School of Computing, Edinburgh Napier University

1. Module number	SET08101
2. Module title	Web Tech
3. Module leader	Simon Wells
4. Tutor with responsibility for this Assessment	Your first point of contact is Simon Wells
5. Assessment	Please see attached.
6. Weighting	**40%**
7. Size and/or time limits for assessment	Please see attached.
8. Deadline of submission Your attention is drawn to the penalties for late submission	**3:00PM on Friday 8th March 2019**
9. Arrangements for submission	Please see attached.
10. Assessment Regulations	This assessment is subject to the University Regulations**.**
11. The requirements for the assessment	Please see attached.
12. Special instructions	None
13. Return of work	We will aim to email marks to you within three **working** weeks.
14. Assessment criteria	Please see attached. With reference to the module descriptor, this assessment covers: **LO1:** Describe Internet and World Wide Web technology standards **LO2:** Identify and apply an appropriate web page development methodology **LO3:** Demonstrate competence in the use of authoring tools & markup languages. **LO4:** Demonstrate competence in Client-Side and Server-Side programming

Coursework Assignment
Web Tech (SET08101)

Overview

The aim of this coursework is to design and implement a website that enables your user to encode simple text messages using at least two different classical cyphers or encoding schemes. On your page (**index.html**) you must provide at least:

1. an area in which your user can type in a message,
2. an area where the encoded message can be displayed,
3. a way to select between different cyphers (for example a dropdown),
4. some method to cause the cypher to be computed (such as a button) and,
5. some mechanism for deciphering your messages to recover the plaintext.

A simple place to start is with the Caesar Cipher. The Rot13 cipher, which we investigated in the labs, is a special case of the Caesar Cipher which was used in Ancient Rome to encipher secret messages. You should research, select, and implement at least two different ciphers for your site.

There are many ways that you can extend these core requirements. For example, you may include additional cyphers and web pages, that document your site, your chosen cyphers, and the use of your pages. It is worth being creative in your extensions, for example, if you implement a Morse encoder then you might consider the use of the WebAudio API as a way to provide audio output of the Morse message.

Your page must be implemented using HTML, CSS, & Javascript and make no use of any additional libraries, templates or frameworks.

Deliverables

The coursework has two separate parts; a **submission** and a **demonstration**.

1. Submission

You must submit a Git repository containing the following:

1. The source code for your page
2. A written report

Your source code and report must all be committed to Git and pushed to your repository before the coursework deadline. Any late submissions that are not authorised by your Programme Leader will be capped at 40%. Any evidence of

plagiarism will be submitted to the School misconduct officer for possible disciplinary proceedings.

1.1 Source code

- All source code (HTML, CSS, JS) and associated files (such as images) required to view your site must be committed to your Git repository.
- **IMPORTANT:** Your Git repository must be named according to the following pattern (all lowercase):

  lastname_firstname_webtech_coursework1

- Your repository must be pushed to a hosting service, e.g. Bitbucket or Github, and you should make your repository private. If your repository is private then you must also add the user ***siwells*** as a collaborator so that your work can be retrieved.
- **IMPORTANT:** Email the Git SSH clone URL (the one that starts with either git@github or git@bitbucket) for your repository to s.wells@napier.ac.uk at least **one week before** the assignment deadline.

1.2 Report

Your report must be no longer than 6 pages in length (excluding appendices). You may consider typesetting your report using LaTeX, in which case the Napier LaTeX template might prove useful:

http://github.com/edinburgh-napier/aux_latex_cw_template

Appendices may be used to include supplemental data, for example test data, screenshots, designs, or documentation, but these must be referenced from within the main body of your report.

The format of the submitted report must be in PDF and should include the following sections:

1. Introduction. An introduction to the assignment stating its scope and content – this should include a brief overview of your site and your choice of codes or cyphers. Reference any background reading that you've done.
2. Software design. You are expected to plan how you will approach your implementation before actually writing any HTML, CSS, or JavaScript. You should describe this plan and the associated artefacts in this section. Artefacts might include lists of requirements, sketches of the layout for important pages, or a navigation diagram showing how pages are organised in relation to each other.
3. Implementation. Short description of your site's implementation including screenshots.

4. Critical evaluation of your implementation. Points to consider discussing in this section are:

    - A comparison against the requirements set out in this document
    - Possible improvements to your application, for example, what did you miss out?

5. Personal evaluation - reflecting on what you learned, the challenges you faced, the methods you used to overcome challenges, and you feel you performed.

6. References (Optional) - If you have used additional resources then these must be cited. Otherwise this section may be omitted. You must provide a reference for every resource used that you have not created yourself – for example, additional image, sound, video, or software library resources.

2. Demonstration

The main goal of the demos is to establish that the work you've submitted is your own. It is also a useful opportunity to provide verbal feedback.

Demos will be held during the period immediately after the deadline. Due to the size of the class additional time may have to be timetabled to ensure that all students can demonstrate their work. Prior to the deadline you will be able to arrange a demo time slot. During your demo you will have the opportunity to show off your app and may be asked questions about your work.

You should aim to be set up and ready to go **before** your demo slot time. It is your responsibility to ensure that you can demo the site that you have developed; this can be via a lab machine or your own laptop, however **without a demonstration your submission will not be marked**.

Assessment Criteria & Marking Scheme

The marking scheme is devised so as to reward those who go beyond the core taught material by integrating their own self-directed learning and discoveries. A reasonable attempt at a difficult application is likely to attract more marks than a complete implementation of a simple application. As a general rule, the more functionality, the better the mark, however your functionality should be consistent with a cohesive overall design, professional presentation, and pleasing user experience.

0-40% There are a number of ways to achieve a mark in this band, but generally you will either have failed to create a working website, omitted major functionality, have used a wholly inappropriate and unjustified approach, failed to include a report, or the report will be wholly inadequate in justifying the decisions that you've made in your code.

40-49% To achieve a mark in this band you must have developed your own set of HTML web-pages with associated CSS and Javascript as appropriate. Your user must be able to navigate between your pages. You will have implemented two ciphers. Your cipher pages will encipher a supplied message. Your design will be rudimentary but a basic usability requirement is that other users (aside from yourself) must be able to navigate your web-site. A submission in the grade band may be based on an extension of the practical work covered in class. Your report will adequately describe your work.

50-59% A submission graded into this mark band will indicate that you have developed a site that is less ambitious in its functionality. You will have implemented at least two ciphers and your site will enable users to both encipher and decipher messages. Your site will have good technical and visual design and provide an acceptable user experience. Your report will be well written and will reference the material you have used.

60-69% To achieve a mark in this band you will have developed a site with very good functionality, offering the user multiple ciphers to use. You may have chosen more complex ciphers to implement, or might have implemented ciphers that require additional keys. Your site will have a pleasing design, making very good use of appropriately selected HTML, CSS, Javascript features in order to provide a pleasing user experience that is underpinned by very good technical and visual design. Your report will address all the necessary sections effectively, be very well written, clearly presented, and will reference all materials you have used.

70-100% A submission in this mark band will demonstrate that you have gone beyond the core learning for the module and have actively pursued your own learning path. Your submission will include a site that goes beyond the core techniques discussed in class and lab sessions and that offers an excellent level of functionality combined with a rewarding user experience. You will have evaluated your design using appropriate techniques. You will have implemented more advanced features that have not been specifically covered in the practical sessions and which you will have investigated for yourself. Your design and code will be excellent. All HTML, CSS, and JavaScript will be well organised. Your report will be

comprehensive, very well written and presented, and will correctly reference all the material you have used. This is likely to include textbooks, online forums and tutorials and some of the suggested reading for the module.