Advanced Web Technologies Coursework Specification

Overview

The assignment for this module is a single project that is split into two parts. This document details both parts. The objective is to demonstrate your understanding of server-side web development and mastery of the Python Flask micro-framework. You will achieve this by completing a personal project in which you design, implement, and evaluate a web application on a topic of your choice. Because this is a project, there is an overall goal that you should be working towards, but you also have considerable flexibility in how you achieve this and which features you choose to incorporate. This is a personal project so the web-app that you will build for this assignment is completely up to you. However it is advisable to discuss your idea with the module leader in advance to ensure that it has sufficient scope to enable you to achieve a decent grade. Some ideas for suitable projects include a simple Twitter clone or a Bookmarking/URL shortening web-app. A list of topics that students have investigated in previous years has been posted to Moodle which might help you to get started.

Once you've decided what you're going to build you should carefully consider the nature of the problem domain, and design a URL hierarchy and user interface that provides your users with a good experience. If appropriate you may also include additional features, functionality and APIs as necessary to implement your project. You may also use an appropriate mix of client and server-side web technologies in order to implement any additional features that you want to include. It is a good idea to research similar web-apps that you can use as a benchmark against which to measure the functionality of your own and which might provide inspiration for how other developers have solved similar problems.

The first part of the assignment is a report which will focus on the features, analysis, design, and plan for implementation of your web-app & is worth 40% of the total mark for this assignment. The second part of the assignment is the implementation and deployment of your design and is worth 60%.

It is expected that what you implement might well deviate from your initial design and that techniques you discover later in the module might cause you to re-think or re-approach decisions or plan you have already made. You might even discover that your initial idea needs to be completely replaced with something more achievable. This is perfectly fine and gives you an opportunity for reflection in your final report.

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.

Part #1

For this part of the assignment you will develop an idea for your web-app alongside a design and a plan for achieving an implementation of that design. These will be presented in a report. You will likely want to develop a satisfying user experience, and utilise a visually pleasing design, so consideration of your users interactions with the site at this stage can be useful.

Before you begin, it is worth doing a little research into the kinds of features that existing, real-world, web-apps in your domain support. Don't get too ambitious though as you have a limited amount of time and are looking at what teams of professional developers have achieved. As a rule it is worth having a simple, core plan that you can supplement with more elaborate functionality in the situation that you make better progress than you expected, but allowing you to fall-back to something more achievable if you experience challenges. Where applicable you should describe how you plan to exploit the web technologies, such as HTTP and other protocols, that we've studied. You should also consider and report upon any data you intend to store, considering the sensitivity of that data, and how it may affect your approach to security and privacy for the wider site.

Your deliverable for this part of the assignment is a single, short, and well written PDF report, that is limited to no more than 8 pages of text (i.e. excluding images) and which includes the following:

1. A description of the site that you are planning with enough background context for your reader to understand what you are trying to do.

2. A summary of any background research you have done with a description of how this has contributed to your project (perhaps through better understanding potential solutions).

3. A list of features and some discussion of why each feature is included.

4. A navigation tree and some discussion of how you plan to organise your site.

5. A sketch of an initial user interface for your system and some commentary on the motivation for your design, i.e. how does your design address the features you've listed. It is at this stage that you should considering which aspects of your design are common and might be translated into a template hierarchy for easier reuse & reduction of repetition. NB. Any designs can be hand-drawn and scanned/photographed for inclusion in your report.

6. As appropriate: any additional sections that you deem fit to describe your project. For example, if your site will provide a data API then you should include an initial description of the API routes and data formats. Similarly if you intend to save data on the server, perhaps in a database, then some description of the kinds of data that you intend to store, and how you will store it, should be reported on.

7. A references section listing any work, that is not your own, which you have cited elsewhere in the report (see the "notes" section below for additional information about referencing). If you do not cite anybody else's work then this section can be omitted.

8. (Optional) You may include additional appendices in your report if you have too much material to fit within the 8 page limit. Appendices are not counted towards the 8 page limit but must be referenced from within the main report. There can be useful for screenshots, code examples, wireframes, etc.

Your report must be uploaded to Moodle by the deadline of 3**PM on Friday 20th October, 2023**.

Part #2

You must implement your planned project from part #1 using Python and Python-Flask. You may use any other technologies as appropriate in addition within your virtual server. If you have requirements that cannot be satisfied using just your VM them talk to the module leader about your plans in advance. You may use additional Python modules, Flask plugins, Javascript Libraries, or miscellaneous software in the production of your project. However you must be able to justify why you used additional software rather than writing a solution yourself. Clear documentation must be supplied in an appendix of your report for anything you use that is not within our default learning environment. This documentation must clearly explain the process of installing and using your choice of additional software. There are four deliverables for this part of the assignment which comprise the following:

1. A deployment of your site either to your personal VM or to another location that supports Flask hosting (such as Heroku's free tier). Your site must be publicly accessible to a Web browser. You **must** check that your site is running correctly in advance of the deadline.

2. The complete source code of your site committed to a Git repository. Your repository must be archived into a zip file for upload to Moodle (although a non-standard usage of Git this is more efficient for the marking and moderation process).

3. A short (limited to no more than 4 pages) PDF report that covers the following:

    3.1. Gives the URL where your site can be found.

    3.2. Explains the differences between your initial plan as outlined in part #1 and your final implementation.

    3.3. Describes any features that you would add or improve to enhance your project, given more time.

    3.4. Reflects upon the challenges you faced and achievements you made during this assignment.

4. A short, approximately 3-5 minute, screencast with voice-over of you demonstrating the features of your web-app (see "Demonstration Screencast" section below for more details).

Your zipped Git repository, report, and screencast must all be uploaded to Moodle by the deadline. The deadline for part #2 of the assignment is **3PM on Friday 2st December, 2023.**

Demonstration Screencast

You will produce short, approximately 3-5 minutes, screencast recording of your running software. Note that your screencast should not be more than 150MB in size. There will be variance in the size of the resulting video, but if your video is >150MB then you must investigate the encoding of your video and select something more efficient.

A screencast is simply a video recording of your screen as you interact with your web-app. This should include a voice-over of you describing what you are doing and why your app works the way it does. Screencasts are straightforward to record on most major platforms, Windows 10 includes the "Game DVR" feature and Mac OS enables screen-recording via the QuickTime Player application. A free, cross-platform solution, with more functionality is the Open Broadcaster Software (OBS Studio[1]). Instructions for producing your screencast will be posted to Moodle to support you in producing this.

Feedback

Feedback is very important to your learning process. During the duration of this module you will reveive feedback using a variety of modes and at various times. The most common type of feedback that you will get is verbal feedback during timetabled contact time. The aim of this is to help you to improve your practical skills and to help you to think critically about your progress. This is why attendance at lab sessions is important as these provide a great opportunity to discuss your ideas with teaching staff in a less pressurised context. You will also get some written feedback after your hand-ins along with your grade. Generally this is more brief than the verbal feedback you will have already received, and is primarily aimed at helping you to see what you did that helped you to achieve your grade. There may also be suggestions for improving things that you should consider in the context of your work. Under ideal circumstances you will be emailed written feedback within three working weeks of the submission deadline. However you will also receive verbal feedback during contact time throughout the trimester. Verbal feedback is as important, sometimes more-so, than written feedback, and should neither be discounted nor disregarded.

[1] https://obsproject.com/

Grade Guide

The marking schemes are devised so as to reward those who go beyond the core taught material by integrating their own self-directed learning and discoveries. As a general rule, the more functionality, the better the mark, however your functionality should be consistent with a cohesive overall design, professional presentation, and pleasing user experience. This means that your project will not be marked solely based upon technical prowess, nor solely upon interface design, but upon a synthesis of all aspects of design and implementation that are brought together in the final realisation of the project.

Note that, because this is a project, rather than a mere test of your capabilities, you have significant leeway for what you include in your final submission (beyond the specified requirements). The following grade guide gives you a coarse description of how to interpret your overall percentage grade (for the entire module). When interpreting your feedback, please don't think in terms of "where did I lose marks?" or "did I get marked down for that?" but instead you should consider that you start off with nothing and incrementally approach perfection. A better question to ask yourself is "what could I have done to improve my work?" or "how could I refine what I have done to make it even better?". Thinking critically about our achievements ourselves is how we develop a professional sense of the quality of our work, rather that relying on external validation of whether it is good enough. Projects like this are designed to encourage that kind of reflection and professional development.

The following scale will give you some idea of the overall grade bands:

0-40% There are a number of ways to arrive at a mark in this band, but generally you will either have failed to create a working practical implementation to a minimal standard in either part of the project, or have failed to submit a report that is written to an acceptable standard in either part of the project, or some combination of both.

40-49% Work in this grade band is considered to be up to an overall, acceptable, but minimal standard and constitutes a bare pass of the module. Practical implementations will cover at least the core requirements in each part and reports will be written to a minimally acceptable standard of content and presentation.

50-59% Work in this grade band is work that has achieved a good standard. This means that there is evidence that you are applying some depth of knowledge to the goals that you set out to achieve and are developing ambition in relation to what you build.

60-69% To achieve a mark in this band you will have produced work that is to a very good standard. As a rule, **most students will achieve in the mid to upper end of this grade band**. This indicates that you are developing significant depth in your understanding of the domain as a whole, as well as significant technical understanding of underlying technologies. You will also be developing reliable critical faculties that enable you to realistically appreciate what you have achieved and how it can be improved.

70-100% A submission in this mark band represents excellent work. Above 80% you should consider your work to be exceptional, and above 90% your work is exemplary and tending towards perfection. To achieve a mark above 70% you will have integrated and 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. Your reports will explain your thinking, in relation to both design and implementation, with clarity . To achieve above 80% then you should be aiming to exceed the taught content of the module and introduce ideas and findings from your self-directed learning.

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. Once you believe that you've secured some work at the pass level then you should iterate over your working solution to improve those features and try not to break things.

Notes

Try to avoid accusations of plagiarism:

- Do not copy and paste text from the Internet.
- If you use code from the internet, please acknowledge it in comments in the code and also in the report.
- After reading reference material, lay it down where you cannot see it and write your own interpretation in your own words.
- Format your references & citation using the IEEE which is fairly widespread & standard amongst computer science publications.

Indicative Marking Scheme (Part #1)

Topic	Criteria	Marks
Core Criteria	Description of project & context	**/15**
	Summary of background research & exploration of how this relates to your project	**/15**
	* List of features * Navigation tree * Initial UI sketch & commentary * Implementation Plan * Consideration of underlying protocols * Consideration of security & privacy	**/40**
Above & Beyond	**Above & beyond (indicative):** * Additional report sections to address objectives specific to your project * Appendices as appropriate * Quality of critical reflection * Quality of presentation of report * Quality of writing within report	**/30**
	Total	**/100**

Indicative Marking Scheme (Part #2)

Topic	Criteria	Marks
Code	Working implementation of server component using Python & Python-Flask	**/20**
	Development & integration of appropriate user interface & associated user experience	**/20**
	Above & Beyond (indicative): * Additional features specific to your project * Use of UI template language instead of raw HTML) * Use of client-side storage to survive browser restart (offline mode) * Use of storage on server-side to survive app restart (file, db, cloud) * Creative use of additional browser APIs or server-side libraries * Design & Implementation of creative extensions & additional features beyond the topics studied in class * Evidence of professional skills: Use of Git, quality of code, project organisation	**/20**
Report & Screencast	* Explanation of differences between initial plan and final submission * Description of features for enhancement * Reflection upon challenges faced * Reflection upon achievement made * Screencast with voiceover	**/30**
	Above & beyond (indicative): * Quality of writing beyond minimum acceptability * Quality of critical reflection on changes between plan & implementation * Depth of reflection & plans for addressing un-met challenges	**/10**
	Total	**/100**

1. Module number	SET09103
2. Module title	Advanced Web Technologies
3. Module leader	Dr. Simon Wells
4. Tutor with responsibility for this Assessment	Your first point of contact is the module leader
5. Assessment	Please see attached descriptor for details
6. Weighting	**Part #1 (40%) | Part #2 (60%)**
7. Size and/or time limits for assessment	Please see attached for details.
8. Deadline of submission *Your attention is drawn to the penalties for late submission*	*** Part#1 is due at 3:00PM on Friday 20th October 2023.** *** Part #2 is due at 3:00PM on Friday 1st December 2023.**
9. Arrangements for submission	Please see attached descriptor for details
10. Assessment Regulations	This assessment is subject to the University Regulations**.**
11. Requirements for the assessment	Please see attached descriptor for details
12. Special instructions	None
13. Return of work	We generally aim to return work with an uncomfirmed grade and summative feedback within three **working** weeks.

14. Assessment criteria	Please see attached. With reference to the module descriptor, part #1 of this assessment covers aspect of learning outcomes **1, 2, & 4** and part #2 covers aspects of learning outcomes **1, 2, 3, & 4**. **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.