COMP1841 (2023/24)	**Web Programming 1**	**Contribution: 100% of course**
Module Leader: **Matt Prichard**	**PHP/MySQL** **CRUD system**	**Deadline Date: TBC**

Learning Outcomes:
1 Develop a web application with relational database connectivity using a variety of technologies, specifically HTML5, PHP PDO and MYSQL
2 Discuss issues surrounding Internet and intranet technologies, such as browser dependency, accessibility, legal and security concerns.
3 Critically analyse and evaluate Internet related business problems.

Developing a functioning CRUD data driven website

This is an individual piece of coursework. It is to be implemented in PHP and MySQL on the CMS student web server (i.e. on your I Drive) or on localhost with XAMPP on your own machine. You must use PHP PDO programming and show an understanding of relational database design including referential integrity.

Do not use MySQLi to develop this system. Marks will be lost for ignoring this !

General Overview

During the lectures and labs, you will be given the basic building blocks of a web-based PHP/MySQL CRUD system based on an online joke site. It will be **up to you** to apply these skills and techniques to a different but similar case study.

You are required to build a system where students can post questions amongst themselves to get help with their coursework. This is a prototype system so the functionality will be quite limited, but able to be built upon in future iterations.

Think of it as a **very simple** self-contained student stack overflow.

Below is an outline of functional requirements for the new system in a loose order of importance.

- A public facing web front end allowing students to view a list of questions or posts by other students.
- The facility for a student to post, edit and delete a question.
- An area to manage authors and module names
- Provide a contact form for students to send a message to a system administrator.

Detailed specification:

A public facing web site that:

- Displays a list of questions/posts.
- Allow a user to add, edit and delete a post.
- Ability to display an image/screenshot for each post (You will be shown methods to achieve this in the class - there are some differences to what you can do on our servers compared to your local host, or managed hosting that we will discuss).
- Facility to add edit and delete users (username, email address for example) and connect them to their posts in the database).
- Ability to assign a question/post to a module name from a pre-existing list.

- Ability to manage list of module names.
- Allow user to send an email message to the admin via a web form.

Additional information and considerations

You must ensure your site meets any **legal accessibility** requirements.

Where appropriate implement form validation though either JavaScript or modern HTML 5 elements.

It is expected that the database part of the system contains more than one table and demonstrates an understanding of relationships between tables. You are only required to enter enough records into the database to demonstrate functionality.

The front end needs to be clear to navigate and consistent across all pages.

Use of CSS frameworks such as Bootstrap is allowed.

Any other additional functionality you may add beyond this scope needs to be clearly identified in the report if you want credit for it.

Additional functionality discussed in the lectures can be included in order to achieve a first-class mark.

All sources of code not from the lectures must be fully referenced. Do not implement anything that you do not understand.

Deliverables

1. **A formal report (2000-3000 words) in PDF format containing:**

- Title page, including the **URL of the web site if it is on the I-Drive**
- Table of contents.
- Introduction
- How you developed your system including design of the pages, navigation structure, data diagrams
- Discussion of the technologies used and why
- Detailed consideration of any legal, social and ethical issues including personal data storage and an understanding of GDPR.
- Overview of your system with annotated screen shots showing sample pages to prove the functional requirements have been met.
- Evidence of a structured and thorough testing schedule including input validation / control.
- Evaluation, conclusion and future recommendations.
- Harvard Referencing

2. **A ZIP file of your entire site, including your database file.**

Assessment Criteria

Functioning CRUD web system	60%
Report including evidence of testing and evaluation	40%
Total	100%

Grading Criteria –

70% + (1st class)

A fully functioning professional solution has been developed in PHP and MySQL

A high quality and complete implementation of the required functionality with no code errors is present.

Additional features showing self-directed learning will also be present

Critical evaluation and specific conclusions are presented.

60-69% (2:1)

A functional Web solution has been developed in PHP and MySQL

All of the required functionality is present.

Good evaluation and conclusions are presented.

50-59% (2:2)

A Web solution has been developed in PHP and MySQL which includes some of the requirements

Most of the required functionality is present.

Some evaluation and conclusions are presented.

40-49% (3rd)

A Web solution has been developed in PHP and MySQL with large gaps in the technical requirements.

An attempt at the required functionality has been made but shows limited technical competence or understanding.

Very little, if any evaluation or conclusions are presented.

<40% (Incomplete)

Web site is incomplete or not meeting the requirements.

No PHP or MySQL functionality at all.

Very poor or no evaluation or conclusions are presented.

Marking scheme

Report (2000-3000 words)	**40**
How you developed your system including design of the pages, navigation structure and a data diagram. Include a discussion of the technologies used and why and also briefly demonstrate an understanding of web standards.	10
Detailed consideration of any legal, social and ethical issues and web accessibility. Discuss personal data storage and show an understanding of GDPR.	10
Overview of your system with annotated screen shots showing sample pages to prove the functional requirements have been met.	10
Evidence of structured testing.	10
Use of English, spelling, grammar, Harvard Referencing. 1 mark removed for each error up to a maximum of 10 marks	
Total	

Functioning CRUD web system	**60**
Marks will be awarded in line with the requirements and additional tasks listed above. This is a holistic process and will be discussed in the lectures.	
Total	