# 'Natter' Report

Chris Reid
40202859@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

## Abstract

For this assignment, a web application was created using Python Flask. The application created is a social media web-app called 'Natter'. This report goes into details about the actions taken to design and create 'Natter', proposed enhancements, with a personal and critical Evaluation.

## 1 Introduction

The objective of this assignment was to display a working knowledge and understanding of server-side web development, using Python Flask. The application of choice was a social media web-app, that enables user to register, log in, follow other users, share comments and edit their own profile. A database was created using 'SQLite3' which stores 3 tables. A 'users' table which stores the users log in credentials and details, a 'comment' table - which is related to the 'user' table - to store a users comments, and a 'followers' table which stored the users id and followed users id.

## 2 Design

The main focus when creating 'Natter' was to create a intuitive and simple web-app, that displays information sensibly, while using the knowledge that was gained throughout the module. As it is a social media web-app, an important aspect is the registration and log-in page. Below is the details of how the web application was created.

### 2.1 Registration

Each user can create their own account on 'Natter' which is stored in a 'Sqlite3' database. When creating an account, the user is prompted for a user-name and password. The user-name is the unique identifier of the accounts, therefore the chosen user-name gets compared to the user-names within the database and prevents any duplicates. The user name must be between 6 and 20 characters long(string and integer values, as well as combinations of both, are valid). The password has to be more than 8 characters long to be valid. The password is hashed and salted using 'bcrypt' before being stored in the database.

### 2.2 Log in

Once a user has created an account, they are redirected to the 'log in' page (this is also the first page displayed when a user visits 'Natter'). The 'log in' page is simply a form that takes their user-name and password and checks they match a user-name and hashed password combination within the database. Once successfully logged in, they are redirected to the main page, known as their 'wall'. The 'wall' requires a user to be logged in and cannot be viewed otherwise. There is also a different navigation bar applied for a logged in user, than that of one who is not. The difference being the logged out user navigation bar has links to the 'log in' and 'register' page, while the logged in navigation bar has a link to the 'profile' page and a link to log out.

### 2.3 The 'Wall'

Like the majority of all social media, each user has their own home page, which for this web-app has been called their 'wall'. The 'wall' is where the user can, post comments, view other peoples comments and follow other users.

#### 2.3.1 User details

On the left hand side of the 'wall', there is a section dedicated for displaying the users details. The user name, profile picture, real name and a 'bio' gets displayed (the 'bio' being a short paragraph about themselves, that any user can add). When the user creates registers, the name and 'bio' are not yet added. Instead of displaying their information, there is a hyperlink text telling the user to update their information or to add a 'bio'. These links take the user to their profile page. The profile picture is a default profile picture that gets assigned to all new users; which can later be changed.

#### 2.3.2 Posting a comment

One of the main features of the web-app, is the ability for users to post comments. At the center of the users

'wall', there is a 'comment' section. Within this section there is a large text area, with a button underneath. A user simply types what they want into the text area and clicks the button and their comment gets stored in the database. These comments are then displayed on the users profile in descending order (to show most recent first) and can be viewed by other users that follow them.

#### 2.3.3 Following users

On the right side of the 'wall', a list of people they may know gets displayed. As this is sample data, and only a few users have been created to test the web-app, this displays all other users that are in the database. This section displays the other users user-name and profile picture within a separate bootstrap 'well'. Underneath, is a 'follow' button. When the logged in user clicks on the follow button, the followed users details get removed from the list and the page gets reloaded.(The followed users id also gets inserted into the database to create a link between the two users.)

### 2.4 Profile page

Each user has a profile page where they can edit their own information, update a profile picture, post a comment and view their comment history.

#### 2.4.1 'User details' section

On the left hand side, all the user information is displayed, along with their profile picture. From this section, the user can update their profile picture. A file input has been added, which enables the user to browse their computer and select the picture the want. This makes it a simple process. When the user then hits the update button, the picture is uploaded to the server and the title of the image is stored in the database, overwriting the last picture title. The new profile picture is assigned.

In the same section, but below, there are 2 text boxes and a text area. The user can then update their user accounts first and last name, along with adding a 'bio'. When this is added, it gets stored in the database and displayed on the users 'wall'. The values are displayed in the text boxes when the user navigates to their profile.

#### 2.4.2 'Post history' section

The right side of the 'profile' page is simply another section for adding comments, with all previously made comments displayed below. The comments are ordered by most recent.

## 3 Enhancements

The web application could be made extremely better with a number of enhancements. In its current state, 'Natter' is a simple social media web-app with basic features. Many of these features were initially planned to be included, but time became a deciding factor on what could be implemented.

### 3.1 Email validation

An important improvement would be to make users authenticate their accounts through an email address. By doing this, it would prevent users from making many accounts and reduce the amount of anonymous accounts being made.

### 3.2 Deleting a post

The feature that was most recently being worked on but never got implemented, was giving the user the ability to delete comments within their comment history. It is a very common feature and one that is almost mandatory for a social media web-app.

### 3.3 Viewing other users profiles

The ability for a user to view another users profile is another common social media feature. To implement this would be relatively easy as there is already a profile template. The only changes that would need to be made are to remove the buttons and have the information only be displayed. The idea was to have the URL be root plus the user-name ('/<username>).

### 3.4 Improve the 'people you may know' section

Currently, this displays all the users that have been stored in the database. Finding a way to improve who gets displayed to a user would greatly increase the efficiency of the web-app. It would also make it significantly easier for a user to find other users they may know.

### 3.5 Improving how information is displayed

Another enhancement would be to limit the amount of posts or other users shown when the page is loaded. Currently, it displays all the posts and all of the users. With more users this is a bad way for the information to be handled. Limiting the posts to a maximum of 20 and enabling the user to load more with either a button or implementing an 'infinite scroll' would be an ideal solution.

### 3.6 Messages

Private messaging is another feature that would improve the web-app. Enabling users to communicate directly and in private is another feature that most users would expect a social media web-app to have.

### 3.7 Search bar

Implementing a search bar that enables the user to find other users is another enhancement that would greatly improve the web-app by making it more efficient for users to find people they are searching for.

## 4 Critical Evaluation

The web-app works well and enables simple communication between users. The main problem is that there are many features that could be added to enhance the experience for the user and improve the functionality of it.

The registration and 'log in' work well, although it could be improved with an email authentication added. The appearance of both pages is decent and the way the text box validation is displayed looks quite professional. Validation is also implemented well and prevents duplicate user-names, only letters or numbers, which have to be between certain lengths. Furthermore, having the password hashed and salted improves the security of the users accounts.

One small feature that was implemented that works well is the uploading of the users profile pictures. In the beginning, this was done through supplying a file path into a text box. This way is much easier for the user and reduces the chances of user errors.

The overall layout of each page looks well, and the use of 'Bootstrap' was well implemented, which means that the web-app is responsive (a few button positions need adjusted when on a small screen).

As previously mentioned, the main problem with the web-app is that it needs more features added. One aspect that needs redesigned is how the user edits their details. Currently, it is just text boxes and a text area, with the values loaded in them, which looks quite unprofessional.

Other than that, the problems are nearly all to do with the appearance. A better colour scheme is needed, which would immediately improve the web-app. The styling of images needs to be fixed as they differ depending on the width, which results in different shaped images when they are circular.

## 5 Personal Evaluation

One of the biggest difficulties came from a personal decision to change what the web-app was going to be, after some time had already passed. By doing this, it made the process a great deal more stressful and time became a huge issue. In the beginning, the web-app was an 'RSS aggregator', but due to difficulties with loading times and spending great amounts of time with no real improvements, the decision was made to make 'Natter'.

The biggest challenge for this project was using 'SQLite', due to being fairly new to python and flask and having no experience with 'SQLite'. This was made much easier thanks to the resources in the workbook, flasks own documentation and previous experience using 'MySQL'.

Overall, personal performance during this assignment was very good. Working on this web-app was extremely enjoyable and a great deal was learned, along with increasing experience working with databases. The only personal problem was making bad choices, reducing the amount of time that could have been spent focusing on the web-app. With more time, the web-app would have been much better than how it is in its current state.