Assessment Brief

Module title	Web Application Development
Module code	COMP2011
Assignment title	Coursework 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 responsive 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 spend around 15 hours working on this assignment.
Weighting	60%
Submission deadline	Thursday 5th December at 2pm
Submission method	Gradescope
Feedback provision	You will receive a marked up rubric.
Learning outcomes assessed	LO2- Integrate principles of database management into web application architecture to ensure efficient data handling and retrieval. LO3- Demonstrate practical skills by designing, implementing, and rigorously testing web applications, covering all stages of development. LO4- Critically evaluate design decisions, and analyse web application architecture to assess effectiveness, scalability, user experience, and accessibility standards compliance, ensuring inclusivity for all users.
Module leader	Amy Beloe

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 a login system
- Use an advanced feature (see section 2)

You can decide what you want your web application to do – the only limits are:
- Not a library system
- Not a to do list tracker

You can design an e-commerce site if you wish, but there is no requirement for you to implement a payment system. If you want the challenge, you can try using payment APIs such as Stripe which have development mode, just make sure to use the demo card information which they provide and not a real payment card.

If you can't think of an idea, you could create:
- An e-commerce site with stock-tracking
- A basic social media 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 also personalise the styling yourself using some of your own CSS code.

You should create the layout of your site yourself and should not use any service to generate this code for you- for example, using drag and drop tools or what-you-see-is-what-you-get design software.

All JavaScript and CSS should be in its own file(s) in the 'static' folder. You can import any external libraries in your base HTML code.

Your python code should be split into multiple files, as you have seen in the module notes. It is not acceptable to put it all in one 'app.py' file.

2. Assessment tasks

Login system

You should create a login system that allows users to register and login. This will then enable the user to have a unique user experience if you add the additional functionality to facilitate this. For example, with an ecommerce site they could have a basket with items in from their previous visit, they could see their orders and the process of these, they would have a section with their account details to change their password. These elements would be unique to that user.

Database

You must create a minimum of 2 models, and there must be at least one many-to-many relationship included.

You should ensure that this relationship **impacts the functionality of your website in a useful way**. For example, an e-commerce site which uses database relationships to keep track of stock levels and shows anything which is out of stock due to being bought by other people as unavailable to the user.

If you're not sure what to do, look at some existing websites and think about the features they contain and how you could try and re-create them using your database.

If you wish to, you can use existing datasets to pre-populate databases – for example if you decide to make a Book or Music recommendation site, you can get datasets online which you could then put into your database using Python.

Complex Behaviour – advanced feature

You should include at least one complex feature - **note that this does not include the 'advanced' HTML from year 1**.

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 library code.
- Using some more complex Python logic to add helpful functions, for example:
  - Having a recommendation system which suggests things based on shared likes/reviews

If you are unsure if your idea is complex enough, please **speak to the module staff in the lab**.
As with your database, this should be **integrated into the user experience** – make this advanced feature relevant to the user.

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 including mobile phones – you will find this easier when using CSS libraries such as Bootstrap which are responsive by design.

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 Web application

You must **deploy** your application, 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 module staff in the lab.

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
- Explain/demonstrate how you tested your website (functionality, responsiveness, accessibility)
- 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.

A template is provided for the written report, and the same content can be followed for the video.

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 including references should be more than sufficient. 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 time.
- 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.

If you put a password on your deployed site, ensure this is submitted with the URL - **this is not your account password!**

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. Includes authentication, and the site is unique for the user.	Website contains a many-to-many relationship and has more advanced aspects included. Authentication used correctly, with some impact on the user experience.	Database has some relationships between entities, with an attempt at including advanced features. Authentication used but minimal impact on the user experience.	Database contains no relationships and only basic functionality. No authentication used.
Layout, Design & Accessibility 20	Suitable and stylish design which is fit for purpose. Accessibility considered throughout. Fully responsive design.	Design is suitable, and there includes multiple accessibility features. Responsive design, may have a minor issue.	Basic design which has some consideration of accessibility. Limited responsiveness.	No styling used, or no consideration of accessibility. Not responsive to different screen sizes, not attempted.
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. Thorough testing.	Good explanation of work done, some attempt at critical analysis of work. Fair amount of testing.	Basic explanation of work done, with some surface level analysis. Very limited testing.	Weak explanation with no analysis. No testing considered.