Edinburgh Napier UNIVERSITY

Advanced Web Technologies SET09103
Coursework Specification #1

1 Task

The objective is to demonstrate your understanding of the Python Flask micro-framework by creating a prototype web application for an online directory about a given subject. Your subject can be music, movies, books, stars, butterflies, rabbits, or any other domain that interests you. If you have a novel idea for a collection then speak to the module coordinator to make sure that it is suitable. You should carefully consider the nature of the problem domain, and design a URL hierarchy that is appropriate for finding and retrieving information about the collection. For example, when we think about a collection of music we often consider genre and artist, but also other kinds of metadata like release dates, formats, number of tracks, track length, album length, and many more parameters (it is worth looking at online music streaming sites & MP3 management tools to get an idea of the ways that music can be organised and discovered).

This coursework should be fun, so use your imagination, and give your creativity a free rein. Invention and originality will be rewarded by the marking scheme. I hope you enjoy working on it.

2 Submission & Deliverables

Your coursework deliverables comprise the following:

1. Source code.
2. Report.

The method of submission for both deliverables is via a single Git repository. Both your report and source code must be committed to a Git repository in "sourcecode" and "report" folders respectively.

- All sourcecode and your report must be placed in a Git repository.
- Your Git repository must be named lastname_firstname_set09103_cw1.
- Your repository must be pushed to a hosting service, e.g. Bitbucket, Github, or the school's internal Git server.
- Email the Git clone URL for your repository to s.wells@napier.ac.uk at least one week before the assignment deadline. This should be the SSH clone URL (the one that starts with either git@github or git@bitbucket).
- If your repository is private then your must add the user siwells as a collaborator so that your work can be retrieved.
- Your Git repository must contain only the files for your web-app, e.g. the sourcecode and report. It is your responsibility to ensure that you have placed all of the source code necessary to run your web-app in your repository. Note that this does not mean you should include either Python or Flask or any of the other tools that are already installed on the module's development server.

If you are in any doubt about any of the requirements for the coursework or any aspect of the submission procedure then please contact the module coordinator for further guidance.

2.1 Sourcecode

Your web-app must not use any tools beyond those supplied in the default learning environment and any code that you have written yourself. This means no additional Python or Flask modules, libraries, or plugins. The rule of thumb is that if any additional software must be installed then it is not allowed in this coursework. External web APIs are also prohibited. However you may use HTML, CSS, & Javascript via the Flask static folder, i.e. You can use Bootstrap or similar design frameworks that are entirely installed into the Flask app's static folder. These files must subsequently be included in your repository and referenced in the appropriate section of your written report.

2.2 Report

Your report must be no longer than 6 pages in length (excluding appendices) and written using the Napier report template:

`http://github.com/edinburgh-napier/aux_latex_cw_template`

It is recommendeed that you use LaTeXto typeset your report, but so long as your report is according to the template format this will not be enforced. The format of the submitted report must be PDF and must include the following sections:

Title of your web-app.

Introduction Describing your web-app and its core features. This section should include at least one screenshot of your web-apps main page.

Design Explaining how and why your web-app is structured the way it is. This section should include a navigation map for the URL hierarchy of your web-app.

Enhancements Describing the features that you would add or improve.

Critical Evaluation This is about *what* you built and should explain the features of the web-app that you feel work well, or work poorly, and why.

Personal Evaluation This is about how *you* perfomed. You should reflect upon what you learned, the challenges you faced, the methods you used to overcome challenges, and how you feel you performed.

References If you have used additional resources then these should be cited. Please acknowledge all sources of help and material. If there is anything included that was not written by you for this module, then you should indicate that within this section.

Appendices (Optional) Appendices may be used to include supplemental data, for example test data, screenshots, or documentation, but these must be referenced from within the main body of your report.

2.3 Demonstration

It is school policy that all coursework must be demonstrated and that without a demonstration your submission will not be marked. Demonstrations will be held during regular lab sessions and all students will have the opportunity to sign-up for a demo slot. The module coordinator will contact the class closer to the deadline to organise demo slots.

2.4 Important Dates

- Clone URL to module coordinator: At least one week before the deadline.
- Submission deadline: 11:45PM on Sunday 21st October.
- Demos: During the regular lab sessions on Monday 22nd October & Monday 29th October.
- Return of work & feedback: Under ideal circumstances you will be emailed written feedback within three working weeks of the submission deadline. However you will also receive verbal feedback both during your demonstration and during contact time throughout the trimester. Verbal feedback is as important, sometimes moreso, than written feedback, and should neither be discounted nor disregarded.

2.5 Assessment Criteria & Grade Guide

This coursework is worth 40% of your overall grade for this module. The remaining 60% come from coursework #2. A broad indication of the requirements for each grade band is as follows:

40-49% This grade band indicates work that is acceptable, that demonstrates basic skills in the core concepts covered by the module, but where there is plenty of room for improvement. To achieve a mark in this band you must have developed your own working web-app with multiple routes allowing the user some, but not extensive interaction. It may be based directly on an extension of the practical work covered in class and your report must adequately describe your work.

50-59% A grade in this band constitutes good work. A submission in this mark band will indicate that you have developed a web-app that is less ambitious in its functionality but will offer the user suitable ways of interacting. Your report will be well written and will reference the material you have used.

60-69% Work in this grade band is at a very-good level. To achieve a mark in this band you will have developed a web-app with very good functionality, for example, offering the user multiple URLs together with some evidence of appropriately designed routing, correct use of requests, redirects, responses, custom error code handling, and appropriate use of static files and templates. Your report will address all the necessary sections effectively, be very well written and clearly presented and will reference material you have used.

70-100% This grade band indicates work at an excellent (70+), exceptional (80+), or exemplary (90+) level. A submission in this mark band will consist of an application that has extended the lab work covered in class to offer an excellent level of functionality, both in terms of the number of features and their quality of implementation. To attract a grade at this level a submission must also have effectively evaluated. Your design and code will be excellent making good use of Flask features and having an exemplary design, application, and URL layout (API). Your report with the sections detailed above will be comprehensive, very well written and well presented and will correctly reference all the material you have used. This is likely to include textbooks, online forums and tutorials and some of the suggested reading for the module.

You should think strategically about how to approach this assignment. The grade guide is cumulative, i.e. to get a higher grade, you must also have achieved the functionality required to attain a lower grade. A lower-risk strategy is to identify the core features that you think will attract a pass mark then aim to complete those features as soon as possible. You can then iterate over your *working* solution to improve those features.

1. Module number	SET09103
2. Module title	Advanced Web Technologies
3. Module leader	Dr Simon Wells
4. Tutor with responsibility for this Assessment	Dr Simon Wells
5. Assessment	See Attached coursework description
6. Weighting	40%
7. Size and/or time limits for assessment	See Attached coursework description
8. Deadline of submission	Your attention is drawn to the penalties for late submissions **11:45PM on Sunday 21st October**
9. Arrangements for submission	See Attached coursework description
10. Assessment Regulations	All assessments are subject to the University Regulations
11. The requirements for the assessment	See Attached coursework description
12. Special instructions	None
13. Return of work and feedback	You will be emailed written feedback within three working weeks of the submission deadline. However you will also receive verbal feedback during your demonstration.
14. Assessment criteria	See Attached coursework description for details. This assessment covers: **LO1:** Understand the role of HTTP and related protocols in the design and efficient exploitation of robust and scalable services and APIs for the Internet and Web. **LO2:** Evaluate the sensitivity of data gathered by your Web app and select appropriate tools and techniques to ensure its security and privacy. **LO3:** Demonstrate effective use of client side scripting languages and libraries at an advanced level to produce a compelling user experience. **LO4:** Demonstrate competence at an advanced level in the design, development, and evaluation of web applications and services using server-side languages, libraries, and tools.