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 catalogue. Your online catalogue can be a catalogue of music, movies, books, stars, butterflies, rabbits, or some other collection 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).

The 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 coursework needs to be demonstrated. 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. 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. Use Bootstrap or similar design frameworks.

- All sourcecode and your report must be placed in a Git repository.
- Your Git repository must be named lastname_firstname_set09103_coursework1.
- Your repository must be pushed to a hosting service, e.g. Bitbucket or Github.
- Email the Git clone URL for your repository to   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 your web-app files and not your entire SET09103 folder of workbook and lab content. 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.

2.2 Report

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

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

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. The format of the submitted report must be PDF and should include the following sections:

Title of your web-app.

Introduction Describing your web-app.

Design Explaining how you architected your web-app.

Enhancements Describing the features that you would add or improve.

Critical Evaluation Explaining the features that you feel work well, or work poorly, and why.

Personal Evaluation Reflecting on what you learned, the challenges you faced, the methods you used to overcome challenges, and how you feel you performed.

References (Optional) If you have used additional resources then these should be cited. Otherwise this section may be omitted.

2.3 Important Dates

- Clone URL to module coordinator: At least one week before the deadline.
- Submission deadline: 11:45PM on Wednesday 25th October.
- Demos: During the regular lab sessions on Thursday 26th October & Thursday 2nd November.
- Return of work: you will be emailed written feedback within three working weeks of the submission deadline. However you will also receive verbal feedback during your demonstration.

2.4 Assessment Criteria & Marking Scheme

This coursework is worth 40% of your overall grade for this module. The remaining 60% come from coursework #2. The mark breakdown for this coursework is as follows:

Report (20%)

Formatting, Citations, Grammar Correct use of Latex template & academic language and style, i.e. third person, concise, and formal.	10%
Content Well written mandatory sections: Introduction, design, enhancements, critical & personal evaluations.	10%

Source Code (20%)

Code Quality Your code should be readable and well commented.	5%
Software Engineering Your code should be well structured.	5%
Efficiency Your software should be performant.	5%
Evidence of good version control practice Good commit messages & commit granularity, use of cloud hosting, good feature branching (if necessary).	5%

Features (60%)

Route, URL Hierarchy, & Navigation (Design & Implementation) Users should be able to navigate a sensible & well designed site.	15%
Requests, Responses, & Redirects Use of these elements should be appropriate to the design of the collection.	15%
Templates Correct usage of templates will reduce instances of HTML embedded in Python code.	15%
Static Files Including use of images, Javascript, CSS, &c.	15%

2.5 Grade Guide

70-100% 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.

60-69% 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.

50-59% 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.

40-49% 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.

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.