Year 3 Computing
Web Framework Development

Individual Project – semester 2 – academic year 2020-21
FINAL CODE DEADLINE: 6pm Friday 23rd April 2021

You will also be required to defend your project in a video meeting during week 12 / revision week (1 x 15minute timeslot)

You are to design, develop, test, publish and defend a database-driven Symfony 5 PHP website, to illustrate the topics of the module. Your project should be an original approach to developing a website based on the **case study** topic for your **assigned lab group** (see separate document).

NOTE:

- Projects submitted on a topic different will score zero (unless a different case study has been agreed and confirmed by email with the module lecturer)

- Each student is strongly advised to create a high-level design for their own approach to the general case study topic
    - For formative feedback from the lecturer, no later than 1 month before the final deadline email PDFs of Use Case Diagrams and ER-diagrams outlining your web application design – the more detail in this design the easier it is to create the website.

Your work will be submitted through your private Github repository:

- DEADLINE 1: **no later than Friday 5th March**
    - You should have:
        - created your private project repository
        - submitted the repository URL to Moodle
        - added the lecturer as a collaborator

- DEADLINE 2: Week 9 commit: **no later than Sunday 29th March**
    - Commit of initial work on project – with commit comments describing features of this version

- DEADLINE 3: Week 10 commit : **no later than Sunday 18th April**
    - Commit of current project status – with commit comments describing features of this version

- DEADLINE 4: Final commit: 6pm Friday 23rd April
    - NOTE: If you have published your website:
        - you need to provide the URL for the published site in the README.md file of your project repository in this final commit
        - you must include a folder "screenshots"
            - containing several screenshots showing working use cases showing database CRUD at the published URL
            - or containing a short MP4 video file of you visiting the working published website
        - the site must be up and running for all of Saturday 24th April
          (which is possible using a sample project on Fortrabbit for example)

Please note:

- You must include database setup Fixtures with at least the following:
    - Top level administrator user, username/password = **admin** / **admin**
        - With ability to CRUD users and ROLES

General criteria for web project grading

- **Originality** (your own work)

- **Correctness** (it works)
  - Your project should be error free

- **Completeness** (features specified are present)
  - Refer to the marking grid and list of use cases

- **Code Quality & Software Engineering Process**
  - good quality and meaningful identifiers (names) files / folders / classes / ids
    - follow correct UpperCamelCase/lowerCamel case for OO PHP programming
  - decent and consistent indentation and code layout – the PHP PSR standards
  - demonstrate automated TESTING, e.g. with Codeception for all use cases etc.

- **Usability**
  - a decent user experience
    - e.g. use Bootstrap CSS
  - it should be straightforward and easy for users to locate and perform each use case

- **Consistency & Coherence** – decent website 'look and feel' (please avoid drop-down nav-bars)
  - a simple but effective website will score better than a hard to understand / messy complicated website – so keep a **non-technical website visitor** in mind at all times …

- **Technical Challenge –** Demonstration of attempting interesting / challenging features
  - NOTE:
    - there is no extra credit for using JavaScript … so don't make work for yourself

- **Value Added** (more than just examples reproduced or on changed in minor ways)

Academic Honesty

By submitting your project for assessment you agree to the following:

"The material contained in this assignment is my own original work, except where work is clearly identified and duly acknowledged. No aspect of this assignment has been previously submitted for assessment in any other unit or course."

In general: For every piece of work you submit to the Institute, your documentation must make it very clear which parts are your own creations; the work of others; and your adaptation of other's work, and what your adaptations were. Work submitted without full and unambiguous acknowledgements is plagiarism. Plagiarism and academic dishonesty can lead to failure of the module and other penalties outlined in the Institute's rules and regulations. For any project or coursework, you should discuss how to best declare the use of work from other sources with your lecturers.

This assignment is an **INDIVIDUAL PROJECT**. The work you submit must be your own (with fully declared exceptions described below). It is fine for you to ask a lecturer or fellow student for assistance with some problem you are stuck on during the project, but the actual final work created and submitted must be your own. While you may get IDEAS for code to solve particular problems from other sources, you must write and user the command line tools to generate all codes yourself. Remember by submitting this project you are declaring that it is all your own individual work unless explicitly and unambiguously acknowledged by you.

You must NOT SHARE YOUR OWN CODE FILES with fellow students! Also ensure your project Github repository is **private**!

You should NOT BE TYPING IN CODE into a fellow student's computer project – talk to them yes, but let THEM design and type-in their own code.

You may NOT use other PHP code inserted into your own classes or scripts - i.e. all your PHP classes must all be your own code, or generated by you at the command line, or imported through the Composer package manager utility.

The text **content** of your web pages should all be prose text that YOU have authored yourself (so do NOT copy and paste from Wikipedia or IMDB etc.) – pages only need a few sentences/paragraphs each, so write and paraphrase them yourself in your own words. Or you may generate random content, e.g. with Faker.

You may use the following **without need of citation**:

(1) you may use and modify any **example codes from the lectures/labs** on Moodle, also any of the lecturer's GitHub sources
(2) you may generate code with the Symfony command line tools
(3) you may (and in fact are encouraged to) use full PHP **components** which you install using **Composer**. These are therefore stored in the **/vendor** directory, and your **composer.json** file lists all project dependencies for third-party components

You may use the following, **with citation** in a **`/public/sources.txt`** document

(4) client-side **media** files, such as images, fonts, and templates for HTML/CSS/JavaScript
- you must declare these in your **`/public/sources.txt`** document
  - give the URL and list what client-side files you used from that location

Talk to the lecturer if you have any doubt about whether it is permitted to use / how to properly cite resources you have not created yourself for this module project.

Declaration of media sources
\public\sources.txt

\public\sources.txt – plain TEXT file declaring sources

- You are to submit a **text file** which states the origin of each image (and font and JS) used in your website (be specific, e.g. saying "Google images" is not good enough! You need to state the URL including the original filename of the image from that source, and also state the image filename as it appears in your website folder)
- If you created an image yourself, say so …
- You should also declare any other sources used in your website, and state clearly how they have been used, so it is clear which parts of your project are your own work, which are the work of others, and the extent to which you have changed any work from others
  - But remember, apart from images all other parts of the project should be your own original work …
- Example of content:
  - File: **\public\sources.txt**

    logo.gif
    http://www.itb.ie/images/itb_logo.gif

    cat.jpg
    http://images.wisegeek.com/young-calico-cat.jpg
  - And so on
  - All you need is to state the name of the image file in your images folder, and the URL of where I can find the original image on the web image
  - HINT: View Source will give you direct links to images that you see on a web page
    - Click that link and you should see the image, and be able to copy and paste the URL into your sources.txt document
- NOTE
  - **Do NOT submit a PDF or Word document !!**
    - just a TEXT FILE with simple contents as shown above
  - when you download an image / CSS file, that is a good time to rename it as you save it into your \web\images directory
    - you'll be marked down for poorly named images like: **p041gljk.jpg**
      from:
      http://ichef.bbci.co.uk/wwfeatures/wm/live/1280_640/images/live/p0/41/gl/p041gljk.jpg
    - a much better file name would be**: cat_face_closeup.jpg**
      from:
      http://ichef.bbci.co.uk/wwfeatures/wm/live/1280_640/images/live/p0/41/gl/p041gljk.jpg

Web Framework Development – project – indicative marking

STUDENT

F (< 35) D (< 40) C (< 55) B (< 80) A 80 ... 100

	Databases	Security	Case Study Use Cases CS = Race Auction	Testing	Github	Advanced Features / Quality / UX
A	for A grade:: site must be published with all roles / login / db CRUD working plus:: broad TESTING of all Case Study roles			❍ ALL Use Cases & Roles ❍ $I->cannot/dont testing ❍ test method names ❍ one test-per-method	❍ Issues ❍ Releases & Versions	❍ Published, Working site DB, UCs all working etc. ❍ File upload / Logging / API ❍ Advanced: other
B	❍ Fixtures associated CS Entities (not just User-Role) ❍ No repeating groups!		❍ CS STAFF use cases	❍ DB via Form-filling ❍ ADMIN User CRUD #PW ❍ STAFF use cases	❍ useful commit messages	❍ Navbar for ALL CS roles ❍ easy to use ❍ consistent look and feel ❍ tidy and professional looking
C	❍ Fixtures for two+ CS Entites	❍ hierarchy for CS roles	❍ CS USER use cases	❍ Login for each CS ROLE ❍ USER use cases ❍ PUBLIC use cases	❍ commits over WEEKs	❍ Navbar for USER role
D	❍ CRUD 2 x CS entity ❍ PUBLIC list DB items TABLE-per-page ❍ PUBLIC item details RECORD-per-page		❍ CS PUBLIC pages (e.g. index / about / sitemap) ❍ CS PUBLIC use cases		❍ Project folder contents (not ZIP. no parent etc.)	
core from provided starter project	❍ role is String not JSON ❍ CRUD Users ❍ DBFixtures: User	❍ ADMIN New Hashed Passwords ❍ ADMIN User CRUD - Hashed Passwords ❍ Log-in protect CS ROLE use cases ❍ Fixtures Hashed Passwords				

COMMMENTS

CS = Case Study / UC = Use Case