COMP1687 (2018/19)	Web Application Development	Header ID: 300487	Contribution: 100% of course
Coordinator: Dr Mahtab Hossain			Deadline Date: Thursday 13/12/2018
This coursework should take an average student who is up-to-date with tutorial work approximately 50 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. Describe and critically discuss the design, engineering, legal, social, ethical and professional issues and considerations involved in web application development.			

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 **11:55 p.m.** on the Deadline Date of **Thursday 13/12/2018** using the link on the coursework Moodle page for COMP1687.
- For this coursework you must submit a single Acrobat PDF document. It MUST contain the self-assessment sheet as the first page [just after the cover page]. Your developed "Web Application" should be hosted inside stuweb.cms.gre.ac.uk for access. One single ZIP file of your web application should also be uploaded.
- In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As ..PDF"). An exception to this is hand written mathematical notation.
- There are limits on the file size (the limit for ZIP file has been set to 2 GB).
- 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 Moodle. 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 coursework 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.

Detailed Specification

This coursework is worth 100% 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 web site for Time Banking management system [not a complete system by any means – only with partial specific functionalities as outlined below] for a community, e.g., the borough of Greenwich. A Time Bank is generally a community-run system where the time (e.g., hours) to deliver a particular task/service is the unit of the account – not the currency as in traditional bank. It tries to convert unpaid time into a valuable commodity that is targeted at building social capital, and greater community bonding. The functional requirements of this web site are outlined as: visitors to the site will be able to register with the site as members and provide information about their skills and required services/tasks via posts. Casual visitors to the site will be able to search through the posts to see if any posts are of interest to them. While any visitor can search through the various posts, full details of time bankers (i.e., registered members) and their posted tasks'/services' details will only be available after registering to this web site.

To implement the site you ***must*** use XHTML 1.1/HTML5, 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 mysql.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*. 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

The required functionality is expressed as a number of levels. The 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 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: 12 marks
Create an XHTML/HTML5 form that allows visitors to create a member's account. The form must require only 5 pieces of information from the applicant; their chosen username, their chosen password, their email address, a CAPTCHA string, and their skills [e.g., plumbing, teaching, programming, etc. or even no skills]. Account details are to be stored in your MySQL database. On successfully completing this form the applicant must be presented with the verification 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 details (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 presents usability issues without JavaScript).

Level 2: Verify account: 10 marks
Account verification will require sending a message to the email address provided in level 1. This email message should include a 5 character activation code which enables an applicant to activate their newly created account. Members should not be allowed to make use of the site's member facilities until they have verified their account.

You are to create an XHTML/HTML5 account verification form providing only a single field allowing an applicant to enter the activation code retrieved from their email. On successful verification, an initial time-bank credit of 100 will be given to the newly registered member. Accounts must remain inactive until the correct information is provided.

Note: The stuweb.cms.gre.ac.uk HTTP server is configured to send email. Keep the email simple plain text, some users may not be able to, or not wish to click links in emails. This page should not require applicants to re-enter information that they have previously entered. 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 agent should be immediately logged into the system and not be expected to authenticate.

Level 3: Authentication: 8 marks
Provide an XHTML/HTML5 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 from 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 will find it useful to implement some form of logout mechanism if you are to be able to test this login process.

Level 4: Member post: 10 marks

Provide XHTML/HTML5 form that allows authenticated members to post information about their needed services/tasks. These forms should allow members to upload structured information including the service/task type [plumbing, accounting, baby-sitting, etc.], status [open/completed], skills required [none, programming, math, etc.], service's/task's location, and the credit/point that may be awarded to the person who completes it. The system must allow for members to post one or more service/task posts. The system must provide for editing [for example, after completion, the member may change the status field from open to completed], and also keep provisions for deleting of a post.

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: 8 marks
Provide XHTML/HTML5 form [it can be together with the form that you have created for Level 4] that allows authenticated members to upload images to accompany their post (e.g., images related to the tasks/services required, e.g., broken boiler, clogged kitchen sink, etc.). The system must allow for multiple images to be uploaded (not necessarily all together, one at a time may be simpler) against each post 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 member may have zero or more uploaded images.

Level 6: Member search: 10 marks
Provide a means for casual visitors (i.e., not registered) to enter a skill or location or combined [skill+location] to search for matching or nearby tasks'/services' posts. Search results must be returned in a paginated list brief format where each entry in the list can be clicked to, if they are registered as a member, show full details of the matching commuter, otherwise (if not registered) redirect to the level 3 login form.

Search results should be filterable to sort by, for example, Euclidean distance from the casual visitor's input location, exclude posts without images, etc.

Note: A casual visitor should not be expected to authenticate with the site. This form can attract members so should be clearly visible from the home page. You should not expect search terms to be an exact match. Consider carefully how you may sensibly match to location. Result lists may become lengthy (e.g, searching with an empty string may return all the existing entries of your database), and therefore must be paginated. Make sure that you have sufficient items in your database to demonstrate pagination.

Level 7: Cookie: 6 marks

Use a cookie to remember the member's username but not the password. In addition use a cookie to remember the last search term. Sites that store cookies must conform to EU cookie law (e-privacy directive).

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: Usability: 6 marks
The developed application should incorporate the contemporary issues of Web development in terms of user experience (i.e. usability). For example, is it responsive/adaptive to address device heterogeneity issues for access? How is the Web interface's information architecture in terms of finding relevant items and navigation?

Level 9: LSEPi consideration: 25 marks
You are to write an article of approximately 2,000 words addressing the Legal, Social, Ethical and Professional issues and considerations (LSEPic) that arise in the specification and implementation of this "Time Banking" Web application. You may wish to extend your discussion to include Political, Philosophical and Economic issues and considerations (PPEic). You are advised to only address these issues in the context of the functionalities that you are required to address for this web application.

You must choose a suitable title for your article. Your article is expected to provide an informed discussion of LSEPic and PPEic arising in this application and include reflective discussion specific to your implementation of this application. This is a short article and there is a great deal that could be discussed, so it is better to provide an in-depth discussion of one or two aspects than to attempt to cover many considerations.

This article must be suitably structured, written in your own words, and develop a clear narrative or argument using appropriate language. All assertions are expected to be supported by references (using Greenwich Harvard formatting) or otherwise justified.

Your essay is required to provide 2,000 words plus or minus 5% (100 words) not including the title page or any preamble and not including the references. A penalty will be applied for word counts that are outside these limits.

If you are in any way unclear about this specification you should discuss this with your tutor.

Use of tools

You are free to use web authoring tools such as Brackets 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.

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 submission. Referencing code sources in your submission 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 22nd November** there will be a peer assessment exercise in which students working in groups of three will assess each other's implementation of levels 1 through to 3. 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. Your final mark for each of levels 1 to 3 is calculated as the ***greater*** of either the peer assessment mark or the final mark of your demonstration with your tutor which will be scheduled after your coursework submission.

B. A PDF document submitted by the due date 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 cover page.
2. A completed self assessment sheet (see end of this document).
3. 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.
4. 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!
5. Level 9's article.

Your developed "Web Application" should be hosted inside stuweb.cms.gre.ac.uk for access. One single ZIP file of your created web application should also be uploaded via Moodle submission page.

C. After you have submitted your coursework, 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. Failure to attend a demonstration will result in loss of marks for levels 1 to 8. You will be sent an email to reserve a demonstration slot with your tutor after the coursework upload.

Your tutor will moderate your self assessment (deliverable B.2) 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

In terms of the developed Web application (Level 1-8):

The 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 though the application clear and straightforward?

The accessibility of the application. Does the application try to follow WAI and Section 508 accessibility guidelines?

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?

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.

You MUST ensure the following:
Your code should run from the required web and database servers as mentioned in this specification.
Attend the demonstration viva with your tutor.
Submit the coursework documentation by the deadline electronically.

In terms of the submitted LSEPi article from Web development perspective (Level 9):

Marks are awarded in ***equal*** measure for the following ***five*** criteria.

The quality of the language used in the article. Is the language clear and unambiguous or is it difficult to follow, with poor sentence structure and grammatical errors? Is the language at an appropriate level using a technical vocabulary or is it too simplistic or overly familiar?

The structure of the article. Is the text in a single paragraph or is it organised into sections and subsections? Are the paragraphs and sections sensibly chosen? Is the text overly compartmentalised? Is there a narrative or argument or is it merely a collection of facts and assertions? Is the content contextualised or just a collection of bullet points?

The quality of the content. Is there adequate discussion of legal, social, ethical and professional matters or perhaps only one of these four? Are they discussed superficially or does the discussion have depth and demonstrate understanding of how these aspects are interrelated? Are more interesting or important aspects discussed?

The scope of the discussion. Does the account provide generic, specific and reflective discussion of the subject or only one of these three? Are these discussed separately or contextualised to demonstrate understanding and insight?

The academic standard of the writing. Is the essay entirely original or is there evidence of pasted content? Are citations provided to support facts and assertions? Are references provided to match the citations? Are all references cited in the essay? Are the references appropriately formatted?

Grading Criteria

The specification is given as nine levels including the article. Marks for each of the nine individual levels are provided with the level specifications above, making a total of 95%. The accuracy of your self assessment is worth up to 5%.

Your mark for each of levels 1 to 3 is calculated as the ***greater*** of either the peer assessment mark or the final mark for that level. The rest of the levels' (level 4-9) marks will only come from the final mark of your submitted work.

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, outstanding in all elements.
6	Upper Second Class, meritorious, 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, achieved some learning outcomes
0 … 2	Fail, does not meet level 6 undergraduate degree standard.

Assessment sheets

The 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. If you find yourself wanting to circle 7 or above then please read ***all*** of the specification document very carefully.

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 201819 Coursework

Assessed student: _____ **Student ID 00** _____ **Sign**_____

Peer student : _____ **Student ID 00** _____ **Sign**_____

Peer student : _____ **Student ID 00** _____ **Sign**_____

URL

`http://stuweb.cms.gre.ac.uk/~`_____

Student Use													
		Total Mark	Please circle ONE of the grades (0 to 10) for each level below [Level 1 to Level 3]										
Level 1	Account creation	12	0	1	2	3	4	5	6	7	8	9	10
Level 2	Verify account	10	0	1	2	3	4	5	6	7	8	9	10
Level 3	Authentication	8	0	1	2	3	4	5	6	7	8	9	10
Comments													

COMP1687 Self Assessment Sheet for the 201819 Coursework
This sheet must be completed and submitted with your coursework

Student name: _____ Student ID 00 _____

URL
`http://stuweb.cms.gre.ac.uk/~`_____

Student Use													
		Total Mark	Please circle ONE of the grades (0 to 10) for each level below [Level 1 to Level 8]										
Level 1	Account creation	12	0	1	2	3	4	5	6	7	8	9	10
Level 2	Verify account	10	0	1	2	3	4	5	6	7	8	9	10
Level 3	Authentication	8	0	1	2	3	4	5	6	7	8	9	10
Level 4	Post	10	0	1	2	3	4	5	6	7	8	9	10
Level 5	Image upload	8	0	1	2	3	4	5	6	7	8	9	10
Level 6	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	Usability	6	0	1	2	3	4	5	6	7	8	9	10
Staff Use													
Self-assessment		5	0	1	2	3	4	5	6	7	8	9	10
Level 9	LSEPi article	25											
Comments													