**Michael Cook report on Login, Registration and topic selection on the index page.**

**Michael Cook work designed and implemented the index.html page for login, Registration and topic selection with Pieter Goos adding the dynamic Flask parts for use with the backend.**

The index.html page which deals with the Login, Registration and topic selection is extended by the layout.html page which allows for the common navbar and all the header information to be included in the index.html page. The index.html page also contains the flask section to where messages are received for display purposes from the view.py page when any of the login or registration information is invalid or otherwise incorrect, this allows flask to visualise the error messages on the page.

All the login details are contained in the flask section of navbarright to allow it to be on the right this request the user to input their username in a text field and the password in a password field. The login button is then selected to login with the information posted to the view.py with the name “log”. The post is handled in the view.py file where the post data is checked to see if the form request is for login. Then the password is requested, and username is also requested and set to lower case. The password for the username is then checked with the database and if correct the profile page is opened but if there is no match then the message 'Your username and or password was incorrect or does not exist' is passed to flask to be displayed on index.html page

Registration is handled on the index.html page where the logo sits on the left for medium size and then on top with small displays. The Registration information is next to the logo for medium and below for small screens. All the data is inputted in a form that requires the user to input their User name, Full name, E-mail, and then there password twice to make sure they don’t mistype it when registering. This is all inputted in the form in a text field expect the email in a email field and the passwords which are in password fields. Below this is also a list of all the topics that can be selected to follow by use of check boxes from a predefined list. All this information is then able to be posted with the “Sign me up!” button. The posted data is then handled by the views.py file which receives the Post data and checks to first see if it is login been attempted otherwise it will know it is registration. It first will pass the first password from the post information to a variable and then it will also pass the username and removes all spaces and converts to lower case. The username and password are validated by checking that the password contains one uppercase character, one number and that the length is 8 or more characters long. The username is just checked to contain only alphanumerical characters. If they are not valid then the message 'Password of incorrect format or username is not alphanumeric' is passed to flask. After this the two passwords are compared to make sure they are the same and if not then the message 'Passwords do not match' is passed to flask. With knowing all the data is validated then the rest of the form is requested so that the new user with their selected topics from the predefined list can be added but if the user does exist the message 'This user already exists' is passed to flask, otherwise the username is available, and the user is added to the database and logged in and redirected to their new profile page to continue with the website. If it is just a get from the page without the valid data to add the user, then they are just redirected back to the index.html page.