Coursework Report

Connor Cunningham

[40340741@live.napier.ac.uk](mailto:40340741@live.napier.ac.uk)

Edinburgh Napier University - SET08101 2018-9 TR2 001

Web Technologies Assessment 2

19/04/2019

# 1.Introduction

This report is about my assessment two on Web Technologies, I have been given the task to design a website that is hosted on webserver where user will be able to Registers/login into their account securely, they will also be able to plane test message to other user that are using the website.

This is the list of dependencies I have chosen:

* "bcryptjs": "^2.3.0",
* "body-parser": "^1.18.3",
* "connect-flash": "\*",
* "cookie-parser": "~1.3.5",
* "debug": "~2.2.0",
* "express": "^4.16.4",
* "express-messages": "\*",
* "express-session": "^1.13.0",
* "express-validator": "\*",
* "jade": "~1.11.0",
* "mongodb": "\*",
* "mongoose": "\*",
* "morgan": "^1.9.1",
* "multer": "\*",
* "my-dependency": "^1.0.1",
* "nodemailer": "\*",
* "passport": "\*",
* "passport-http": "\*",
* "passport-local": "\*",
* "serve-favicon": "^2.5.0"

I have done a lot of research to find out what Extensions to use to accomplish this Assessment.

## 2. Software design

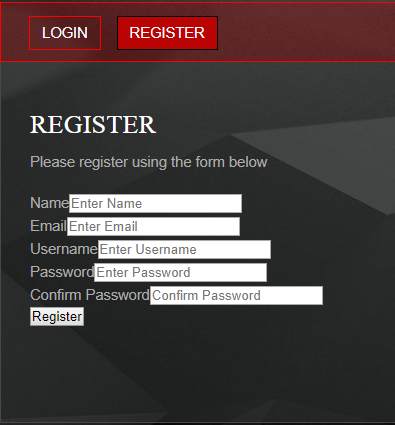
For my website I used the structure I had from Assesment one but implemented it into Node.js, I used Node.js because of the the extesions that could be used to help the funcionality of the website. The main requirment of the assesmnt is to “provide a web interface to enable your users to sign-up for 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.”

By looking at the Assesment spec I started off by doing research on the main requirements that I need to accomplish, so for enabling Login/Request I used MongoDB with passport.js, for encripting that data I used bcryptjsm, for the chat I used node-mailer.

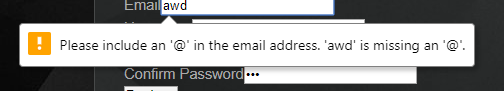
## 3. Implementation

I used the CSS from my first assessment as my Main colour scheme, When you arrive at the site, the first page you will see is the login and register page, to be able to access the ciphers, the user will have to create an account using the field give.

The account information will be encrypted using bcrypt.js and will be held on the MongoDB server.



The user will have to fill in all the field, if the user doesn’t fill in the fields, they will have an error depending what field is missing

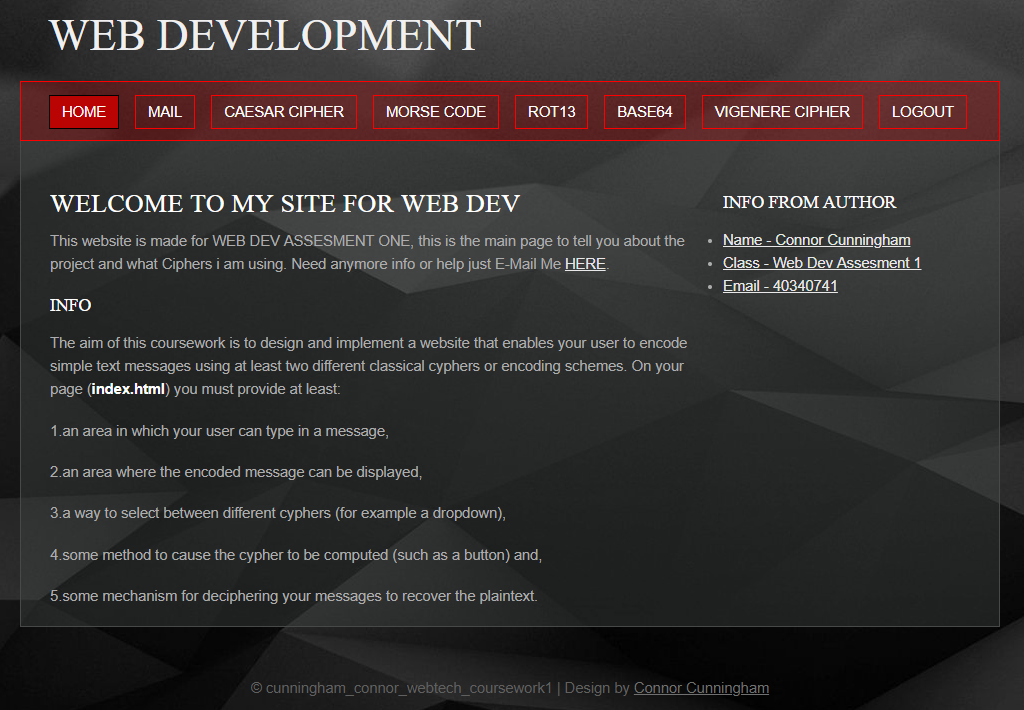




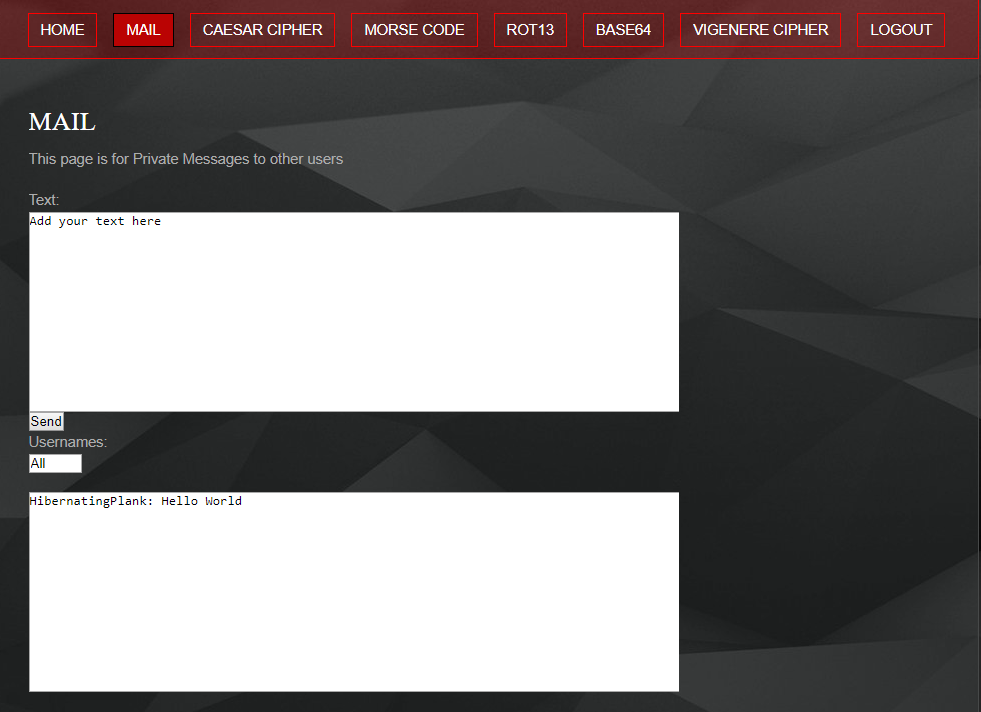
When Logging into your account, the two below will have to be filled in using Username and Password, When the fields are completed and the user presses login, they will be redirected to the main website page.

# 

This is the main page of the website, the user will be able navigate to the other pages to find out the cipher they want.



In the nav bar there is a mail Tab, this will allow users to Private Message each other so that they can share Cipher text or plane text



# 4. Critical evaluation

The requirement of this assessment was to create a website that could allow user to create a account so that users could send messages to other users while being secure and send ciphers from the site, I designed to website so that the user could create and login to their account easy and fast while still being secure, user could message other users of the site while sharing ciphers.

My website is up to date and looks Modern, the layout of the text with the structure of the nav bar makes the site stand out.

I completed the task to create the website register/log page that was hosted on a webserver, the requirement of users Private Messaging was my downside and needed more research in Mongo-Mailer as it doesn’t 100% work.

5. Personal evaluation

# By doing the course work I have learnt how to create a webserver using Node.JS and how to use its extensions to host a website, by doing the course work provided and while researching websites like W3School, GitHub and Stack Overflow for more extra learning, it has helped me build my knowledge exponentially on how to create and house a webserver. If I had more time on this course work I would like to do more research on certain topics to fully understand the how to implement extensions onto my webserver. This assessment has helped me learn a lot about how HTML, JavaScript and CSS are implemented into a webserver and how they are created and build from the beginning to the end.

6. References

<https://nodejs.org/en/>

<https://expressjs.com/>

<https://www.npmjs.com/package/bcryptjs>

<https://www.npmjs.com/package/cookie-parser>

<https://nodemailer.com/about/>

<https://github.com/expressjs/multer>

<https://expressjs.com/en/guide/database-integration.html>