**University of Leeds School of Computing**

**COMP3011, 2019-2020**

**Web Services and Web Data**

A RESTful API for

Rating Professors

By

insert your name here

insert your university number and email here

**Date:** insert date here

# Introduction

Explain how much of the coursework you have been able to achieve (for example, have you been able to implement all features or just the database).

Clearly state that you have uploaded the Django server code to pythonanywhere.com and that you have written the client in Python 3 and thoroughly tested it on a DEC10 machine. **If you do not upload the server code to pythonanywhere.com or test your work on a DEC10 computer, it will not be marked, and you will score 0 in this coursework.**

Also, state the following:

1. The URL of your pythonanywhere account.
2. A superuser name and password that the assessor of your coursework can use to access the admin site of your service.

(Restrict this section to a maximum of half a page)

# The Database

Clearly but briefly explain the design and implementation of the Django database model of the service. You must specify all required tables, the fields of each table and the data type of each field, and the relationships between tables. You should also indicate which fields have unique values.

(Restrict this section to a maximum of one page)

# The APIs

Clearly yet briefly explain your API. For each API, you must specify: the purpose of the API, the URL, the HTTP method, the request data, and the response data. Please use the examples at the end of Lecture 8 as templates for this.

(Restrict this section to a maximum of 2 pages)

# The Client

Clearly but briefly explain how you implemented the client.

(Restrict this to a maximum of half a page)

# Testing

Clearly indicate that you have tested the service using the testing protocol explained in Appendix A.

(Restrict this section to a maximum of half a page)

# Using the client

You should provide clear and complete instructions on how the client can be used and the syntax of the commands. Ambiguous or incomplete instructions will make it difficult for the assessor to test your client and could result in the loss of marks.

State if there are some libraries that should be pip-installed to run the client.

(Restrict this section to a maximum of half a page)

**Appendix A: Testing Protocol**

To test your service, the assessor will login to your admin site using the superuser name and password that you have provided. The assessor should then be able to use the admin site to:

1. Add the following two modules:
2. Computing for Dummies.
3. Programming for the Gifted.
4. Add the following three professors:
5. J. Excellent.
6. V. Smart.
7. T. Terrible.
8. Create the following module instances:
9. Computing for Dummies, Semester 1, Year 2017, taught by J. Excellent and V. Smart
10. Computing for Dummies, Semester 2, Year 2018, taught by J. Excellent.
11. Programming for the Gifted, Semester 2, Year 2017, taught by T. Terrible.
12. Programming for the Gifted, Semester 1, Year 2019, taught by J. Excellent.

The assessor will then use your client to:

1. Register two new users (students).
2. Login in as the first user and use the *list* command to see a list of all module instances.
3. Use the *rate* command to:
4. Give professor J. Excellent a rating of 2 stars in module instance a), and

a rating of 5 stars in module instance b).

1. Give professor V. Smart a rating of 3 stars in module instance a).
2. Give professor T. Terrible a rating of 1 star in module instance c).
3. Give professor J. Excellent a rating of 4 stars in module instance d).
4. Logout, then login as the second user, and use the *rate* command to:
5. Give professor J. Excellent a rating of 3 stars in module instance a), and

a rating of 4 stars in module instance b).

1. Give professor V. Smart a rating of 2 stars in module instance a).
2. Give professor T. Terrible a rating of 1 star in module instance c).
3. Use the *view* command to see the rating of all professors. For the above data, professor J. Excellent would have a rating of 4, V. Smart 3, and T. Terrible 1. Note that fractions are rounded to nearest integers.
4. Use the *average* command to see the rating of professor J. Excellent in the Computing for Dummies module. For the above data this would be 4.