

Course: ICT461 Web Systems and Technology		Contribution: 50% of CA
Course Lecturer: Dr Sinyinda Muwanei	Due: 10/10/2023	
This coursework should take an average student who is up-to-date with tutorial work approximately 50 hours		
Learning Outcomes: A Demonstrate familiarity with open source web servers and web clients. B Use open source client-side technologies for building web sites. C Use open source server-side technologies for building multi-server, multi-tier web applications. D Select and employ appropriate open source technologies for the development of web applications. E Describe and critically discuss the issues surrounding the design and implementation of 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 placed in quotation marks and in italics, with a full reference to the source directly underneath the material.

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 should be fully uploaded by midnight (local time) on the Deadline Date.
- The last version you upload will be the one that is marked.
- 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").
- For this coursework you must also upload a single ZIP file containing supporting evidence.
- There are limits on the file size.
- Make sure that any files you upload are virus-free and not protected by a password otherwise they will be treated as null submissions.
- Comments on your work will be available from the Coursework page on the Intranet. The grade will be made available on Moodle
- You must NOT submit a paper copy of this coursework.

Coursework Regulations

1. **If no submissions were made before the deadline, coursework submitted up to two weeks late that meets the criteria for a pass will be treated as a referral. It will be subject to university regulations for referral work.**
2. **Coursework submitted late without an Extenuating Circumstances claim will receive a ZERO grade.**
If you have extenuating circumstances you may submit your coursework up to two weeks after the published deadline without penalty but this is subject to acceptance of your claim by the School Extenuating Circumstances Panel. If your claim is rejected then you will receive a zero grade for your work.
3. **Coursework submitted more than two weeks late will be given feedback but a grade of non-submission will be awarded regardless of any extenuating circumstances. However, if your Extenuating Circumstances claim is accepted then the Extenuating Circumstances Panel will recommend to the Progression and Award Board that you be permitted to retake a different item of assessment at a future assessment point.**
4. **All courseworks must be submitted as above.**

Detailed Specification

This coursework must be completed individually.

You are to create a web site for an apple farm.

John Smith has been a farmer for a number of years and he has been using an offline inventory system to manage his day to day farming business. The offline software system that he is currently using allows to record sales and keep track of his apple fields but that's about it. Having heard of the wonderful online solutions that are available to he has decided to hire you to build a better system for him.

His requirements in a nutshell are:

- He plants, grows, harvest and sell fifty three apple varieties – he needs to use the website to allow customers to see the available varieties of apples and express interest.
- As more than one customer can express interest for an apple variety, customers are to be served on a first come first served basis.
- Customers can be new or returning customers (never registered or registered with the site/have bought produce before). Apple produce is collected onsite (cash payment) although there is thought for a future option that would allow shipping of produce.
- Visitors to the site should be able to search and browse produce varieties for items on offer.
- Only registered users should be able to use the site and register interest for apple produce.
- The farmer should be able to search through the sales history.

To implement the site you should use HTML 5. The site *must* run from a web server and make use of a MySQL database.

In completing this coursework it is recommended that you *adhere to the specification* and *keep it simple*. You are required to produce web sites that pass HTML 5 validation using <http://validator.w3.org>. You are also required to produce web sites that pass at least the Priority 1 W3C web content accessibility guidelines (Cynthia Says at <http://www.contentquality.com/>). Passing such validation is easier with simple pages. When designing your web pages you should give serious consideration to security, usability and how CSS and JavaScript can be used to enhance usability. Your sites are required to display and operate correctly and consistently on all popular web browsers, i.e. Microsoft Internet Explorer, Mozilla, Firefox, Opera, etc. Client side (JavaScript) and server side (PHP) scripts should be used to validate input data from all forms.

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 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

Create an HTML5 form allowing visitors to create an account with the site.

The form must mandate basic contact details but should also allow for personal information to be included (think facebook). Account details are to be stored in your MySQL database. Information should be stored in a secure way.

Visitors should be able to choose a username and password. The system must prevent duplicate usernames being chosen. Newly created accounts should remain inactive until they are verified by handshaking the email details (Level 3).

Note: Remember that some characters (notably the apostrophe) can cause problems with your SQL strings. The member passwords should ideally be stored in the database in an encrypted format. Authentication credentials should be protected from interception in transit.

Level 2 : Authentication

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

Note that you will need to initiate some form of session state to prevent unauthorised access to further activity. New members will also need to verify their account details (Level 3) before being allowed access to further activity with the site.

Level 3 : Verify account

Account verification will require sending a message to the email address provided in Level 1. This email message should include some sort of key that enables a member to activate their newly created account after they have authenticated with the site. Members should not be allowed to make further use of the site until they have verified their account.

Members should only be required to verify their account the first time that they authenticate.

Level 4 : Editing details

Provide HTML5 forms that allow authenticated members (who have verified their account) to add to and edit existing information stored about themselves.

Note: Editing information is not the same as re-entering information, the user may only be seeking to correct a spelling mistake and so should not be required to re-enter complete data. The simplest way to implement this is to re-use the form that you created in Level 1 but populated with default values extracted from your database.

Level 5 : Offer apple varieties and express interest

Provide HTML5 forms that allow our farmer to advertise apple varieties he grows - These forms should allow to upload images associated with the produce. Images may be stored as either files on the server or as records in the MySQL database.

Level 6 : Allow customers to express interest on an apple variety

These forms should allow members to be able to express interest on an advertised apple variety. The system should allow for editing and deleting of an offer. An offer can be in the form of “I am interested in 500 kilos of apples variety type 1 and I am willing to offer £1000 pounds”. These offers are to be sorted displayed based on price per kilo.

Level 7 : Keyword Search

Provide a means for our farmer to search the order history of all orders placed in the system. Search results should be initially returned in a brief list format where each entry in the list may be clicked to show full item details. Result lists may become lengthy and should therefore be paginated.

Note: A casual visitor should not be expected to authenticate with the site. Obviously this means that he/she cannot make an offer for produce.

Use of tools

You are expected to use web authoring tools to aid your productivity. If you wish, you may make use of WYSIWYG. Be careful when using code generators that you understand the code that is being generated and that the tool that you use doesn't insert components that require specific client or server support.

Remember that your application needs to run in a range of browsers such as Mozilla, Netscape, Opera and Internet Explorer and that the servers that your sites run on may not have FrontPage server extensions installed.

Borrowed material

You are expected to borrow code, text content, images, etc. in creating your websites. All borrowed material *must* be clearly identified and copyright *must* be acknowledged where appropriate. Failure to correctly reference your sources may be considered as plagiarism!

Deliverables

- A. You will be required to give a demonstration of your system in operation and to 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 complete the self assessment sheet at the end of this document and submit it the beginning of your report. We want to see that you can evaluate your product and that you understand what level of functionality you have achieved. After you have submitted your report you will be asked to demonstrate your work to your tutor.

If you don't demo your work you will automatically fail this coursework.

You should 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 and test your work as opposed to working offline and then porting your work.

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

- B. A report submitted electronically consisting of the following sections **IN THE ORDER** given below. This report should contain sufficient information to allow another developer to understand and maintain your work.
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. The report should contain clear evidence of testing, including a test plan and a test log
 4. A list of any bugs in your program (if you don't think there are any then say so!).
 5. A brief (up to 300 words) reflection on the strengths and weaknesses of your program. This is your opportunity to show the tutor what you have achieved, why you have done things in a particular way and what you have learned from the process.
 6. Brief design documentation including diagrammatic schemas for all databases used and a list of all files that you have created with a description of what the files are for.
 7. Screen shots of your programs in operation. Include notes with your screen shots to explain how they illustrate the functionality that you have implemented. Note that it is more appropriate to have all screenshots numbered in the appendix of your report.

Grading Criteria

The specification is given as seven levels.

- Levels 1 - 7 worth 7% each
- The documentation (report – deliverable B) is worth 51%

Note that the level you achieve as defined in the specification sets the **maximum** possible mark. The other factors that will be taken into account when awarding a grade are described below in the assessment criteria. Remember you may get a mark **lower** than the maximum possible for the level you implement depending on how well your assignment meets the assessment criteria.

Your tutor is there to guide you in completing your coursework but not to implement it for you. You should seek regular feedback from your tutor to ensure that you are progressing in the correct direction at the correct speed.

Assessment Criteria

Marks are awarded for:

- The functionality that you have achieved. Have you achieved all required 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?
- User interface design and overall 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, attractive and easy to read? Is navigation through the application clear and straightforward?
- Does the application follow WAI accessibility guidelines? Is it secure?
- Reliability of the application. For example, if it crashes 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.
- Quality of the code. For example; inclusion of meaningful comments, use of the 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 an unstructured tangle? Most importantly, how many of your pages pass W3C validation?
- Does your application operate correctly on all of the required browsers? If any features fail on a particular browser, does it fail gracefully or simply crash?

- Appropriate use of technologies, for example, is user data validated on both the client and the server? The specification is intentionally open so that you can decide to a certain extent how to implement each feature.
- 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?
- 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 adequate?

You will not pass if any of the following are true:

- Your code does not run.
- You do not attend the demonstration.
- You do not electronically submit your documentation by the deadline

Assessment sheets

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

The assessment sheets require a circle to be drawn around one or more of the following records for each row on the sheet.

none – missing or may as well be, no real attempt made

poor – fail, some evidence of work but falling way below the required standard

weak – a bare pass, lacking in essential aspects but not a fail

ok – acceptable, largely meets the specification but lacking in quality

good – first class, meets all of the specification to a high standard

excellent – faultless, difficult to criticise, outstanding work

The difference between these six categories should be perfectly clear. You should seek timely guidance from your tutor if you require clarification. Try where possible to refine your assessment by perhaps circling two adjacent records.

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.

Student name: _____ **Student ID 000** _____

By D.F Page 10 13/10/2023