**Extra Feature**

As a group we have decided to implement two extra features: email notifications and the ability to ‘like’ other users.

Likes:

The current logged in user can “like” other users’ profiles. All users (apart from the current logged in user) have a like button attached to their profile. The logged in user can choose to press this button and like the users’ profile. This will increase the number of likes that that user has on their profile. This information is retained in the database. To implement this feature, we created a third model in ‘models.py’ called Like. This model has two variables; “user” and “likedUser”. These variables are both foreign keys that point to the “UserProfile” model. The idea is that when the current user (user) decides to like another users profile (likedUser), this is stored in the database. The id of the current user is stored in the “user” column and the id of the profile that the current user liked is stored in the “likedUser” column. This new entry is then stored in the database. We have also included validation to ensure that a logged in user cannot like the same profile more than once. If they attempt to do so, an alert is printed on the screen informing them that they have already liked the users’ profile.

Email Notifications:

We also decided to implement email notifications in our web application. Each user when registering must also provide their email address. If their profile is “liked” by another user as discussed previously, the user will receive an email notifying them that their profile has been liked. Django provides wrappers for existing Python functionality included in the smtplib to make it efficient to send emails. To do this, we had to modify the settings.py file and the views.py file. In the settings.py, we defined some email features such as; address, password and host. To notify the user once their profile has been liked, we “EmailMessage” feature, providing a title, body and recipient email address. We obtain the recipients’ email address by searching through the UserProfile models in the database and matching the correct user.