Assessment Brief

Module title	Web Application Development
Module code	XJCO2011
Assignment title	Assignment 2
Assignment type and description	Programming assignment You will design, plan, develop and deploy a functional web application which utilises a many-to-many relationship, authentication and logging, with a simple and accessible design.
Rationale	You will be combining a number of your development skills, designing objects and databases, and developing queries and algorithms which use the data you collect from your users in an interesting way. You will top this off by creating a simple front-end which complies with the WCAG requirements for websites.
Word limit and guidance	You should spent around 15-20 hours on this assignment.
Weighting	60%
Submission deadline	31 December 2025
Submission method	Gradescope
Feedback provision	You will receive a marked up rubric and some short comments on your code
Learning outcomes assessed	LO2 - Improve knowledge of Python in regard to programming skills and web development LO3 - Apply database knowledge and consider architecture in the web application. LO4 - Demonstrate practical elements such as designing, implementing, and testing a web application. LO5 - Evaluate design decisions and analyse the web application.
Module leader	Amy Brereton

1. Assignment guidance

You will be designing, developing and deploying a web application of your own choosing.

Your app needs to:

- Use a many-to-many relationship
- Have an authentication system
- Use an advanced feature (see section 2)

You can decide what you want your web app to **do** – the only limits are:

- Not a library system
- Not a budget tracker
- Not a to do list tracker

You should also ensure that the many-to-many relationship you implement has some sort of impact on the user experience- it should not just be token and there to get marks, use it in an interesting way to provide some benefit to the user.

You can design an e-commerce site if you wish, but there is **no requirement for you to implement a payment system**.

If you can't think of an idea, you could create:

- An e-commerce site with stock-tracking
- A basic social media site
- A film/music review site

Your website should have a professional layout which is easy to navigate and complies with the WCAG standards for accessibility. You can use open source CSS libraries such as Bootstrap, but should not rely on them for all of your styling.

You must develop your layout yourself using HTML, CSS and JavaScript if appropriate. You cannot use any software or services to create your layout for you, the code should be written by you.

All JavaScript and CSS should be in its own file(s) in the 'static' folder.

2. Assessment tasks

Database

You must create a minimum of 2 models, and these models should have a many-to-many relationship implemented between them.

This relationship should be used to **improve the user experience**- it shouldn't just be there to get marks, you need to think about why it exists and what it adds for the user.

For example, a **bad** many-to-many relationship would be:

You create ingredients, and add them to a recipe. If an ingredient hasn't already been added, you create it and it can be used by anyone in their recipes.

This is bad because:

- It makes the process of adding a recipe more difficult for users
- There is no actual use for the 'ingredient' entity
- There is a risk of misspellings in the ingredients created by users
- If a user edits or deletes an ingredient, it affects everyone else who is using it in their recipes

So this relationship is both pointless and actually detrimental to the website.

To make this a more **suitable choice**, you could:

- Pre-populate the ingredients database with a range of ingredients and not allow editing
- Create a feature which filters recipes by ingredient
- Add a feature which finds recipes which contain only ingredients the user has

This adds much more usability, and allows you to **demonstrate your database querying skills** – remember, you need to demonstrate some more complex behaviour!

In general, for this assignment you will probably find that you want to **pre-populate your database**- you can use datasets found online for this, and there are many for things such as films, books and music which you should be able to access for free. You should be able to use a simple Python script to do this.

Complex Behaviour

In this assignment, you need to demonstrate some 2nd year programming skills- basic queries aren't enough. You need to use some of your algorithm design skills to use the data you get from your users in a more interesting way!

There are a few options available to you:

- Adding some AJAX features – a good choice is a 'like' button.
    - Adding a significant amount of JavaScript or other scripting language – this must be **written by you** and not solely library code.
- Using some more complex Python logic to add helpful functions, for example:
    - Having a recommendation system which recommends things based on shared likes/reviews
    - Implementing a search or sort function from scratch

If you are unsure if your idea is complex enough, please **ask on the Teams channel** or **speak to me in the lab**.

Design, Layout & Accessibility

You also need to create a layout for your website, and style it appropriately. As with assignment 1, you should ensure that you consider users who are accessing the website on assistive technologies, and users with different disabilities. Your site should also be **responsive** and work on a range of devices.

All public-facing websites need to comply with the WCAG guidelines, so you should, as far as possible, try and ensure that your website is compliant.

Your design should be suitable for the purpose- marks for the design will be based on the suitability of your design for purpose, the level of effort you have put in to making the site look good, and the ease of navigating the website. Again, it is recommended to use a horizontal navbar which is consistent across all pages as this is the most accessible.

Deploying your Site

You must **deploy** your website, meaning publish it online.

It is recommended to use PythonAnywhere as this is free and simple. PythonAnywhere has a clear guide on how to do this: https://help.pythonanywhere.com/pages/Flask/
Or instructions are provided in the Module notes.

The deployment process can take some trial and error; you will need to change settings in your code and in the PythonAnywhere interface. If you are struggling, please post on the Teams group or speak to me in the lab.

Your deployed website needs to remain live for **at least 3 weeks** after the final deadline so that it can be marked. No special action is required from you, just don't deploy other code on the same account until 3 weeks are up!

Report/Video

Part of the assessment of this coursework will involve you presenting your work. You have a choice in how you do this.
You should create **one** of:

- A video presentation, or;
- A written report.

Both methods will involve conveying the same information:

- Explaining the purpose of the website
- Walking through a user's journey on the site (signing up, logging in, and using the site)
- Justifying the design choices made
- Explaining the many-to-many relationship and how it adds to the user experience
- Demonstrate your advanced feature(s) and explain the code behind it
- Explaining how accessibility has been considered
- Explaining how security has been considered
- Explain/demonstrate how you tested your website (accessibility, functionality, responsiveness)
- Analysing your web app in terms of:
    - What you did well
    - What you struggled with
    - Any changes you would like to make in future

The video should be no longer than 10 minutes, while the report should be 5-10 pages. In both, you should show your website 'in action' via screen recording or screenshots, as well as showing key parts of the code. Both options have the same weighting- it is up to your own personal preference which method you choose.

3. General guidance and study support

You should use the module website, accessed via Minerva, as your primary source for information.

If you do wish to use any other websites, books or sources, this can introduce some issues as the way that files are set up and names may be different from the module notes.

4. Assessment criteria and marking process

You will be assessed on:

- The functionality of your website
- Your design and layout
- How successfully you deployed the site
- Your report/video

You will receive feedback in the form of a marked up rubric (see section 8) and some comments on your code within 3 weeks of the final deadline.

5. Presentation and referencing

Your submission will include **one** of the following:

- A video presentation, or;
- A written report.

A template is provided for the report which covers the key areas you should discuss- there is no word limit, but around 5-10 pages is expected. You can use screenshots of your website and your code.

If you chose to present your work through a video, you must ensure that your video is high quality enough for any text used to be visible, and the audio is clear– it is highly recommended that you use screen recording software such as OBS Studio (https://obsproject.com/) which can record both your screen and your voice.

You should reference any copyrighted materials used in your website by comment in your code, or any additional sources used for your report using Harvard referencing.

The quality of written English will be assessed in this work. As a minimum, you must ensure:

- Paragraphs are used
- There are links between and within paragraphs although these may be ineffective at times
- There are (at least) attempts at referencing
- Word choice and grammar do not seriously undermine the meaning and comprehensibility of the argument
- Word choice and grammar are generally appropriate to an academic text

These are pass/ fail criteria. So irrespective of marks awarded elsewhere, if you do not meet these criteria you will fail overall.

6. Submission requirements

You will submit **three** things to Gradescope:

1. A zipped folder containing your code
2. A link to your deployed web application
3. Either a video or a report.

Your zipped folder of code should be ready-to-run, with the database set up and pre-populated with any required data. This will be used if, for any reason, your deployed site does not work. You **should not** upload your venv folder.

You need to ensure that your deployed site is live for **at least 3 weeks from the final deadline** to ensure that it is marked – you don't need to do anything special, just don't deploy anything else on the same account.

Make sure that your report/video follows the guidance set out above, and covers all the required sections.

7. Academic misconduct and plagiarism

Leeds students are part of an academic community that shares ideas and develops new ones.

You need to learn how to work with others, how to interpret and present other people's ideas, and how to produce your own independent academic work. It is essential that you can distinguish between other people's work and your own, and correctly acknowledge other people's work.

All students new to the University are expected to complete an online Academic Integrity tutorial and test, and all Leeds students should ensure that they are aware of the principles of Academic integrity.

When you submit work for assessment it is expected that it will meet the University's academic integrity standards.

If you do not understand what these standards are, or how they apply to your work, then please ask the module teaching staff for further guidance.

By submitting this assignment you are confirming that the work is a true expression of your own work and ideas and that you have given credit to others where their work has contributed to yours.

8. Assessment/ marking criteria grid

	Outstanding	Very Good	Good	Poor
Functionality 40	Website contains a many-to-many relationship, an advanced feature, and a number of useful features for the users.	Website contains a many-to-many relationship and has more advanced aspects included, with some functionality for the user.	Database has some relationships between entities, with an attempt at including advanced features.	Database contains no relationships and only basic functionality.
Layout, Design & Accessibility 20	Suitable and stylish design which is fit for purpose. Accessibility considered throughout.	Design is suitable, and there includes multiple accessibility features.	Basic design which has some consideration of accessibility.	No styling used, or no consideration of accessibility.
Deployment 10	Website deployed and working correctly.	Website deployed with some minor errors.	Website deployed with some major errors.	No attempt to deploy.
Report / Video 30	High quality explanation of work done, and a good critical analysis of work.	Good explanation of work done, some attempt at critical analysis of work.	Basic explanation of work done, with some surface level analysis.	Weak explanation with no analysis.