CSC3123

Aims

To practice and enhance your web development skills

Learning/Skills Outcomes

- Gain experience of making design boards
- Gain experience of using the features of the PHP language
- Practice following strict development QA guidelines
- Develop your independent learning skills

Brief

You are to design and build a website meeting the requirements listed below that provides a logging tool for developers. The site should :

- support multiple users each working on multiple projects,
- allow users to log the time they worked on a particular project,
- let users store notes about their work to include text, links, relevant images (e.g. screenshots), and documents (PDF, Word etc.)
- be secure
- have a RESTful interface

Requirements

- The site ***must*** be built using PHP and the Framework – see the website for installation instructions on various platforms,
- Database access ***must*** be done using RedBean (installed with the Framework),
- Page rendering ***must*** be done using Twig and Bootstrap (also both installed with the Framework),
- Where appropriate, use icons from Font Awesome (again, supported by the Framework),
- All code ***must*** be formatted according to the rules specified below,
- All code ***must*** have suitable embedded documentation (PHPdoc, JSdoc)
- Where appropriate, all data sent or received ***must*** be comprehensively error checked,
- You are free to use external packages that provide facilities not already provided by the Framework. Any external code must be clearly referenced in your files if you are not simply using the original source or loading them from a content delivery network,
- As indicated above the interface must be RESTful.

Notes

- Links to documentation for all these tools are available at the Framework website and online introductory material will be available.
- If you have any questions or need clarifications, then ask. Don't wait.
- The Framework already provides you with multiple user support, you do need to implement it. It also supports file uploading and storage, again, you should not need to reimplement this.
- This is nowhere near as hard or scary as it sounds. If you have a good initial design for the UI and data storage, then implementation is straightforward. The Framework API provides you with support to simplify things like error checking, data handling, use of AJAX etc. etc. You will have access to support material on SEO (search engine optimisation), cacheing, and security.

Formatting

All code (PHP and JavaScript) must ***strictly*** follow these rules, some of which are (intentionally) not those that are familiar to you. The code for the Framework is all formatted using these rules so look at it for examples of how to proceed. Failure to follow these rules will lose you marks. If you are using a suitable code IDE then it should be easy to set it to format your code correctly.

- Strict separation of content : NO HTML in PHP files.
- All code must be indented 4 spaces to start with – only /** **/ comments can start at the left hand side. DO NOT USE TAB CHARACTERS.
- Braces occur on their own directly under the construct they are forming a block for. Indentation increases after the line with the brace. Braces should always be used – no single line blocks.
- A single space after reserved words (e.g. if, foreach, for etc.)
- Spaces around operators., i.e $j = $k + $b; not $j=$k+$b
- Class, variable and method names should be sensible in length and meaning. You can use camelCase if necessary but you ***must not*** use snake_case,
- Class names start with a capital letter, variables with a lower case letter and constants should be all upper case,
- *In PHP code* use TRUE and FALSE for boolean values. Do not use true or false.
- Every PHP files starts with a comment identifying the content and with suitable PHPDoc commentary. PHPDoc must be used where appropriate throughout the PHP code, i.e. every class variable and method must be documented.
- HTML should be standard HTML5 but should be styled as XHTML, i.e. lower case tags, /> at the end of singleton tags (e.g.
)
- If possible, you should SASS when writing any extra CSS code.
- SASS/CSS **must** be formatted according to these guidelines.
- Please try to use JSDoc for any JavaScript you write.

Deliverables

There are two deliverables for this module:

1. The design for the website, which must include:
    a. The design for the website, which must include
    A moodboard,
    b. A fontboard (this must be a PDF to ensure typefaces are rendered correctly),
    c. A comprehensive storyboard showing all the main pages and an outline of their layout and content. At least one of these should be shown using the colour scheme you decide on for your site.
    d. A description of the database structures you will use to represent the logged data. This can be an ER diagram or text. Also a brief description of the RESTful URL scheme that your site will be using.

    N.B. This submission is to allow you to develop your design skills. You will ***NOT*** lose marks if your final website appears different from your initial design, though you should familiarise yourself with the look and feel of Bootstrap sites before deciding on anything too complex! (You can make complex sites using Bootstrap, but this is just a 10 credit module!)

2. The final website, to include:

    a. The source of the website – remove the "vendor" directory
    b. A short essay reflecting on your experience of the design and implementation – *NOT a description of your code* – including also a summary of how you approached SEO, caching of pages and the security of the website.
    c. An SQL dump of your site database.
    d. Some screenshots of your site in use.

Marking

The first submission is worth 25 marks:

- Moodboard – 3 marks
- Fontboard – 2 marks (PDF)
- Storyboard – 10 marks
- Data storage – 10 marks (text, docx)

Submit a zip or tar archive of these files to NESS. Your fontboard should be a PDF file to preserve the typefaces you use. Do ***not*** use Microsoft Publisher for any documents – I cannot open them!

The second submission mark will be broken down as follows :

- Essay – 25 marks
- Website – 50 marks

- Adherence to style rules – 10 marks
- Use of support packages and their features : (RedBean, Twig, etc.) – 15 marks
- Use of PHP and its features – 15 marks
- Overall design of your solution 10 marks

Submit a zip or tar archive of these files to NESS.

N.B. This is a programming exercise not a graphic design exercise so there are no marks allocated for the look of the site. There are no specific marks for either JavaScript or CSS as Bootstrap (and the Framework) should provide you with everything you need to build this site. If you do have some extra CSS or JS then they can contribute to the marks for the overall design.

Comprehensive error checking and handling is extremely important and I will be looking specifically for that when evaluating your code.

Marks will be given along the scale Outstanding, Excellent, Very Good, Good, Basic, Borderline Fail, Fail, Absent which is equivalent to the degree classification scale with Outstanding being an extra category in the first class range and being equivalent to professional quality work. I am looking for you to show me your ability to take a specification and build a high quality site (coding aspect) using a specified set of tools that are thoroughly documented online.

Again, let me stress that none of this is as hard as it may sound. It is meant to be challenging but should all be within your skill set. The Framework is designed to provide the tools you need to build a RESTful website quickly and easily – make sure that you use it to its full potential.

Please ask for help if you are stuck!