Business Application Programming   INFO102
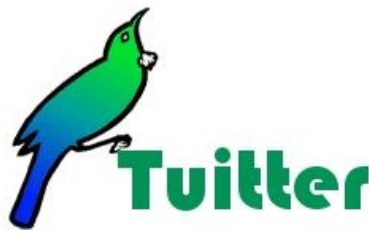

VICTORIA UNIVERSITY
OF WELLINGTON
Te Whare Wananga o te Upoko
o te Ika a Maui

School of Information Management (SIM)
Te Kura Whakaipurangi Korero

Assignment 3
Trimester 2 2011

A microblogging website

Due Date: **Friday 14 October 2011 - 5pm**

Overview

This C# assignment requires you to create a small microblogging website that has some of the features of the well known Twitter website (www.twitter.com).

A New Zealand software company has decided to follow the successful example of the TradeMe.co.nz website which has become the most popular website for online auctions in New Zealand. The software company has decided to ask INFO102 students from Victoria University of Wellington to design a Twitter-like website prototype following some predefined requirements.

Please note:
- *Read the whole assignment before you begin.*
- *The best source of information to help you complete this assignment are:*
  - *Textbook: Bradley and Millspaugh, Chapters 1-9 and 11 (chapter 10 is excluded).*
  - *Lectures 1 to 20.*
  - *Workshops 1 to 8.*

Learning Objectives

In this assignment you will be showing an understanding of the following:

- C# built-in classes, objects, and methods.
- Creation of an ASP.net website using C# code and Visual Studio.
- Design of web pages using HTML tables and ASP.net controls.
- Interface controls including image links, textboxes (single and multi-line), images, ASP.net buttons.
- Variables, constants, and calculations.
- Programming concepts: decision structures, arrays and loops.
- Using session variables.
- Creating and applying CSS style sheets.
- Manipulating files (reading and writing).

Warnings

1. You must write your program in C# 2008. Programs written in C# 2005 or C# 2010 will not be marked.

2. You are responsible for backing up your own work.

3. Plagiarism is not accepted in this course. Please read the course outline for guidelines on the University plagiarism policy.
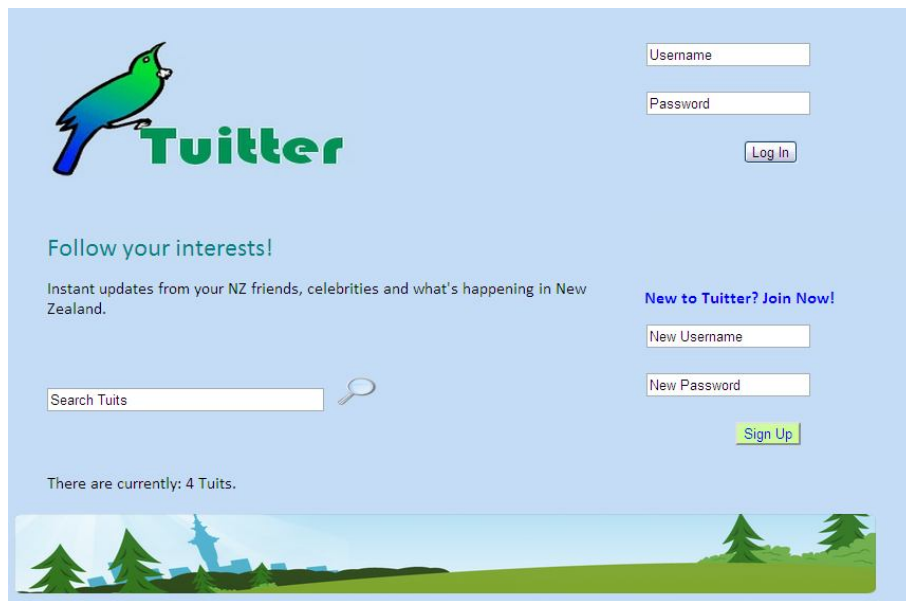
Submission Instructions

1. Name your project using this format: StudentID_TuitterProject_A3 (e.g. 3000123456_TuitterProject_A3).

2. You will need to zip your website project folder and then use file transfer protocol (FTP) to submit the zipped folder to the SIM web server. Instructions for zipping project folders and for using FTP are available on Blackboard. Ensure you know how to do this because if you do not submit your assignment correctly we cannot mark it.

3. The submission system closes at 5pm on the due date. You are strongly advised to submit your work one or two hours before this time.

Technical Notes

1. You can use C# Visual Express 2008 or Visual C#.Net 2008 Professional edition to create this application. Both versions are suitable for this assignment. Instructions for downloading these products for home use are available on Blackboard. For downloading at home you will find the Express edition is smaller.

Specifications

1. You are required to create an ASP.net website using Visual Studio 2008. The website is a microblogging prototype. The website source files must be all stored in a website folder entitled 'TuitterProject'.

2. The website will contain three web pages:

    - tuitterMain.aspx: This is the main page of the website. In this webpage, users will be able to either log in or sign up. They will also be able to search through the existing tuits by entering text in the Search text box of the page and clicking on a search image button.

tuitterMain.aspx

- tuitterDisplay.aspx: In this page, users will be able to see the tuits from all the users who submitted tuits using the system. Users will also be able to send new tuits which will then be displayed in the tuit list.
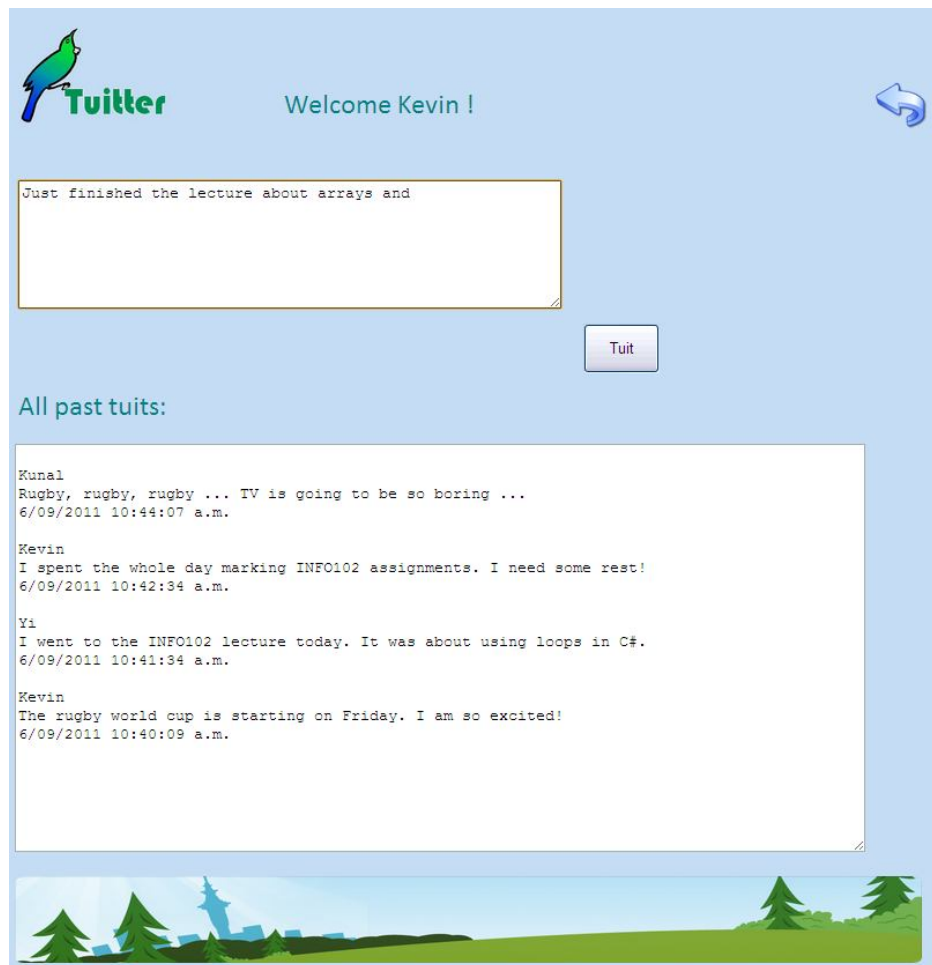
tuitterDisplay.aspx

- TuitterSearch.aspx: This page will display the results of all search queries. Only the tuits containing the searched string will be displayed.
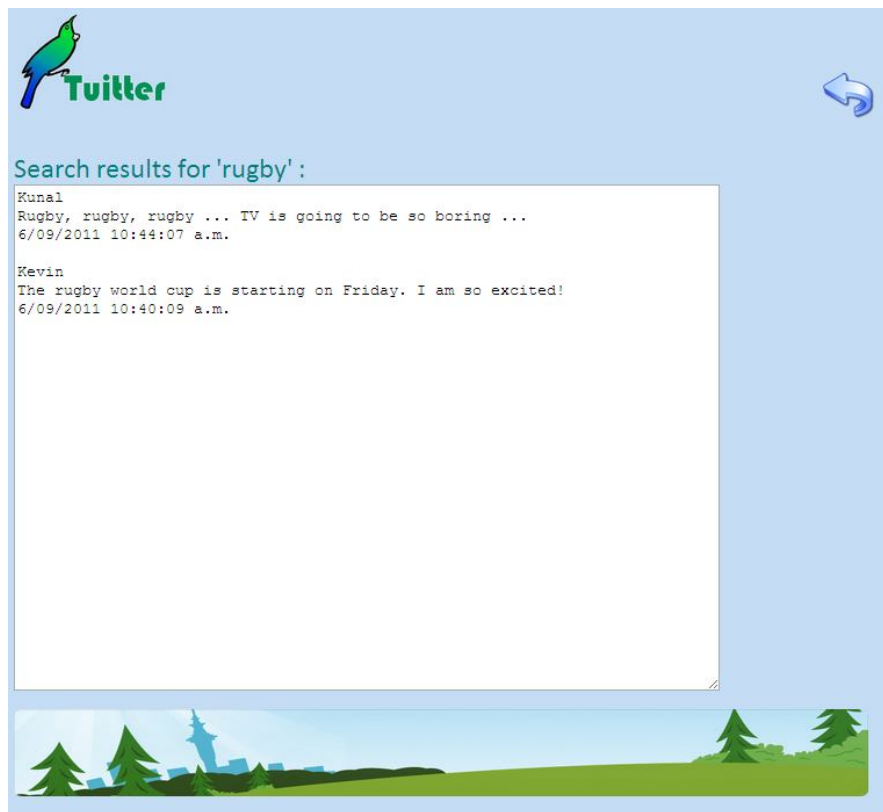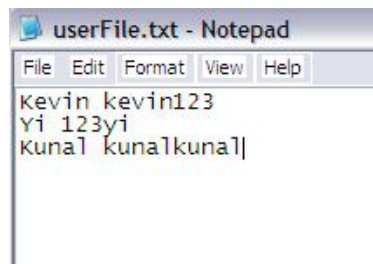
tuitterSearch.aspx

3. Each tuit consists of:
   a) The username of the person who submitted the tuit.
   b) The content of the tuit message.
   c) Date/time when the tuit was submitted.
   d) A blank line (used for display purposes).

```
1  Kevin
2  I spent the whole day marking INFO102 assignments. I need some rest!
3  6/09/2011 10:42:34 a.m.
4
```

1 Tuit

4. In the App_Data subfolder of the website, there will be two files:

   - userFile.txt: it will contain the list of all the usernames and corresponding password (see below):
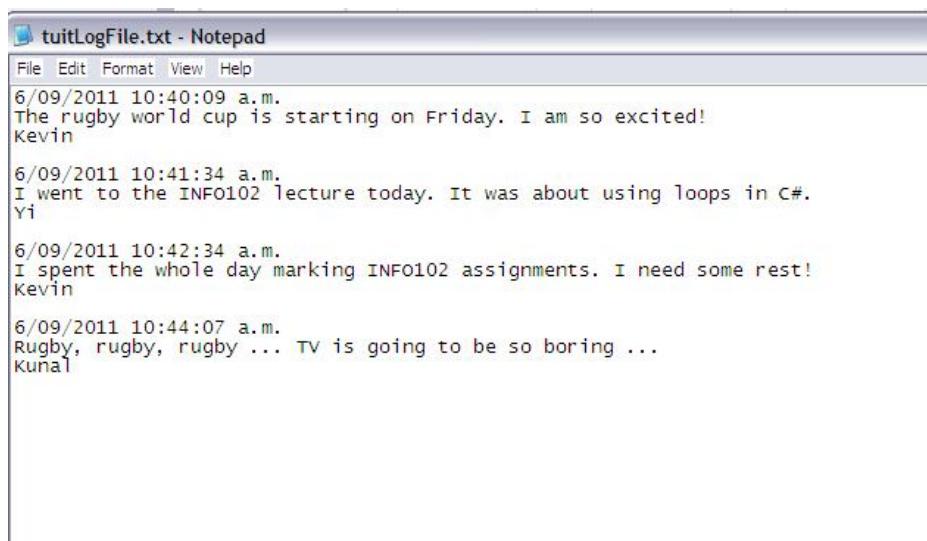
- tuitLogFile.txt: this file will contain all the tuits that have been submitted by the users of the website (see below):

Important: the tuitLogFile.txt will store the past tuits **in the reverse order**. The content of the tuits will be stored using the following sequence: blank line, date/time, tuit content, username. The reason to do so is not to make things more complicated but simply because it is only possible to add text in the end of a file (it is not possible to directly insert text in the beginning of a file).

Tuit in the TuitterDisplay.aspx page:

```
Kevin
I spent the whole day marking INFO102 assignments. I need some rest!
6/09/2011 10:42:34 a.m.
```
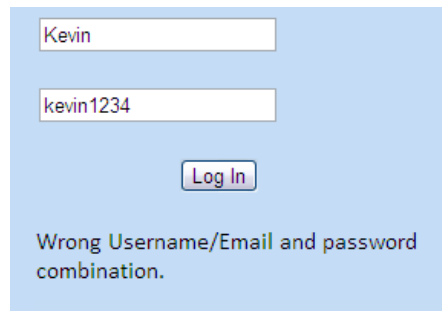
Corresponding Tuit in tuitLogFile.txt:

```
6/09/2011 10:42:34 a.m.
I spent the whole day marking INFO102 assignments. I need some rest!
Kevin
```
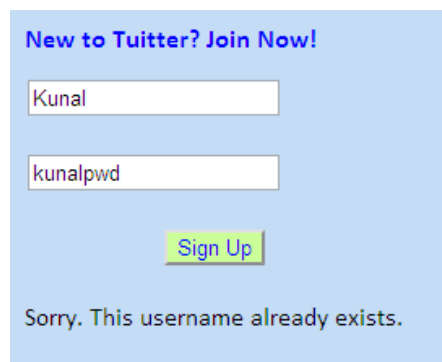
5. Sign in/ Sign up:

- Users whose username is already present in userFile.txt will be taken to tuitterDisplay.aspx only if they enter the right password. Otherwise, an error message will be displayed. The same error message will be displayed if the entered username does not exist.

- New users will be able to create a new username/password only if the username does not exist in the username list (from userFile.txt). If a new user chooses a username that is not already used then the new username/password will be added into userFile.txt. The user will then be taken to tuitterDisplay.aspx. Otherwise, an error message will be displayed:

6. Use of CSS styles: in order to keep the website easy to maintain when adding extra webpages in the future, a new CSS style sheet must be created and attached to each of the project web pages.

The style sheet shall be called TuitterStyleSheet.css and shall contain three styles:
   o .tuitterText: for all controls (see image below).
   o .tuitterHeading: for all three pages' headings (see image below).
   o .tuitterBg: for the blackground colour of all three webpages.
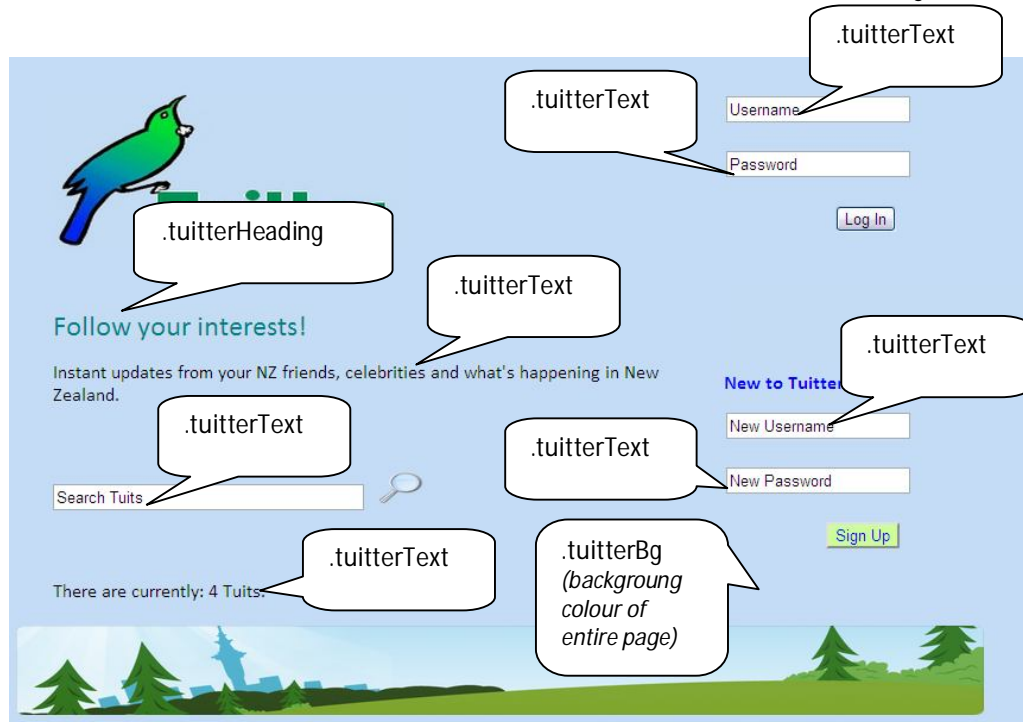
Requirements are as follows:

1	**Coding** All program objects including variables, constants, and methods are named following good programming standards: code layout, naming convention, comments (as specified in the text book).
2	**Webpages** The application will contain 3 webpages: 1. `TuitterMain.aspx`: the main page in which the user will log in, sign up, or else perform Tuit searches. 2. `TuitterDisplay`: a webpage allowing to consult all past tuits and to add new tuits. 3. `TuitterSearch`: a webpage that display the results from a tuit search. • The overall design for all three pages must be based on the use of HTML table in order to locate the various controls within each webpage.
3	**TuitterMain.aspx** The main webpage must contain the controls displayed in the screen captures used in this document. The page will contain: • An Image for the Tuitter logo. • An Image for the page footer. • A textbox and an image button for the search functionality.

|	
	• Two textboxes (for username and password), a 'Log In' button, and a label to display error messages for the log in functionality. • Two textboxes (for username and password), a 'Sign Up' button, and a label to display error messages for the sign up functionality. • A label that will display the current number of tuits. • The 'Title' property of the page will be set to 'Welcome to Tuitter'. **Important**: Some sample images are provided to you for the Tuitter logo, footer, and search button. However, you are **totally free** to use your own images. You are also free to decide the layout and design of the webpage as long as the required elements (or controls) are present.

4	**TuitterDisplay.aspx**
	 The display webpage must contain the controls displayed in the screen captures used in this document. The page will contain:
	 • An Image for the Tuitter logo.
	 • An Image for the page footer.
	 • A multi-line textbox and a 'Tuit' button to submit new tuits.
	 • Another multi-line textbox to display all the past tuits (in read-only mode).
	 • A 'Back' link button taking the user back to TuitterMain.aspx
	 • A label next to the Tuitter logo which displays a welcome message to the logged in user. The current username must be displayed in the message.
	 • Each tuit will be displayed as follows:
	     o The username of the person who submitted the tuit.
	     o The content of the tuit message.
	     o Date/time when the tuit was submitted.
	     o A blank line.
	 • The 'Title' property of the page will be set to 'My Tuitter'.
	
	 Important: You are also free to decide the layout and design of the webpage as long as the required elements (or controls) are present.

5	**TuitterSearch.aspx** The display webpage must contain the controls displayed in the screen captures used in this document. The page will contain: • An Image for the Tuitter logo. • An Image for the page footer. • A multi-line textbox to display the result of the search query (in read-only mode). • A 'Back' link button taking the user back to TuitterMain.aspx • A label indicating the word (or text) that the current user is searching among the existing tuits. • The 'Title' property of the page will be set to 'Search Tuits'. **Important**: You are also free to decide the layout and design of the webpage as long as the required elements (or controls) are present.

6	**`Username/password file: userFile.txt`**
	• A file called userFile.txt must be used to store all the users.
	• The file must be located in the App-Data subfolder of the website folder.
	• Each line of the file will consist of:
	    ○ Username
	    ○ A 'space' character
	    ○ Password

7	**`Tuit log file: tuitLogFile.txt`**
	• A file called tuitLogFile.txt must be used to store all past tuits.
	• The file must be located in the App-Data subfolder of the website folder.
	• Each tuit will consist of four lines:
	    ○ The date/time the tuit was submitted
	    ○ The content of the tuit
	    ○ The username
	    ○ A blank line

8	**Session variables** • You must use a session variable called `Session["User"]` to pass the current username from the TuitterMain.aspx to the TuitterDisplay.aspx page. • You must use a session variable called `Session["Search"]` to pass the searched words from the TuitterMain.aspx to the TuitterSearch.aspx page.

9	**CSS Styles**
	• Using the Visual Studio Style Manager, you must create a style sheet entitled TuitterStyleSheet.css that contains three styles:
	    ○ .tuitterText
	    ○ .tuitterHeading
	    ○ .tuitterBg
	
	• These styles must be applied to the appropriate controls of the three web pages.
	• You are free to define these styles according to your own preferences (font type, font size, colour...).

10	**Extra Feature** • You are expected to add an extra feature of your choice. You are free to add any feature which you think would improve the application. • The extra feature must be briefly explained in a multiple line comment at the top of the Tuittermain.aspx.cs file.

Important Help

1. **Design of the webpages**. Here is a screenshot that may give you some ideas about how to use HTML tables when setting up the layout of a webpage.
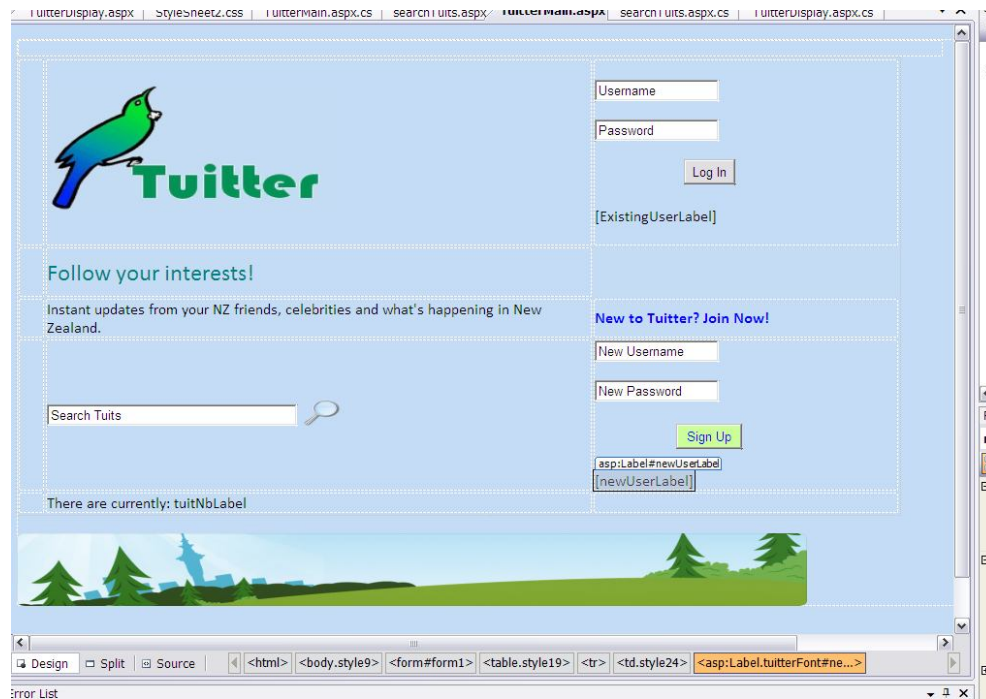
In addition, do not forget to tick the following option in the website option windows:
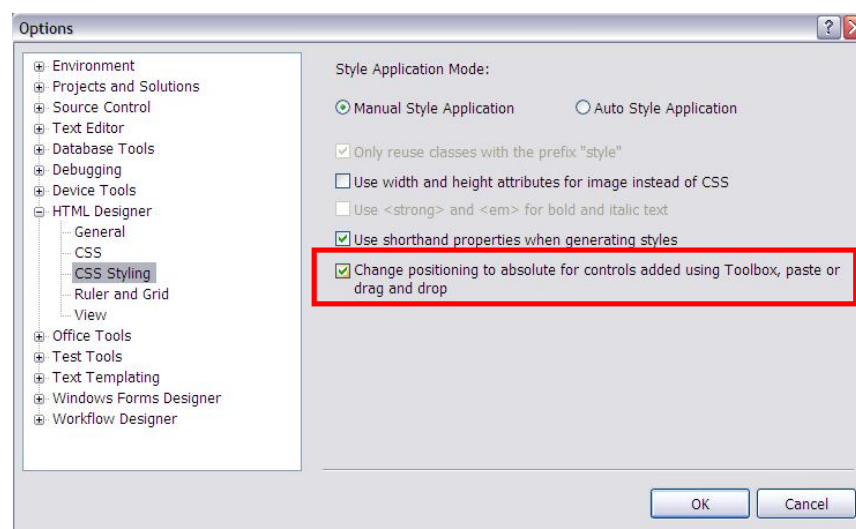
2. **Web page title**. To verify whether you have properly set up the Title property of each aspx web page, make sure that the title is displayed in the Internet browser when running the website:
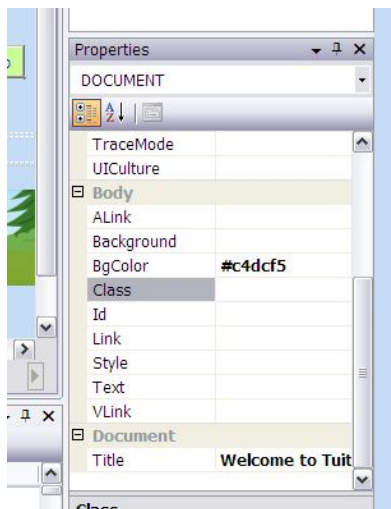
3. **Session variables**. You are free to set session variables in any of the .aspx.cs pages. Once defined, the variable is accessible in any method of the three .aspx.cs pages.

4. **General advices about how to manipulate the user and tuit files**.

    a. Location of the file: you may use the following instruction in order to locate both files:

```
string userFilePath = Server.MapPath("~") + "/App_Data/userFile.txt";
string tuitFilePath = Server.MapPath("~") + "/App_Data/tuitLogFile.txt";
```

    b. From file to array: It is recommended to use the `File.ReadAllLines(filePath)` method in order to read a file and generate an array which items are each line of the file. A simple while/for loop can then be used to:

        o Check the content of each line (to check each username/password)
        o Display the content of a file in a text box (to display the tuits)

    c. Checking username/password: a recommended strategy to retrieve the username and password from a certain line within a file is to use the following method:

```
string[] userPasswordPair = fileLineString.Split(' ');
```

    Suppose
```
fileLineString = "tanya tanyazzz";
```

Then the `Split(' ')` method will return an array containing two elements. The first one will be the username, and the second one will be the password.

```
userPasswordPair[0] = "tanya";
userPasswordPair[0] = "tanyazzz";
```

    d. To add a 'new line character' at the end of the string use the following:

```
currentString += System.Environment.NewLine;
```

- **A strategy for proceeding:**

It is recommended to start with the design of all three web pages before dealing with any file operation.

1. Sketch the three web pages including the HTML tables for each page, indicate the name you will use for each control.
2. Create the project and rename the main web page.
3. Add an HTML table in the page and configure the appropriate number of columns and rows (following your sketch).
4. Add all the required controls in TuitterMain.aspx without taking care of colour, font type and size…
5. Create the two other web pages and add all the required controls (again without setting any style property such as font size/style, colour ….).
6. Add the HTML tables and configure the appropriate number of columns and rows for each page (following your sketches).
7. Add all the required controls in TuitterDisplay.aspx and without TuitterSearch.aspx without taking care of colour, font type and size…
8. Add the necessary code so that you can navigate from one page to another when running the website (back buttons and log in/sign up buttons.
9. Manually create userFile.txt and tuitLogFile.txt and add some random data for both users and tuits. Make sure you follow very precisely the syntax indicated in the requirements.
10. Start with the code in TuitterMain.aspx that deals with clicks on the Sign Up button. Make sure that each time you click on the Sign Up button, the new username/password is added to the file (without considering existing usernames).
11. Once it works, add the functionality that tests whether the new username already exists in the file and display an error message in case it is so.
12. Continue with the code handling clicks on the 'Sign In' button. In both 'Sign Up' and "Log In' cases, do not forget to set `Session["User"]`.
13. Next, take care of displaying the tuits contained in tuitLogFile.txt within the TuitterDisplay.aspx page.
14. Then, continue with the add Tuit functionality.
15. Once everything works fine without any error, implement the Search functionality. Do not forget to set `Session["Search"]`.
16. Finish by creating the CSS style sheet, attaching it to each page, and applying the styles in all 3 web pages.

<u>Indicative Marking Guide</u>

	Guidelines	Total Mark
1	Overall website • Website opens. • Each web page opens and closes according to requirements. • Project and webpage names are correct. • Files in right location • Quality of the design/layout of the website	/4
2	`TuitterMain.aspx` • Correct controls (including correct location). • HTML table set properly. • Images and image buttons are properly displayed. • Log in functionality works (including error message). • Sign Up functionality works (including error message). • Current number of tuits is correctly displayed.	/8
3	`TuitterDisplay.aspx` • Correct controls (including correct location). • HTML table set properly. • Past tuits are correctly displayed (based on file entries). • Add tuit functionality works. • Welcome message is displayed. • Back button works.	/8
4	`TuitterSearch.aspx` • Correct controls (including correct location). • HTML table set properly. • Search results are correctly displayed. • Back button works.	/6
5	CSS style sheet • Style sheet is created and correctly configured. • All styles applied to all three web pages.	/4
6	Code comments are present and appropriate • Main comment • Code and method comments	/2
7	Naming standards correct and consistent throughout • Form and other controls • Variables • Constant	/2
8	Clean code • Correct layout (indentation) • No empty methods • Proper use of blank lines and code blocks	/2
9	Extra feature • The suggested extra feature is relevant. • The suggested extra feature provides value to the application.	/4
		TOTAL: 40 marks

20% of total course mark