Introduction

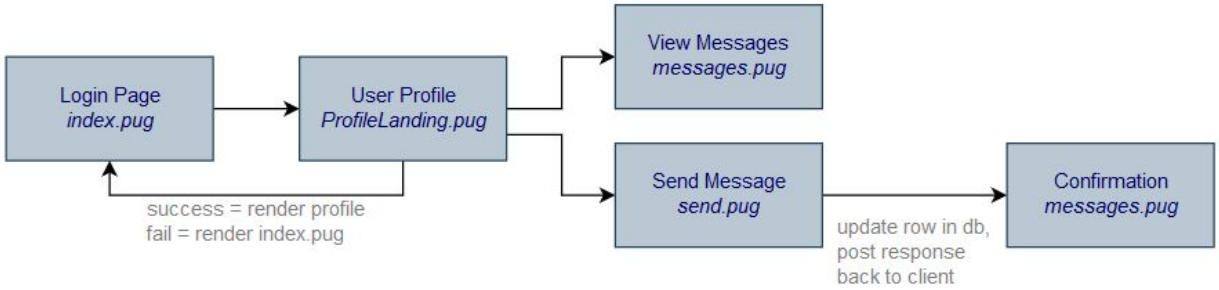
The assignment required a server based program similar to a e-mailing client. The specification required a user to create an account, sign in to their profile, send and view messages. The time scale restricted the scope of the website, however the main requirements of the specification were implemented. The background reading done was mostly too long to list, but was mostly wrestling with getting packages installed and files working properly as a lot of the concepts were new to me.

Software design

I had hoped to take the most primary features from my previous coursework, recycling the files and the styling. However, with this project I chose to use pug files instead of html files, so copying over the code was not feasible as I found myself spending too much time trying to integrate it and make it work. Furthermore, the entire layout of my previous site was not suitable for the intent of this one. Thus, I started from scratch, and unfortunately ran up very short of time to present a complete design as I have never used pug files before, so it was not as easy for me to implement something appealing in a short space of time.

Routes

I started with the basics, doing one page at a time. I wrote out and programmed all the links and the routes, however did not have time to implement the extras, so for now they're just a visual. I knew that as a minimum, my website required a Login page, a profile page, and a page where the user could view or send messages. As routes were still very new to me, I found myself going back and forth a lot before being satisfied with the layout of the website, especially as i often found myself needing more routes as there was more javascript processing to do than I had initially anticipated. I had attempted to create a navigation system however with some pages this did not work, as certain variables were created in one route, and were not passed between them. For example, I tried creating a back button, rendering the page before, however the variables would not be passed over thus throwing an error. Below is a basic layout of the current navigation of my website. The user is expected to just press the back button should they want to go back to the previous page, however some sort of prominent navigation would have been preferred.



Data processing

Despite designing a form in my previous website, I hadn't actually implemented it, so this was my first time creating a functioning login. Using forms posed another challenge as this required a POST route, which behaved differently from get routes, as the POST route expected feedback to be sent back from the server. I did this in the form of rendering the pug file in response to the POST either being a success or an error. For the following step, I needed the POST route to obtain the variables that were entered by the user, and compare these against the database. I had never used databasing with html before, apart from mysql and php, so this was a chance to learn from scratch how to do it myself without an already dedicated server handling it.

Initially I had looked at using local storage and using json files, however when I was reading up on potential approaches i could take for the implementation i was doing, there were a lot of results of implementations with sqlite which seemed a lot more feasible. It took several tries before I created the functioning database that i ended up using for my website. Finding that you cannot drop rows from the table stunted the development as it meant I could not freely delete data from the table (such as a password if a user was changing it, or removing an account if a user were to delete their account), however it was possible to update tables. I was also unable to get the output to print the messages on different lines. I had attempted to use variations of newline character and return carriages with the variables being printed but javascript seemed to ignore these altogether.

Finding a way for the function to take in the username and compare it against the database was perhaps one of the biggest challenges. There were many ideas of how to go about this however it was a struggle finding anything specific to using sqlite in express. There were a lot of bugs, such as the select statement not being able to differentiate between values in the database and the variables the program was attempting to compare, which was overcome with hours of background reading and experietnation, and moving around code until it gave a different type of error that I could work from.

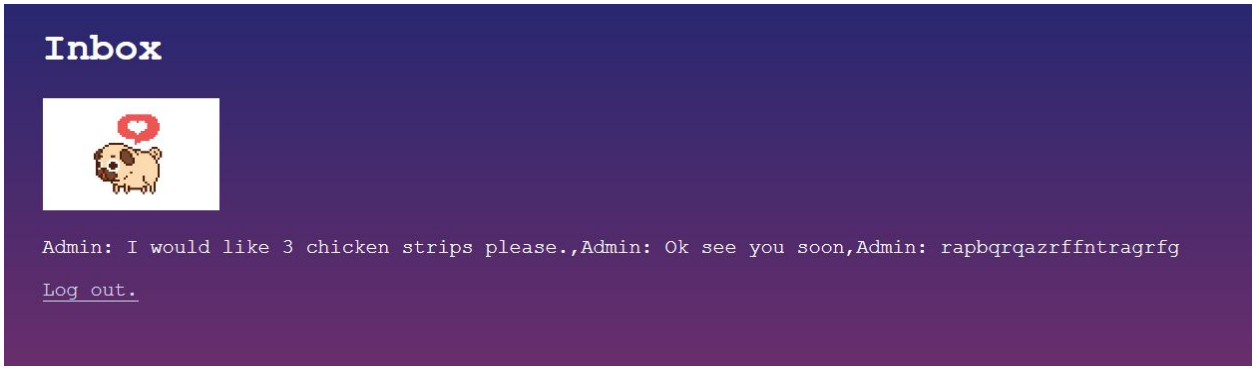
Cookies

I have had no previous experience implementing cookies, which was another challenge as they worked differently server side than they did client side. Digging around on stack overflow, they turned out far easier to implement than I had thought. I needed to use cookies for the website as a means of keeping the user “logged in” - a case of merely making key data available to grab and process from the page rather than having the user enter it any time they wanted to receive a response from the server (i.e retrieving messages). Initially I had made the cookie so that it, in it’s own way, encrypts the data it is storing. However, this was causing a challenge I did not have time to solve when it came to using the variable stored. My cookies stored the username of the logged in user, and for routes such as viewing or sending a message, I could not find a way to retrieve an unencrypted version of the username, so currently the cookies are in plaintext.

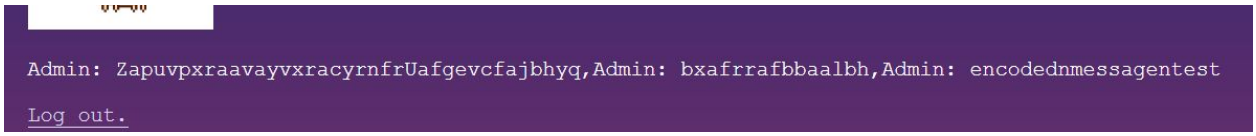
Encryption

In the early stages of development I had the website sending the messages to the database in plaintext. These messages would be retrieved based on the sender and receiver, and displayed upon the user’s request. Later on, I implemented the Rot13 encoding technique I had developed in my initial coursework. This is why currently, the first couple of messages appear encrypted, but the others do not - It is presuming all the messages in the database are encrypted. There appeared to be an encoding issue that I encountered - the algorithm didn’t work out of the box. The shift seemed to be taking in the space bar and ciphering it to the letter “a”, so come decoding and it would decipher it to the letter “n” instead of just a space. I was unable to resolve this issue as the shift was the same as in my previous coursework. I have speculated that perhaps the server side encoding is different. Currently it stores the message as encrypted in the database, so the program is sending the text incorrectly formatted (inserting “a” where spaces should be”.

Displaying plaintext messages:

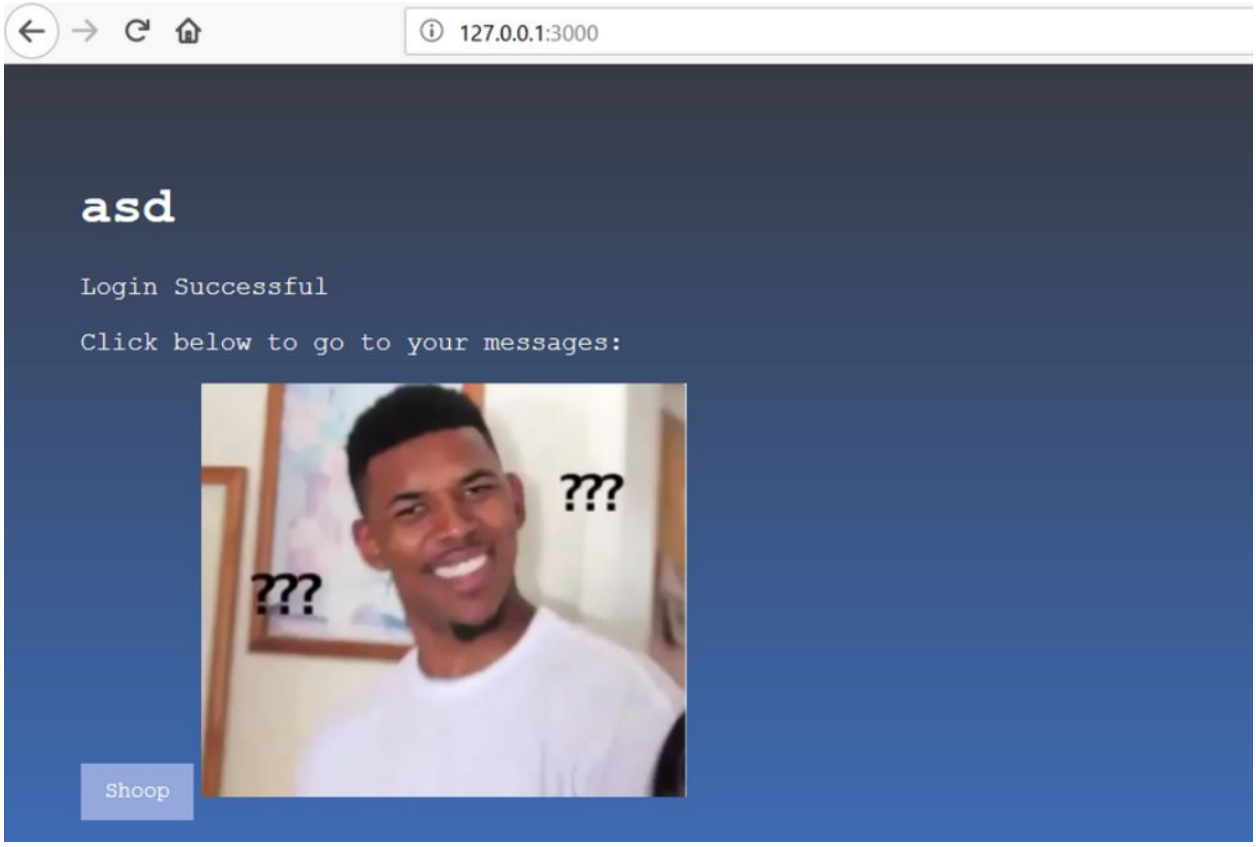


Decoding all the messages:



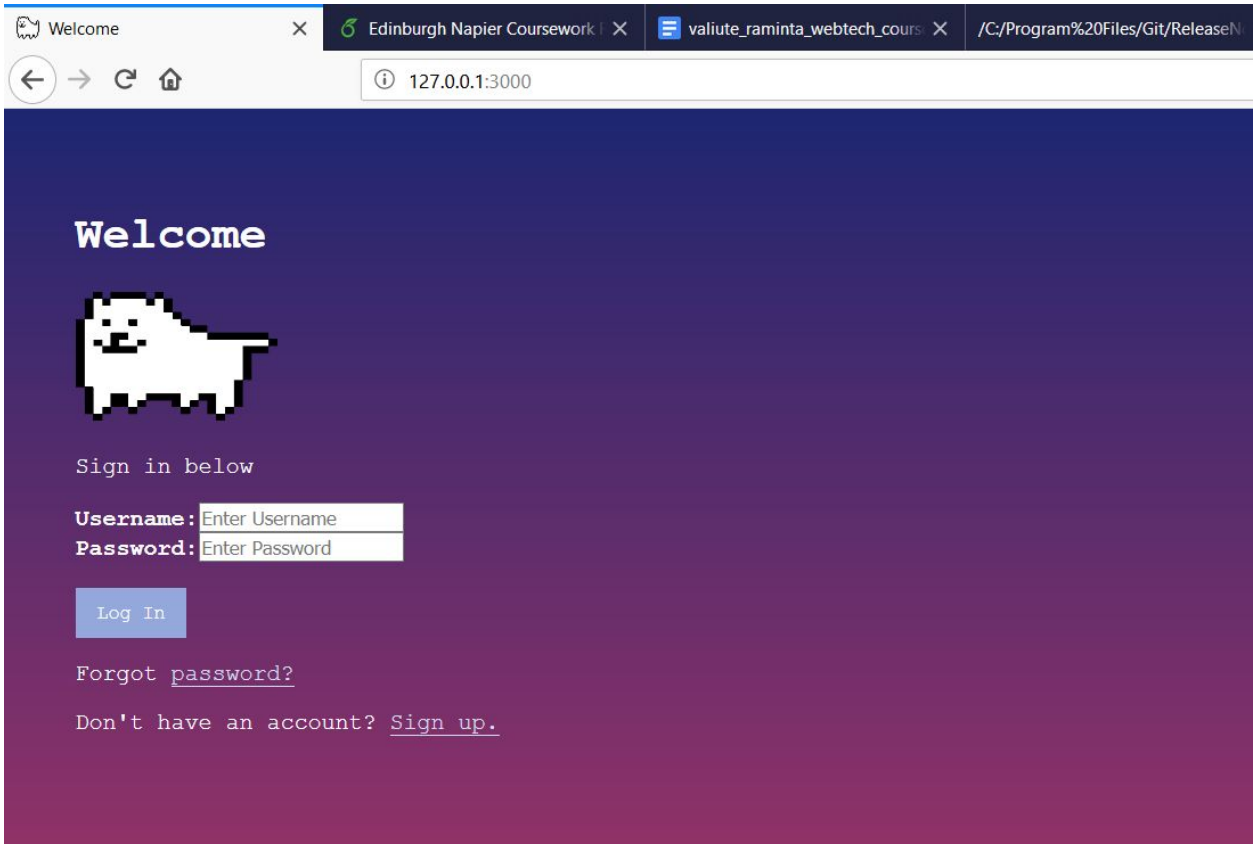
Implementation

Early development stage:

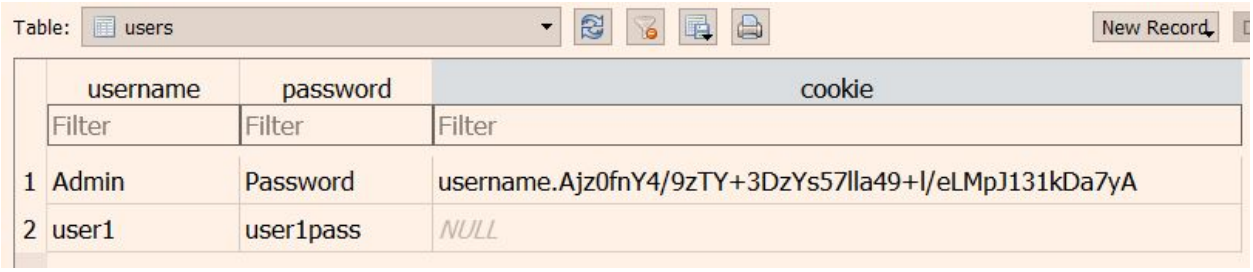


I was throwing together code in early development to see what worked. At this point, the name of the user (in this case asd), was being retrieved from the cookie rather than the database. The hopeful intent was to create a profile page for the user, displaying an image that they picked (rendered using a route depending on choice), however this ended up being left unimplemented. As the POST required to receive a response for the server, I thought to create an in-between page to show the user, however realised it would be far easier just rendering the pug files differently. The photo used in the screenshot reflected my opinion of using pug at the time.

Final design (features different dogs for each page!):



Initial encrypted cookies generated upon the user logging in:



	username	password	cookie
	Filter	Filter	Filter
1	Admin	Password	username.Ajz0fnY4/9zTY+3DzYs57lla49+l/eLMpJ131kDa7yA
2	user1	user1pass	*NULL*

Critical Evaluation

Currently, the website meets the points of the specification, however not perfectly. There are currently no security implementations, however an encryption of the database has been attempted. General cleanup is desperately required, however I ran short of time. It was difficult to balance design and functionality, as I like to make a design then base the functionality around it.

The site has a few bugs, such as outputting the messages in a very poor format, and currently you cannot send a message with apostrophes in it otherwise it will crash. However, this should redirect the user to an error page.
It would be good to find a way to encrypt the data stored in the cookies, as well as add a functionality to display all the messages in a list labeled by sender and receiver, instead of just viewing received messages.

References

https://i.imgur.com/pKra7We.gif white dog gif
https://i.gifer.com/RMmw.gif pug gif
https://thumbs.gfycat.com/PlaintiveSeparateBarebirdbat-small.gif shiba gif

https://codeburst.io/june-27-2017-62b9b5416a7f create user authentication
https://codereview.stackexchange.com/questions/160252/express-routing-with-a-login-action-using-sqlite - form action, learning to handle routes

There were countless pages used so only the ones with similar code have been listed.