H2GO

# 1. Planning the website

After the user has logged in, they will no longer see the buttons to sign up or log in, however, will instead see an update and log off button.

## Step by step plan of how much I understand?

1. I don’t understand how in-depth this plan should be

2. I will design my website using fluidui.com, showing which sites link to which as well as possible, keeping in mind what I can and can’t do due to my limited knowledge of web design. The design will include which pages I will have, how each page looks, where the navigation link will be, how the content will be structured to ensure simplicity for the end-user, where text will be, where the headings will be and other design elements such as a draft colour scheme. This will make sure that I won’t have to think up designs for the site on-the-go and it will help ensure consistency.

3. After creating the design I will select the software that I will use to create the website. This includes the browser that I will use to view the website with, the software that I will use to run and maintain the database, the software that I will use in to program the website and possibly any image-editing software that I may use.

4. Next I will create a thorough test plan that I will use to test my website for data integrity, functional (and non-functional) features, content, links and browser compatibility. This will be done in a word document as a checklist that I will check-off once I have completed the product.

5. After that I will create a log in which I will record what I have been doing and at what times I have been doing them. For the log, I will be using GitHub. To log what I do I will be pushing the updated documents to GitHub and commenting on hardships and troubles that I came about and fixed eventually.

6. I will now search for the images that I will use in the website, have details on each product that I will be selling VIA the H2GO website. I will store the images in a folder and I will store the details of each product in a word document. The details will include the name of the product, the price of the product and a description of what the product is. I will also reference in a separate document where I got the content from to avoid copyright.

8. Now I will begin creating the website, staying as close to the original design as possible. Firstly, I will create the homepage that I will name “index.php”. Next, I will create the catalogue page, which will be named “shop.php”. This page will have details on each product. After that I will create a sign-up page which will be called “registration.php” which will have a form that the user will have to fill out. This form will also have safety features such as confirming your password and choosing an original username. You will be able to sign up to an account with H2GO through there. Then I will make a log in page called “login.php”. It will have also have safety procedures, such as needing to entering a password. After that I will add the update profile page which will be called “update.php”. You will be able to update your personal information on that page, including your password. Afterwards I will add the cart page which will show all the products in your cart and the amount. It will be called “cart.php”. ***While creating all these pages, I will constantly be checking for data integrity and functionality and other testing. If something is incorrect, I will fix it in a short manner of time.***

9. During the process of creation of the website I will create the database for it. This database will feature user information and product information, and through forms the user will be able to add or update information. This is done during the last step because I will most likely only be given instruction for the database during the creation of the website.

10. Once the end-product has been finished, I will test every single feature, look for errors in data integrity and follow the previously created test plan to test the functionality of the website. I will also use the W3C validation tools for my CSS and HTML.

11. After I am certain that everything works, I will ensure that my website meets all moral, ethical and legal obligations. I will ensure that it meets moral obligations by not creating or using content that I personally find offensive, and I will do my best to ensure that that content also doesn’t offend the rest of the public. I will meet all legal obligations by following the law whilst creating the website.

12. I will double check that I have done all the work necessary for an excellence grade.

13. I will now hand in my work and wait for my final grade.

# 2. Software selection

I will use this software for each category:

Browser – primarily Mozilla Firefox. For compatibility testing I will also use Google Chrome, Internet Explorer and Microsoft Edge.  
I will primarily use Firefox to view my website because it is my default web browser, due to being lighter on CPU and RAM usage (in my experience) than Chrome (my previous browser). Aside from that, it is no different from Chrome, although it has less extensions, however, has all the extensions that I need. I will use Chrome, IE and Edge to test for compatibility since they seem to be the biggest browsers on the market now. Also, all these browsers are free to use.

Virtualised server – XAMPP.  
I will use XAMPP since that’s what the teachers recommended and it seems to work so far. XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with several other modules including OpenSSL, phpMyAdmin (which is what we will use for the database), MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version). We are using phpMyAdmin, which XAMPP supports, therefore making it my virtualised server of choice.

Code editor – atom.io.  
I will use atom.io because it is the best text-editor software that I have used so far. It is very light-weight, has an intuitive design and best of all, it works! Also, it has super cool packages which you can download to personalise the code editor for your needs. For example, you could add an in-built browser allowing you to preview your website without opening an actual. You can also customise the layout of atom.io (split screens top/sides/bottom), customise your theme/colours to make the design more appealing and even download other people’s designs. atom.io also has a CSS, HTML5 and PHP library which will be useful for writing code. It can also push to Git without needing to open up Github, which will be very useful for me since I will be using Git as my log.

Log – Git  
I will use Git as my log because it synchronises with atom.io and makes it very clear where I made changes to my documents. It also allows Josh and Johnson to pull my files straight from my repository without me needing to email them. And if I ever need/want to edit the files on a different platform I can just pull the files from my repository and being editing. Git also allows me to add comments and descriptions describing my changes, which is what I need to do for my log. It also has timestamps, which also need to be included in the log. It is also super easy to push files since I am using atom.io and I barely have to think about logging. Makes life simple that way 😊.

Image editing – Microsoft Photos  
I will use Photos for very simple editing tasks, such as cropping an image or applying a filter, however, I cannot use Photos for more than cropping and light editing. Photos is already preinstalled which I why I will use it. It can perform basic tasks such as cropping, filtering and other small retouching which is all that I need.

# 3. Develop a test plan

This is done in a separate word doc