

Coursework 2

Mohammed Almouhanna
40332133@live.napier.ac.uk

Edinburgh Napier University - Web Technologies (SET008101)

1 Introduction

The aim of this coursework is to develop the previous website that I made. The website is going to have a sign in, registration, send encrypt messages to users, and view them with the as a decrypted message. I used some features, which is a requirement for this coursework, for the website, such as Nodejs, express and MongoDB. I have done some reading from the reading list to get started with Nodejs [1] and express [2]. As well as, going over the documentation of them [3] [4]. I have watched videos to get the idea of how MongoDB works with Nodejs, as well as, how to make the database react as a service [5]. I have used extra modules with Nodejs, which helped me in implementing the requirements of this submission, such as body-parser and http-errors, etc.

2 Software Design

I have started designing the new implementation, by sketching in a paper, for the home page which included a log-in link in the side navigation bar and the title area has a background colour as the side navigation bar. It will also have a button for the user, when they log in, that has a drop-down list of links that will take the user to view messages, send messages, and log-out page. See figure

1

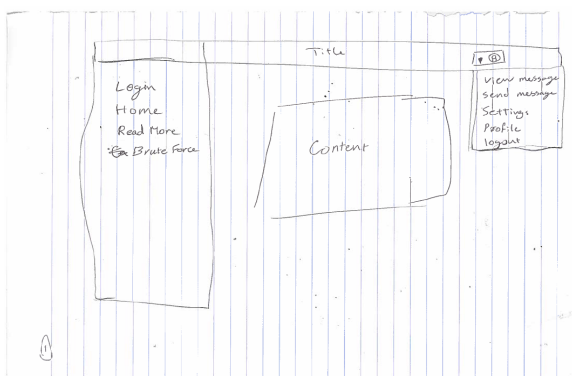


Figure 1: Home Sketch

I have sketch the send and view message page. The send page will have an area with a user-name that is entered by a user to send to the targeted user and an area for the text, which will be sent and load to the database. See figure

2

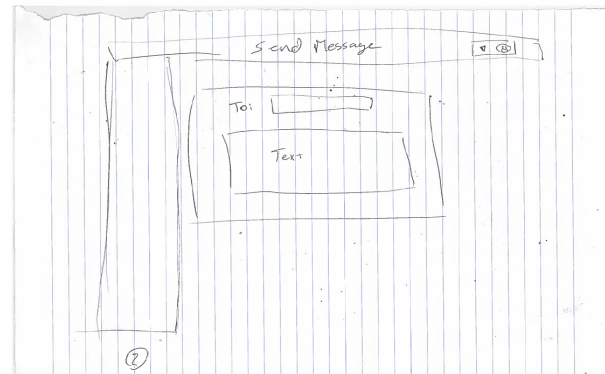


Figure 2: Send Sketch

For the view message page, it will have an area to view the users, who sent a message, and an area to view the encrypted message. As well as, a button to decrypt that message and an area to choose which cipher to decrypt with. See figure

3

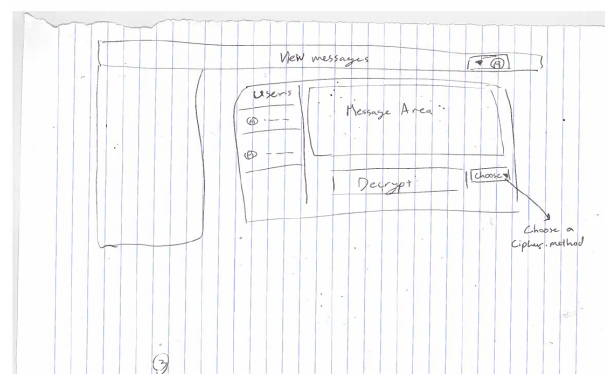


Figure 3: View Sketch

The user can navigate between all the pages from the navigation bar. However, the send message and view message page will not be shown to users that have not logged in. The login will be a pop-up that can linked from any page, as well as the sign-up.

3 Implementation

The implementation of developing the site have been reached by using the CSS and Nodejs modules. The first implementation is developing the home page, which will have a log in link in the side navigation bar and a button in the top next

to the title. However the button will not be available for unregistered user. The button will have a 5 links, which is view messages, send messages, settings, profile and log-out. See figure.

4

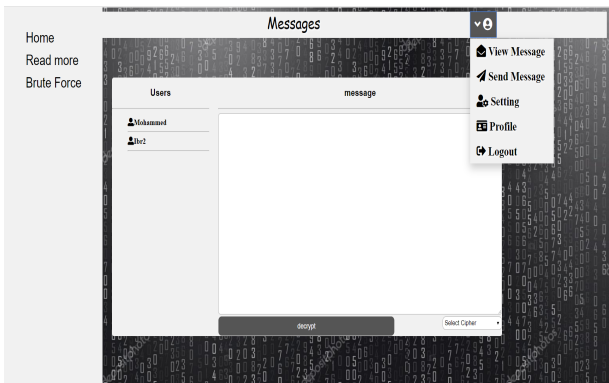


Figure 4: After Log-in

The second implementation is the log-in which I implemented it as a pop-up, which the user can access it by the log-in link in the side navigation bar. This will have a user-name input, password input and log-in button, which will make the authentication check in the database, and sign up link. See figure

5

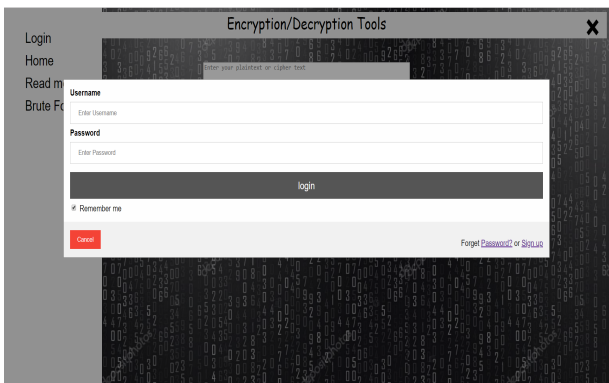


Figure 5: Log-in pop-up

The other implement is the sign up, which I design it as a pop-up as well. I got both of the design style from w3schools [6]. This will have user-name, email, password and confirm password fields, which will create a user-name in the database after clicking the sign-up button. There are also a cancel button which simply take you to the home page. See figure

6

The design for the view messages is very simple that will have a in the left side of the content a list of the user-names that sent a message to the logged-in user. In the right it has a text-area which will be used to view the messages after clicking the user-name in the left. In the button it will have a decrypt button and drop-list which the user will be able to choose, which cipher method to decrypt the message with.

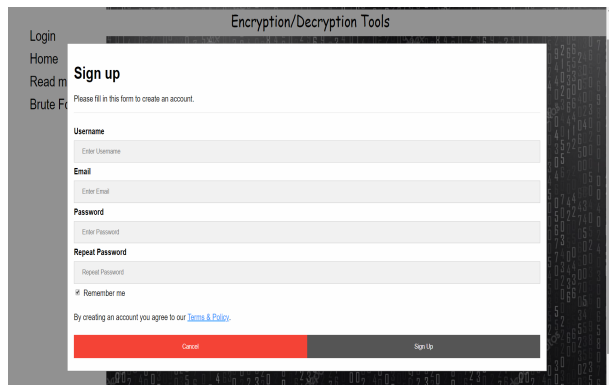


Figure 6: Sign-up pop-up

After clicking the button a pop-up will display with decrypted message. See figure

7

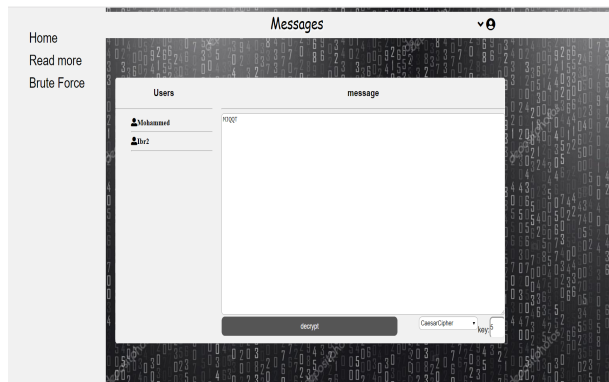


Figure 7: View message page

The design for send message its very simple, the design for most send email sites. It has two fields with the user-name of what user to send to, and for the text that wanted to send. See figure

8

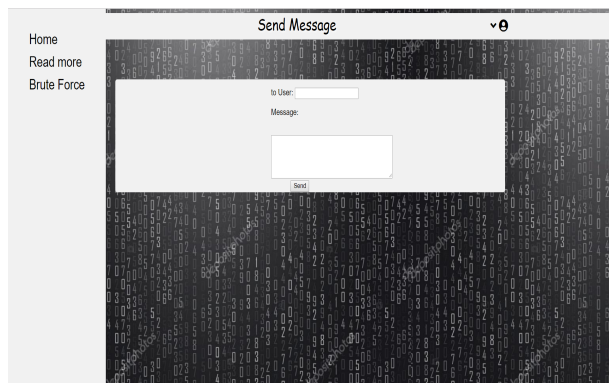


Figure 8: Send message page

4 Critical Evaluation

4.1 A comparison against the requirements set out in the assignment

Most of the requirements that I have used in this assignment has met the requirement, such as, Nodejs, express, HTML, JavaScript, and CSS. I have used an advanced no SQL database, called MongoDB.

4.2 Possible Improvements

Things that could be improved in the website is that, an automated decrypted message can be done, when viewing a message. This means that, no need to choose which cipher to use to decrypt the message that have been received from a user.

Another possible improvement that could be done, is editing the user profile, such as changing the name and adding a profile picture. This feature will make the website more efficient. Or the user could make changes to how the pages arranged in the side navigation bar.

5 Personal Evaluation

I have learned from scratch, how to use Nodejs with its modules such as, express, body-parser and passport. Therefore, it was difficult in the beginning because I did not get the style and how it works. After watching videos about how to build a server with Nodejs from scratch, this gave me a strong idea about it. I Learned SQLite3 and MongoDB in the same time, because in the beginning MongoDB did not work for me. However, after learning SQLite3 and how it works with Nodejs I changed to work with MongoDB. This shift made working with the database much easier.

The first challenge I faced is that I had booked my flight before the extension to have a holiday for a week. So, after the extension I postponed working in this coursework to work while I am in the holiday, which I find it a challenging. Because the travel is getting in the way so I do not have a time to put more effort in the coursework.

Another challenge I faced is checking the authentication for each user and how to give for only the users a control access to do special methods. I have used passport module for the authentication checking which I learned from the YouTube and it was helpful. To give a control access is a quite difficult.....

What I have done with the challenges I faced, I think I have done very well with tight time. I have met most of the requirements, and learned more than just the requirements, such as MongoDB.

References

- [1] <https://link.springer.com/book/10.1007/978-1-4842-0187-9>.
- [2] <https://link.springer.com/book/10.1007/978-1-4842-0037-7>.
- [3] <https://nodejs.org/api/documentation.html>.
- [4] <https://expressjs.com/en/api.html>.
- [5] <https://www.youtube.com/watch?v=k0ZzvHbNBQlist=PLillGF-RfqbyYRpji8t4SxUkMxfowG4Kqp>.
- [6] <https://www.w3schools.com/>.