

Assignment 2 & 3: Group Project – Build A Web Application

Class	HDWD
Stage / Year:	1
Module:	Server-Side Web Development
Assignment No:	2 & 3
Deadline:	See Moodle, for Assignment 2 & 3 deadlines. Late submissions will be penalized
Submission:	Upload to Moodle
Weighting:	Assignment 2: 15%, Assignment 3: 25% of module

Introduction

Working in a group of 3 people you will plan, design, and created a dynamic PHP web application with a MySQL backend containing the following features: C.R.U.D. pages, user access control using sessions, dynamic navigation bars, CSS and modular code design.

Build a web application for a Fitness Centre (pick a name). The website will have 3 user levels:

- Public
- Member
- Admin

Build the public pages, then the member pages, then the admin pages.

Web Application requirements:

- Handle form data in a secure manner e.g. use HTML and PHP validation techniques when handling user inputs.
- Make the website look professional by using good CSS techniques (frameworks) and appropriate content.
- Your HTML, CSS, and PHP code should be original and commented. You may be asked to explain any piece of code. Failure to understand and explain code may be taken as evidence of plagiarism. You may use frameworks such as bootstrap. All HTML and CSS should be validated.
- PHP code should be modular e.g. include header and footer files as opposed to copying and pasting the same code to several pages.
- The home page of the site should be called index.php
- Your website should follow best practice in terms of security.
- Store its data in a MySQL database.
- Connection to the database should be made using one file (e.g. connect.php).
- Database table names should be singular e.g. *user* not *users*.

Pages

The following pages must be created. You have the choice to create additional pages if so desired.

Name	Function	Main Table	Public	Member	Admin
index	Basic information about Fitness Centre plus 2 feature boxes. A feature box should highlight new classes, events, special offers etc. Each feature should contain a title, detail, an image and an optional link to another page in the website (for more information).	page	Y	Y	Y
index_edit	Allows the editing of the feature information presented on the homepage.	page	N	N	Y
registration	Displays fee information and allows member to register. There should be 3 fees levels with differing title, benefits and price. Member must choose which classes they are registering for.	user & fee	Y	Y	Y
registration_edit	Allows the fee information to be updated	fee	N	N	Y
class	Displays a list of class summaries , and their associated photo, and link. All photos should be stored in a folder not on the database. The link brings the logged in user to the relevant class page (using GET) and sends them to the login page.	class	Y	Y	Y
class_details	Show 1 attractively formatted row from the class table. This is shown when a link is clicked on the class page. This should display a class title, detailed description, a link to some document.	class	N	Y	Y
class_details_edit	Allows the editing of the class information	class	N	N	Y
testimonial	Shows a list of admin approved testimonials given by members	testimonial	Y	Y	Y
testimonial_add	Allows a logged in member to give a testimonial about a class. Each testimonial should contain the class name, the date, the members first name and their comment.	testimonial	N	Y	Y
testimonial_manage	Allows the admin to decide which testimonials are displayed	testimonial	N	N	Y
contact_us	Form enabling members of the public to leave a message which should contain: - Name - Email - Phone no - Message	contact_us	Y	Y	Y
contact_us_manage	Allows the reading of the messages	contact_us	N	Y	Y
header	Session start, connect.php, links to all pages appropriate to the user level. Prevent unauthenticated users from accessing in appropriate content. The header should contain a navigation bar The header should query the page table to populate a dropdown with the name of all accessible pages A user login box supporting both member and admin login You can choose to have one header which programmatically adapts to who is logged in or have 3 different header files.	n/a	Y	Y	Y
logout	A user should be able to access a logout script from a link in the header. When this logout script executes the session should be destroyed and the user should receive a logout confirmation message and then be returned to the index page.		Y	Y	Y

Deliverables

Both assignment 2 and assignment 3 must be submitted via Moodle

Assignment 2: Initial documentation (15% of the module)

- A completed version of **assignment2_Memberno1_Memberno2_Memberno3.doc** (template available on Moodle)

Assignment 3: Final Documentation, Application and Database (25% of the module)

- One zipped file called *assignment3_Memberno1_Memberno2_Memberno3.zip* containing the following:
 - A completed version of **assignment3_Memberno1_Memberno2_Memberno3.doc** (template available on Moodle).
 - Complete web application folder containing all PHP, CSS, images etc.
 - MySQL database dump file

This zipped submission to Moodle is essential even if you choose to host your project on knuth and share the link with the lecturer.

Grading Summary

Marking Criteria:

- To pass, Members must present a working and usable application that covers the basic specifications provided above. To receive higher grades, you are expected to use innovation.

Marks will be awarded for:

- Implementing features, Quality of HTML/CSS, Quality of PHP/Database design, Usability, and Innovation.

Commenting and Debugging

- Ensure that you code is well commented so the lecturer can see what is happening
- Include debugging code for SQL statements etc. and then comment them out so lecturer can debug easily if there is an issue.

Overall marks breakdown:

- Initial Document – 15% of the module
- Final Application – 25% of the module

Final application marks breakdown:

- 20% Visual Design: Navigation / Layout / Imagery / HTML / CSS
- 10% Database Design including Normalisation
- 10% System Architecture
- 50% Code Quality
- 10% Usability