# 1. Updated set of requirements

### Requirements achieved:

* Store customer details:   
  Personal details can be stored including name, email address, username, password and a short biography. Purchase history and progression with courses cannot currently be stored.
* Provide secure log-in:   
  This does not employ any SSL or any other encryption but customers/members can log-on with email and password. Details are held in localStorage and customers can then view and update details.

### Requirements not achieved:

* Retain customer purchase history:   
  Currently there is a 'basket' which is held in session storage. This would then be transferred to the database once payment had been made, but has not yet been implemented
* Process payments:   
  There is currently no facility for processing payments.

# 2. Discussion of site and technologies used

Currently the website can allow users to register; their name, email, a username (which should be unique) a password and a short biography is saved to a database. When a user registers they are automatically logged on, which means that their details are saved in local storage. If a user then chooses to view their profile, these details are retrieved from local storage and use to populate a form. This form then can be used to update their details. Both the 'register' and update 'profile forms' prevent any updates if required fields are blank or if the email address is not in the correct format. A specific error messages appear to advise which fields should be populated or if the email address needs to be corrected.

If a user logs out, the session storage is cleared and the 'log-in' box reappears in the top right of the page; this is a PHP include, and is included in most pages. If a user successfully logs on using an email address stored in the database along with the matching password, the log-in box is replaced by a box stating who is logged with buttons to either log-out or go to the user's profile page.

The form validations and hiding or revealing of the log-in box are implemented using javascript. Writing, retrieving and updating data to and from the database is implemented using a combination of javascript, JQuery (using the AJAX method) and routes from the SLIM php microframework.

The site currently has a limited number of products for sale. Currently these are only in the 'Equipment' section. The index page has links to three different items. These are displayed within three separate divs which each have a background image of the relevant item. Layered over the top of each image, but containing all the the other elements, is another div with a white background of 50% opacity. When the mouse cursor moves over the div, the opacity changes to zero. This is implemented with a javascript ready function.

The 'shopping basket' can be used regardless of whether or not a user is logged in. This is implemented with session storage so is lost whenever the browser is closed. Items/products are stored in the session array as an array of serialized objects. These php objects have two attributes: product name and price. Adding a product to the basket uses an AJAX request via javascript/JQuery and a SLIM route; once added, the link in the top right of the page asynchronously updates to indicate the new number of items. The 'basketview' page includes a php function which loops through all the items in the array and displays their names and prices in a table along with a button for removing the item from the basket. The sum of the items prices is displayed at the bottom of the 'Price' column. There is a 'Empty basket' button at the bottom of the the page which attempts to use a SLIM route to unset session storage, however this does not work.

Attempts were also made at a SLIM route to retrieve the item objects from the session array and transfer them to the database, but these were also unsuccessful.

# 3. Platforms tested on

The application has been tested on Firefox, Chrome and Internet Explorer. All features appear to work on all three platforms, however styling does not carry across particularly well.

# 4. Appraisal

Extensive use has been made of javascript, SLIM and php. Form validation and CRUD for users/members is quite successful, although measures need to be put in place to ensure the uniqueness of usernames which is used as a primary key in the database. A shopping basket has been created, but if an item is added more than once it appears as individual items in the list rather than once with a counter to indicate the number. The ability to actually make a purchase is missing entirely. A user's profile page should show a list of all products bought, but this was not achieved.

The styling of the site is aesthetic although relatively minimal. Button elements have made greater use of CSS and extension of this to other elements could perhaps give a slicker feel to the website.

Overall, the website has a pleasing look and feel and some robust, if basic, functionality.

## Folder Structure

