Ruaraidh Curran / 40209895

40209895@live.napier.ac.uk

Abstract

A social media platform made using the python flask framework

Social media platform

Set09103\_cw02

# Introduction

For my advanced web-tech portfolio I decided to make my own social media platform using the flask framework with SQLlite3 databases. The application allows users to create accounts to chat to other users, edit their own profiles and also to create, edit and publish posts. All the sites functionality is easily accessible to the user at all times. View Figure 1 for a preview of my website and the ease of accessibility to its functions. The following report explains my design choices and the structure of which, as well as enhancements I would like to implement at a later date, given some extra time and learning. I will be critically evaluating my work picking out points that I like about my application and criticising points I would have liked to improve on given some more time. The report concludes by discussing challenges I faced during the applications creation and then evaluating my planning and implementation methods. Finishing with a personal evaluation of how I feel I performed.

# Design

When a user accesses my website their normal landing location would be the home screen. They will be able to view information about the website here and also have access to the navigation bar at the top of the screen. My website holds many pages, please view Figure2for my sites navigation map.

## User Interface / User Experience

I wanted my application to be easy to use for all people at all technological levels; from older generations and children who may lack experience using websites, to the average person who uses the internet and web application every day. The website needed to have great functionality with the key features easily accessible. I believe that this is achieved with a design and user interface (UI) that is simple and clean. I found that the best way to implement this and allow users to access everything they need through the navigation bar that is located at the top of every page. From this navigation bar the user has complete control over where they want to go and what they want to see.

Having the option to verify a user’s email gave me the opportunity to restrict certain functionality of the website without taking away from the whole experience. So for your average user who may not find it difficult to access their emails and will most likely want the full functionality of the website I made it effortless to verify an account by simply clicking on a link in an email I sent to the email address entered during registration. This means that those less technically inclined, who might simply want to view posts and have the site as simple as it can be, have that opportunity without the hassle of going into their emails.

Having a pleasant and uniformed UI throughout the website as a whole is key to a sites success in attracting new users and holding on to existing ones. I implemented a consistent content section on each page application wide. This section runs down the centre of the screen with a dark yet slightly transparent background to make content easier to read but with leaving gutters on each side of this section as not to make the user need to strain their eyes too much moving from one side of the screen to another. When a user hovers over a functional button I want them to know about it so to help with this I had nearly all the functional links and buttons have an animation or some sort of change when the user hovers over them along with the default change in the cursor.

## Security

User security was one of the main features that I wanted to implement into my site as holding significant amounts of personal data is very high risk and must be done so with extreme care and precaution. I wanted to minimise any risk without taking away from the user experience (UX).

### Database

Holding information on people as I said comes with large risks in terms of possible attacks to capture this information. That is why sensitive information like the email is encrypted in the actual database meaning that not even I can see what any of the users emails are. Along with that I have made sure that the connection between the application and the database when running queries or inserting/updating information is as short as possible, making sure that the connection to the database is closed as soon as an operation is completed.

### Registration

During the registration process, once the user hits enter the name is parsed to create a separate first and second name. The email address is encrypted using the flask mail library token utility which encrypts the email using a 12 figure secret key that is held in a separate document as to not be accessible by anyone other than myself (the creator). Once this is complete, all the data is inserted into the database and is held in there. When the user decides the verify their email address all they need to do is access their email within an hour (the confirmation link in the email expires in one hour) and click on the link within the email I sent them. Once complete the database will update the users information and they will be acknowledged as a verified user and unlock all features of the website.

### Login/logout

When the user logs into their account as stated before they enter their login details (email and password) the database with the encrypted emails and passwords is then queried using the form’s input data. If the query comes back false, with no data matching what the user inputted, they are redirected back to the register screen with a message to try again. If however, the user entered correct details the data is entered into a python flask session which is also encrypted with a 12 character secret key and they will be successfully logged in. This session is popped as soon and the user logs out.

# Enhancements

Given some extra time on my project there are a number of things that I would like to implement which I believe would enhance the user experience and give extra functionality to the application.

1. Allowing users to create blog posts that hold images as well as plain text.
   * I think I could do this but given the time restraints I was unable to fit it in*.*
2. Allowing users to publish comments on other users blog posts.
   * I would need to create a new table and link comments to the individual posts using the post’s primary key.
3. I would like to make chats window accessible across the entire site with the use of a pop up window.
   * I would need to delve quite a bit further into unfamiliar technologies such as to be able to carry this out. Unfortunately time restraints prohibited me from doing this.
4. I would like to add an administration account to the site which had exclusive privileges.
   * This could be done by creating new pages that can only be accessed by a certain log-in. These pages will have access to administration tools allowing him/her to change information in the database without the normal user related restrictions Unfortunately time restraints prohibited me from doing this.

# Critical Evaluation

## Navigation bar

This navigation bar gives the user different options to choose from determined by whether the user is logged in or not. If the user is not logged in it will display the login or register buttons which would take the users to the login and register screens when clicked. If the user does not have an account and does not want to register the functionality of the site is limited to that user. The nav bar then only gives them the option to view users posts made to the site, search for posts and search for other users and view their profile. Once a user has made an account or already has one, the navigation bar changes, allowing the user to access a number of different features with things like viewing current chats they might have between other users, view their own profile, post their own blog posts, and of course the option to logout. I believe that this simple design with the navigation bar being consistent across the whole site which makes it very easy for the user to become familiar with it and user it efficiently.

## Login/Register System

When the user decides they want to utilise some more of the sites features they will need to start by registering an account or logging in. When a new user registers an account they need only to click on the register button at the top of their screen in the navigation bar and they will be directed to a register form. This is going to ask the user to input some basic information before allowing them to continue. The user is then required to input their name; preferably first and second. The user is then asked to enter a valid email along with their chosen password, typed in twice to verify that is entered correctly. Once complete, the user is prompted to verify their email or log-in to the website using those details although access to a number of site features are limited until the account has been verified via email. The following features are blocked from a non-verified user:

1. Edit their profile pictures
2. Chat to other users
3. Make blog posts

Once the user has simply clicked on the link sent to their email from utilising the flask mail library, they will unlock these features. This feature is a little more tricky for someone who is not technologically savvy so I left it as an option for users. I believe that I made this process and simple as possible for those who wish to use the applications full range of features.

Logging into the system Is easy as all a user needs to do is enter their valid details into the login screen press enter then database is then queried using the imputed information and if valid, the user will be successfully logged in where they can access all the sites features that are unlocked to them. In the future, I may consider using a log in system that is built into the navigation bar itself to maybe make this process a little more user friendly.

## User Profiles

A user is allocated a profile upon registering using the details they entered during the registration process. Each user can access their profile and view each other’s. These profiles comes with a default profile picture and a default biography. Profiles have links to the users’ other social media sites such as Facebook, Twitter and Instagram. There is also a link to their Github page and if a user is logged into the session they get the option to chat to other users via their profiles. All this information can be edited and changed by the user when they view their own profile and click the button saying to “Edit Profile” or “Upload Profile Pic”. Uploading a profile picture is simple and straight forward where the user selects a picture from their computer and clicks the “Upload” button. If the user clicks the “Edit Profile” button they will be directed to another screen giving them the option to update all of their information including their password. If a user wanted to search for someone else’s profile in particular they can use the “Search Users” button in the navigation bar which will give them the opportunity to search for different users using their profile names.

Having the user profile button on the navigation bar allows the user to get to his/her profile page with as little thought process as possible. I believe that they profile is laid out as simply as possible making it easy for the users to customize it as and how they want to.

## Blog Posts

Verified users have the option to create and edit their own blog posts. Each blog post created is held in a separate table in the database. These posts have the users unique identification attached to them and a timestamp along with the post’s title and content. Having the users ID attached to the blog posts allows me to easily see which user made which post and grant rights to edit that post. All blog posts can easily be viewed in the “View All Posts” section of the website or if a user is wanting to search for a particular one they can use the “Search For Posts” button on the navigation bar. This allows users to query the database for posts using a number of different variables. When a user has found the post they can click on them the users to access the full post data. From here the users will see all the posts content and can also access the post authors profile. On each user’s profile, people can view posts made by that user underneath their profile details. The use of different colours for the authors name to indicate that it has some click functionality I think is executed well, as other popular sites use this method and it is second nature for people to associate different colours on a website with different functionality. The use of differentiating colours makes it simple for users to distinguish between varying profiles.

## Direct Messaging

When a verified user views another user’s profile they will be given the option in the form of a button to chat to the user. Upon clicking the button the application will create a new entry in the chat database. Each entry uses the users ID in the session and the users from the profile that the chat button was clicked on to distinguish which users are in each chat. The user will be taken to a new page where they can simply enter text into a textbox and click submit. This data is then put into the database and the chat box with the user’s name next to it, the page is updated each time the user enters a new message. This message is viewable for the other user that is connected to the chat using the user IDs. from where the users can chat back and forth without ever losing their chat history. If a user wanted to quickly access previous chats they may have had with other users all they do is click the “View chats” button on the navigation bar and they will be directed to a new screen containing all their chats where they can simply click on one to enter the chat screen for each. As stated in the ‘Enhancements’ section of this report, I like the implementation I have, but if I was wanting to criticise it I would like to make the chats accessible site wide in the form of pop up window.

# Personal Evaluation

Throughout the development of this project I had to adapt and learn some new techniques with some libraries and extensions I had previously unheard of and some that I needed to practice with before utilising properly. I enjoyed the learning experience and getting to try out some new techniques and extensions such as “CSS grids”, “Flask\_mail” and the “itsdangerous” library. I am glad that I made myself use these as I feel like it has helped me better come to grips with the flask framework and development as a whole. I feel like I still have areas to improve, mostly with the planning and pre-production. Creating even simple lists of site functionality and sketching ideas I feel would have helped me in the long run. Another area that I feel like I could improve on is maybe clean code. Although my code, I believe, is adequately commented, I feel there may be better practical ways of splitting code into smaller more manageable sections I would like to practise this before taking on my next project.

Overall I am happy with my work and am looking forward to expanding my skills as a developer.

# References

## External Learning

Udemy Course. Web Accessed between 15/10/2018 and 20/11/2018: <https://www.udemy.com/python-flask-course/learn/v4/overview>

Davidism. Itsdangerous. Web Accessed between 15/10/2018 and 20/11/2018: <https://pythonhosted.org/itsdangerous/>

Youtube Tutorials. Web Accessed between 15/10/2018 and 20/11/2018:

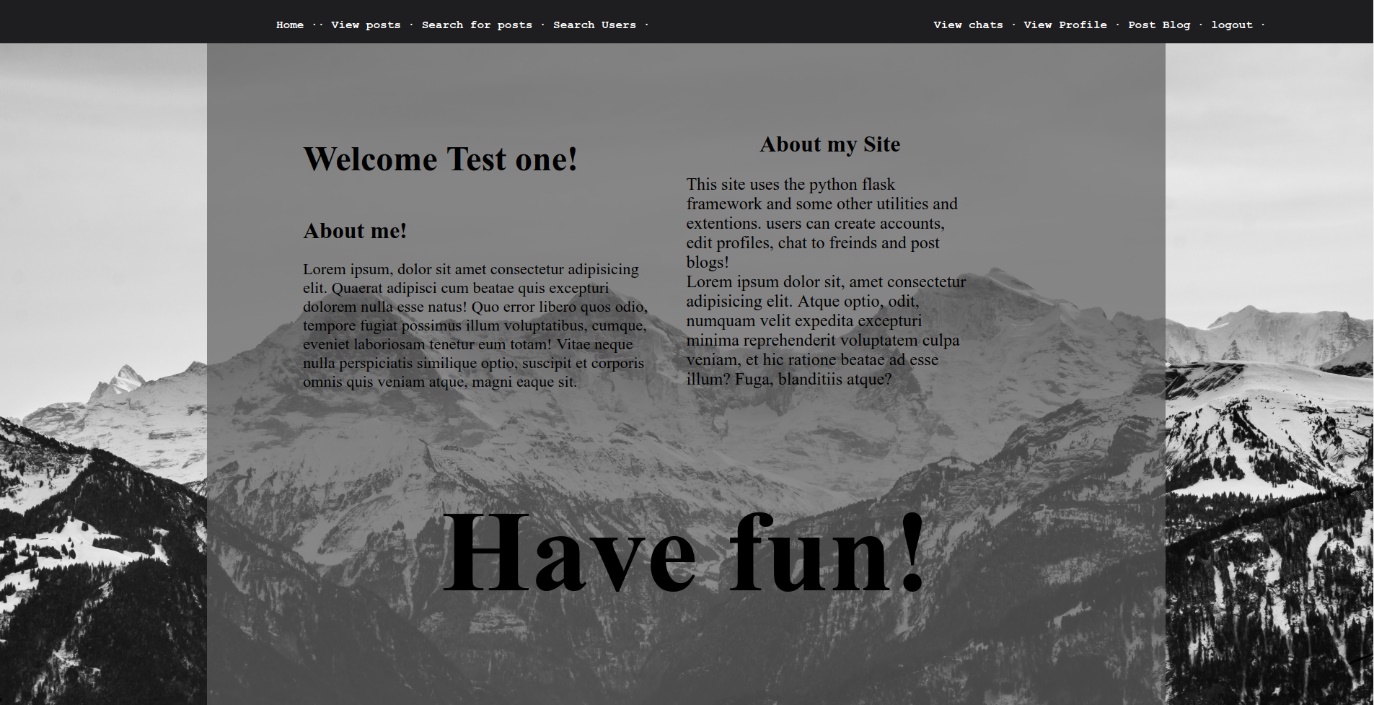
1. TraversyMedia: <https://www.youtube.com/user/TechGuyWeb>
2. PrettyPrinted: <https://www.youtube.com/channel/UC-QDfvrRIDB6F0bIO4I4HkQ>

## Software used

URL MAP: creatly.com. Web Accessed 20/11/2018: <https://creately.com/app/>

# Appendices

## Figure 1



## Figure2

