

COMP1687 (201516)	Web Application Development	Header ID 227630	Contribution: 75% of course
Course Leader: Dr Kevin McManus	Release Date Saturday 03/10/2015		Deadline Date Saturday 05/12/2015
This coursework should take an average student who is up-to-date with tutorial work approximately 38 hours			
Feedback and grades are normally made available within 15 working days of the coursework deadline			
Learning Outcomes: Use client-side technologies for building, usable, accessible, standard compliant web pages. Use server-side technologies for building secure, stateful, database driven web applications.			

Plagiarism is presenting somebody else's work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student's coursework; stealing or buying coursework from someone else and submitting it as your own work. Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University.

All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using.

Your work will be submitted for electronic plagiarism checking. Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.

Coursework Submission Requirements

- An electronic copy of your work for this coursework must be fully uploaded by midnight on the Deadline Date of Saturday 05/12/2015 using the link on the coursework Teachmat page for COMP1687.
- For this coursework you must submit a single Acrobat PDF document.
- In general, any text in the document must not be an image (ie must not be scanned) and would normally be generated from other documents (eg MS Office using "Save As .. PDF"). More details are on the [IT Support pages](#). An exception to this is hand written mathematical notation, but when scanning do ensure the file size is not excessive.
- There are limits on the file size (current values are on TeachMat and the Student Intranet).
- Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.
- Your work will be marked online and comments on your work and a provisional grade will be available from the Coursework page on Teachmat. A news item will be posted when the comments are available, and also when the grade is available in BannerWeb.
- You must NOT submit a paper copy of this coursework, or include the Banner header sheet.
- All courseworks must be submitted as above. Under no circumstances can they be accepted by academic staff

The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences. See <http://www2.gre.ac.uk/current-students/regs> for details.

Detailed Specification

This assignment is worth 75% of the total marks for this course.

This coursework must be completed individually.

Please read this *entire* specification very carefully so that you are fully aware of the requirements.

You are to create a simple freecycle web site for the Royal Borough of Greenwich. Visitors to the site will be able to register with the site as members. Freecycle members can post information about items they wish to donate and search for items posted by other members.

To implement the site you **must** use XHTML 1.1, CSS and JavaScript for the client-side coding. HTML5 extensions are permitted but should be identified and justified in the documentation. PHP **must** be used for the server side coding. The site **must** run from the Unix Apache web server `stuweb.cms.gre.ac.uk` and the MySQL database server `studb.cms.gre.ac.uk` provided by the department.

In completing this coursework it is recommended that you *strictly adhere to the specification* and *keep it simple*. You are required to produce web sites that largely pass W3C XHTML 1.1 validation using <http://validator.w3.org> noting exceptions, for example arising from the informed use of HTML5 elements. You are also required to produce web sites that comply with W3C web content accessibility guidelines at least to WCAG 1.0 priority 1 and WCAG 2.0 success criteria level A. Passing such validation is easier with simple pages. When designing your web pages you are expected to give serious consideration to usability and how CSS and JavaScript can be used to enhance usability. Your sites are required to display and operate correctly on all popular web browsers, i.e. Mozilla Firefox, Chrome, MS IE, and so on, and on all platforms; desktop, laptop, tablet and cellphone. Client side (JavaScript) and server side (PHP) scripts **must** be used to validate input data from **all** forms. Client side validation may be supplemented with HTML5 elements. Your site **must** operate correctly with and without JavaScript and CSS. Please take time to read carefully the grading and assessment criteria that follow.

Functionality to be achieved

This is expressed as a number of levels. The level of functionality implemented in your application will determine the **maximum possible** mark that you can achieve. The actual mark awarded depends strongly on the quality of your work. Make sure that you fully understand the grading and assessment criteria.

It is recommended that in designing your websites (and databases) you should allow for all of the features to be implemented. In building the websites each level should be attempted in increasing order. Starting with level 1 you should incrementally enhance your work to include the next level.

Level 1 : Account creation : 18 marks

Create an XHTML form that allows visitors to create an account as a member. This form must require only 4 pieces of information from the applicant; their chosen username, their chosen password, their email address and a CAPTCHA test. Account details are to be stored in your MySQL database. On successfully completing this form the applicant must be immediately presented with the activation form (level 2).

The system must prevent duplicate usernames being chosen. Do not use the email address as a username (to avoid spamming). Newly created accounts must remain inactive until they are verified by handshaking the email address (level 2).

Note: Authentication credentials should be protected from interception in transit. member passwords should be stored in the database in an encrypted format. You may use one of the many open source CAPTCHA systems or write your own (beware reCAPTCHA is less than friendly and becomes clunky when you turn off JavaScript).

Level 2 : Account activation : 12 marks

Accounts must remain inactive until the email address provided in the registration form (level 1) can be verified. This is achieved by sending a message containing a 5 character activation code to the email address. You are to create an XHTML account activation form, providing only a single input for the applicant to enter the activation code retrieved from their email. Members should not be allowed to make use of the site's member facilities until they have activated their account.

Note: The stuweb.cms HTTP server is configured to send email, your own server may not be. Keep the email simple plain text, some users may not be able to, or not wish to click links in emails. This form should not require applicants to re-enter information that they have previously entered (this must persist somehow, perhaps as session data). This page should deny access to visitors who have not applied to be a member using the form that you created in level 1. If the correct information is provided the member should be immediately logged into the system and not be expected to authenticate.

Level 3 : Authentication : 10 marks

Provide an XHTML login form that allows returning verified members to authenticate with the site using their username and password. These credentials should be compared with the information recorded in your MySQL database.

Users who have already applied to be members but have not yet verified their account and attempt to log in using this form must be presented with the verification form you created in level 2.

Note: You will need to initiate some form of session state to prevent unauthorised access to member activity within the site. Authentication credentials should be protected from interception in transit. You are expected to implement a logout mechanism. This will assist your testing of the login process.

Level 4 : Member post : 12 marks

Provide XHTML forms that allow authenticated members to post information about themselves (e.g. name, location) and the items they have to offer (adverts). These forms should allow members to upload structured and unstructured information. The system must provide for editing and deleting of all information. Adverts should be structured to include a brief heading (item title) and a more complete multi-line description. This information must be stored with a posting date and be editable by the member who posted it and nobody else. Members should be able to delete items they have listed (to remove them once they have been freecycled).

Note: Editing information is not the same as re-entering information, the member may only be seeking to correct a spelling mistake and so should not be required to re-enter complete data. Some of this data is best handled with HTML form elements other than input type text. Consider carefully the most appropriate form element to gather this user input. Remember that some characters (notably the apostrophe) can cause problems with your SQL strings.

Level 5 : Image upload : 10 marks

Provide XHTML forms that allow authenticated members to upload images to accompany their postings. The system must allow for multiple images to be uploaded (not necessarily all together, one at a time may be simpler) against each member's service with a means of deleting or replacing uploaded images.

Note: Images may be stored as either files on the server or as records in the MySQL database. Remember that images need suitable alternate text content when included in a web page (the file name is seldom appropriate as alternate text). Do not force the member to upload an image, each item may have zero or more uploaded images. It may be helpful to resize uploaded images before storing. It may be useful to generate and store thumbnail images (see level 6).

Level 6 : Item search : 10 marks

Provide a means for casual visitors to search for items. Search results must be returned in a paginated list format where each entry in the list can be clicked to show full details including images. Search results should be filterable to; search headings only or headings and full text, exclude items without images, and sort by, for example, price or distance from the visitor.

Note: A casual visitor should not be expected to authenticate with the site. You should not expect search terms to be an exact match. Consider carefully how you may sensibly match to location. Consider how location could operate with Google maps to implement, for example, a search by radius around a specified location. Result lists may become lengthy and therefore must be paginated. Make sure that you have sufficient items in your database to demonstrate pagination. You may wish to include thumbnail images in the paginated list results.

Level 7 : Cookie : 6 marks

Use a cookie to remember the member's username but not their password, so that they do not need to re-enter their username when re-visiting the site. UK sites that store cookies must conform to EU cookie law (e-privacy directive). Use a cookie to remember previous search terms and use these to enhance the search facility.

Note: This could be implemented using either client side or server side code. While cookie handling is arguably implemented rather better in PHP than JavaScript, you must bear in mind that the cookie is stored on the client and server side manipulation of a cookie can therefore be problematic.

Level 8 : Report : 16 marks

A brief report documenting your implementation of levels 1 through to 7 of this specification as detailed in the following section under Deliverables A.

Note: This report should contain sufficient information to allow another developer to quickly understand and maintain your work.

Use of tools

You are expected to use web authoring tools such as EditPlus or NetBeans to aid your productivity. If you wish, you may make use of WYSIWYG tools such as Dreamweaver or Expression Web. Do not become distracted into spending valuable time on the appearance of your work or gold plating the specification. Be careful when using code generators that you understand the code that is being generated.

Remember that your application must operate correctly in a range of desktop, pad and mobile browsers such as Mozilla, Chrome, Android and Internet Explorer, with and without JavaScript enabled and as far as possible in a text only browser such as Lynx (no CSS, no images, no JavaScript).

Borrowed material

In creating your websites you are expected to borrow code, text content, images and so on. Be careful when using borrowed code such as PHP or CSS frameworks (e.g. CodeIgniter, Fusebox, Baseguide, Bootstrap) that you understand the code and it functions correctly from the specified deployment server. All borrowed material *must* be clearly identified. Include comments in your source code to clearly identify what code you have borrowed and where you borrowed it from (even if you have adapted the code for your own use). Your code sources must also be identified in your report. Referencing code sources in your report is not sufficient on its own. Copyright *must* be acknowledged where appropriate. Failure to correctly reference your sources may be considered as plagiarism.

Deliverables

- A. On Thursday 26th November there will be a peer assessment exercise in which students working in groups of three will assess each others implementation of levels 1 through to 5. A peer assessment sheet is attached to this document, this will be provided in the exercise. This sheet must be completed by you and your peer assessors and handed to your tutor during the peer assessment exercise. Your tutor may ask to see a short demonstration to confirm the accuracy or otherwise of the peer assessment. Your tutor

may decide to moderate the peer assessment. Marks from this assessment will contribute to your final overall grade as described below under Grading Criteria.

Participation in this exercise is compulsory. Non-participation will inevitably result in losing possible marks. Your application is expected to be deployed on the specified web and database servers for the purposes of this exercise. Localhost implementations will inevitably result in a loss of marks.

- B.** A short report as described in level 8 submitted by the due date. This report must be a single PDF file containing the following sections **IN THE ORDER** given below. Do not include any other information. Do not include all of your source code.
1. A completed self assessment sheet (see end of this document).
 2. A statement of the functionality that you have achieved as described in the specification. If you have not achieved all of a certain level then specify the sub-parts of it e.g. all of level 1, 2 and 3 plus some of level 4.
 3. A description of any bugs in your program (all software has bugs!). Bugs declared in here will lose fewer marks than ones that you don't declare!
 4. Reflection on the strengths and weaknesses of your application and development process including discussion of your approach to evaluation and testing. This is your opportunity to draw attention to what you have achieved, why you have done things in a particular way and what you have learned from the process.
 5. Brief design documentation including a *diagrammatic* schema for the database, a list of all files that you have created with a description of what the files are for and supporting UML as appropriate.
 6. Screen shots of your programs in operation. Provide notes with your screen shots to explain how they illustrate the functionality that you have implemented.
- C.** After you have submitted your report you are required to attend a viva (demonstration) to examine your system in operation and answer questions about it. This will be used to both assess the level of functionality and the authenticity of your work. You are required to attend a viva by the end of week 13. Use email to be proactive in scheduling your viva. Failure to attend a viva will result in marks being capped at 30%.

Your tutor will moderate your self assessment (deliverable B.1) during your coursework viva. Marks are available for the accuracy of your self-assessment.

Guide notes on completion of the assessment sheet are included in this document. When completing the assessment sheet you should bear in mind that your tutor is looking for honesty and accuracy.

Be advised that you will be required to set up and run your demonstrations from the specified web server and database server. You should therefore make sure that your work is set up and tested well in advance so that you do not waste time trying to make it work during the demonstration time. You are strongly advised to develop your work directly on the specified deployment servers as opposed to working offline and then porting your work.

Assessment Criteria

Marks are awarded for:

The functionality that you have achieved. Have you achieved all specified functionality or only some? How well have you achieved the functionality? Have you incorporated any features that were not explicitly included in the requirements but add value to the site? Have you added features that contravene the specification? Have you added features that were not explicitly included in the requirements but detract from the usability of the site?

The usability of the application. Is the application easy to use? Is it obvious to the user at each stage what the user needs to do next? Are all messages to the user clear and unambiguous? Is the layout consistent and easy to read? Is navigation through the application clear and straightforward?

The accessibility of the application. Does the application follow WAI and Section 508 accessibility guidelines? Which guidelines have you considered, WCAG 1 or WCAG2? How well does it conform to the guidelines? Priority A or AAA?

The reliability of the application. For example, if it throws an exception every time the user enters invalid input you will lose marks. Faults that you admit to on your bug list (see deliverables) will be looked on more kindly than those that are not declared.

The security of the application. For example, is the database protected from unauthorised access and alteration, is it open to SQL or script injection. Is sensitive data protected in transit? How difficult is it to hack your application? Security holes that you admit to on your bug list (see deliverables) will be looked on more kindly than those that are not declared.

The scalability of the application. For example, is the database appropriately normalised. Will the system be usable with one thousand entries in the database? Will the system be usable with one million entries in the database? Are queries paginated at the database, in the middleware or at the client?

The quality of your code. Have you included meaningful comments, used sensible naming standards (e.g. for variables, functions and files) and code layout (e.g. indentation to make the structure clear). Is the code well structured or a tangle? Have you clearly identified borrowed code with the original source? How many of your pages pass W3C XHTML validation?

Does your application operate correctly on all of the required browsers? Is the page layout elastic, responsive or adaptive? If any features fail on a particular browser, does it fail gracefully or become unusable? Is it usable without CSS? Without JavaScript? Without images?

Appropriate use of technologies, for example, is user data validated on both the client and the server? Has a sensible choice of validation priority been made? Is the validation effective? The specification is intentionally open so that you can decide to a certain extent how to implement each feature.

The quality of the design and of the design documentation. Is the design flexible? Would it be easy to add to or amend the application to support additional functionality? How much of the application is hard coded and how much comes from structured file or database resources?

The quality of the report. Are all the required sections included and completed appropriately? Is the report lengthy and verbose or is it concise but full of significant information? Is the standard of English appropriate? Does the design match the implementation?

You will fail if any of the following are true:

Your code does not run from the required web and database servers.

You do not attend the viva.

You do not electronically submit your documentation by the deadline

Grading Criteria

The specification is given as eight levels including the report. Marks for each of the eight individual levels are provided with the level specifications above, making a total of 94%. The accuracy of your self assessment is worth up to 6%.

Your mark for each of levels 1 to 5 is calculated as the greater of either, the average of the peer mark and final mark for that level, or 75% of the final mark for that level.

Note that the mark you achieve as defined in each level specification sets the maximum possible mark, you may get a mark lower than the maximum possible for each level that you implement depending on how well meet the assessment criteria. Factors that may be taken into account when awarding a grade are described above in the assessment criteria.

The self assessment sheet below requires that you record a grade for each level as a number between 0 and 10. The weighting for each level is applied later.

7 ... 10	1st Class, distinctive, excellent or outstanding in all elements.
6	Upper Second Class, meritorious, good or very good overall standard
5	Lower Second Class, adequate, largely meets the requirements
4	Third Class, pass, acceptable, largely achieves the learning outcomes
3	Compensatable fail, not acceptable, partly achieves the learning outcomes
0 ... 2	Fail, unacceptable, does not achieve the learning outcomes.

Assessment sheets

The peer assessment and self assessment sheets are to be completed by the student. The factors that should be taken into account when completing the assessment are described above in the grading criteria.

The assessment sheets are an eleven point Likert scale requiring a circle to be drawn around one of the records for each row on the sheet.

- 10** – faultless, or at least difficult to criticise
- 9** – exemplary achieves all of the specification to a high standard
- 8** – outstanding, achieves all of the specification
- 7** – excellent, achieves most of the specification
- 6** – good, largely meets the specification but lacking in quality
- 5** – acceptable, a clear pass but barely acceptable, lacking aspects of the grading criteria
- 4** – a bare pass, lacking in essential aspects but not a fail
- 3** – an attempt has been made but insufficient to pass
- 2** – clear fail, some evidence of work but falling way below the required standard
- 1** – some limited evidence of an attempt
- 0** – missing or may as well be, no real attempt made

The difference between these eleven categories should be perfectly clear. You should seek timely guidance from your tutor if you require clarification. Should you find yourself wanting to circle 7 or above then please read ***all*** of the specification document very carefully, submit your pages to validator.w3.org, consider WCAG 1 and 2 and consider what is meant by faultless.

Clearly there is a degree of academic judgement in making any assessment. The assessment sheets are intended to help in making an objective assessment so please consider each record carefully. It is in your interest to be honest and accurate in your assessment.

COMP1687 Peer Assessment Sheet for the 201516 Coursework
This sheet must be completed in the peer assessment exercise

Assessed student: _____ **Student ID 000** _____ **Sign** _____

Peer student : _____ **Student ID 000** _____ **Sign** _____

Peer student : _____ **Student ID 000** _____ **Sign** _____

URL

<http://stuweb.cms.gre.ac.uk/~>_____

Student Use													
Level 1	Account creation	18	0	1	2	3	4	5	6	7	8	9	10
Level 2	Verify account	12	0	1	2	3	4	5	6	7	8	9	10
Level 3	Authentication	10	0	1	2	3	4	5	6	7	8	9	10
Level 4	Member post	12	0	1	2	3	4	5	6	7	8	9	10
Level 5	Image upload	10	0	1	2	3	4	5	6	7	8	9	10

Comments

COMP1687 Self Assessment Sheet for the 201516 Coursework
This sheet must be completed and submitted with your report

Student name: _____ Student ID 000 _____

URL

<http://stuweb.cms.gre.ac.uk/~>_____

Student Use													
Level 1	Account creation	18	0	1	2	3	4	5	6	7	8	9	10
Level 2	Verify account	12	0	1	2	3	4	5	6	7	8	9	10
Level 3	Authentication	10	0	1	2	3	4	5	6	7	8	9	10
Level 4	Member post	12	0	1	2	3	4	5	6	7	8	9	10
Level 5	Image upload	10	0	1	2	3	4	5	6	7	8	9	10
Level 6	Item search	10	0	1	2	3	4	5	6	7	8	9	10
Level 7	Cookie	6	0	1	2	3	4	5	6	7	8	9	10
Level 8	Report	16	0	1	2	3	4	5	6	7	8	9	10
Staff Use													
Self-assessment		6	0	1	2	3	4	5	6	7	8	9	10
Comments													