University coursework requiring to create a website based around an artist or a band. All the images and materials were used for educational purposes only.

Below are indications of the requirement and the corresponding file and line number where they were implemented.

**A clear use of HTML5**

HTML5 is used throughout all pages, examples of HTML5 tags such as nav and footer can be found in index.php (line 37, line 324)

**Use of the Bootstrap framework providing a responsive layout**

Used throughout all pages, some examples are index.php (lines 15-18, line 37, line 130, line 190)

**Use of JavaScript to manipulate the DOM based on an event**

In script/navbar-overlay.js on button click (line 7) different classes are added to DOM elements to change their appearance. The navigation menu becomes full screen which makes it easier to use on small screens.

**JavaScript loading of dynamically changing information**

script/countdown.js. We substract the current time from the end date in miliseconds (line 12) and display the result (line 20), it is updated each second (line 31) This creates a countdown timer for each tour date.

**Use of jQuery in conjunction with the DOM**

script/navbar-colour.js When the user scrolls past the image carousel the navbar colour is changed from transparent to dark in order to improve visibility (line 10) and the changes are reverted if they go back up (line 13).

**Use of a jQuery plugin to enhance your application**

I have used a plugin that measures password strength when the user is signing up. The plugin settings are configured in script/jqueryPluginStart.js. The source files for the plugin are script/password.min.js and the css for it is css/password.min.css.

**Use of AJAX (pure JavaScript i.e. without the use of a library)**

In script/tourDatesAjax.js (lines 2-24) an AJAX request is sent to a php script (script/getTourDates.php) which fetches the tour dates from the database and sends them back in JSON format. Javascript then recieves the response (line 8) and outputs the data in Bootstrap card elements (lines 11-20)

**Use of the jQuery AJAX function**

script/requestDataAdmin.js When a button in the admin section is clicked (lines 3 and 16) an AJAX request is sent (lines 5-13 and 18-23) to the corresponding php script which fetches info from the database.

**Use of cookies**

A Javascript cookie is used for setting the user’s favourite colour in the file script/cookie.js. The script is included in welcome.php line 30. PHP cookies are used to maintain login sessions (script/logIn.php line 42, script/signUp.php line 66, index.php lines 3-6).

**User login functionality (PHP/MySQL)**

A Bootstrap modal is displayed when the user clicks on the ‘Log In’ button (index.php line 55). The modal has a tab for either loggin in or signing up (index.php lines 64-127). When the submit button is clicked the form is sent to script/logIn.php or script/signUp.php. The log out functionality is provided by script/signOut.php which removes cookies and session variables.

**Admin section of the website (PHP/MySQL)**

admin.php Here the admin can browse, edit, add or delete information from the users or tour dates databases. The credentials needed to access this section of the website are provided in the file adminCred.txt in the ZIP submitted to Blackboard.

**Ability to select, add, edit and delete information from a database**

**Select**: By clicking on the corresponding button the admin can view either the users table (script/getUsers.php lines 12-15) or the tour dates (script/getTourDatesAdmin.php lines 8-11). The tables with the data are dynamically created by the php scripts: getUsers.php (lines 22-50), getTourDatesAdmin.php (lines 18-47). Each table row has buttons which on click send the record’s id to the script for editing or deleting.

**Add**: The admin can add tour dates by clicking on the Add button above the table, which redirects to script/addTourDate.php. A form is displayed (lines 23-46) upon submission the input is validated (lines 53-62) and if everything is alright, the data is added to the database (lines 65-70).

**Edit**: A user’s first and last name can be edited (script/editUser.php lines 79-83). Tour dates can be edited as well (script/editTourDate.php lines 88-93). The scripts display a form with the existing data associated with the record in text fields which then can be edited and submitted.

**Delete**: Users can be deleted (script/delUser.php lines 53-59), as well as tour dates (script/delTourDate.php lines 53-59). The scripts first display a confirmation message and if the admin clicks on the submit button the query is carried out.

**Appropriate consideration of relevant laws**

There is a privacy and cookie policy modal (index.php lines 278-321), it is accessible through a link in the footer (index.php line 326). Also, the user has to tick a box when signing up to confirm they agree to the policy (index.php lines 116-118). The policy is short and easy to understand. It explains what data is collected, how long it is kept for, how it is stored and secured and who to contact if the user would like to request the data to be edited or removed.

**Security measures**

**SQL queries should be written as prepared statements**

script/logIn.php (lines 26-33), script/signUp.php (lines 40-47). The aforementioned scripts for the admin section also feature prepared statements.

**Passwords should be salted and hashed**

When signing up the password is hashed script/signUp.php (line 57). The password\_hash function creates both a salt and a hash. When logging in the password\_verify function is used to check if the provided password matches the hash stored in the database (script/logIn.php line 39).

**Validation of user input**

script/dbCred.php has a function called sanitizeInput (lines 18-23). It takes input as a parameter and passes it through the trim, stripslashes and htmlspeacialchars functions. This script is included and used in all other php scripts. Some examples: script/logIn.php lines 17, 24. In script/signUp.php name and email fields are additionally filtered (lines 19-22, 33-36).

**Any other relevant security features**

The admin page and the scripts used modifying database records, check if the user has the credentials associated with the admin account. For example, script/delUser.php lines 6-9, admin.php lines 4-6.

Log In cookies and sessions expire after an hour.

The admins input is validated and sanitized before modifying db records addTourDate.php lines 53-56.