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	**60%**
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 12th April 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 goal of this assignment is to extend your cypher site from the first coursework to design and implement a coded messaging platform. This platform

Your solution will include server and client elements. The client element will provide a web interface to enable your users to sign-up for an account, to write plain text messages, to encode those messages (using the cyphers that you previously implemented), to send those messages to other users, to access message send to them, and to decipher any received messages. The server element will persist data about messages and users and will support the client-side functionality. This should be achievable using a simple CRUD (create, retrieve, update, delete) approach.

There are many additional features that you could design and implement to give your site additional functionality. For example, pages to enable users to manage their own account, administrative pages for managing users, security features to restrict access only to authorised and authenticated users. It is critically important that you decide on the essential core features and plan your time accordingly. Your aim should be to produce a well-designed, well-engineered, and well-implemented, robust, and reliable platform that provides a pleasant user experience.

You must use HTML, CSS, & Javascript for the client interface. On the server side you may use Node.JS with the Express.js framework and SQLite. You may not use any additional libraries, templates, or frameworks (beyond those that are installed as standard with Node and Express).

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 app
2. A written report

Your source code and report must be committed to Git and pushed to your repository before the coursework deadline. Any late submissions that are not authorised by either 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 static files (such as images), and datastore information required to run your site's Node.JS server and access the client web interface must be committed to your Git repository. You should include a README.txt in your repository that explain the set-up process for installing and running your software.

- **IMPORTANT:** Your Git repository must be named according to the following pattern (all lowercase):

  lastname_firstname_set008101_coursework2

- 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 your must 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. An introduction to the assignment stating its scope and content – this should include a brief overview of your site and your choice of features and functionality. 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. A 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.

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 Node application, 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 a working Node.JS server component that serves up an appropriate HTML interface. Your user must be able to navigate between your pages, to sign-up for an account, to send encoded messages to other users, and to view decoded messages from others. 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 framework that is less ambitious in its functionality. Your user must be able to navigate between your pages, to sign-up for an account, to send encoded messages to other users, and to view decoded messages from others. Messages will be persisted using an appropriate strategy. Your site will have solid 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 the ability to navigate between your pages, to sign-up for an account, to send encoded messages to other users, and to view decoded messages from others. Your server component will persist messages using an appropriate strategy. Using an appropriate methodology, you will have protected users data and any features that could be detrimental to your users experience if misused. You will have implemented functionality to allow administration of your site. 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. 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 functionality that goes beyond the core techniques discussed in class and lab sessions and that offers an excellent level of functionality 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 in your repository. 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.