

Monash University

FIT5192 Enterprise and Internet Applications Development – S2B 2016

Assignment 2: Website Development Using ASP.NET

This assignment is worth **25%** of the final mark of this unit, and it is a **group** assignment of maximum **two** students for each group. You can also choose to do this assignment on your own as an individual work.

This assignment will have a submission only (**NO demonstration**), due on **Friday, 26th August 2016 by 11:55pm**. In this assignment, you are asked to construct **TWO websites for a business or organisation**, using different ASP.NET technologies. The business or organisation may be real or imaginary. However, if you choose a real business, you must ensure that you do not break copyright.

The assignment 2 contains the following two major tasks:

- **Develop an ASP.NET Web Form-based Website** (70%, which is worth 17.5 marks of the final mark of this unit)
- **Develop an ASP.NET MVC 4 Website** (30%, which is worth 7.5 marks of the final mark of this unit)

Task One – Develop an ASP.NET Web Form-based Website

The site will be made up from XHTML and ASPX (using C#) documents as indicated below:

Please note the following requirements:

- The name of the company shown here is indicative only. Make up your **own name, graphics and colour scheme**.
- The site is to use a **master page**. The master page contains the material across the top, down the sides and along the bottom. The screen shot below shows the home page loaded.
- Use a **skin and a style sheet** to set display features.
- Use a **fly out menu** (using a site map) in the master page for site navigation. The menu items must be set out as shown: a *Home page*, *Registration page*, *About Us* page with two sub-pages (*Our history* and *Frequently asked questions*), a *Site map* page and a *Documentation* page. Set tooltips for menu items.
- Add breadcrumbs as shown. Use either a small graphic or conventional arrow (-->).
- Add a calendar control to simply display a calendar as shown.
- Use at least three images as part of the layout. Make sure the images are copyright free.

NOTE: ONLY THE MASTER PAGE, HOME PAGE AND REGISTRATION PAGE WILL BE VIEWABLE UNTIL THE USER HAS LOGGED IN (SEE LOGIN SECTION BELOW)

1. Master Page

It is recommended to use only one ASP.NET Master Page in this assignment. The menu must be contained within the Master Page. Use a skin and a style sheet to set display features.

The **menu** must be provided via an ASP.NET menu navigation control and provide links to five other pages, which are displayed in the main screen area. The order of pages in the menu must be the same as the **order** of pages given below. The five other pages will be as follows:

2. Home Page

This is the home or title page. It will have a graphic and some display text to **introduce the topic of the website**. It will clearly indicate what the site is about and will contain an **AdRotator control to display three banner advertisements** appropriate to your website.

At the bottom of this page include the following:

- A link to the standard Monash course disclaimer:
<http://users.monash.edu.au/~sgrose/msh/disclaimer.htm>
- Email links to the page author and the server webmaster (use your own Monash email address for both).
- A copyright notice. The notice must only refer to the material to which you, the author, own copyright. This will include all the text and your original graphics.
- Acknowledgements. Acknowledge the sources of any material which is not owned by you, and which is either copyright free or for which you have permission to use: e.g. clipart. However, original sources of all images used in your site need to be put in your documentation web page.
- **AdRotator** Place this image at the bottom of this page. Make the image clickable to open up a new tab in the browser, in which is displayed the full page code for this page and its associated XML file. Use ASP.NET to dynamically read and display this code, ie DO NOT copy and paste the code into a file for display.
- **Master Page** Place this image at the bottom of the page. Make the image clickable to open up a new tab in the browser, in which is displayed the code used for the Master Page as described above.

3. Registration

On this page present an information gathering form for the user to register with the site. The form must include at least **eight** data entry fields, and include at least one of each of the following: dropdown list, listbox, textbox, radio buttons list and checkbox. Use a **RequiredFieldValidator** validation control to ensure that every text input field is entered by the user. Also include at least one of the following for each text input field: CompareValidator control, CustomValidator control, RangeValidator control and RegularExpressionValidator control. Make sure you use each of these controls at least once. DO NOT use a ValidationSummary control for this. Please make one of the fields contain a valid URL (any URL will do, it does not have to be relevant to your site topic).

In addition to displaying the user-entered data as required above, this page must also provide for the contents of the form to be written to an **Access database**. Just use one table in the database, which will include all the fields of the form. Make sure that the database has a key field, e.g. a customer ID, which is unique for every record and which the user does not have to enter themselves and that each record contains an **email field**.

Add Records Place this image at the bottom of the page. Make the image clickable to open up a new tab in the browser, in which is displayed the code used for this page as described above.

Add a button or link to the page, with suitable formatting attributes. The text on the button/link will be: "Click here to see all the xxx", where "xxx" is replaced with a term, like "customers", appropriate to the website. When the user clicks this button/link a new tab is opened in the browser, and all of the records in the database are displayed, with suitable heading and formatting. Use a GridView to display the records. Use the formatting features of the

GridView control, including **BoundField**, **HyperlinkField** (for the URL field of the database), and a **ButtonField**.

The **ButtonField** is used to display the data of a selected row of the GridView, under the **GridView** using label controls contained within a Panel control (Format via a CSS to create attractive output). Again, please make one of the fields contain a valid URL.

Use an **Access DataSource Control** for this page. Marks will be deducted if you do not use a GridView and/or do not use BoundFields and at least one ButtonField and HyperlinkField.

Display Records

Underneath the output place this image, which, when clicked opens a new tab in the browser and displays the code, as indicated above. Add another button/link to this page, which when clicked, will open a new tab in the browser and allow the user to **select a single record and have it displayed**. The user can enter part of the contents of any of three suitable fields (e.g. customer ID, customer name or customer address field) and have the whole record displayed, with suitable formatting. If there is no record that matches the user input, then a suitable message is displayed to the user.

Search Records

Underneath the output place this image, which, when clicked opens a new tab in the browser and displays the code, as indicated above.

4. About Us

The About Us page, has 2 sub pages -Our History and Frequently Asked Questions. Make up some information for each of these 3 pages, which is relevant to your site (around 15-20 lines).

About Us

Place this image at the bottom of the page. Make the image clickable to open up a new tab in the browser, in which is displayed the code used for the About Us, Our History and Frequently Asked Questions pages.

5. Site Map

The site map page shows a site map of the site, using a **TreeView control** and the **Web.sitemap** file.

Site Map

Underneath the output place this image, which, when clicked opens a new tab in the browser and displays the code, as indicated above.

6. Documentation

This page will provide assignment documentation.

- Details of the author: name, student ID, this unit's name, unit provider (Monash University, The Caulfield School of Information Technology), assignment number, date of submission, lecturer/tutor's name, an email link to the author, a link to this assignment specification.
- **CSS** Place this image at the bottom of the page. Make the image clickable to open up a new tab in the browser, in which is displayed the code for the Cascading Stylesheet.

Skin

- Place this image at the bottom of the page. Make the image clickable to open up a new tab in the browser, in which is displayed the code for the skin file.

7. Other Features

Construct a **login system** for your site. Any user may load the title page, but to go to any of the subsidiary pages, the user is first diverted to a login page. This login page will have two text boxes (one for a username and one for a password), and a button. Both boxes need to be checked that the user does not leave either empty before clicking the button.

When the user enters a username and password, these are checked against a set of usernames and passwords in an **Access database**. If the user logs on correctly, then the user is taken to the page originally intended, if not then they are required to try to login again.

In the database, please set up an **administrator account** as the following, where the username and password are the same: "admin".

Login

Use this image to display the code for your login page, and the **web.config** file. Note: in implementing your login system, place all the pages that require a login challenge, in one sub-directory, called "locked" for example. Then place the two web.config files given below in the appropriate directories.

Place the following web.config file in the root directory of your application: i.e. in your root directory, above the "ass2" sub directory. Also place the login page in your root directory.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <customErrors mode="Off"/>
    <authentication mode="Forms">
      <forms name = "ass2"
        path="/"
        loginUrl="login.aspx"
        protection = "All"
        timeout = "10" />
    </authentication>
  </system.web>
</configuration>
```

Create a directory called "locked" inside your "ass2" directory. Place this web.config file in the "locked" sub-directory.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
  </system.web>
</configuration>
```

```
</authorization>
</system.web>
</configuration>
```

Optional Requirements - High Distinction

Completing the requirements stated above may earn you up to a Distinction grade for this assignment. In order to earn up to a High Distinction grade you must complete the following optional requirement.

8. Email

This page will be reached by a separate menu item and will display a list of customer, clients, registered users (or whatever you have called them) from your database. Display the name, email address and a checkbox for each record. The user will be able to select one or more clients, enter a subject and email and send the message.



Place this image at the bottom of the page. Make the image clickable to open up a new tab in the browser, in which is displayed the code used for the Email page.

9. Calendar

You will need a table in your database, which holds information regarding events. There needs to be an EventDate, EventTime and an EventDescription.

When the page loads the database should be checked to determine if there any events scheduled for today. If so, today's date should be highlighted in the calendar control by using a different background colour and a label below the calendar should display the event description and the event time. If there are no events for the current date, the label should display appropriate text such as "No Events Scheduled for Today".

NOTE: Events can be manually entered into the database. The only requirement is to display events. It is not necessary to be able to add/edit/delete events from this table.

HINTS:

- A Calendar control cannot be bound directly to a DataSource control. You will need to specify an *onDayRender* event handler for the Calendar control. The method signature for this is:

```
void dayRender (object sender, DayRenderEventArgs e)
```
- You can use a DataSource control to connect to the database and then populate a DataView object with the results of the SQL SELECT statement:

```
DataView dv = (DataView) dsEvent.Select
                (DataSourceSelectArguments.Empty);
```

The DataView object is contained within the System.Data Namesapce.
In this example the AccessDataSource has an ID of dsEvent.
The DataView.Count method can be used to determine if any rows have been returned.
- Dates in Access are delimited with the # character. So a valid SQL statement would be

```
SELECT * from Event WHERE EventDate = #10/8/2016#;
```

NOTE: Feel free to implement this using a method of your choice but the functionality must be as described.

Task Two – Develop an ASP.NET MVC 4 Website

In the second part of the assignment 2, you or your group is asked to construct an ASP.NET MVC 4 application. **30%** of marks of this assignment will be awarded for the development of an ASP.NET MVC 4 application. It is worth 7.5% of the final mark of this unit. You are not allowed to change your partner to do this task, if you choose to do the assignment 2 in a group.

Construct an ASP.NET MVC 4 application. The application must connect to a data store, which contains **at least two** related tables/entities, e.g. Customers and Orders, Students and Classes, Car Manufacturers and Cars etc.

1. Master Page

The site must contain a Master Page, which contains a site banner and menu links.

2. Home Page

The Home Page will contain some text describing the site and an appropriate logo/image.

3. Content Pages

These pages must provide full **CRUD** functionality for the database tables to which they refer. The initial view should be an **alphabetically sorted list**. This view will also provide an option to allow the user to **add a new record**. The user can **"drill-down"** from the initial view for further details of a record. The details view will allow for **editing and deleting** of the particular record. When adding or editing a record, the built in **MVC jQuery validation** script library should be used to display appropriate error messages.

4. Other Features

Marks will also be awarded for the use of any other appropriate jQuery or MVC functionality, such as jQuery UI, **up to a limit of 5 additional features**. For instance, a search function for the database tables to which the content pages refer, can be developed using an MVC Custom Template. The search function will be implemented via a textbox, which allows the user to enter a search term and display any records that match this term (or part thereof).

Besides, the following jQuery functionality can be the choices for implementing the search section, such as:

- **row highlighting**
- **alternating row colouring**
- **date-picker for entering dates**

5. Documentation – Reached via the Main Menu

Construct a web page that will provide assignment documentation as the following:

- Details of the author:
 - Name, student ID, this unit's name, unit provider (Monash University, The Faculty of Information Technology), assignment number, date of submission, lecturer/tutor's name, an email link to the author, a link to this assignment specification.
- Image details. You must include the source of all images used in your assignment. This should take the form of a table with 2 columns (Image and Author).
 - The Image column should contain the original image and the
 - Author column should contain the organisation, website address or the person who own the copyright to the image. If you have received permission to use an image from the owner, this should be included here

- Details of any additional MVC functionality you have included, including:
 - Page which contains the functionality
 - Description of the functionality
- Details of any jQuery functionality contained within your application. This should include:
 - Page which contains the jQuery functionality
 - Description of the functionality
 - A link to the source of the jQuery library or libraries you have used

Submitting Assignment 2

The site you develop must conform to all copyright requirements. Failure to observe copyright will cause the assignment to gain a maximum of 50%.

Ensure that all ASP.NET Server Controls are formatted via CSS - there must be NO formatting contained within the control specifications.

The assignment will be corrected using **Google Chrome** web browser. Design your web pages to be viewed with **1024 x 768** screen resolution and higher resolutions. The assignment may be corrected at any screen resolution of 1024 x 768 or higher.

There is NO hard copy submission for this assignment. You are required to submit your assignment as a **“.zip”** file named with your student number. For example, if your student number were **12345678** then you should submit a file named **12345678a2.zip** - **marks will be deducted if this requirement is not strictly complied with.** The softcopy of your assignment submission is to be submitted via the Assignments link provided on the FIT5192 unit's Moodle site by the deadline specified in this document. The only exceptions will be in cases of sickness or other serious cause, for which documentary evidence (e.g. a doctor's certificate) must be provided. **Students must request an extension prior to the submission date.**

Your submission must include the following:

1. Include an electronic copy of your assignment marking sheet (an Excel spreadsheet), which can be downloaded from the unit's Moodle site. The spreadsheet has three worksheets. The first is a cover sheet for the assignment. Read this carefully and then fill in your details in the boxes at the top of the worksheet. The second and the third worksheets are for staff to enter your marks. **DO NOT make any changes to the second and the third worksheets.** Do not rename the assessment sheet file name under any circumstances. After correction, your marks will be added to the assessment sheet. You will receive your resulting sheet via your Monash email address to see your marks. Please note: 20% marks will be deducted for failing to fill out and submit the assessment sheet. Go through the assessment sheet carefully and check that you have done everything for which marks will be awarded. Remember, marks will be awarded for the assignment that has been set, not the assignment you might like to do. Make sure that the various features are on the pages as detailed above. If you place a feature on the wrong page you will be given zero marks for that feature.
2. Only in the case of teamwork, include your completed team contribution evaluation form (FIT5192-A2-2016_TeamContributionForm.docx), which can be downloaded from the unit's Moodle site. Note: In the case of teamwork, not submitting the team contribution form will result in mark reduction of your mark.
3. Include all application files of the two websites. Please setup the IIS server on the lab machine that you use for development. The IIS server is required to run ASP.NET v4.0 or v4.5. Please deploy your websites onto this IIS server, test it thoroughly, and ensure that all files to be submitted will work under these versions. Do not assume that just because the assignment works properly from the Visual Studio IDE, that it will also work properly from

the IIS server you setup. For example, if you have spaces in the names of your files, they may not work properly from the IIS server, but may work from your IDE. To view the two websites after the applications have been deployed, point your browser to (Note: insert your ID number in place of "12345678").:

<http://localhost/12345678/index.aspx> (ASP.NET Web form-based website), and

http://localhost/12345678_mvc/ (ASP.NET MVC 4 website)

The author (or the two authors in case of a group task) must write all submitted web pages. Where there is evidence of similarity with other assignments, students will be heavily penalised.

Plagiarism

Before submitting your assignment, please make sure that you have not breached the University's plagiarism and cheating policy. It is the student's responsibility to familiarize themselves with the contents of these documents.

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students would be severely penalized, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with the following policy from the Plagiarism Procedures of Monash, available at

<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-managing-plagiarism-collusion-procedures.html>