

The Department of Computer Science

**CIS2169**

**Web Application Development**

Level 5

Coursework

2019/2020

### Module Leader: Dr. Hari Mohan Pandey

🕿 01695 65 6411

**Email: Pandeyh@edgehill.ac.uk**

## *Administrators:*

### 🕿 01695 65 7603

|  |
| --- |
| Coursework one – Portfolio (LO1, LO2) |

Weighting: 20%

Deadline for final documentation: For all hand-in dates and times please see the appropriate Blackboard dropbox. (Week wise submission 9:00 am)

Learning Outcomes Assessed:

LO1 Demonstrate a critical understanding of the fundamental models for structuring web applications.

LO2 Demonstrate the ability to work on a shared code-base

**1.1 INTRODUCTION**

This assignment is a major part of the formal assessment for CIS2169 and is, therefore, compulsory. You will be required to work individually, so time management is of the essence. Read this document carefully and make sure that you are clear about what you have to do, and what you have to hand in, before you attempt the assignment.

The aim of this assessment is to build up a portfolio of work that will demonstrate the skills and technologies that you have learned. Read this document carefully and make sure that you are clear about what you have to do, and what you have to hand in, before you attempt the assignment.

**1.2 TASK**

The aim of this assessment is to give you the opportunity to construct a portfolio, which will demonstrate your skills in a number of areas of web application development. During the course of the year, you will be given a total of six tasks (five questions per task) to complete and submit individually, which will showcase the skills you have learnt in this module.

Each portfolio task will be posted on VLE each week, so that you can read through the task before the lecture and lab sessions. By doing this, you will be able to bring any questions about the task to the sessions and ask your tutor for clarification. Each task will be discussed in detail in the lab to ensure you understand what you have to do and what the assessment criteria is. You should make use of this opportunity in the sessions to speak to your tutor about the assignment tasks!

1.3 CW1 ASSESSMENT CRITERIA

* Number of individual tasks and questions in those tasks that have been seriously attempted and answered correctly.

1.4 GUIDELINES FOR SUBMISSIONS

The following table shows when each task should be submitted:

|  |  |  |
| --- | --- | --- |
|  | Week handed out | Week due |
| Task 1 | Week 1 | Week 2 |
| Task 2 | Week 3 | Week 4 |
| Task 3 | Week 4 | Week 5 |
| Task 4 | Week 5 | Week 6 |
| Task 5 | Week 6 | Week 7 |
| Task 6 | Week 7 | Week 8 |

The solutions to each task must be submitted to the appropriate dropbox on the VLE by the deadlines given on the appropriate dropbox on BB. Details on how each of the tasks should be submitted will be included in each of the task descriptions.

If you have any questions/queries, please contact your tutor via email with the following as the subject line:

CIS2169: CW1 Question

1.5 POINTS TO CONSIDER

It is important that you attempt all of the tasks and all of their questions, as doing the tasks provides important practice for the work in coursework 2. If you attempt all of the tasks, then you will find coursework 2 significantly easier.

|  |
| --- |
| Coursework Two – Web Application Development (LO2, LO3) |

Weighting: 80%

Deadline for submission of final application: For all hand-in dates and times please see the appropriate Blackboard dropbox (Submission deadline week 12 middy 12:00 noon).

Learning Outcomes Assessed:

LO2 Demonstrate the ability to work on a shared code-base

LO3 Demonstrate an understanding of the processes and the ability to deploy an application to a server

|  |
| --- |
| Introduction |

This assignment is a major part of the formal assessment for CIS2169 and is, therefore, compulsory. You will be required to work individually, so time management is of the essence. Read this document carefully and make sure that you are clear about what you have to do, and what you have to hand in, before you attempt the assignment.

The aim of this assessment is to give you the opportunity to construct a small application using programming languages for the web. You must select the appropriate tools to use in order to achieve this, and communicate your ideas clearly and concisely.

|  |
| --- |
| Task |

The University runs a web-based academic management and tracking application, which in its current form tracks all academics research activities. It is your task to extend this application so that it also tracks teaching-related activities. This extension must support the following information:

The core concept for teaching is the module. Modules have an identification code, name, number of hours, learning outcomes, and number of credits. Each module has one or more assessments assigned to it. Each assessment has a title, number, learning outcomes it covers, volume, and weighting. The weightings of all assessments of a module must sum to 100%.

In each academic year a module is led by exactly one academic. In each academic year a module is time-tabled for one or more time-slots. Some modules have one slot each week, some have multiple slots each week, and some have irregular slots throughout the semester or year. Each slot is taught by one academic. Regular slots are frequently taught by the same academic each time, but there are also modules with regular slots that are taught by a different academic each time. Each time-slot is also assigned a room. As with academics, regular slots are frequently taught in the same room, but can also be in different rooms. In each academic year each assessment is given one submission date.

Modules are organised under degree programmes. Each module belongs to one or more degree programmes. A degree programme has an identification code, name, and learning outcomes. A degree programme also has a number of exit awards that define the degree that the student is given if they complete a certain number of module credits. A degree programme is led by exactly one academic.

In addition to supporting the concepts listed above, your design must also be able to support the following user stories:

* An academic creates a new module that is added to one or more degree programme(s).
* An academic creates a new degree programme and adds existing modules or new modules to it.
* Academics schedule the dates of the assessments for modules that they are leading in the current academic year.
* Head-of-department schedules the module time-slots for an academic year.

|  |
| --- |
| Required Content |

You must design the extensions to the application, build them, show the implementation of interface, write deployment plan and demonstrate the plan being actioned (use the repository: "**https://github.com/profharimohanpandey/CW2**" and modify it to demonstrate features. This repository is given just to provide the information about how to start the CW2). Your design must be submitted in the form of a report, which must contain the following sections:

1. Modelling techniques - Description of the modelling techniques used in your design and how they work together.
2. User modelling - Use-case models for the user stories listed above including explanatory text for each use-case.
3. Structure modelling - Class diagram modelling the problem space described above and Explanatory text for the class diagram.
4. Process modelling - Activity diagrams showing the application flow for the user stories and problem space described above and explanatory text for each activity diagram.
5. Design, Develop, deploy - Design and develop interfaces for one of your extension features (JavaScript, Html, JSON and CSS); Write a deployment plan and demonstrate the plan being actioned.

|  |
| --- |
| Initial Planning |

Before you start work on your application, it is important that you focus on the design aspect first you need to think about a few things before you start building something! To help you with this, consider the following design process/workflow:

* Coursework Assigned - Document/clarify your understanding of the assignment, ask questions for clarification and take notes.
* Design - design the system using the appropriate UML diagrams and ensure you know how each builds/links to the others.
* Implementation - Based on the UML Designs create your application for the extension feature pages (you can use Java-script, HTML, JSON and CSS to implement required features).
* Testing - Test the application meets the required standards.
* Deploy - Plan, document your deployment plan. Then set up and deploy your mockup according to your plan. (using netlify.com as a free service may be of assistance).
* Document - write report up using your detailed notes from throughout the process, ensure you use the CW brief and marking scheme to .

|  |
| --- |
| What will be Assessed |

* Your report

|  |
| --- |
| CW2 Assessment Criteria |

* Design Quality.
* Design & Report Coverage.
* Readability.
* Referencing.
* Development and deployment planning and practice

|  |
| --- |
| Guidelines For Submission |

When completed, you will need to upload a copy of your report to the relevant drop box on Blackboard.

The report file must be named:

LastName\_FirstInitial\_StudentNumber\_CW2

e.g. Pandey\_H\_12345678\_CW2

If you have any questions/queries, please contact your tutor via email with the following as the subject line:

**CIS2169: CW2 Question**

|  |
| --- |
| Points to Consider |

It is important that you not only design the extension, but also clearly document the process. The documentation represents a major aspect of the overall mark for the work and should not be left to the end, but created as you develop your design.

|  |
| --- |
| Late Submission |

If you have missed a submission deadline for a written (individual) assessment, at the scheduled first sitting, the University will accept Late Submissions where the submission is received within five working days of the original deadline (aside from where PSRB requirements preclude this). [These] late Submissions will be subject to the normal capped mark for reassessment. Work handed in late [after this five-day period] without approval through a Deferral, Extension or authorised late submission (Academic Regulation H4.4) will be marked at zero. (Late submission deadline is middy 12:00 noon)

|  |
| --- |
| Extension |

If you are unable to submit to the stated deadline, you should follow the extension application guidelines below. Only in extreme cases will extensions be granted for coursework.

**Student guidelines for applying for an extension**

* All applications for extensions should be submitted to the Department Administrator. (Forms are available on Blackboard / Organisation area)
* Any application for an extension of a deadline should normally be made at least twenty-four hours before the due deadline for the assessment. In exceptional circumstances outside of this time scale students are permitted to submit an extension form without supporting evidence. The supporting evidence must be submitted within 7 working days or the extension request will not be approved.
* All applications for an extension should be accompanied by verified independent documentary evidence.
* Extensions may be granted for a period that extends up to **two weeks before** a Module Assessment Board. If a period longer than this is required the student should be advised to apply for consideration under exceptional mitigating circumstances.
* The timing of student feedback against published deadlines will be affected if an extension is granted.

|  |
| --- |
| Malpractice |

When researching and writing-up your coursework you must ensure that you fully reference any elements within it that are not written or conceived by yourself. In doing so you will respect the intellectual property of other writers, and acknowledge the importance of their work in your area of research. If you fail to attribute a source for the work of other people and present it as your own, you are stealing their work. Academic malpractice of this kind (’plagiarism’) is seen as the most serious offence in academia, and if you are found guilty of such activities it may lead to the termination of your course of study or the non-conferral of your degree.

As such, you should ensure that you are familiar with the appropriate ways in which to avoid academic malpractice, using the appropriate referencing system. The department of Computer science, like many departments at Edge Hill University, use the Harvard referencing system (2014).

**Avoiding malpractice when using electronic research mechanisms**

 You will be encouraged to use a range of electronic and online resources such as electronic journals, online database and the Internet, which, collectively are a ubiquitous resource. However, students need to be cautious when using these research tools, as whilst they streamline the process of research/information gathering, they simultaneously generate a depth of materials that need to be filtered and ***read***.

In addition, caution is needed as use of these electronic research tools often generate materials for your scope of study which are not relevant. Consequently, you should ensure that you search within the relevant parameters of investigation as framed by the recommended reading and research materials indicated in the module handbook of each module.

Remember: you ***read*** for a degree – you cannot demonstrate you understand academic theory in an appropriate manner if you have not read and **understood** ***relevant*** resources.

If you have any concerns please check with your tutor or Learning Services

<https://www.edgehill.ac.uk/ls/uni-skills/?tab=referencing>

**Remember, it is as easy for the department to uncover any information taken from the internet and passed off as your own work, as it is for you to find it!**

|  |
| --- |
| Marking Criteria |

Coursework1

|  |  |  |
| --- | --- | --- |
| Mark | Questions seriously attempted (50%) | Questions correctly answered (50%) |
| Fail (0 - 29) | Less than 30% of the task questions seriously attempted | Less than 30% of the task questions correctly answered. |
| Narrow Fail (30-39) | Between 30% and 39% of the task questions seriously attempted | Between 30% and 39% of the task questions correctly answered. |
| Pass (40-49) | Between 40% and 49% of the task questions seriously attempted | Between 40% and 49% of the task questions correctly answered. |
| Good (50 – 59) | Between 50% and 59% of the task questions seriously attempted | Between 50% and 59% of the task questions correctly answered. |
| Very Good (60 – 69) | Between 60% and 69% of the task questions seriously attempted | Between 60% and 69% of the task questions correctly answered. |
| Excellent (70 – 84) | Between 70% and 84% of the task questions seriously attempted | Between 70% and 84% of the task questions correctly answered. |
| Outstanding (85 -100) | Between 85% and 100% of the task questions seriously attempted | Between 85% and 100% of the task questions correctly answered |

Coursework2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mark |  |  | Modelling technique | | | | | | | User Modelling (15%) | | Structure | | | modelling | | Processing modelling | | | | | Development & Deploy­ | | | |  |
|  |  |  | (20%) | |  |  |  |  |  |  |  | (15%) | |  |  |  | (20%) | |  |  |  | ment plan (30%) | |  |  |  |
| Fail (0-29) | |  |  |  |  |  |  |  |  | Nothing submitted, rudimentary content only, or substantially plagiarised. | | | | | | | | | | | | |  |  |  |  |
| Narrow | | Fail | Basic | | documentation | | | | | Design is not readable | | Design is not readable | | | | | Design is not readable | | | | | Very Basic Interface design | | | |  |
| (30-39) |  |  | exists. | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | no deployment plan | |  |  |  |
| Pass (40-49) | | | Overview | | |  | of | mod­ | | Occasional | comments | Occasional comments | | | | | Occasional comments | | | | | Very Basic Interface design | | | |  |
|  |  |  | elling | |  |  | techniques | | | have been added | | have been added | | | |  | have been added | | | |  | with deployment plan | |  |  |  |
|  |  |  | given, | | without | | |  | in­ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | cluding explanatory | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | text |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Good (50-59) | | | Overview | | |  | of | mod­ | | Some elements of deign | | Some | | elements | | of | Some | | elements | | of | Good interface design with | | | |  |
|  |  |  | elling | |  |  | techniques | | | are explained but not in | | deign | | are | explained | | deign | | are | explained | | basic deployment plan | |  |  |  |
|  |  |  | given, | | including | | |  | very | detail. |  | but not in detail. | | | |  | but not in detail. | | | |  |  |  |  |  |  |
|  |  |  | brief |  | discussion | | | | of |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | explanatory text. | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Very Good (60­ | | | Overview | | |  | of | mod­ | | Most elements of design | | Most elements of de­ | | | | | Most elements of de­ | | | | | Good interface design with | | | |  |
| 69) |  |  | elling | |  |  | techniques | | | are explained | in detail | sign | are explained | | | in | sign | are explained | | | in | detailed | deployment | plan, | |  |
|  |  |  | including | | |  | role |  | of | but difficult to follow | | detail | | but | difficult | to | detail | | but difficult | | to | creating | repository | and | |  |
|  |  |  | modelling | | |  | techniques | | |  |  | follow | |  |  |  | follow | |  |  |  | adding file into repository | | | |  |
|  |  |  | over | explanatory | | | | | text |  |  |  |  |  |  |  |  |  |  |  |  | with commit | |  |  |  |
|  |  |  | is detailed. | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Excellent | |  | A very good discus- | | | | | | | Models design easy to | | Models design easy to | | | | | Models design easy to | | | | | Good interface design with | | | |  |
| (70-84) |  |  | sion |  | of |  | modelling | | | follow and explained in | | follow | | and | explained | | follow | | and | explained | | detailed | deployment | plan, | |  |
|  |  |  | techniques | | |  | and |  | its | detail |  | in detail | | |  |  | in detail | | |  |  | creating | repository | and | |  |
|  |  |  | association | | |  |  |  | with |  |  |  |  |  |  |  |  |  |  |  |  | adding file into repository | | | |  |
|  |  |  | explanatory | | | | test | | is |  |  |  |  |  |  |  |  |  |  |  |  | with commit and pushing | | | |  |
|  |  |  | detailed. | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | changes |  |  |  |  |
| Outstanding | | | Excellent | | | description | | | | Model design is easy to | | Model design is easy | | | | | Model design is easy | | | | | An excellent interface de­ | | | |  |
| (85-100) | |  | of | modelling | | | | tech­ | | follow, clearly explained | | to | follow, | | clearly | | to | follow, | | clearly | | sign and deployment plan. | | | |  |
|  |  |  | nique | | used | | to design | | | each term used, discus­ | | explained | | | each term | | explained each term | | | | |  |  |  |  |  |
|  |  |  | and | how | | they | | work | | sion on how the mod­ | | used, | | discussion | | on | used, | | discussion | | on |  |  |  |  |  |
|  |  |  | together. | | |  |  |  |  | elling is associated. | | how the modelling is | | | | | how the modelling is | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | associated. | | |  |  | associated. | | |  |  |  |  |  |  |  |

Each section is assessed independently and the overall mark is the weighted average of the section marks.