your code.

5. Software re-use is important, you should aim to include all database functionality in a PHP function library and endeavour to have generic functions that could be re-used both in this and other applications.

6. Comment your code as much as possible to explain your code including documenting database connectivity.

7. Refer also to the “Assessment Criteria” section for details of what will be deemed satisfactory in this assessment task (this can be found in the “Assessment Instructions for Students” document). This will give you an idea of what needs to be completed or attempted at a satisfactory level to pass this assignment.

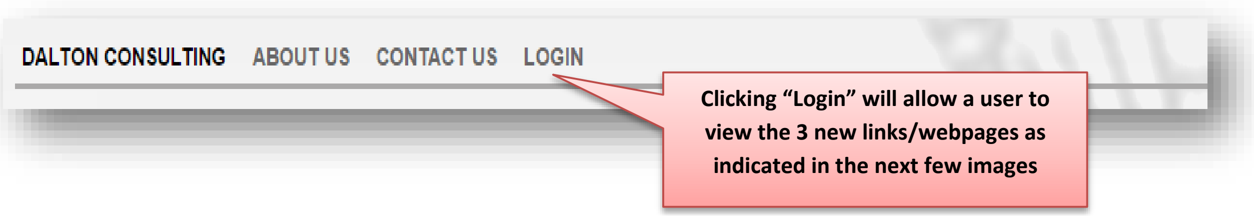
## Part A: (Build the Database)

Import into your UniServer the provided SQL (dalton.sql) script located on Brightspace. This script creates the database and tables and inserts some test data into the database.

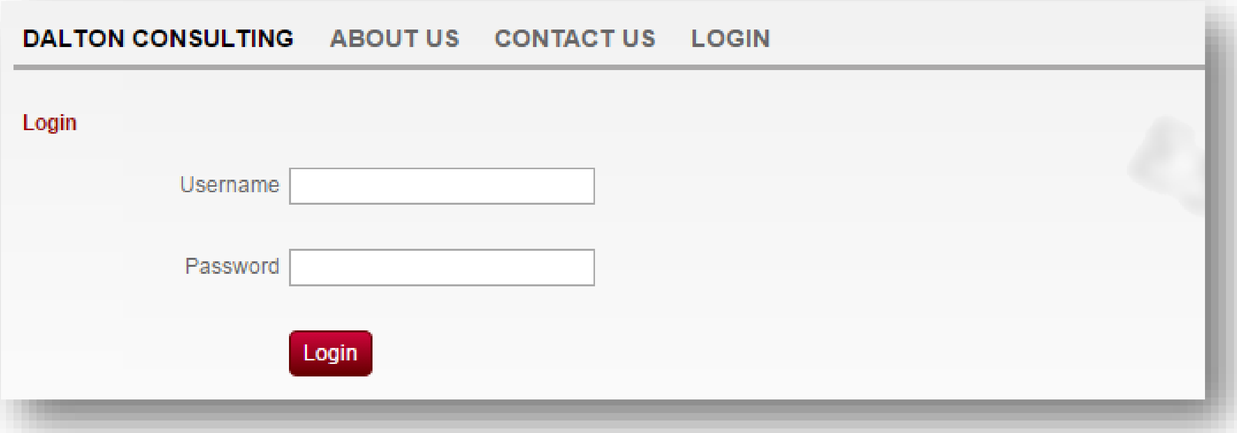
## Part B: Log in Link/Page

Create a new link in your website navigation bar called “Login”. This will allow authorised users to login to the websites restricted area (by connecting to the Login database) to see new links that provide access to the following webpages that you will create.

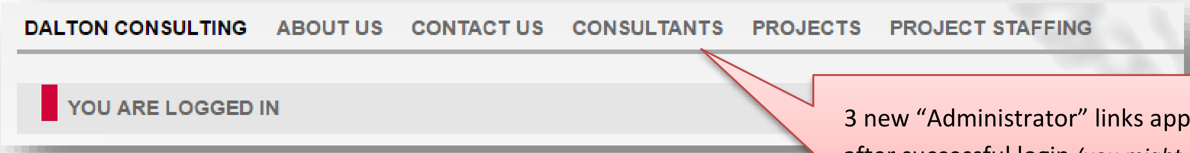
You will be required to create a new database and table and then add some users. One of the users for testing purposes will have the username of “**admin**” and a password of “**pass**”.



Clicking “Login” will allow a user to view the 3 new links/webpages as indicated in the next few images



*Note: Three new links appear in the Navigation bar (Consultants, Projects, Project-Staffing) after successful login.*

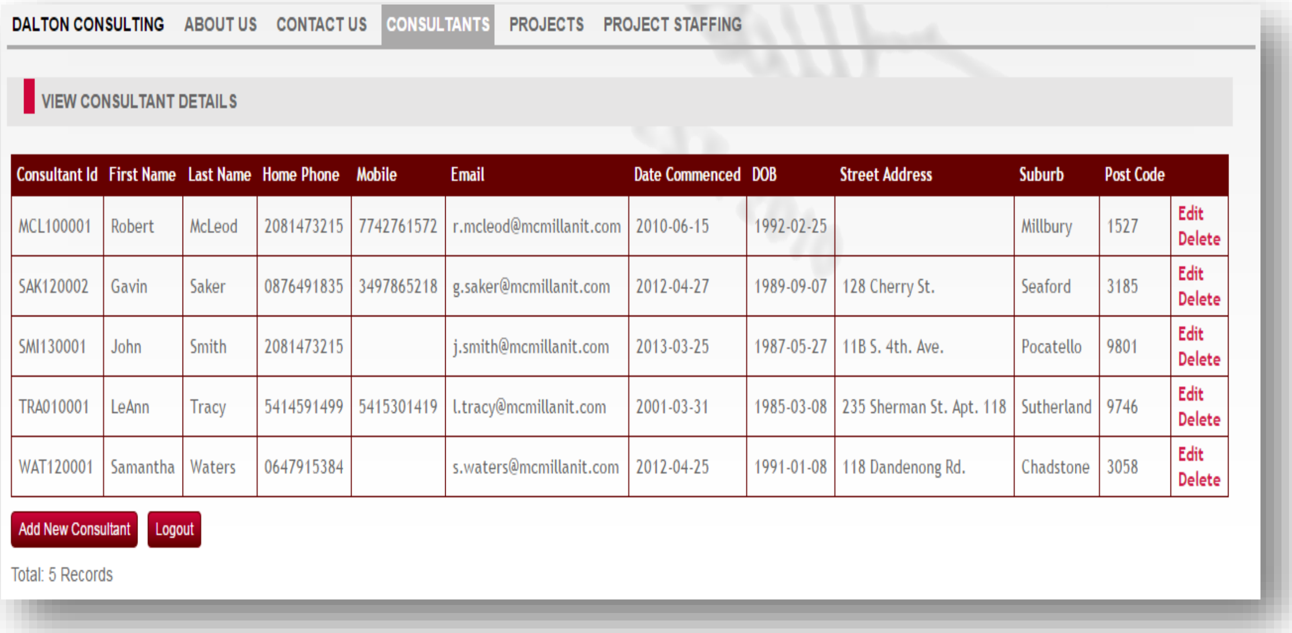


3 new "Administrator" links appear after successful login *(you might also include a logout link/button)*

## PART C: Create 3 new database linked webpages.

### "Consultants" Webpage

1. View Consultant -> A table/form that displays all records from the **d_consultant** table. It should be structured in the following way although styling, colours and fonts should be changed according to your liking.



| Consultant Id | First Name | Last Name | Home Phone | Mobile | Email | Date Commenced | DOB | Street Address | Suburb | Post Code | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MCL100001 | Robert | McLeod | 2081473215 | 7742761572 | r.mcleod@mcmillanit.com | 2010-06-15 | 1992-02-25 | | Millbury | 1527 | Edit<br>Delete |
| SAK120002 | Gavin | Saker | 0876491835 | 3497865218 | g.saker@mcmillanit.com | 2012-04-27 | 1989-09-07 | 128 Cherry St. | Seaford | 3185 | Edit<br>Delete |
| SMI130001 | John | Smith | 2081473215 | | j.smith@mcmillanit.com | 2013-03-25 | 1987-05-27 | 11B S. 4th. Ave. | Pocatello | 9801 | Edit<br>Delete |
| TRA010001 | LeAnn | Tracy | 5414591499 | 5415301419 | l.tracy@mcmillanit.com | 2001-03-31 | 1985-03-08 | 235 Sherman St. Apt. 118 | Sutherland | 9746 | Edit<br>Delete |
| WAT120001 | Samantha | Waters | 0647915384 | | s.waters@mcmillanit.com | 2012-04-25 | 1991-01-08 | 118 Dandenong Rd. | Chadstone | 3058 | Edit<br>Delete |

Total: 5 Records

2. The form should also include a button/link that has the text "Add New Consultant" positioned below the last record.

3. When the user selects the edit link from a particular record in the table a separate form is called and the details for that consultant are displayed ready to edit. The data fields should be displayed one under another. When the user has updated the details they click a button to save the changes and the web page allows the user access back to the view consultant table data which is refreshed to show the new details.

4. If the user selects the delete link from a record in the table a dialog is displayed asking the user if they want to delete the record. It should indicate in the message the Consultant that will be deleted. You should not allow deletion of the Consultant if the Consultant has existing projects or skill records attached. Provide a message to the user if deletion cannot occur.

5. If the user clicks the "Add New Consultant" button a new form is run that allows the user to add a new Consultant. The fields should be displayed one under the other in a similar fashion to how the edit form is arranged. A button to save the record should be included under the last field. When it is clicked the data is saved to the database (after it has been validated) and the user is provided access to go back and view all the records once again.

6. The "Consultants" form showing a list of all consultants (staff) will be called from a menu option on the Home Page. The other pages (add, edit, etc…) will only be available from the links contained in the Consultant form.

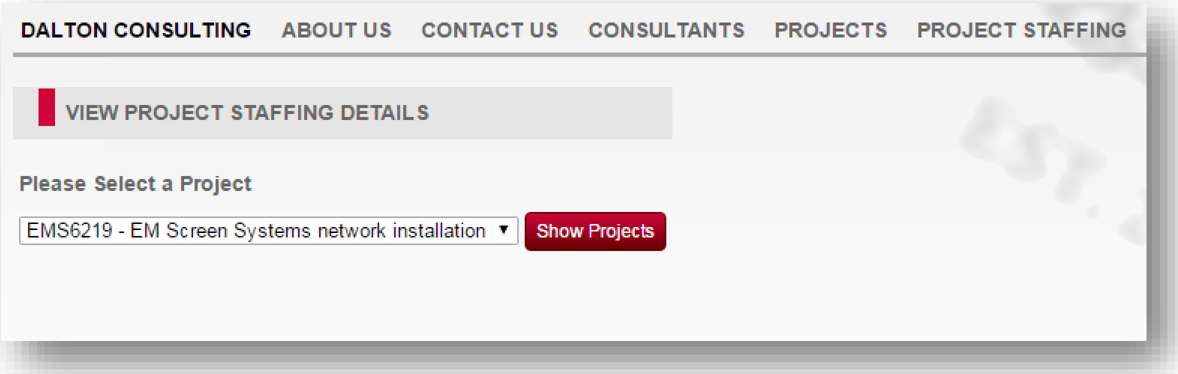
### "Projects Webpage"

A set of Web Pages that work in the same way as the consultant Web Pages that you created above and allow the user to View, Edit, Update and Delete records from the **d_project** table.

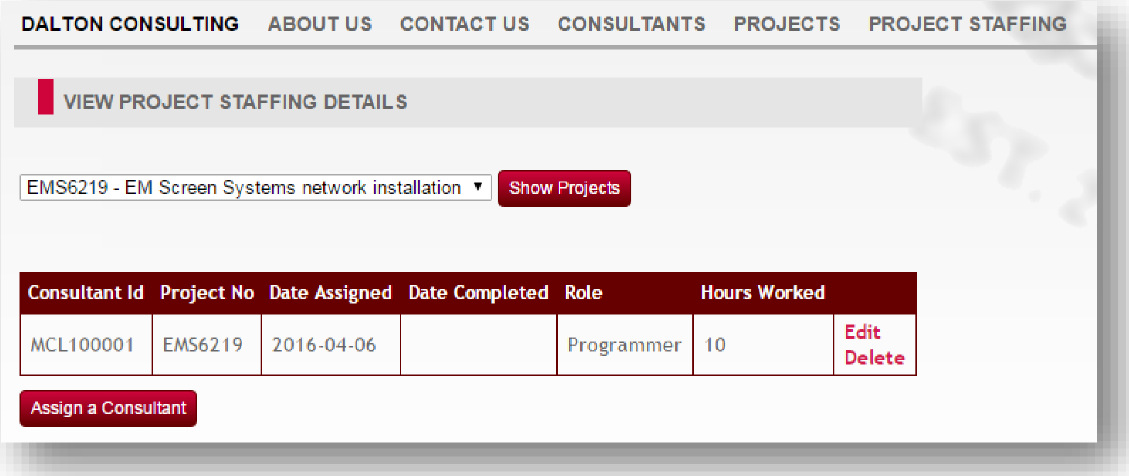
### "Project-Staffing Webpage" *(View Projects and Staff (consultants) together)*

A page that displays a list of all current projects (list is populated from the d_project table of the database), allows the user to select a project and then click a button to display all the d_project_consultant records from the database for that selected project.

The web page should initially be structured in the following way but with your own styling.



After clicking the "Show Projects" button, only the consultant(s) for the selected project are displayed.



VIEW PROJECT STAFFING DETAILS

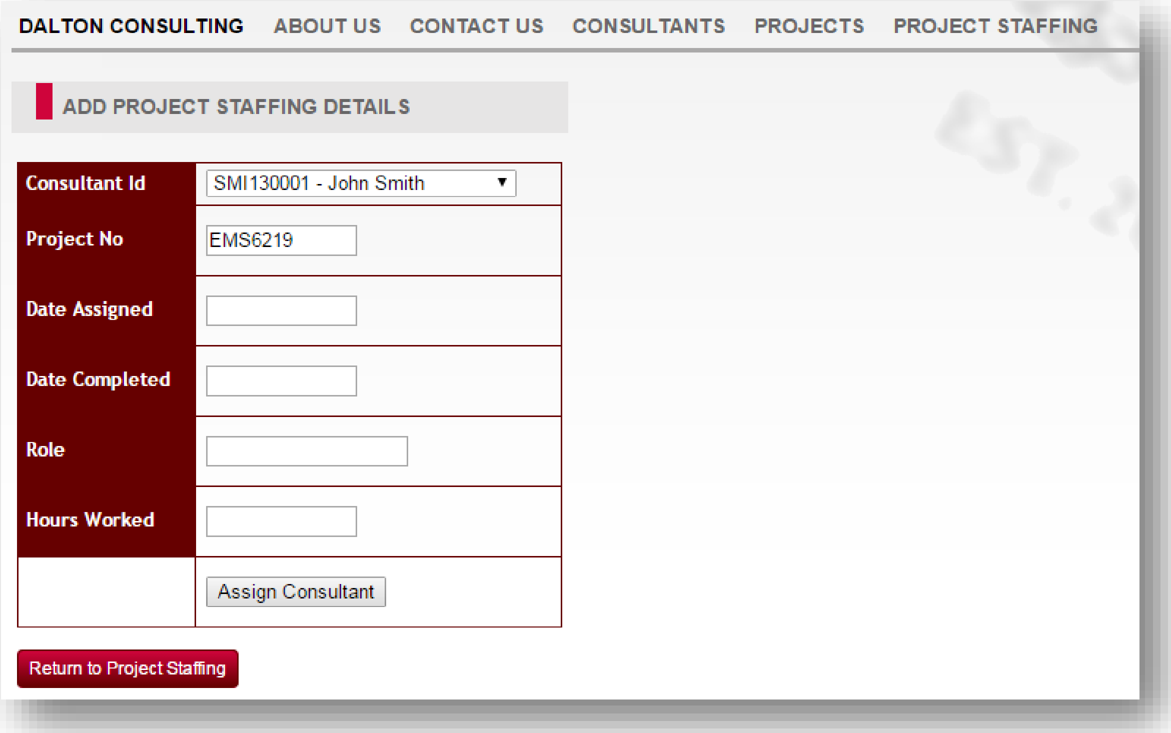
EMS6219 - EM Screen Systems network installation ▾ Show Projects

| Consultant Id | Project No | Date Assigned | Date Completed | Role | Hours Worked | |
|---|---|---|---|---|---|---|
| MCL100001 | EMS6219 | 2016-04-06 | | Programmer | 10 | Edit<br>Delete |

## Assign a consultant to a project

A form is displayed when the "Assign a Consultant" button is clicked with some inputs already populated. This allows existing staff to be assigned to the project (adding a record to the **d_project_consultant** table). The page should include a selection list for the Consultant ID (and Consultants name appended) that is populated from the database. It should also include a selection list (based on static values) for the consultant's 'Role'. The roles in the list should be Programmer, Analyst, Software Architect, Project Manager, Database designer, Network Engineer, and Database administrator. *(**note**: this is not implemented in the next screenshot)*

The web page should be similarly structured in the following way although as mentioned above the "Role" should be a selection list rather than a textbox. A link at the bottom of the page should return the user to the "Project-Staffing" webpage/table.



ADD PROJECT STAFFING DETAILS

| | |
|---|---|
| Consultant Id | SMI130001 - John Smith ▾ |
| Project No | EMS6219 |
| Date Assigned | |
| Date Completed | |
| Role | |
| Hours Worked | |
| | Assign Consultant |

### Update consultant details

A page that is called from the 'Edit' link found next to each record. Clicking this link allows existing staff details to be updated. When the link is selected the update form should open with the matching record from the **d_consultant_project** table (consultant's record)

The page should include list for the Consultant ID that is populated from the database. It should also include a list (based on static values) for the consultant's 'Role'. The roles in the list should be Programmer, Analyst, Software Architect, Project Manager, Database designer, Network Engineer, and Database administrator.

The web page should be structured in a similar way to the 'Assign a Consultant to Project' form/webpage and provide similar feedback and the link at the bottom of the page working in the same way.

All the PHP coding above will be carried out in the data access layer (DAL folder in your website). This will have a function to establish a connection to the Dalton database on UniServer.

This function will:

- Create and manage a connection string that will include the name of the server, database, authorised user of the webserver and his/her password.
- Connect to different data sources by using different databases – in addition to the Dalton database there will also be a database of authorised users that can log into the website.
- Code must be created to handle any connection exceptions such as webserver not online, incorrect connection string data, incorrect username/password, etc…

The authorised users will be allowed to execute commands on the data source such as retrieving and managing data as a result set and displaying it in a HTML table, insert new records into the database and update and delete records. Code will have to be written to handle any exceptions or entry of invalid data.

Your program must manage disconnected data by recognising that the server is offline and that an appropriate message is displayed to the user of the website.

You will have to fully document your code with comments to allow other programmers an idea of what each function does and how the PHP code interacts with the database data.

## Part 3 – Testing

Your web application needs to be thoroughly tested during the construction/coding phase and also when complete. Before you start testing you will need to plan the testing phase of this project before the application is complete. You should produce two documents for this part (use the templates provided to you this semester).

1. A Test Plan (you can have a separate document for the test cases)
2. A Test Report

The suggested tasks are as follows:

Create a test plan after determining the requirements of the application and gathering the relevant information in regards to testing roles, testing types, etc… Create test cases and relevant test data and document these in your test plan (or in a separate test case document).

Test your application and record the results of your testing. Analyse these test results and produce a test progress report document indicating what went wrong and what defects where detected and how you will manage these defects in the code/application.