A Matching Site

Group: Four students
Mark: 15%
Deadline: Friday 14 December

Description: Your task is to develop a **matching website** using the Django framework. The website should provide the following basic functionalities:

1. Users can create an account on the site and login.
2. The user's profile should contain (at least): profile image, email, gender, date of birth, and a list of hobbies.
3. The overall list of hobbies should be defined in advance by the application developer, so the users of the site can only select one or more hobbies from the given list. (i.e. on your DB you should have a table for User and another for Hobby, with a many-to-many relationship between them)
4. Users should then see a list of other users who have the most similar set of hobbies, i.e. for each two users you should count how many hobbies in common they have, and then list those users in descending order (users with most common hobbies first).
5. From the list above users should be able to filter by gender and/or age, e.g. only females with ages between 30 and 50. Searching and filtering should be done using Ajax and jQuery.
6. Frontend should use Bootstrap, and be responsive.
7. Apart from the basic features above, you should implement at least one extra feature. Feel free to include any extra feature you can think of. Here are two examples of possible extra features:
   a) Users are able to request to connect with another user. The other user would then need to approve the request.
   b) Users are able to "like" other users — and users receive alerts or emails when someone likes them.

Outcome: Once fully tested, your application should be deployed to the school's OpenShift web servers (to be discussed in week 10) — one deployed app per team. Each group should submit the **code** together with a **one-page report** describing their extra feature. Submit these to QM+ as a single zip file. Remember to include in the report the URL of your deployed application, and the username and password for the admin page.

Marking criteria:

1. [10%] Application is deployed on OpenShift. DB is populated and ready for testing.
2. [35%] Basic functionalities implemented and fully working.
3. [10%] Good modelling of the application data, making use of Django's ORM.
4. [10%] Appropriate uses of Ajax and jQuery.
5. [10%] Code is well-written and appropriately commented.
6. [10%] Django/python features fully explored (e.g. decorators, filters, forms,…).
7. [15%] Extra feature included, and report succinctly and clearly explains the feature.