Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code:	CMT120
Module Title:	Fundamentals of Programming
Lecturers:	Federico Liberatore, Martin Chorley, Natasha Edwards
Assessment Title:	[redacted]Web Application Development’
Date Set:	17th July 2023
Submission date and Time:	7th August 2023 at 9:30AM
Return Date:	18th August 2023

This assignment is worth 60% of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1. If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;

2. If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:
`https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf`

Submission Instructions

All coursework should be submitted via upload to Learning Central.

Description	Type	Name
Cover sheet	`.pdf` file	`[Student number].pdf`
*Source file: **complete website code in one .zip file****	.zip	`[Student number].zip`

* More specific instructions on what you need to include in the .zip file are given in Section *’– Submission to Learning Central’* on p. 3.

The following should also be pushed to the **School’s GitLab server**:

- **Complete project folder**, which includes all directories and sub-directories, necessary for the project deployment. DO NOT push a single .zip file - instead, this should be uploaded to Learning Central.

- Any ’supplementary’ files, e.g. virtual environment or bytecode files or folders (`venv`, `.git` folders, `.pyc` files, etc.), which are not vital for the deployment **should be excluded**.

- Full instructions on what you need to push your project code to GitLab are found in Section *’– Submission to the School’s GitLab’* on p. 4. Please make sure you read these carefully.

Any code submitted will be run on a system equivalent to the laptops provided to the students, and must be submitted as stipulated in the instructions above. The code should run without any changes being required to the submitted code, including editing of filenames.

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a deduction of 25% for the corresponding task.
Staff reserve the right to invite students to a meeting to discuss coursework submissions.

Assignment

This coursework is comprised of one tasks which you need to complete for this assignment.

Website on Security, Quality and Usability

For this task, develop a website, which provides your visitors with information on security, quality and usability.

Your website should be developed using Flask [1] and SQLite database, and have a number of pages to display content to visitors. Each web page should have:

- **header** with navigation bar to enable the visitors to access all pages;
- section that displays the **main content**, appropriate for each page;
- **footer** containing relevant links to additional information, e.g. contact details, privacy policy, disclaimer, etc.;
- you can also include a **side bar** to display any information you think is relevant and makes your website more attractive and usable.

More specifically, your website's organisation and functionality should be as follows:

1. **Home page** should allow visitors to:
    - see a preview of each individual page available on your website - the preview should contain a title, image and a short summary of each page (3-4 lines);
    - access each individual page by click on each page's title or image;
    - access **login** or **registration pages** *(see explanation of content and functionality of these pages later)*;
    - see personalised greeting in the top navigation bar if they are logged in, e.g. a logged in customer will be greeted by their first name (e.g. `'Hello, Jane!'`). The visitor, who is not registered or logged in will see `'Hello, Guest!'` message.

[1] https://flask.palletsprojects.com/en/2.0.x/

2. **TWO individual pages** - one page for the topic of **'Security'**, and one for **'Quality'** and **'Usability'**.

    - It's up to you to decide on the actual **content** on these pages, but it is expected that for each topic your present: a brief explanation of main aspects of the topic, best practices in the industry, and your own reflection based on your experience in developing software systems.

      NB: You are allowed to use external sources for some of the content, but full references must be provided.

    - Each individual page should allow a registered visitor to leave **a comment**.
    - Additional web pages on each topic are allowed, but will not attract additional marks.

3. **Registration page**, which should:

    - Display a registration form, which asks the the visitors, who wish to register, to provide their first and last name, email address (which will they use to log in) and password.
    - Perform a simple check that the visitor input their email address in the expected format, e.g. `my_email@some_domain.com`.

      Note: this should be a simple check for the email address format - there is no need to check if this email address actually exists.
    - On **successful registration**: your website should redirect to the Home page, and display a message to the newly registered user confirming the registration succeeded.
    - If **registration is unsuccessful**: the system should display a meaningful error message, and will not redirect to any other page.

4. **Log in page**, which should:

    - Enable a registered visitor to log in, and if successful, the visitor is redirected to the home page.

    - Display an error message if there is a problem with a visitor's login. **NB:** make sure that error messages do not jeopardise your website's security.

    - Allow a logged-in user to log out.

Submission Instructions

For this task, you need to submit your work to two places: **Learning Central** AND **GitLab**, as follows:

– Submission to Learning Central

- This **must be a single .zip archive** that includes the complete source code of your website, including:

  - **README.txt** (or README.md) file, which includes: **your student number**, and **specific instructions on how to run your website** from the command line. You may also include any other information you consider relevant to your submission. Any references that are not on your website should also be included here.
  - **requirements.txt** file, which contain a complete list of all python packages required to run your website project.
  - Your **SQLite database**.

- All files need to be in appropriate folders so that your website can be compiled without any need to modify files or folders, e.g. the actual web pages in `templates` folder, images, JS, CSS files in appropriate subfolders of `static` folder, etc.

– Submission to the School's GitLab The School's GitLab server is located at `https://git.cardiff.ac.uk/`.

- All project code should also be pushed to a repository on our School's GitLab server. You will then need to share your GitLab repository with Federico Liberatore and Natasha Edwards as follows:

  > Click on '**Members**' on the left sidebar. This will open '**Project members**' page.
  >
  > In '**GitLab member or Email address**' field, search for Federico Liberatore (username: **scmfl2**)
  >
  > In '**Choose a role permission**' field, select **Maintainer** from the dropdown menu.
  >
  > Click '**Invite**' button to confirm.
  >
  > Repeat the above steps for Natasha Edwards (username: **scmne**), making sure the role permission is also set as **Maintainer**.

- **NB**: You **must** use the School's GitLab, using external git servers is not allowed.

Learning Outcomes Assessed

- LO4: Develop secure web applications
- LO5: Critically evaluate the role of security, quality and usability within software projects

---

Criteria for assessment

Credit will be awarded against the following criteria.

Usability: Is the navigation intuitive, consistent, and self-explanatory? Do all the pages have clear navigational menus that provide access to all other pages without breaking the logic of operation?

Presentation and content: Is the website professionally presented? Is the content laid out logically and consistently? Are HTML and CSS used effectively? Is the content high-quality?

The mark breakdown for each criterion is given in the following.

Task [Total: 100 marks]

- Functionality: programmatically generated content (i.e. pulled from the database, not ‘hardcoded’ manually). **20 marks**
- Functionality: Home page. **20 marks** (content displayed correctly - 5, access to individual pages - 5, access to registration and login - 5, personalised greeting - 5)
- Functionality: Individual pages. **30 marks** (for each page: content - 7, ability to leave a comment - 4, comments displayed correctly - 4)
- Functionality: User registration. **11 marks** (registration form - 2, input validity check - 2, on valid input user account created - 2, confirmation message - 2, redirect to Home page - 1, on invalid input error messages displayed - 2)
- Functionality: User login and logout: **9 marks** (successful login - 3, error messages on invalid input - 3, logout - 3)
- Usability and design/presentation: **10 marks**

For each item evaluated, partial marks can be assigned as illustrated in the following table. The rubric applies across all of the questions, with the split in functionality, code quality, usability, presentation and content defined above.

	Distinction (70-100%)	**Merit** (60-69%)	**Pass** (50-59%)	**Fail** (0-49%)

Content	The content is of high quality: provides an interesting, insightful, focused information on both topics, and demonstrates an excellent ability to choose relevant information and reflect on past experiences.	The content is of good quality but needs some improvement. Good ability to choose relevant information and reflect on past experiences.	The content is of reasonable quality, but requires substantial improvement. Limited ability choose relevant information and reflect on past experiences.	The content is either missing or very limited and is in need of major improvement, e.g. information is unfocused or irrelevant.
Usability	Navigation is intuitive, consistent and self-explanatory, with clear navigational menus on all pages providing access to all other parts of the site without breaking the logic of operation.	Navigation is mostly intuitive, with a few limitations in ease of navigation or with use of menus and links.	Means of navigation are somewhat confusing, some links not working. Consistency of design and presentation needs substantial improvement. Limited content.	Means of navigation are very confusing, most links are missing or not working. Using the website is a chore.
Presentation	Website professionally presented, with content laid out logically and consistently; effective use of HTML and other ’display’ elements, with CSS control of styling.	Reasonable presentation of content, although not yet of professional quality. Generally well laid out using CSS where appropriate but some limitations in overall coherence of design and use of space on page.	Consistency of design and presentation needs substantial improvement.	Inconsistent design across website with poor presentation: failure to employ CSS appropriately to control presentation. Text is often difficult to read (*e.g.* poor colour contrast, obtrusive background patterns, text too small or unnecessarily too big).

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on the return date via email.

The feedback from this assignment will be relevant for any future programming tasks.