|  |  |
| --- | --- |
| Assignment summary: Ten Green Bottles Non-Alcoholic Wine Shop | |
| What is it? | The assignment is to create a working “web application” that fulfils the specification on these pages. *Please read this document IN FULL.* |
| How is it assessed? | The final submission (including the online, functioning site and the archived code) will be marked, and you’ll also have a group “mini viva” (demonstration and Q&A).  You will also be asked to give an indication of how much your partner has participated in the process, and I will also factor in my own view of your participation throughout the coming weeks. |
| When is it due? | The final hand-in date is **17th March 2017**. As noted above, however, your group’s viva may well be some time before this! You will not be able to engage in last-minute scheduling here! **Assume that the practical deadline might be as much as TWO WEEKS before this date!** |
| What do we need to hand in? | The URL where your work is hosted.  A ZIP file containing all the files that comprise your application.  You may also decide to include a SQL database dump in case the marker needs to get your work up and running on an alternate server. |
| How much is it worth? | 25% |
| Who’s in my group? | You can choose your own pairs and you will need to advise me who you are working with. Ideally, you should continue to work in the pairs from the earlier coursework in Teaching Block 1. If you do not notify me of your pair you will be assigned to a pair by me. |
| Can I work on my own? | No\*. When you leave here and go into the real world, you will often be expected to work in teams. Consider this coursework training for this aspect of real life! You work in a pair or you get zero. End of discussion. |
| How can we get 100%? | By giving 100%. Participate every week both in your group and by attending scheduled classes, complete all of the stages described below, meet all of the requirements fully using good, well-documented code, and/or do some “bonus” work. |
| What help can we get? | There is a workshop every week! This is your main help forum! I expect to see EVERYONE at workshops. If you haven’t been turning up to workshops, don’t bother asking for help by email – I won’t be turning up via email for people who don’t turn up in person!  Don’t forget that fellow group members and class mates are a valuable resource – *use them*!  Only use code from a 3rd party if you document its use fully (with references) – you must be able to explain your code in the “mini viva”. |

\* unless you have an appropriate SOSN. (If you don’t know what a SOSN is, you don’t have one!)

# Project procedure

Your group project assignment is to create a database-driven ‘web application’ for a (hypothetical) “client”.

1. A suggested starting (or “common”) schema for the database will be given to you on StudySpace. You can use (and expand upon) this, or you can use the schema from your Oracle work.
   * Implement your chosen schema as a database in MySQL, paying particular attention to the “core” tables
   * If you expand upon the “common” database model further, please describe the changes with justification in your report.
2. Implement the web-based system using HTML5, CSS, PHP, JavaScript and/or JQuery and MySQL to the best of your group’s ability.
   * You may **NOT** use
     + CSS layout frameworks (e.g. Bootstrap)
     + Third party ORM frameworks (although you may write your own ORM/DAO classes; in fact if you are separating your concerns properly you *should* be doing this anyway!)
     + Third party MVC frameworks (although you may choose to adopt MVC as a design pattern that informs the structure of your own code – in fact if you are separating your concerns properly you *should* be doing this anyway!)
     + Off-the-shelf content management systems
     + **IF YOU BREAK THESE RULES, YOUR MARK WILL BE CAPPED AT 50%.** If you are unsure whether a piece of technology is “legal”, ask me first!
   * Your web interface (client *and* server) should *validate* the input of data wherever possible to protect the database (and server) from invalid input.
   * The client-side code should be *valid* HTML5and *CSS*.
   * Bonus marks may be given for good database design techniques, good, accessible web design practices, nice use of CSS, JavaScript, JQuery *etc.* so if you satisfy the overall requirements rapidly, spend some time polishing the details. (But bear your other modules in mind – the top mark is 100%.)
   * You are free to use KUNet (with PHP) *or* your own web server and hosting space but the latter is your responsibility to maintain (particularly when assessment is taking place – if the server is down your work can’t be marked).
   * During the “mini viva” all groups will be asked to edit code on their live site to demonstrate mastery of the MySQL, web server and client-side environments.

## Additional information

Be aware of the University’s rules on plagiarism. Electronic submission of your work amounts to a declaration that it *is* your work and *everyone* should submit work to StudySpace.

## Marks breakdown

The project is 25% of the overall module assessment marks. There is an additional sheet which breaks down how the marks within this assignment are allocated.

# Business requirements outline:

The first coursework specification explained in detail the requirements of a superset of the system (in order to fully specify the database) so this is not repeated here – please refer back to it after reading the following outline.

One small change has occurred in TGB’s[[1]](#footnote-1) business model: they have decided to refocus on non-alcoholic wines. They believe there is a large untapped business opportunity in this area as there are many people who enjoy the taste of wine but object to alcohol for various reasons such as health, religious/moral or others. TGB intend their new web application to be the launch platform for their new alcohol-free business model.

Ten Green Bottles would now like you to implement the customer-facing web application intimated in coursework 1. However, due to financial difficulties they can no longer afford the costly licence fees for Oracle and have imposed the non-functional requirement that the database backend must be MySQL. The web application must be written in PHP, CSS, HTML, Javascript and JQuery. Third party frameworks other than JQuery may NOT be used. Any contravention of these non-functional requirements will result in the company rejecting the work (and your mark will be capped at 50%).

Your application should include the following basic functionality (these are the MoSCoW “must haves”)

**Basic Functions**

* + - 1. Customers can browse and search for wines in across several categories and subcategories. In this company’s world, wines come in red, white and rose varieties. White wines can be dry or sweet, red wines can be light or full-bodied.
      2. Customers can add one or more items to a shopping basket, without necessarily logging in
      3. Customers can go through and complete an order. The system should lead the user through supplying the appropriate information required to fulfil their order, and finally inform them that their order has been received. (Handling actual payment e.g. integration with credit card APIs is not required in this prototype).
      4. Administrators can add a new wine, update an existing wine’s details (e.g. name, categories, description, amount in stock, etc), or delete a wine from stock (and thus remove it from what the customer sees)

If you are successful in implementing the basic functions, the company has also asked for the following extended functions to be implemented (consider these the MoSCoW “should haves”)

**Extended Functions**

1. Customers have some mechanism for being notified when new wines are added to the catalogue – e.g. through an RSS feed, an (automatically updated!) “new wines” section, or email notifications.
2. The site uses AJAX to enhance interactivity. NB: AJAX involves both server and client side code! Just using a bit of Javascript to manipulate the page is NOT AJAX!
3. Customers can create a wishlist that persists across sessions. (You could store this client side, e.g. in cookies or web storage, or server side within the database.
4. Administrators can create promotions that automatically appear on the home page to customers, including expiry dates.

100% is possible by creating a “modern web application” that fulfils the “core” functions (1−4 inclusive) **and** *one* of the “extended” features (5−8) described above. **To get 100% everything would need to be perfect, i.e. the application provides a flawless user experience that conforms to each and every single concept and technique taught in the module**. (In the exceptional case where there might a group of three people, you will need *two* of the “extended” features)

As perfection is not always possible, an imperfect app can top-up and approach 100% through the addition of further “extended” features. You can also make a case that something you have done that is not listed above should count as an extended feature – discuss this with Paul.

To facilitate individual working, the “core” thing that should be created immediately is the MySQL *database*; this should be done as a team. Once the database is implemented and the functional requirements have been analysed it is possible for various requirements to be done independently, facilitating a more individual approach to “group” working.

# Project submission:

Please submit your work to the module coursework 2 drop box on StudySpace by the deadlinebut feel-free to submit *earlier* if you’re ready! (Drop me an email if that’s the case, we may be able to bring the “mini viva” forwards.)

1. Create a Word document in which you should put the URL where your application can be run, any logins/passwords required to access the various parts of your site, and anything else you might think the assessor will need. Remember: it is up to you to make sure we have everything we will need to ensure we mark your work fairly! If (for example) we cannot log into your admin area then your work will be assessed as if there was no admin area!
2. ZIP all of your code – PHP, CSS, HTML, Javascript, etc – everything that comprises your application, add the Word document to the archive and then submit the archive file to the drop box.

# Project groups:

You have until week 3 to confirm your pairs. After that I will assign people into groups using my own algorithm (which will probably involve a class list and darts ☺)

Your group should meet together at mutually agreed times to work on the project – I recommend you exchange mobile numbers ASAP!

You must have weekly meetings to discuss progress and ***take notes* *every time****.* I’ll ask for these notes if problems arise. It may be helpful to mutually sign-off all the notes at each meeting.

This procedure will be followed in the event that your group is not working together smoothly:

* In the first instance, you should try to discuss and deal with the problem in your group.
* If you cannot resolve the problem through discussion within your group, a group member should email me notifying me of the problem. You MUST notify the other member(s) of the group that you are escalating the issue to me.
* The group will be invited to a joint meeting with me in my office to discuss the issue. I will:
  + Ask about your discussions prior to escalating your issues to me
  + Ask to see the notes from your weekly meetings
  + Offer suggestions on how to resolve the issue
  + Potentially, give mandatory tasks to group members which MUST be performed in order to break any deadlock
  + Potentially, reallocate group members to different groups
  + In extreme cases, I may choose to deduct marks from group members where the situation could have been avoided. IT IS THUS IN YOUR INTEREST TO TRY TO MAKE YOUR GROUP WORK!

Most of you have done group project work before so I expect things to progress smoothly. I do NOT intend to reallocate group members except where the group dynamic has broken down to such an extent that it is irredeemable. If I end up having to reallocate group members, I will re-allocate according to the effort in evidence. An individual who has been putting in the effort but chafing against a partner who is not contributing will be placed with another high-performing group. An individual who has not been putting in the effort will be placed with, shall we say, a “weaker” group.

**THERE WILL BE NO FREE RIDES ON MY WATCH! ☺**

# Group marks

The project is worth 25% of your final module grade. You are asked to rate how much effort you believe each member put into the project (so on average/ideally each member of a group of 2 would receive 50%.)

**NB:** Those who do not submit a report, web site and site URL with their project risk getting 0% so please pay attention when submitting your work…

**“Everybody submits, no one forgets: If you don’t submit your work you’ve given 0% to yourself!”**

**NB:** In the event that a group member has made *no discernible contribution* to the group’s work he/she will receive zero marks, irrespective of what any calculations might say!

Good luck and I hope you enjoy the assignment!

Paul

1. I will be using the abbreviation TGB rather than Beryl’s 10GB. As a techie, when I see 10GB I read “ten gigabytes” and that was doing my head in ☺ [↑](#footnote-ref-1)