University of Westminster
School of Computer Science and Engineering

Coursework 1 (2022/23)

Module leader	Sriyan Fernando
Unit	Coursework 1
Weighting:	40%
Qualifying mark	30%
Description	Design of web application and essay
Learning Outcomes Covered in this Assignment:	L01, L03
Handed Out:	03/10/2022
Due Date	**1pm Thursday 03 rd November 2022**
Expected deliverables	Electronic document
Method of Submission:	online via Blackboard
Type of Feedback and Due Date:	Feedback will be given during viva in tutorials following deadline submission and via Blackboard rubric. **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Coursework Description

Introduction

In this coursework, you will submit a report in which you will design a simple web application and critically discuss the technologies needed to implement it.

1. Choose what application to develop

You should choose which web application you will design from the following list:

1. A simple social networking application based on a shared interest: either photography, cartoons, books or holiday destinations
2. A technical question and answer website, where people can pose questions on technical issues or problems, and other people can help answer them
3. An investment forum where people can post about their favourite stock and shares, and predict whether a stock will go up or down
4. A quiz website that allows users to create quizzes and to take other people's quizzes on various topics

Having decided what application you will design, you should write a short introductory paragraph setting out why you chose this application and your idea about the main aim of the application.

2. Decide requirements

The exact requirements for the application you choose are up to you. However, I suggest that your application should reasonably be expected to include:

a. account management – people should be able to register and create accounts, and login and logout
b. creating content in some way (for example, in the social network application, a user creates a post and others comment, or someone asks a question and another person answers, or someone upvotes an answer)
c. finding (in different ways, such as keywords or tags) and browsing content

(Real-world sites also include management or administrations interfaces - for example to ban users or delete content - but you do not need to do this for this coursework to keep it manageable).

Your requirements can be a short set of bullet points or a table, but should be grouped into those that are essential (those without which the application will not work), desirable (those that, while not essential, still add real value), and luxury (nice to have if there's any time to add them).

If there is anything non-obvious about a requirement, you should add an explanation to the bullet-point. Note that since this is a single coursework, the requirements should not resemble a full project! If you are unsure about whether you are doing too little or too much, then please ask. As a rule of thumb, your requirements should cover the basic points (a) – (c) above, but not go beyond that in any non-trivial way. Remember, you will have to implement your requirements in the next coursework, so don't set yourself too much to do.

3. Create an outline design

You should create an outline design for your application using (i) class diagrams, (ii) a database schema, and (iii) screen mockups. This design should be sufficient to allow a reader to know

a. What classes will need to be written
b. What database tables will be needed and what information they will contain (this includes field names and types)
c. What, roughly, the pages will look like and what information they will contain, as well as how, roughly, how one page is linked to another

The class diagrams should show controllers, models and views as classes, together with (a) the relationships between them, and (b) the main methods or functions in the controller and model classes.

The database schema can be represented diagrammatically (as shown on this page: https://database.guide/what-is-a-database-schema/), with relationships between tables where you think they need to exist, or you can use UML notation in the form of class diagrams (as you may have learnt to do on 5COSC020W Database Systems).

Screen mockups should resemble the wireframes on this page: https://www.smartdraw.com/website-wireframe/. This page also discusses sitemaps, which you will also find useful. Your wireframes should include some example data, rather than be entirely generic.

Again, if anything is non-obvious to a reader, you should briefly explain it.

4. Discuss technologies

First, read the following articles.

1. Crawford, T. & Hussain, T. 2017, *A Comparison of Server Side Scripting Technologies*, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Athens. (https://www.proquest.com/docview/2139472343?pq-origsite=gscholar&fromopenview=true)

2. K. Lei, Y. Ma and Z. Tan, "Performance Comparison and Evaluation of Web Development Technologies in PHP, Python, and Node.js," *2014 IEEE 17th International Conference on Computational Science and Engineering*, 2014, pp. 661-668, doi: 10.1109/CSE.2014.142. (https://ieeexplore.ieee.org/abstract/document/7023652)

3. Purer, Klaus. "PHP vs. Python vs. Ruby–The web scripting language shootout." *Vienna University of Technology* (2009). (http://online.verypdf.com/u/8925/api/20140616-010712-3660462726.pdf)

4. 8 Reasons Why PHP Is Still So Important for Web Development (https://www.jobsity.com/blog/8-reasons-why-php-is-still-so-important-for-web-development)

5. Is PHP Worthy of Developers' Hate (https://www.altamira.ai/blog/is-php-worthy-of-developers-hate)

Note (i) these sources are all trying to convince you of their argument – ask yourself if you agree, (ii) some of the information might be a little old but might still be useful.

Second, write a short essay of around 800-1000 words with the title "What Makes a Programming Language Suitable For **Server-Side** Web Development? Is PHP a good language for server-side web programming?"

Your essay should answer the questions in the title, referencing the articles above, in an original way. By "original", I mean in your own words, using the references to back up or support the points you make yourself. If the authors of the articles make points you disagree with, you should make your disagreement clear and support it with an argument. Note that the first question is not about any particular language such as PHP, but what makes ANY language good for web development. You can then apply this to PHP to argue whether or not it is good for this purpose. Note that just because we use PHP on this module, this does not mean you should decide it is good for server-side web development.

To be submitted

A single document in Word or PDF containing:
1. Your choice of web application with discussion (including scope)
2. Requirements (as list or table)
3. Outline Design (as described above)
4. Essay
5. References

Indicative Mark Scheme

Marks out of 100 will be allocated between the different components as follows:

1. Choice of web application - 5 marks

2. Requirements - 15 marks
3. Outline design - 40 marks
4. Essay - 40 marks

An excellent submission (between 70-100%) will clearly and precisely explain the choice and purpose of the application, clearly describe a focused set of requirements, with categorization, that are linked to the stated aim of the application, and present a design in terms of class diagrams, database schema and screen mockups that clearly supports the requirements with no omissions nor aspects that are unrelated to the aim/requirements. Accompanying discussion will be provide useful commentary. The essay will feature an original argument (in the sense of being the student's own view), be clear, focused and organized, with good grammar and structure. The conclusion(s) will be well supported by strong arguments and good use of references. Higher marks will be awarded on the basis of quality (for example, in class modelling, and/or novel arguments. *Note that simply increasing the amount of work you do (by adding more requirements) will not, in itself, increase your mark.*

A good submission (60-69%) will explain why the application has been chosen, perhaps in slightly generic terms, as well as a description of the aim. Requirements will be related to the aim, but may omit some important features. Design will cover all aspects (classes, database, mockups), but may omit some aspects. Discussion will be good but some features requiring explanation may not be well explained or may be omitted. The essay will feature an argument in the student's own words, but may be generic in parts, or conclusions may not be well supported by argument. Some use of unsubstantiated assertions (for example,""PHP is slower than other languages").

A fair submission (40-59%) will address the components of this coursework specification and will follow the same guidelines as a good submission. However, it will rely more on generic content, use of language that features poor use of grammar, and may be missing components or components will be sketchy or lacking necessary features.

A failed submission (less than 40%) will not address all components of this specification. Elements that are present will lack necessary features. The essay will be generic, lack a coherent argument and feature discussion that is unrelated to the title. Overall, rammar, structure and organization will be poor.

Finally

All work must be done on an individual basis - group work is NOT sanctioned.

If you need any clarification about any of these issues, please contact me by email (at sriyan.f@iit.ac.lk) or in the tutorials or make an appointment to see me in my office.

END DOCUMENT