SET09103 Coursework 2

40183396

**Introduction**

The aim of this coursework was to design and create a prototype web application. The purpose was to show an understanding of the topics that were taught in this module. From this demonstration a mastery of the Python Flask approach to web development was shown. Also it highlighted using appropriate skills and tools in the Linux-based server set up. The design of the project was meant to be personal. So the scope of the project was to be anything that would be deemed to achieve a decent grade. It was assumed the app would be user friendly to someone who was not familiar with the language.

The website design is a twitter clone app named ‘MohsBook’. It allows users to log in and register details, and allows them to post messages to their walls. This project allowed creativity. Most of the topics in the workbook were covered including databases, password hashing and user logins.

A lot of these new things were learnt since doing the first coursework. The use of databases heavily featured in this app. So tutorials on ‘SQLAlchemy’ and ‘SQLite’ were used to create user tables and tables for wall posts.

There were difficulties involved. This mostly was due to user logins not working correctly. This meant it was difficult to get features such as the wall posts working. A lot of these problems were solved by looking up tutorials and searching for problems on the internet. When there was a problem that was too difficult to solve it was left for later so other features such as design could be done.

The purpose of this report is to give a background on how the app was designed. It is also to provide critical evaluation on how it functions and looks. There is a part on any enhancements that could have been added. And finally a personal evaluation describing what was learnt, what challenges were met, and how they were overcome.

**Design**

The app is stored in the ‘src’ folder. The python app is run from here. In the ‘src’ folder there are the ‘templates’ and ‘static’ folders. The app uses ‘Bootstrap’ as the CSS design. This is also in the static folder.

The website has a welcome page, login page, a profile page, and a registration page. It has templates for each of these. These templates all extend from a base template. On the welcome screen the user logs in or registers a new account. After this they are taken to profile page where their wall posts can be posted and displayed.

The app is designed so it can be run with the command ‘python app.py’ and the user can then load the site **‘**[**http://localhost:5000**](http://localhost:5000)’ on their browser. From here the login page loads up and the user enters their details for the username and password. This page was added to highlight the understanding of using requests in the app. From here the user can also register their details and make a new account. These details are stored in the database.

From here the profile page loads up. The design is in a table format at the top with a profile picture. And below is their wall with wall posts. The user can update their status by posting to the wall which gets added to the database.

The design for this was inspired from the website ‘Twitter’. The app was to display a profile page for each user with the statuses of each user.

**Enhancements**

The app is a simple twitter clone. However if it were to be improved on the feature which would be most important is the use of having friends and followers. This would make the app more of a social network.

Right now the app is a simple status poster. This is good for a prototype. The overall layout of CSS could be better. But this was left as the module is not focused on CSS. The overall feel of the app is good. It works as intended and with some improvements could be uploaded to a cloud. For example Heroku.

It would be good to add an information about the profile. Also having the user able to connect and follow other profiles. And also having working profile pictures to give a more social media feel.

**Critical Evaluation**

The app itself is reasonably simple. There could have been several things added in future to actually make it a much more usable prototype. With showing what has been learnt in the module so far it gives a good show of that. The addition of more features such as profile information and profile pictures could have been used.

The features that could have been added have already been mentioned in the enhancements paragraph. Overall though the site is pretty good for a simple prototype.

**Personal Evaluation**

The app is a good indicator of showing what the module has taught. It shows a mastery of using Python Flask in the web environment. It also shows using skills and applying them in the Linux text based server. There has also been attempts to cover all the things taught in the work book.

An exception to this is the use of password hashing. An attempt was made on this for user logins but it caused the app not to register login correctly. So this was removed. Other things that were used were databases. This was to store user details, and the wall posts they make. It also allowed the user to register new accounts.

The data storage however was done using an ORM. Instead of SQLite, SQLAlchemy was used. This was due to it providing a much easier way to upgrade and migrate databases.

Overall a lot knowledge was gained from this assignment. The freedom of choice allowed to explore features that were deemed initially difficult. Creating something similar to a social media app has been a valuable experience, and shows a lot can be done from the Python Flask set up.

**Summary**

Grinberg, M. (2012, July 16) *The Flask Mega Tutorial, Part VI: Profile Page And Avatars.* Retrieved from <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-vi-profile-page-and-avatars>

Grinberg, M. (2012, June 9) *The Flask Mega Tutorial, Part V: Profile User Logins.* Retrieved from

https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-v-user-logins

*W3.css Templates.* Retrieved from

http://www.w3schools.com/w3css/w3css\_templates.asp

Egil, J. and Refsnes, B. *w3schools css.* Retrieved from

<http://www.w3schools.com/lib/w3.css>

*Discover Flask.* Retrieved from

<https://github.com/realpython/discover-flask>

SQLAlchemy. Retrieved from

<http://flask-sqlalchemy.pocoo.org/2.1/>

Flask Migrate. Retrieved from

<https://flask-migrate.readthedocs.io/en/latest/>

Flask Bcrypt. Retrieved from

<https://flask-bcrypt.readthedocs.io/en/latest/>