Institute of Technology Blanchardstown
Institiúid Teicneolaíochta Baile Bhlainséir

Year 2 Computing
COMP H2026 Web ServerSide Development

Individual Project – semester 1 – 2016-17

OVERVIEW: You are to create, publish, and defend a website illustrating the topics from the module:
A database-driven e-commerce website selling merchandise for a movie

See Moodle for submission date and upload link. Use **your** name and student number for uploaded files!

Create:

- Website part 1: A ZIP folder containing all your website files
  - **SMITH_Matt_B000007_website.zip**
- Website part 2: Textfile containing an SQL-dump of your database structure and contents
  - **SMITH_Matt_B000007_SQL.txt**
- Website part 3: Textfile **sources.txt** declaring your image/font sources
  - **SMITH_Matt_B000007_sources.txt**

Defend:

- Website part 4: A short video screencast, demonstrating and defending your website (maximum duration 5 minutes – submit a .swf Jing video screencast)
  - **SMITH_Matt_B000007_video.zip (containing SMITH_Matt_B000007_video.swf)**

Publish (if aiming for an 'A' grade)

- Website part 5: The URL of your published website
  - Note this should be available to view online from the submission deadline until at least 31st January 2017
  - Published pages should exactly match the contents of your submitted ZIP folder (part 1)
  - **SMITH_Matt_B000007_URL.txt**

SUMARY OF WEBSITE PROJECT REQUIREMENTS

Core features:

- (public pages – no login)
    - user can view general information pages (index / about / sitemap)
    - effective horizontal and/or vertical navigation bars (with current section/page highlighted)
    - user can see a list of products (or ideally product categories)
        - TABLE-per-page
    - user can select a product and see details of that product
        - RECORD-per-page´´
        - User can click "Purchase" button and go to a purchase form for that one item
- (admin login)
    - administrator can CRUD products and customers
        - CRUD – Create – Read – Update - Delete

Every student must create a shopping website for a **different** movie, so you need to **communicate with your lecturer**. The lecturer will publish a list of the approved movie choices for each student – **only start work on your website once your project choice has been published by the lecturer**.

NOTES:

- You may be asked to demonstrate/defend your website in person to the lecturer in January
    - (if required you'll be given several weeks notice)
- Projects submitted on a topic different to that published by the lecturer for the student will score zero
- less is more …
    - 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 …

Please also refer to the provided appendices, and the indicative marking scheme.

Meta-criteria for programming / web development projects (keep these in mind all the time)

- **Correctness** (it works) & **Completeness** (features specified are present)
- **Code Quality & Sofware Engineering Process**
    - good quality and meaningful identifiers (names) files / folders / classes / id's
    - decent and consistent indentation and code layout – the PHP PSR standards
    - illustrate MVC and Front Controller architecture & directory structure
    - use of tools (Composer, phpDocumentor, PhpUnit, Github/Lab/CI)
- **Usability –** a decent user experience
- **Consistency & Coherence** – decent 'look and feel' to the website
- **Technical Challenge –**Demonstration of attempting interesting / challenging features
- **Originality** (your own work) & **Value Added** (more than just examples reproduced)

APPENDIX A: Indicative Marking scheme – B/C/D

To get an A grade (excellent) project needs to demonstrate all criteria for a B grade and the following:

- **MUST submit: Part 4: Defence video**
- majority of features excellent quality
- Passwords must be hashed (i.e. not storing plain text password in the database)
    - Note – include README textfile telling lecturer the passwords for users in your database
- (testing) at least one class 100% unit tested & testing coverage report generated
- demonstration of achievement of technical challenges (through advanced features)
- (session / authentication) memory between sessions and login
- several advanced features implemented (see next page)
- all PHP code **Object-Oriented** (apart from **index.php** and setups/configs)
- Front Controller

To get a B grade (above average) project needs to demonstrate all criteria for a C grade and the following:
MUST submit: Part 3: Database SQL dump

- majority of features above average quality
- (DB) PDO communication with MySQL database
- (form handling) validation of data received
- (doc) classes fully documented, and **/docs** folder generated (with very few errors – ideally **zero**)
- separation of view contents and main logic (controller – view of MVC)
- organised database connection code (either functions or classes – model of MVC)

To get a C grade (basic pass) project needs to demonstrate the following:

- majority of features good quality
- (table-per-page) list of product items on one page (from object repository or DB)
- (record-per-page) details of a single item on one page (from object repository or DB)
- (interactive form), data received from form by server and confirmed back to user

To get a D grade (below pass quality) your project needs to demonstrate the following:

- **MUST submit Part 1: ZIP of website folder/files**
- **MUST submit Part 2: ‘NAME_NUMBER_sources.txt’**
- reasonable subset of features implemented and working
- website allows user to complete several core tasks
- runs without any syntax or runtime ERRORS

APPENDIX A: Advanced features for project grade A

Examples of advanced features for extra credit (talk to lecturer about other ideas you have):

- Published and working website and database at public URL
- User can UPLOAD their own image to change their profile picture
- Continuous Integration testing (e.g. Github TravisCI / Gitlab CI)
- Version control your website on Github or Gitlab (and provide me with the URL for the repo)
- Data providers in one or more tests
- Custom Exception throwing and handling
- Use of Twig templating system
- good attempt at advanced feature Use Cases (see below)

USE CASES:

Website advanced features

- (public pages – no login – values remembered during the 'session')
    - user can change display preference (e.g. background colour / font / font size)
        - and this change is remembered for the remainder of the session
    - user can add to 'wish list' or 'shopping basket' of items
        - These should be remembered during a shopping 'session'
        - User can click "Purchase" button and go to a purchase form for all items in 'shopping basket
- (customer login)
    - user can editor their profile details (choose profile picture from given selection)
    - user can change their password
    - contents of 'wish list' or 'shopping basket can be stored in DB (and retrieved next time they log in)

APPENDIX B: Academic Honesty

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 all lines of code yourself and full understand every line of code in your classes and index.php files. 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 CODE with fellow students!

You may NOT use other PHP code inserted into your own classes or scripts - i.e. your PHP code in index.php/setups and all your OO classes must all be your own code.

The text **content** of your web pages should all be prose you have written yourself (so do NOT copy and paste from Wikipedia or IMDB etc.)
– pages only need a few sentences/paragraphs each, so write them yourself.

All work submitted should be your own, except for:

(1) media files, such as images, fonts, and templates for HTML/CSS
    a. you must declare these in your sources document (see relevant Appendix)

(2) you may use and modify any example codes from the lectures/labs/lecturer GitHub sources without any need to reference such sources.

(3) you may (and in fact are encouraged to) use full PHP components which you install using Composer (and is stored in the **/vendor** directory). Your **composer.json** file lists all project dependencies for third-party components, so there is no need to acknowledge any such components any further.

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."

APPENDIX C: declaration of media sources NAME_NUMBER_sources.txt

"NAME_NUMBER_sources.txt" – text file declaring sources

- You are to submit a **text file** which states the origin of each image (and font) 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: **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

APPENDIX D: Defense movie requirements NAME_NUMBER_video.zip/swf

Overview

Instead of an on-campus demonstration, explanation and defence of your website in person, you are required to submit a video ‘screencast’ with voice-over recording, in which you demonstrate the website, some of its source code, and argue its quality and how well it has met all the project criteria.

Content

The video should include 2 main kinds of content – both with a spoken narration to explain what you are doing / what the viewer is seeing:

- Demonstration of the usability, design and content quality of the website pages
  - by you navigating to different pages in the role of a website visitor, perhaps seeking specific information from the website
- Technical description and quality arguments:
  - Through you showing the HTML and CSS, PHP and Twig source code behind some of the pages. Also show directory structure and SQL/composer.json etc.

In both cases refer to the project assessment criteria, so that your demo/defence video forms a strong argument for how your project meets the different requirements.

Format/software

Your submission should be created using the free “Jing” software

`www.techsmith.com/jing.html`

Hardware: microphone-headset

Always use a microphone-headset for recording demos (headphones with a little ‘stick’ microphone – you can pick them up in the 2euro shop!) – NEVER use the built in mic. of a laptop (it means every move of the mouse or key press becomes a BANG in the recording). You will be marked down if audio quality is poor.

Duration

The maximum duration of your movie should be 5 minutes (the free Jing software limitation).

Suggestion – make use of the project rubric-marking grid

Use the project rubric marking grid, either as a visual reminder to yourself as you record you video, or from the rubric create a ‘script’ for your video, listing the sequence of actions you’ll undertake, with bullet points for points you wish to be sure to explain in the voice-over narrative during recording.