# COMP2011 – Web Application Development

# Report

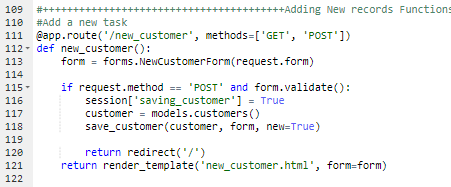
The website implemented is an online shopping or e-commerce website. The website has two main areas; the admin area and the end user or customer side.  
To use the website got to <http://themagicalorange.pythonanywhere.com/> . The system will initially ask for the user to log in, there are pre-existing accounts that can be used to view the system; there is a basic customer account with user number 123 and password ‘ABC’, this will allow access to all customer features. To get access to the admin account log in with the user number 9999 and password ‘admin’. There is also the option for a new user can also create a new account if desired.

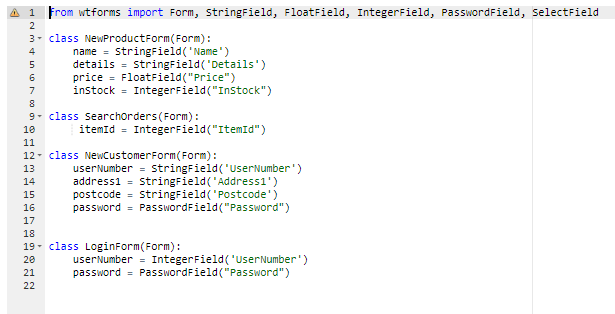
The two main roles have the functionality of:

|  |  |
| --- | --- |
| Admin | Customer |
| Add new products to the database | View all products in stock |
| View all orders made by customers | View all products out of stock |
| Search orders by product id number | Create an account if necessary, can also change password. |
| View all the products in stock and out of stock | Create an order of products |
|  | View all of their orders |

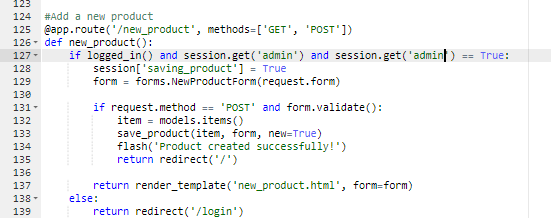
# Web forms

The web forms are created by using the following code. It uses a combination of the HTML page ‘new\_customer’ and python to create the form and to then use the forms to create entries. In this example the page will get the posted data, validate it, then save the customer using the save\_customer(…) function.

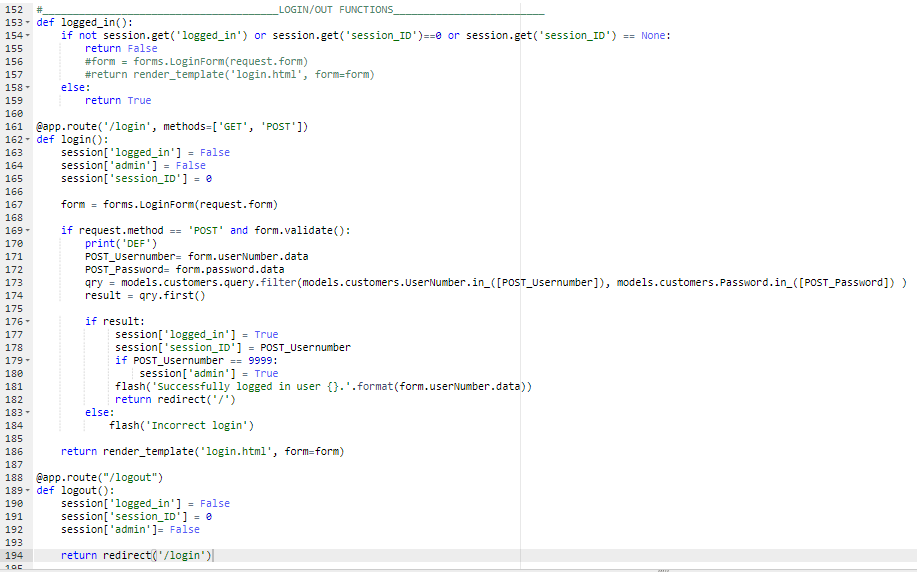
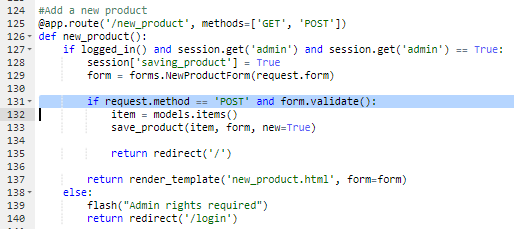




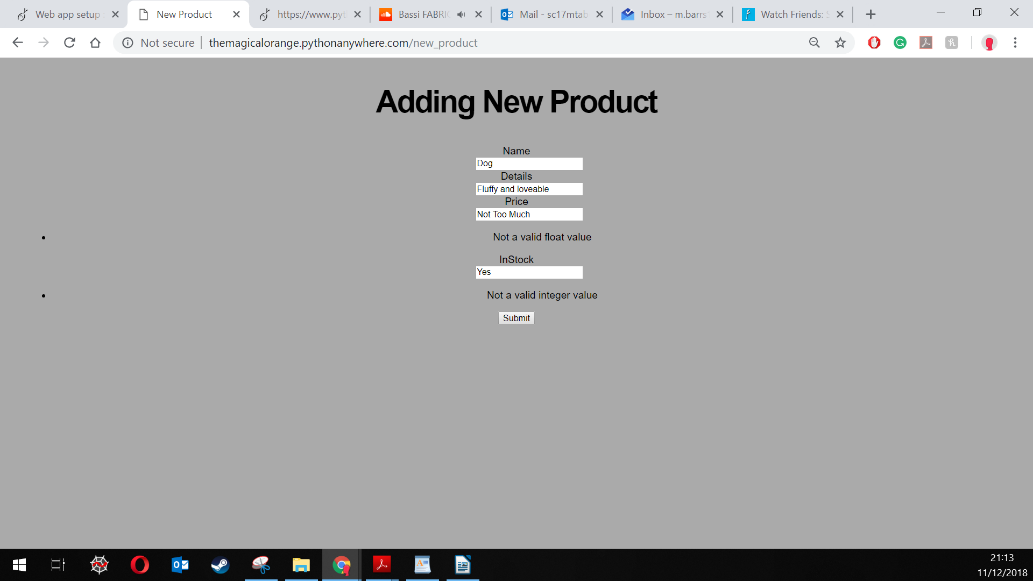
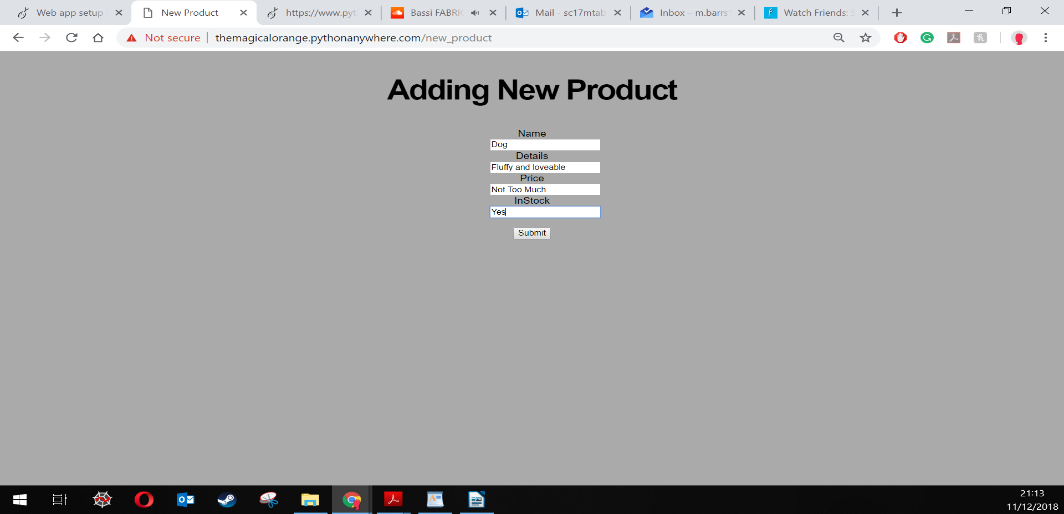
Validation

The system utilises session variables to allow the system to validate if a user is logged in or not, almost all the pages require the user to be logged in and in some instances the pages are admin protected, meaning that the administrator account must be logged, for the system the admin account has user number 9999. the use of sessions and/or cookies.

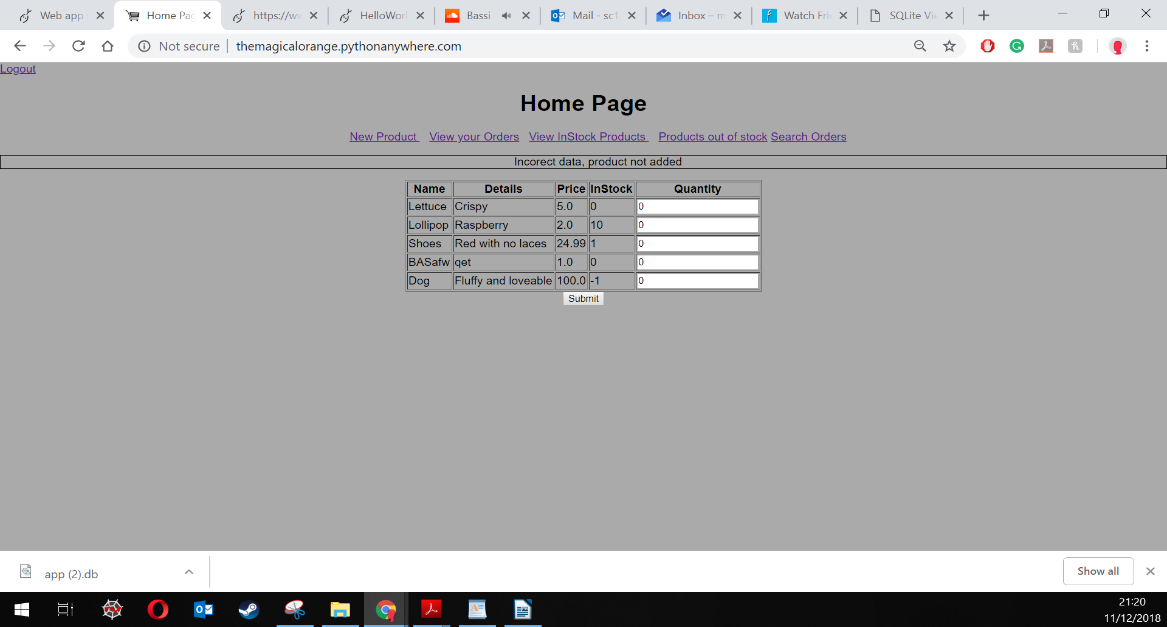
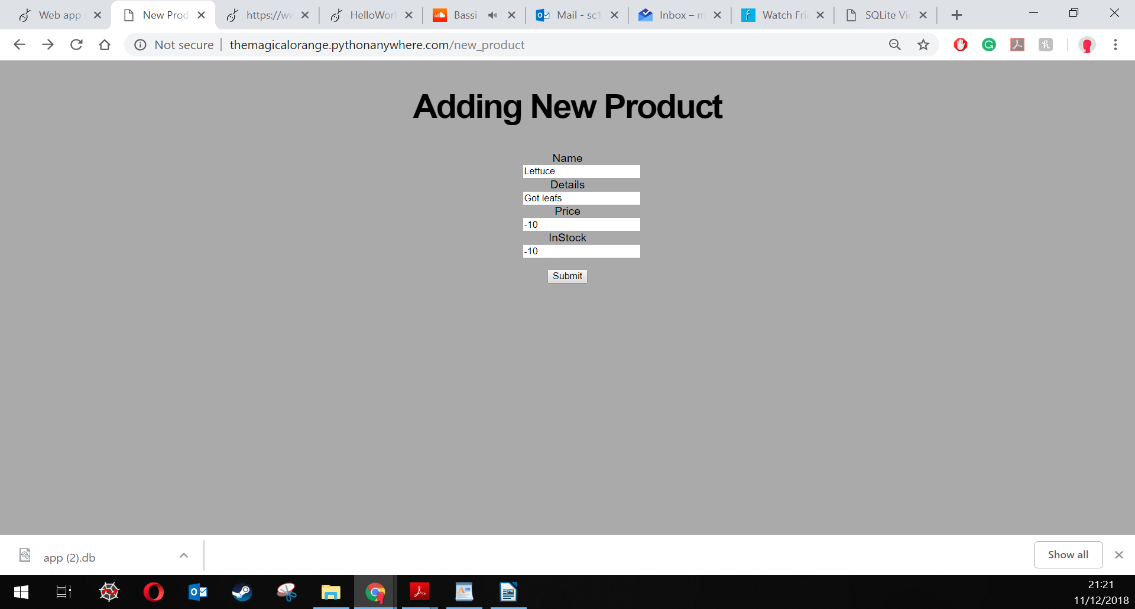
The logged\_in() function will check to see if the log in session variables have been set, or if they are True or False and return a boolean depending on their output.  
The login page first sets all the variables to False or 0, then if data has been posted it will attempt to find the user in the database. If the user is found then the user is logged in, otherwise the system redirects to the login page with an error message.

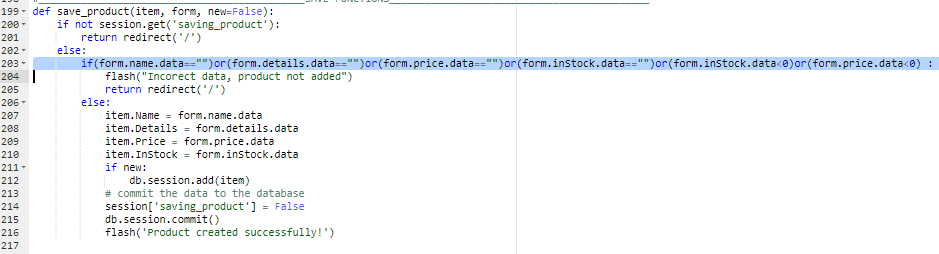
The system has two types of authentication, server side and client side. The screenshots and code below show client-side authentication. It is tested through entering strings into an integer and float input box.

The code has the if statement which checks to see if data has been posted to it as well as ‘form.validate()’ which means it will run the if statement code if the validation of the form comes back True/Ok.



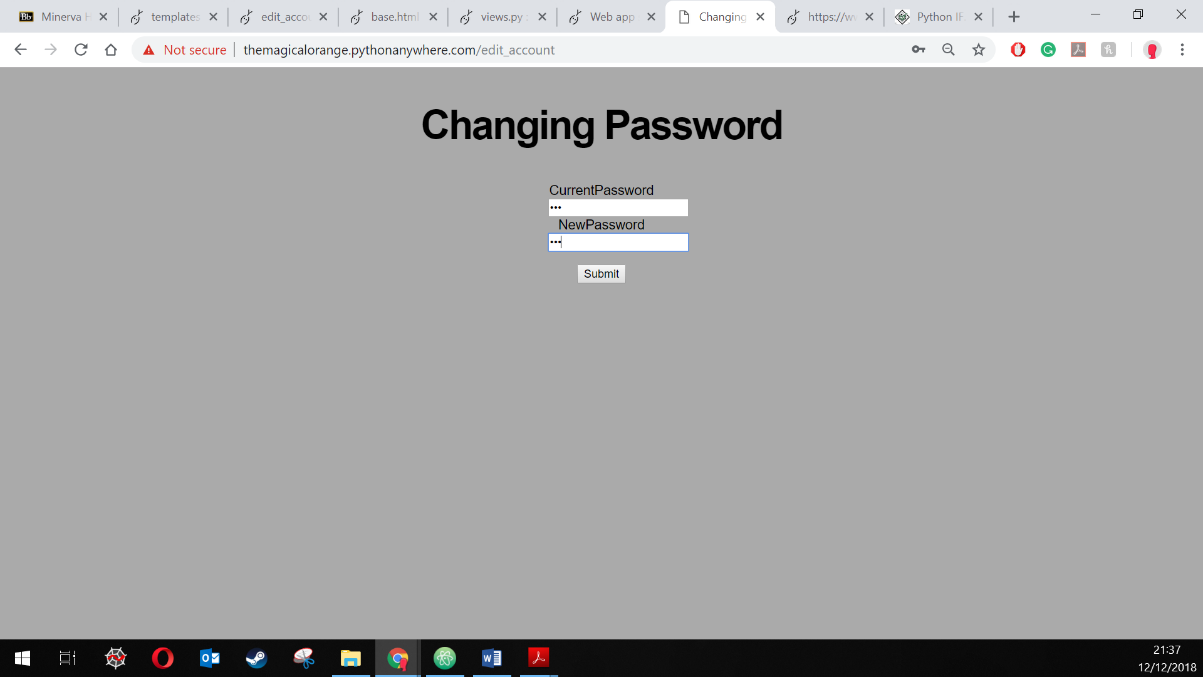
The system also has server-side validation, it checks to see if the values are valid. For example, when creating a new product, the system will check to see if all the variables are not empty and it will also check ‘inStock’ and ‘price’ are not negative. If any constraint is failed the product won’t be added and the system will output a message saying so.

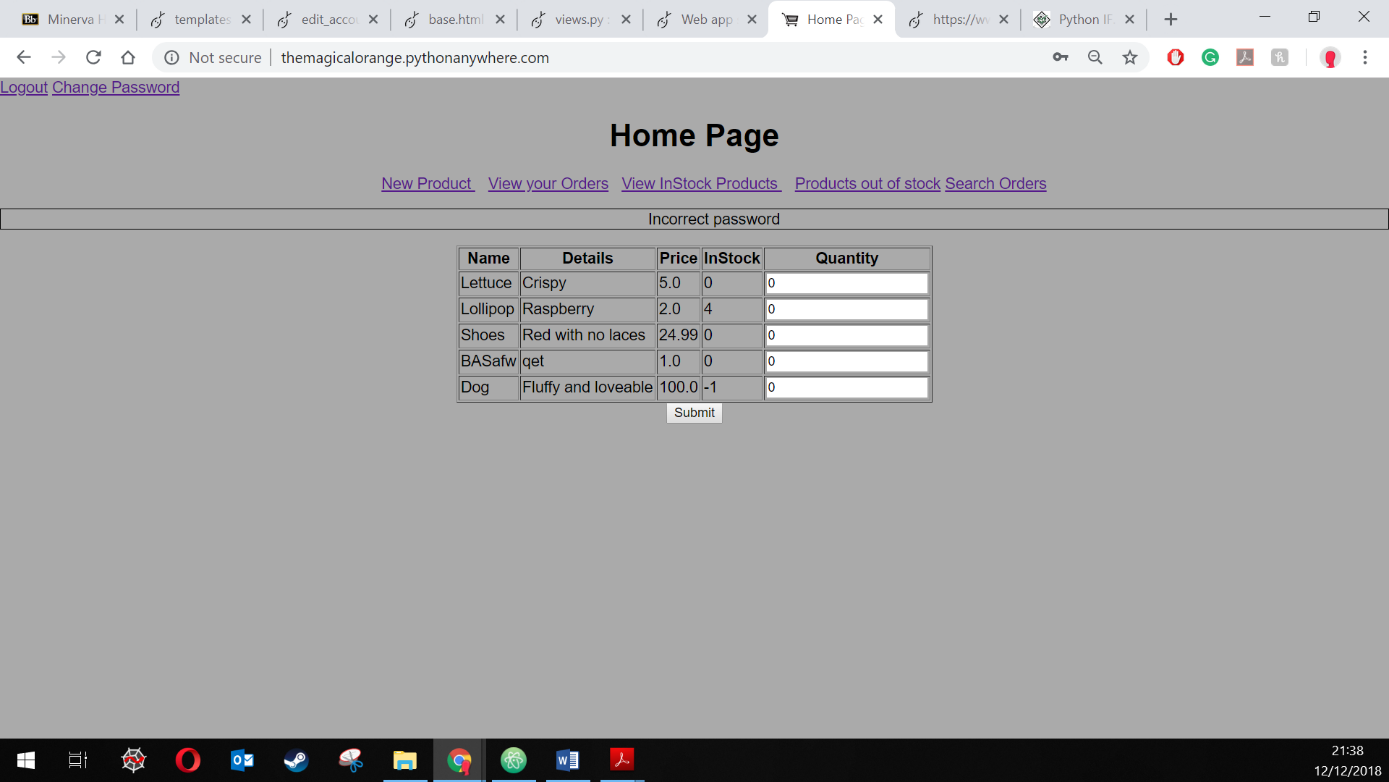


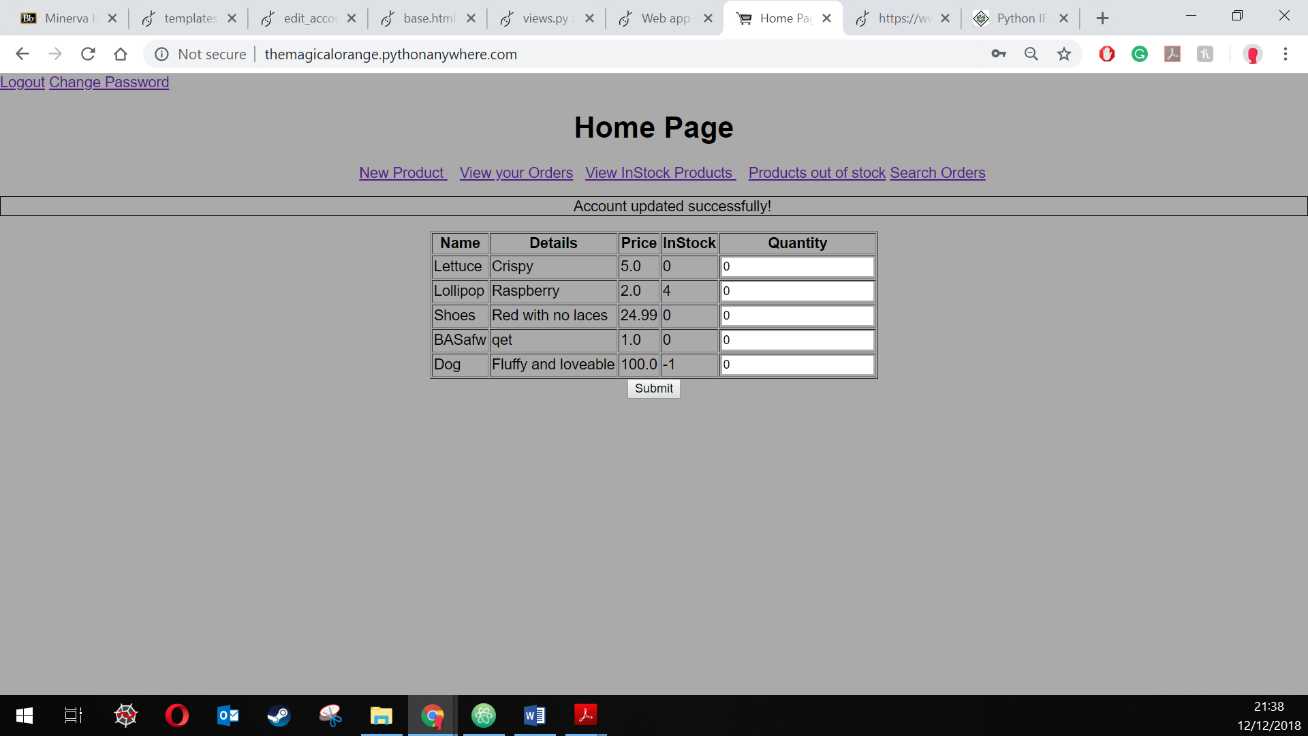


# Authentication

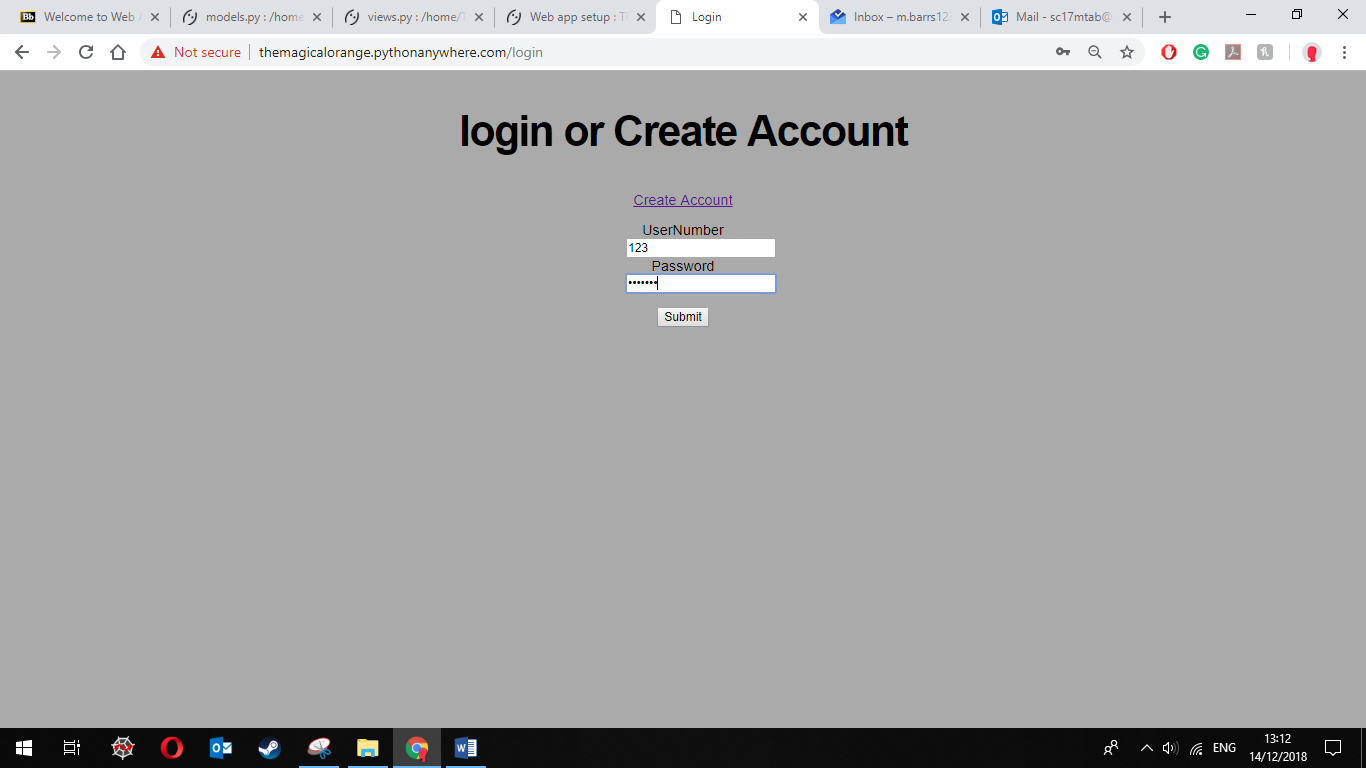
Below shows screenshots of a user changing their password. There are two outcomes shown, one where the user gets their current password correct and the password is changed and one where the user gets their current password wrong and the system does not change the password.



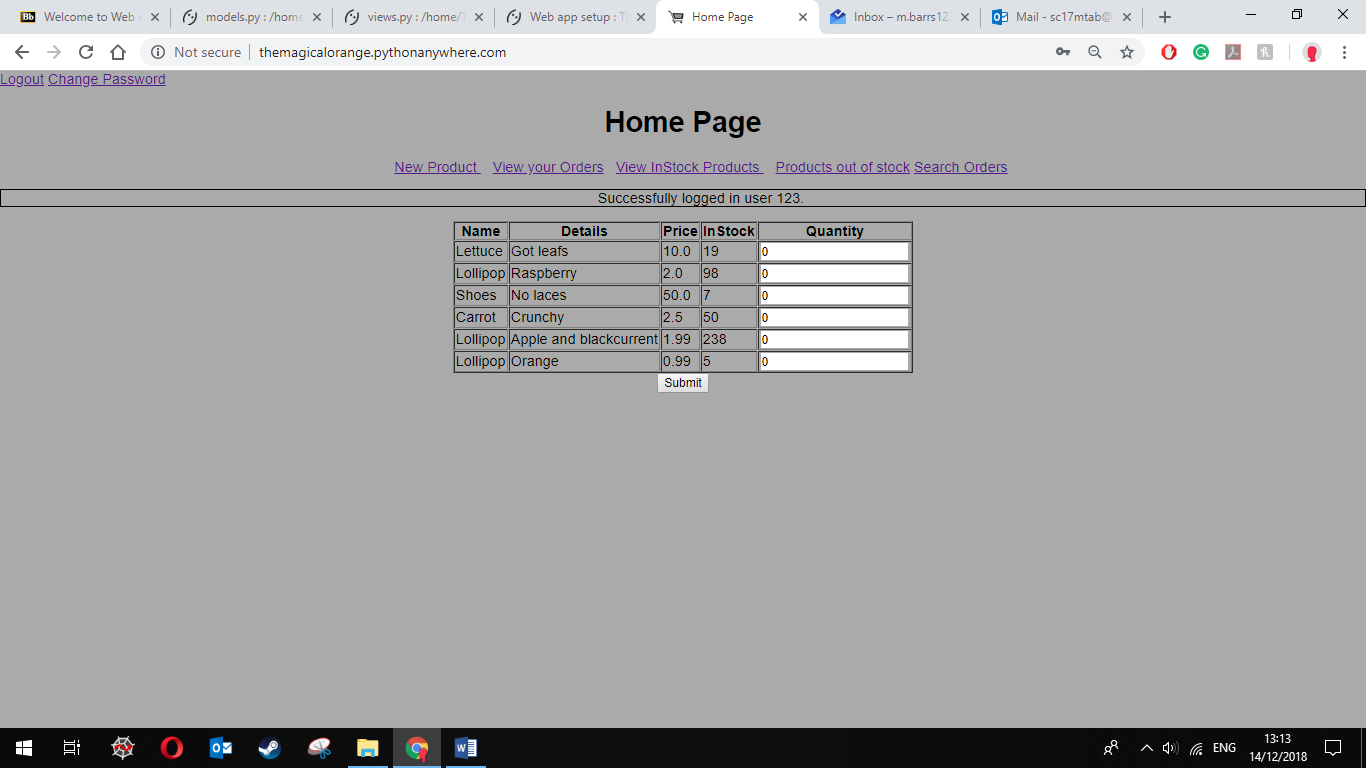
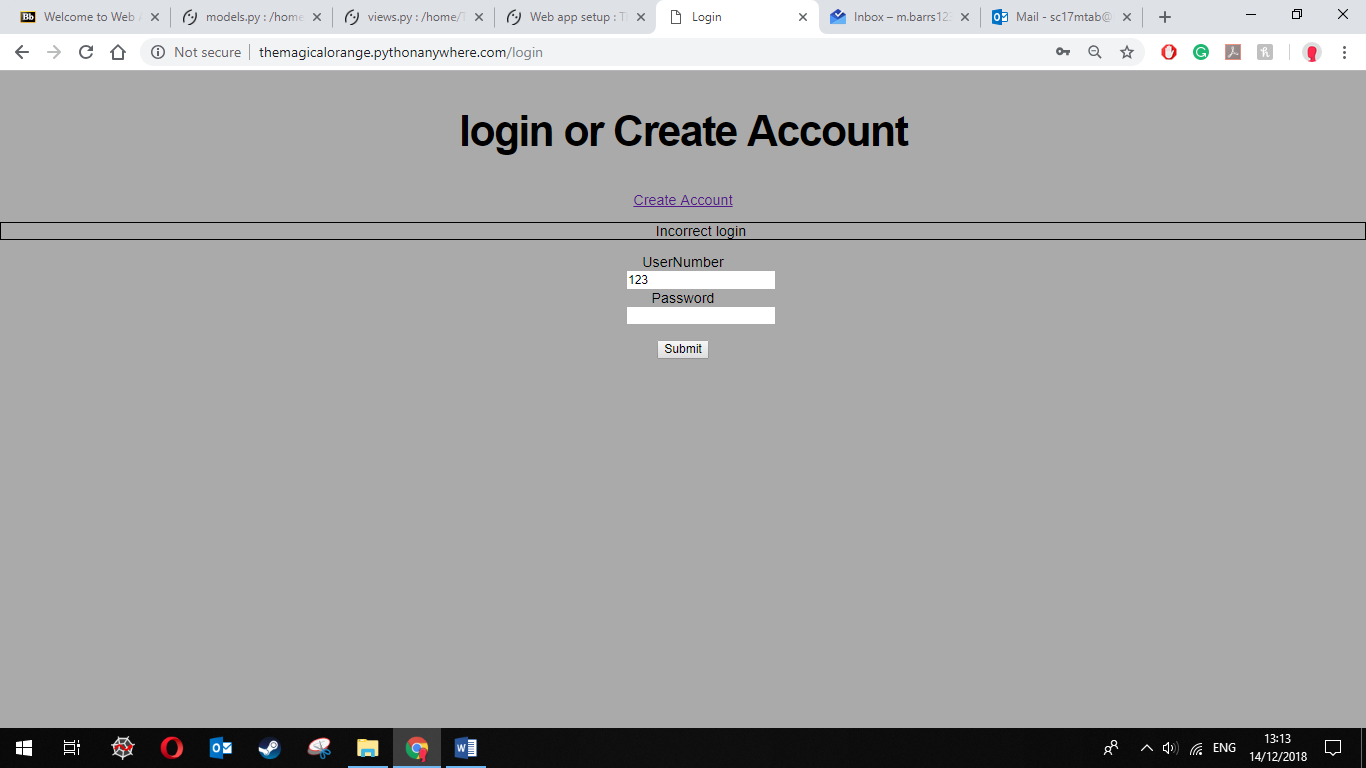




Every page of the system is password protected, if an attempt is made to access a page whilst not logged in then the attempt is logged in the .log file and the user is redirected to the page below.

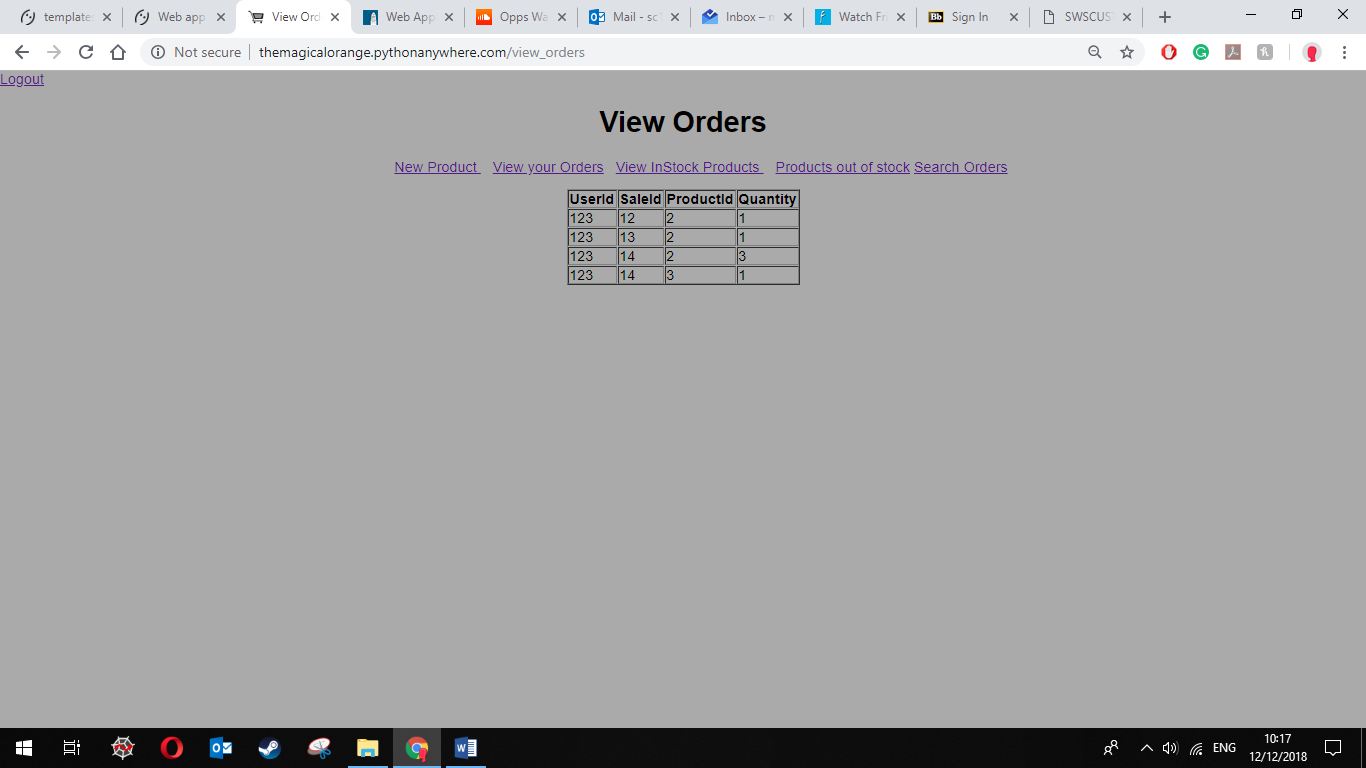


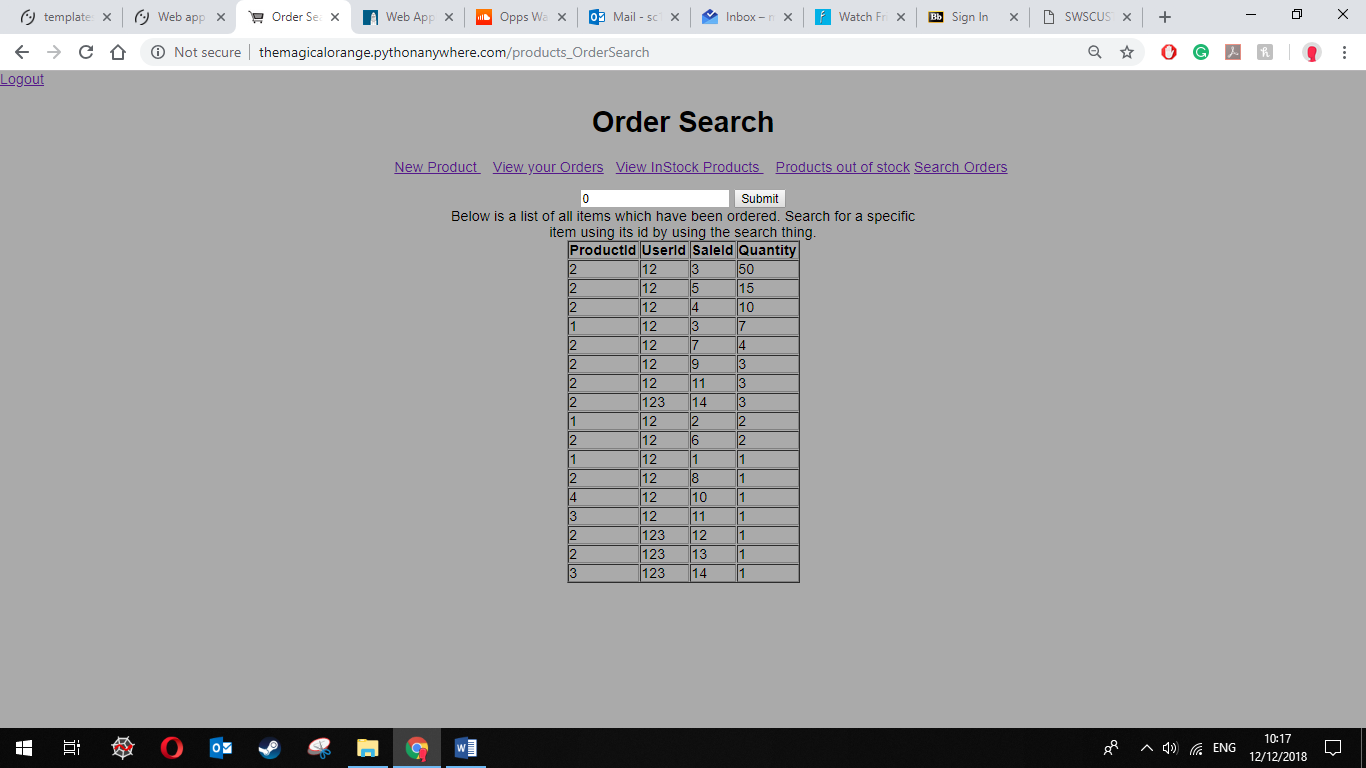
The user is given an output screen depending on what details they enter. If the details are correct then the user is allowed access to the system otherwise the user is redirected back to the log in screen.

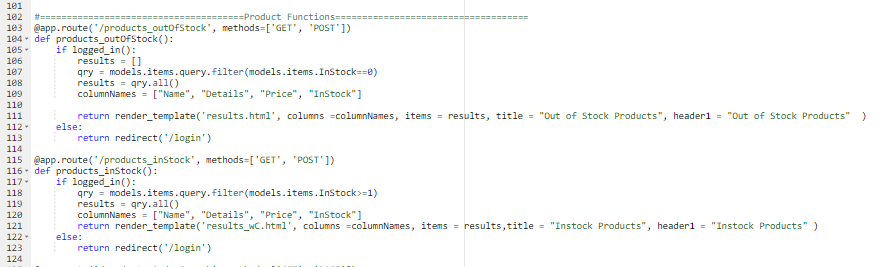


# Database

The database holds a many to many relationship as, a user’s order can hold many items in it, and an item can be in many orders. Below shows that an item can be in multiple orders and also that a single user can have multiple orders. This is shown by the view orders page, which shows that the user with Id 123 has multiple orders. Then the order search shows how products have been bough multiple times.



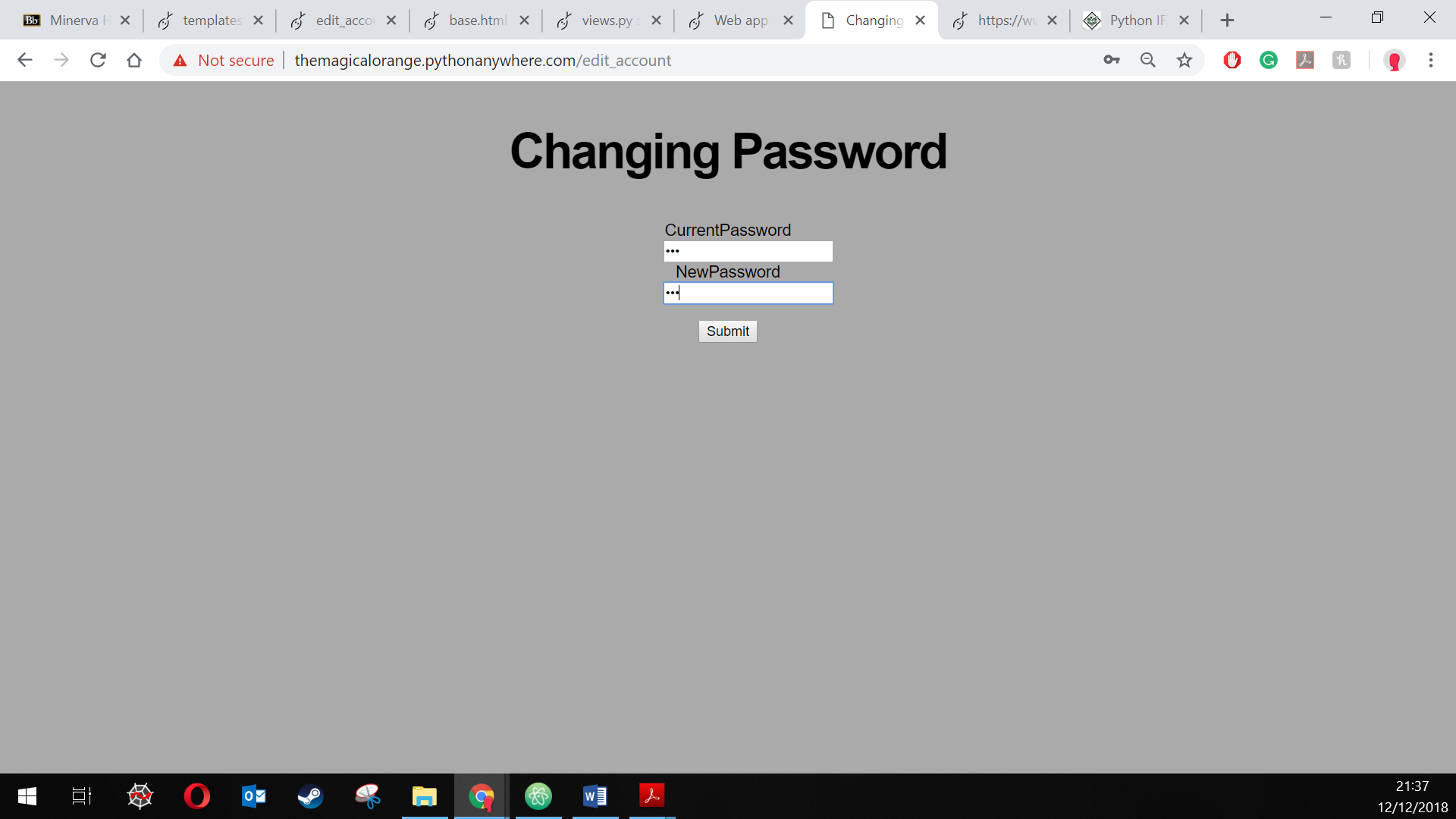


Below shows an example of two pages, one which queries the database and shows products which are in stock and available to buy and then products which are out of stock. They both use a similar query and function.

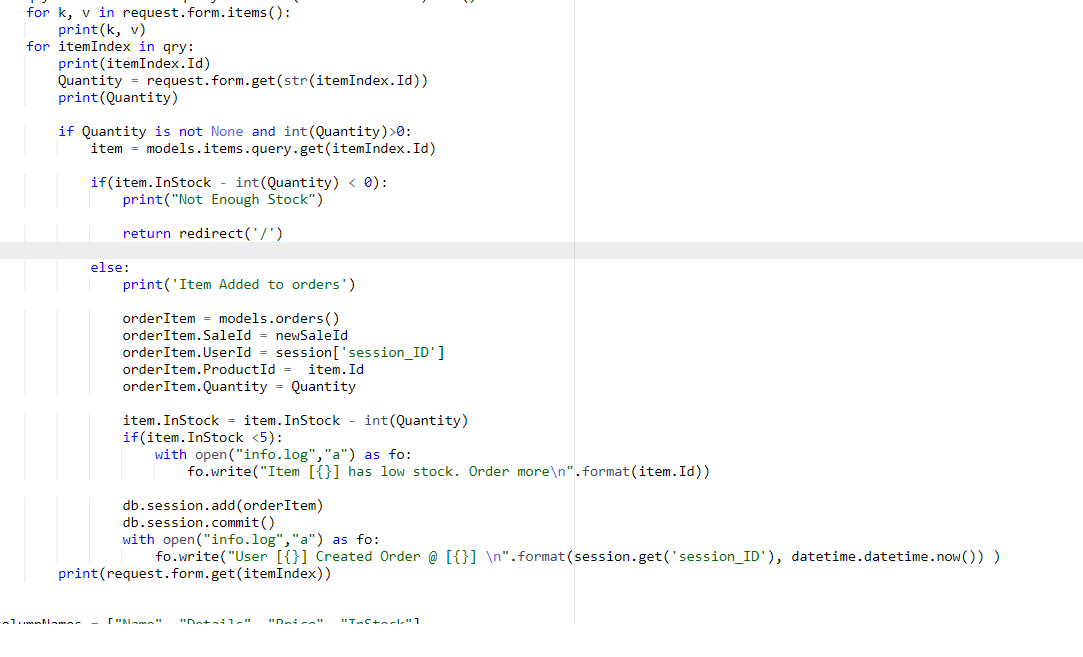
# Styling

The website is set to be centred meaning no matter what device views the pages the content will always be central, this makes the website a lot easier to use. All forms are labelled with information telling the user what is required.

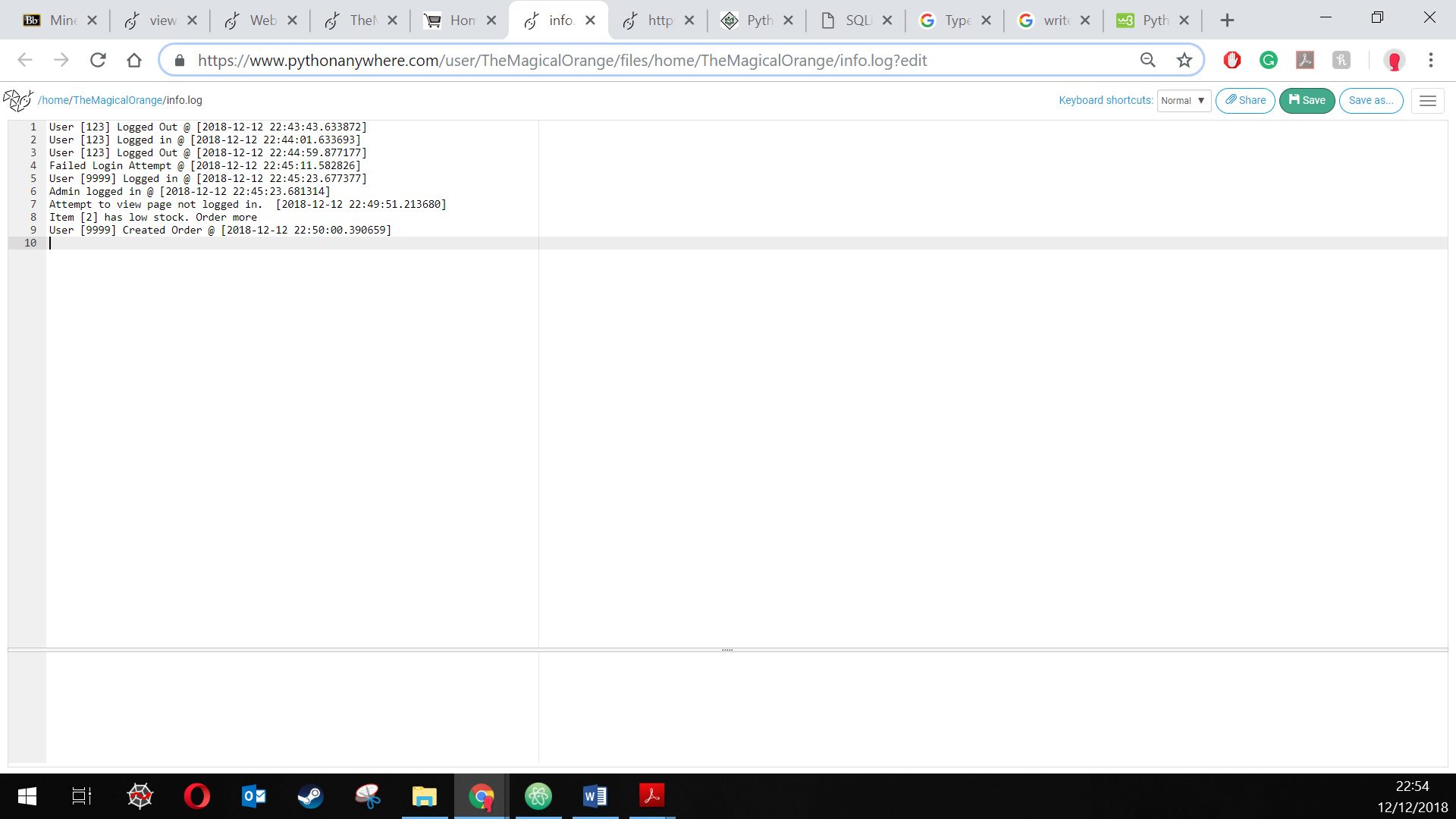
The web site is responsive and feels natural to use on both tablet and desktop computers

All content relies on the same CSS file, this makes the site consistent and appear more professional to the user. The website logout and change password links are in the top left corner, this is common in many websites to have the account functions in the top corners, this makes the users experience better as this makes learning the system a lot faster.

# Logging

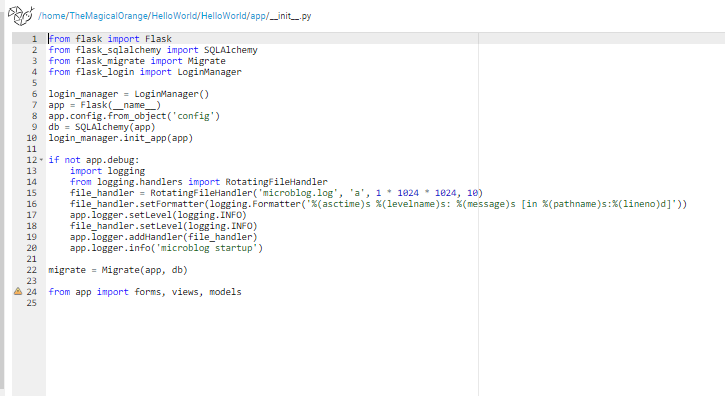
The system harnesses python file writing to help maintainers of the site by writing important events to a file called info.log, the file can be used to monitor activity on the site. The example below is of when a user creates an order, for each item that they have chosen it will check to see if the number of items in stock is less than 5. If so, it will write to the log with the Id of the item saying it’s in stock quantity is low. Once all items have been added successfully the system will then output the user id to the log file and a message saying they created an order.

Below shows the log file generated by the system, if has information like when and which user logins in and out. When a page is accessed without a login and similar events such as when a user changes their password or attempts to change their password but fails, this could be useful to help monitor security of the system, multiple failed attempts and the account could be locked.



The system has also been setup to create logs about other things such as errors.

# Deployment



The web application was deployed on the site python anywhere. The website is accessible through <http://themagicalorange.pythonanywhere.com/> There is the option of using 123 and password ‘ABC’, customer account, or 9999 and password ‘admin’, admin account, or create a new account if they so required. All of the specified functions are completed and available to use.   
The system can be used to:

* Login and admin protection
* Add new product
* View orders
* Search orders
* View products, inStock and out of stock
* Create customer account
* Create order
* Change customer password

# Evaluation of Implementation

The system which has been deployed has all functionality it was set out to have. The system has security measures in place to reduce human error and to ensure the integrity of the system. The system has a consistent styling; however, the colour scheme of the website may not fully suit the site as it is a very bland colour scheme, many e-commerce sites employ a light or bright colour scheme as this is seen as being more inviting to customers and make them more likely to spend more time on the site and then eventually buy more goods.   
The security flaw of the system may have been that when the user wanted to change their password, the system initially did not require the users current password to change the password which meant anyone who had access to the logged in account could have changed it, however this was amended and now the system requires the users current password to change their password, also any failed attempted at changing the password is recorded and kept on a server file, this could be used to spot security threats such as a mass attempt to change all the passwords.  
Another security risk was showing the users the product Id of the items in the view orders section, this was amended by creating an inner join on the items and using that to show useful information about the order.