# CO1706 Block Mode Assignment (Part 2): 75% of module grade

## Release Date: 19th February 2020

## Due Date: **Friday 24th April 2020 (submit on Blackboard by 5pm)**

| Learning Outcomes: | |
| --- | --- |
| **1.** | Discuss the suitable types and format of interactive media |
| **2.** | Identify relevant constraints for different types of interactive media across a range of platforms |
| **3.** | Design and evaluate the interface to an interactive application using appropriate criteria. |
| **4.** | Appreciate the different interaction methods and identify appropriate contexts of use for interactive applications |
| **5.** | Apply industry standards and current approaches to implementing interactive applications for the web |
| **6.** | Interface an interactive application with a database using server side scripting |

## Scenario:

You work for a multimedia software agency and have gained a contract to design and implement a prototype interactive **mobile** application for an online music streaming service.

For this part of the assignment you must add functionality in order to create a working prototype web application. The client wishes you to develop the application using **HTML5** and **CSS** so that it is suitable for use on a range of mobile devices. The interactive application must allow users to interact with a database using **PHP**.

A template database has been provided (see assignment resource files on Blackboard). You are expected to use this for your assignment, although reasonable changes can be made if necessary (e.g. you can certainly add new tuples or change data such as passwords; you may wish to add some further attributes or relations if they assist with your design).

Space has been provided for you on the Computing server (vesta.uclan.ac.uk) and you should use this to host your site – instructions on connecting and transferring files are provided in the lab worksheets, but if you forget your password or have difficulties with using vesta you should contact LIS ([liscustomersupport@uclan.ac.uk](mailto:liscustomersupport@uclan.ac.uk) or ring 5355, or see your local technical support in Cm107).

**Please note: you can use your work from Assignment 1 as a starting point, but you do not need to follow the same design if you prefer not to.**

Important: this is an individual assignment and you are required to write the code for the PHP, HTML and CSS yourself. We expect students will use tutorials or sample code to aid your understanding, but if you include code from a tutorial or any other source in your work then this needs to be referenced appropriately, just as you would when quoting a source in a written report (e.g. you can reference the source through comments in your code). If you have used any frameworks (e.g. Bootstrap) then these must also be referenced appropriately. Please ask your tutor if you need any clarification on this.

Your work will be assessed through a demonstration (see section below for more details).

## Grading Criteria:

For a third-class grade (40%+) you must have:

* Content marked up in HTML with a linked CSS stylesheet, using a layout and style suitable for a mobile app
* All pages must use the PHP file format and run on the web server provided
* App connects to the provided database using PHP and MySQLi (if you prefer to use an object-oriented approach you can use either MySQLi or PDO)
* App must contain a functional login page – this should check the user’s username and password against details stored in the database in order to provide access to the rest of the site, and must NOT allow access if the details are incorrect
* App must contain a homepage displaying the current special subscription rates (retrieved from the database)
* App contains a tracks page listing the tracks available (read from the database using a loop)
* Submission includes a brief ‘readme’ file which details what functionality has been completed, the address of the site on the Vesta server, and any usernames/passwords needed to access the site.

For a 2:2 class grade (50%+), you must also have:

* Use of sessions to ensure that only logged-in users can access content
* The tracks page displays album artwork (using database queries to retrieve the filenames)
* Users can play tracks using HTML5 media features
* Users can browse for music by genre
* Provide a personal greeting to users when they log in, based on their username or name retrieved from the database
* Site checks for likely user errors and provides informative feedback (e.g. form validation on the login page)
* If the user enters a URL incorrectly, a custom 404 Error page is displayed that fits with the branding of the site
* Code should be neatly structured, including comments where appropriate

For a 2:1 class grade (60%+), you must also have:

* A registration page, allowing new users to be added to the database. Users should be able to enter their name, choose a password and select a pricing plan (retrieved from current plans available in the database). Duplicate usernames should not be allowed.
* Individual track description pages containing more detailed information (you should not create a page for each track, but instead generate content from the database for the track selected)
* Logged-in users can submit reviews of tracks, which are added to the database
* Likely errors are handled appropriately throughout the site (e.g. validation of all user input against errors and simple attacks such as HTML injections)
* Passwords should be stored and transmitted securely using a current industry standard (e.g. hashing and salting with a suitably secure algorithm such as bcrypt)
* The content retrieved from the database fits with the design of the page, so that the page still would display suitably on a mobile screen

For a first-class grade (70%+), you must also have:

* App has playlist functionality, including random generation of tracks
* Users can browse for music by artist and/or album
* Album description pages (generated from the database, not created individually) allowing the user to see all tracks on an album together
* App incorporates text search function or page
* The review page for a track should calculate the average rating for that track from all users and display it in a helpful format
* Your code should not produce any bugs when tested by the lab tutor
* HTML and CSS content should validate without errors (e.g. through https://validator.w3.org)
* A professional looking prototype app

For the highest grades (~90%+), as well as all the above, you should also have:

* A recommendation system which suggests tracks that the user might like, based on their ratings of other tracks

# Submission

You **must submit** via the appropriate link on Blackboard. You need to submit a single .zip file containing all the files required for your page to display (i.e. the full contents of your web folder). **Please also remember to submit a brief ‘readme’ file along with the contents as specified above.**

# Late work

Except where an extension of the hand-in deadline date has been approved (using extenuating circumstances forms), work that is handed in within 5 working days late will receive a maximum mark of 40%. Work handed in later than this will receive 0%.

# Extensions and extenuating circumstances

If you believe that there are circumstances that justify an extension of the hand-in deadline for assignment work, you can apply for an extension via Starfish or the School Hub. Extensions (to a maximum of 10 working days) can be granted when there are serious and exceptional factors outside your control. Everyday occurrences such as colds and hay fever do not normally qualify for extensions. Requests for extensions should always be made before the hand-in date. For more information see <https://www.uclan.ac.uk/students/study/examinations_and_awards/extensions.php>

The School considers extenuating circumstances to be conditions that significantly impact on your work. Typically these will cover more than one module. Requests for consideration of extenuating circumstances in respect of assignment work submission should be made using the extenuating circumstances forms online. You are advised to speak to your Course Leader/Academic Adviser prior to completing these forms. Whilst extenuating circumstances are being considered, you are advised to inform relevant staff members, and continue with the assignment.

You must submit your documented extenuating circumstances, using the official extenuating circumstances form, to the C&T Hub as soon as possible. Extensions of up to 10 working days can be granted in order that you can submit the current piece of assessed work. If the circumstances are too serious to be resolved by such an extension, alternative arrangements involving a different piece of work may be made.

If you have any questions about this process, please contact the C&T Hub (C&T Building Room 235, [CandTHub@uclan.ac.uk](mailto:CandTHub@uclan.ac.uk)) or talk to your Academic Adviser.

# Referral work

If you do not receive a passing grade for this assessment, you may need to complete referral work in order to pass the module. This will be offered to you later in the year, and your tutor will give you details of the work to be undertaken. Marks for referral work are capped at 40%.

# Cheating

The consequences of cheating in assessments are serious - you will fail the module. Cheating is using or attempting to use unfair means to enhance performance. This includes plagiarism (presenting someone else's work as if it was your own), collusion (working with others on an individual assignment), and allowing other students to access your work. Make sure that you do not give someone the opportunity to steal your work (e.g. by asking them to print it out for you). We tell students about cheating both during induction and in your student handbook, but if you have any doubt about what cheating is or how to reference material properly, please ask a tutor.

For any more details about assessment procedures and policies, please see the Assessment Handbook or ask your Academic Adviser:

<https://www.uclan.ac.uk/study_here/assets/assessment-handbook-0916.pdf>

# Detail of Work

Week 1:

Started work on learning bootstrap and designing the public (no login required) pages. I started with the home and package pages, then moved onto registration and login pages.

Week 2:

Recapped PHP work from when we were up at UCLan and started adding code to my registration page. Added database integration to the offers page. Added some more PHP code and DB integration to the registration page, as well as started session implementation.

The registration page ensure that all information is correct when signing up, including a valid email address that is not already taken, two matching passwords, and an option box for the package selection. I also went back after this code was written to tidy up and add comments as it was pieced together while I learned.

Added password hashing and added functionality to the login page. Users can now log into the service, where a session will be created and will be re-directed to the song list page. I committed the code at this point.

Week 3:

Added a custom 404 error page. Started work on the song list using bootstrap cards. I decided to use bootstrap cards to display all the available albums in a horizontal row that is scrollable. Each card is clickable and will take the user to a new page where they can listen to each sample track. This of course is all only accessible to logged in users with a valid session.

I found out about prepared statements at this point and decided it was worth looking into.

<https://websitebeaver.com/prepared-statements-in-php-mysqli-to-prevent-sql-injection>

I then changed all my database query’s to be prepared statements to prevent SQL injection.

Next I wanted to have one audio player on the site, so that the site looked more like a music streaming service! I used some JavaScript to do this, adding event listeners to buttons that, when clicked, would change the source of the music player and start playing the ‘music’.

Week 4:

Added reviews and average ratings to each track’s page, and re added an “all tracks” page which displays all the available tracks with details such as album and average rating.

Also, I added playlist functionality. At this point I wished that I had done OOP php for this as I now have a bunch of very similar pages that could have been made a lot easier using a base class for them all (album, playlist page, artist page etc all display a heading and a table of tracks)!

Added a new SQL statement for getting average review rating for the all tracks page, as before I was looping through all the reviews and calculating the average (oh dear). Now it uses SQLs AVG function which is much more efficient!

Todo

Fix login error messages

Add you need to be loggen in etc messages to pages where session is invalid.

Clean up form code

# Questions

* Can packages be on a separate page?
* Do images need to be used?