Blah

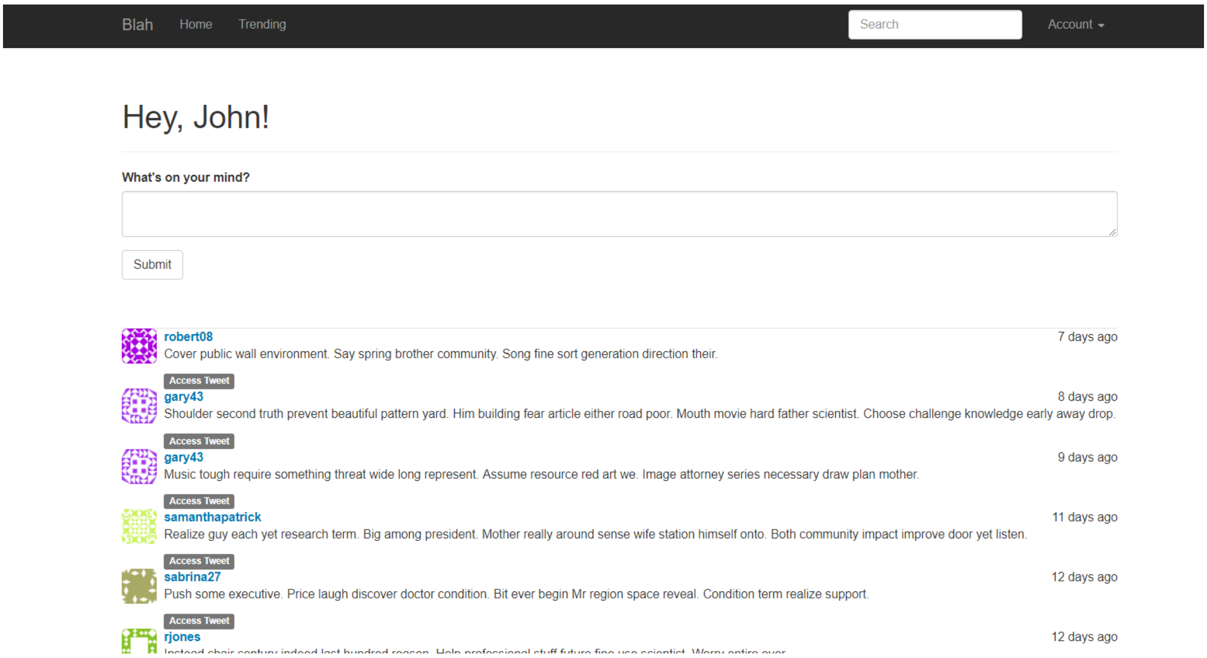
Enrique Belenguer

40406742@live.napier.ac.uk

Edinburgh Napier University – Advanced Web Technologies (SET09103)

Introduction

Blah App is a simple Twitter clone which allows registered users to share their thoughts with others. The app is built using Python Flask Microframework, Bootstrap and Html. It uses a secure login and registration system. Blah sends an email to new users with a token that must be confirmed to finish the registration. If not confirmed, the user won't be allowed to use all of the app. Users are allowed to change their passwords, email addresses, as well as to follow and unfollow people. There is, however, one difference from Twitter- there is a limit to the number of users they can follow. They cannot follow more than 100 users at the same time, if users want to follow someone new when they reach their limit, they should decide who to unfollow. In addition, users can edit their profile data. The Administrator is allowed to edit the data for the rest of the users too.



Hey, John!

What's on your mind?

robert08 7 days ago
Cover public wall environment. Say spring brother community. Song fine sort generation direction their.

Access Tweet

gary43 8 days ago
Shoulder second truth prevent beautiful pattern yard. Him building fear article either road poor. Mouth movie hard father scientist. Choose challenge knowledge early away drop.

Access Tweet

gary43 9 days ago
Music tough require something threat wide long represent. Assume resource red art we. Image attorney series necessary draw plan mother.

Access Tweet

samanthapatrick 11 days ago
Realize guy each yet research term. Big among president. Mother really around sense wife station himself onto. Both community impact improve door yet listen.

Access Tweet

sabrina27 12 days ago
Push some executive. Price laugh discover doctor condition. Bit ever begin Mr region space reveal. Condition term realize support.

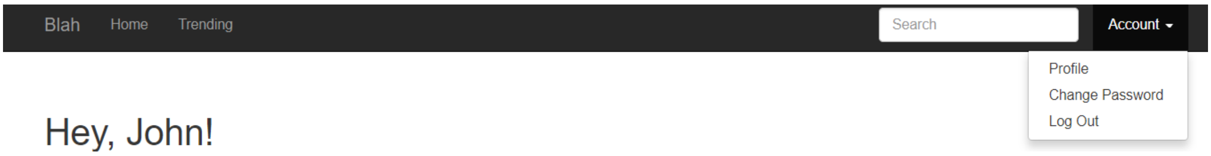
Access Tweet

rjones 12 days ago
Instead chair century indeed last hundred reason. Help professional stuff future fine use scientist. Worry entire over.

Figure 1. Home Page

Design

The design of my app is quite simple and designed to facilitate the user's experience. In the upper part, the user finds a 'Top-Menu' with the different sections of the web page. In 'Home' the anonymous users will be able to see all the posts. Registered users will see the posts of people who they follow. When a user logs into the app, a 'Trending' buttom appears and it allows the registered users to see the posts from everyone-even if they don't follow them. In addition, on the right side of the bar, there is a drop-down with different options. It allows the registered users access to: their profile, to change their passwords and to log out.

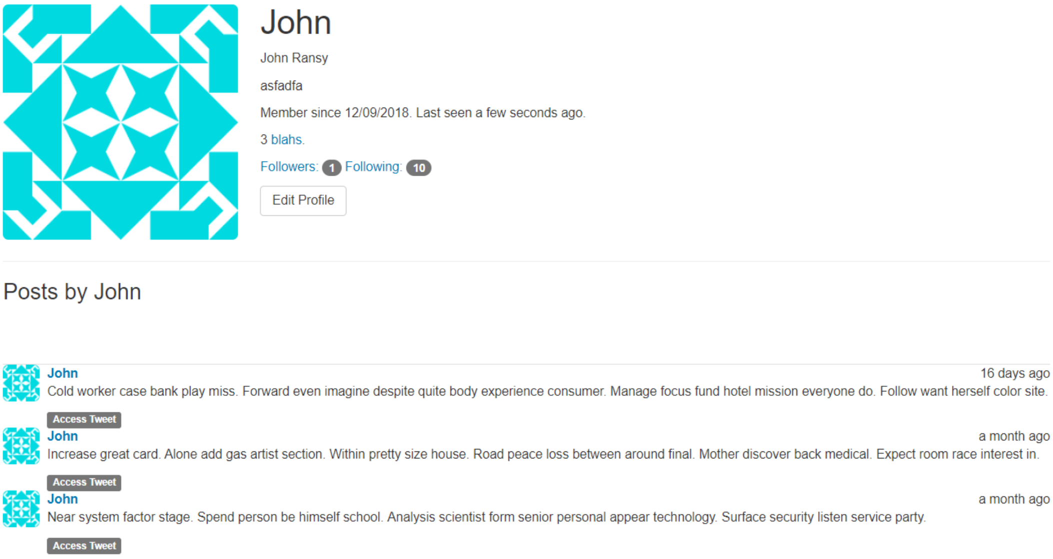


Hey, John!

Figure 2. Top Menu

The characteristics of each page are quite similar, and the colour scheme is the same too, which helps to give consistency to the web. It helps the users to feel comfortable and to not feel any distractions while they are navigating it.

The same happens with the information it contains. The user can move from the feed to any profile or tweet. From the profile they can go to a page where they can see their followers and the users who follows them and also have the access to all their tweets.



John

John Ransy

asfadfa

Member since 12/09/2018. Last seen a few seconds ago.

3 blahs.

Followers: 1 Following: 10

Edit Profile

Posts by John

John 16 days ago
Cold worker case bank play miss. Forward even imagine despite quite body experience consumer. Manage focus fund hotel mission everyone do. Follow want herself color site.
Access Tweet

John a month ago
Increase great card. Alone add gas artist section. Within pretty size house. Road peace loss between around final. Mother discover back medical. Expect room race interest in.
Access Tweet

John a month ago
Near system factor stage. Spend person be himself school. Analysis scientist form senior personal appear technology. Surface security listen service party.
Access Tweet

Figure 3. Profile Page

Another potential problem was trying to always keep the users well informed about their situation within the website. For this I have used an easy and intuitive URL pattern.

Even so, in case the users type a bad URL, he will receive an appropriate message to communicate its error.

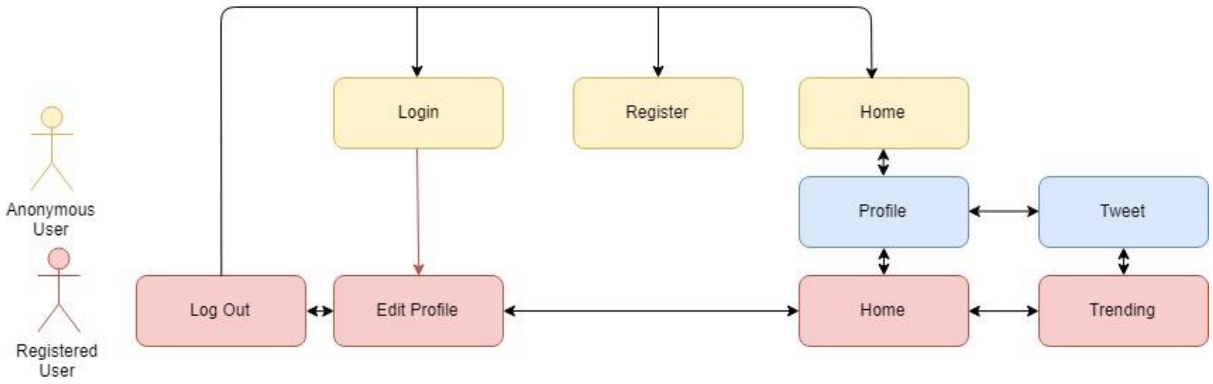


Figure 4. Navigation Map

Enhancements

The page could have many improvements; foremost, the functionality of the search bar. For almost a week I have tried to implement it and even if it is thoroughly implemented and the code is still in the app, it does not work. I cannot figure out why, but during my Christmas holidays, with more time and without pressure, I will investigate.

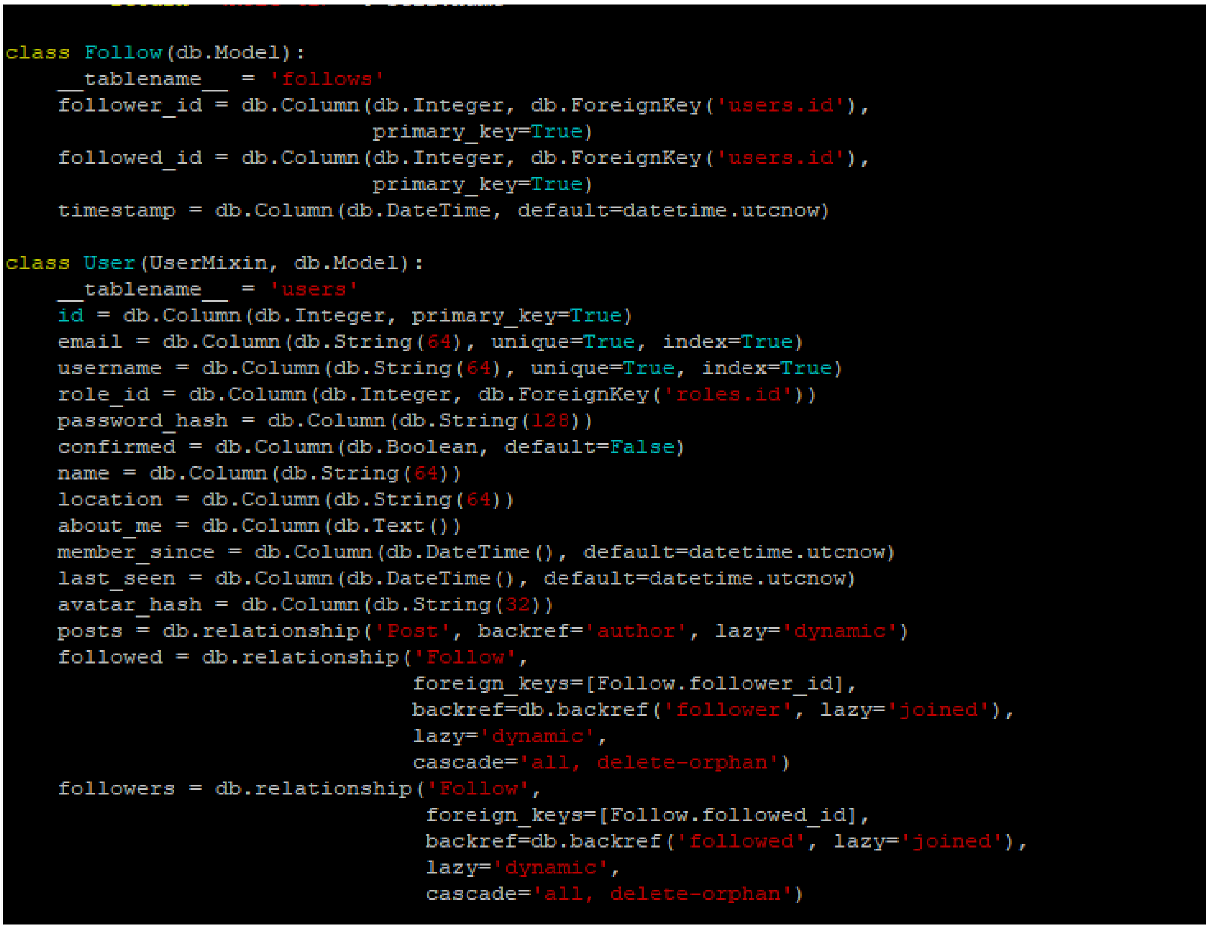
Also, a good Bootstrap template would add a lot to the website, but I could not find any appropriate design and I have ended up leaving the default one, which does not look great.

Critical Evaluation

One of the features that works the best on the web, and which I feel the proudest, of is the ‘Followers’ and ‘Followed’ implementation. I had a serious problem to understand the way that SQLAlchemy works and I could not figure out how the followers table could work.

In addition, I must say, that the confirmation email with the token gives professionalism to my web app.

On the other hand, the fact that I have not changed the Bootstrap template, makes my app appear a bit less than the way it should.



```
class Follow(db.Model):
    __tablename__ = 'follows'
    follower_id = db.Column(db.Integer, db.ForeignKey('users.id'),
                            primary_key=True)
    followed_id = db.Column(db.Integer, db.ForeignKey('users.id'),
                            primary_key=True)
    timestamp = db.Column(db.DateTime, default=datetime.utcnow)

class User(UserMixin, db.Model):
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(64), unique=True, index=True)
    username = db.Column(db.String(64), unique=True, index=True)
    role_id = db.Column(db.Integer, db.ForeignKey('roles.id'))
    password_hash = db.Column(db.String(128))
    confirmed = db.Column(db.Boolean, default=False)
    name = db.Column(db.String(64))
    location = db.Column(db.String(64))
    about_me = db.Column(db.Text())
    member_since = db.Column(db.DateTime(), default=datetime.utcnow)
    last_seen = db.Column(db.DateTime(), default=datetime.utcnow)
    avatar_hash = db.Column(db.String(32))
    posts = db.relationship('Post', backref='author', lazy='dynamic')
    followed = db.relationship('Follow',
                               foreign_keys=[Follow.follower_id],
                               backref=db.backref('follower', lazy='joined'),
                               lazy='dynamic',
                               cascade='all, delete-orphan')
    followers = db.relationship('Follow',
                                foreign_keys=[Follow.followed_id],
                                backref=db.backref('followed', lazy='joined'),
                                lazy='dynamic',
                                cascade='all, delete-orphan')
```

Figure 5. Follow and Users Table

Personal Evaluation

I have improved many of my skills during the development of the project, however I am not happy with what I have achieved during this coursework.

Overall, the application meets the coursework criteria. I have improved my skills with Vim and with Python Flask Framework. I have learnt how a Large Application Structure must be implemented, also how to use Forms and Databases, but I think I could have done way more than what I have.

References

Grinberg, M (March 2018). Flask Web Development (2nd). O'Really

Miguel Gringberg. 2018. Miguel Gringberg Blog. [ONLINE] Available at: https://blog.miguelgrinberg.com/. [Accessed 1 December 2018].

Flask Web Develpment. 2018. Flask Pocoo. [ONLINE] Available at: http://flask.pocoo.org/. [Accessed 1 December 2018].