Zenlike

Project Report

Name: Steven Gaynor

Student Number: 14108038

Email: [x14108038@student.ncirl.ie](mailto:x14108038@student.ncirl.ie)

Higher Diploma in Science in Cloud Computing (HDCLOUD)

#### **Project URL: www.zenlike.ie GitHub Repository: https://github.com/StevenGaynor/Zenlike**

20 May 2015

# Zenlike: Executive Summary

This project is called Zenlike, and aims to offer people an intuitive and easy way of organising the things that they like to do on a daily basis.

*Zenlike is a website that offers an innovative way to plan and review your daily life. It is a life management app for busy people that want to structure their fitness, diet and relaxation.*

It is a dynamic website written in HTML5, CSS3, JavaScript and PHP. The database utilised is a relational model and uses MySQL.

In this project report, I will outline the analysis, design and development of the website.

Contents

[Zenlike: Executive Summary 2](#_Toc419893110)

[1. Analysis Phase 4](#_Toc419893111)

[1.1 Objectives 4](#_Toc419893112)

[1.2 Background 4](#_Toc419893113)

[1.3 Technical Approach 5](#_Toc419893114)

[1.4 Special Resources Required 5](#_Toc419893115)

[1.5 Project Plan 5](#_Toc419893116)

[1.6 Technical Details 6](#_Toc419893117)

[2. Requirements Specification 6](#_Toc419893118)

[2.1 Requirements 6](#_Toc419893119)

[2.1.1 Purpose 6](#_Toc419893120)

[2.1.2 Project Scope 7](#_Toc419893121)

[2.1.3 Definitions, Abbreviations 7](#_Toc419893122)

[2.2 User Requirements Definition 7](#_Toc419893123)

[2.3 Requirements Specification 8](#_Toc419893124)

[2.3.1 Functional Requirements 8](#_Toc419893125)

[2.4 Non-functional requirements 14](#_Toc419893126)

[3. Analysis and Design 15](#_Toc419893127)

[3.1 Entity Relationship Diagram 17](#_Toc419893129)

[3.2 Colours 19](#_Toc419893130)

[3.3 Responsive Design 19](#_Toc419893131)

[4. Development Phase 20](#_Toc419893132)

[4.1 Writing the Code 20](#_Toc419893133)

[4.2 Testing Process 24](#_Toc419893134)

[5. Conclusions 25](#_Toc419893135)

[Bibliography 26](#_Toc419893136)

# 1. Analysis Phase

## 1.1 Objectives

For my project, I decided to develop a website that integrated fitness, diet and relaxation. There are many potential benefits to be derived from a tool that can help people living in a busy and hectic world to plan their daily life. There are of course many apps out there that allow the user to plan and monitor their fitness and diet, but not one that combines three important life disciplines into one package.

## 1.2 Background

For this reason, I came up with the idea of a website called *Zenlike.* The name of this package derives from the old saying that a person can have a “Zen-like calm about them”. I asked myself how we could make everybody have a Zen-like calm, yet also be well prepared for the stresses and strains of modern Western society. Well, the answer lies in giving power to the user to manage how they can apply centuries-old philosophies of mindfulness and planning to their own lives. Primarily, it was envisioned that this balance would be achieved by planning and monitoring relaxation events, healthy diet and a course of bespoke physical exercise that fits in with their lives.

At a highly conceptual level, the overall vision of *Zenlike* was that of being a one-stop shop for doing all of these things and potentially expanding to more.

In terms of potential marketability of the product, it’s clear that there is a large potential audience for this given the recent upsurge in interest in healthy living and lifestyle websites and mobile applications.

## 1.3 Technical Approach

In terms of the technical approach to this project, the objectives of *Zenlike* required the following elements of functionality:

* Choose from healthy foods from a list to work into your diet and attempt to get your recommended daily amounts of vitamins and essential minerals
* Design a fitness plan that fits around a busy life, involving a calendar and reminders
* Choose to do a certain amount of meditation every week depending on how busy your life is, and get reminders based upon how many meditation sessions you choose
* Review stats of your dietary intake, exercise and meditation sessions. These would most probably be in the form of graphs or graphical statistics
* A database behind the package would have a bank of quotes related to mindfulness to help encourage the user and motivate them

## 1.4 Special Resources Required

The main special resource required would be hardware such as a Windows/iOS computer and Android or iOS phone to be used for testing. Other than this, some sort of database hosting was needed. I opted to host the website on the cloud with a company called Hosting Ireland.

## 1.5 Project Plan

Below is the plan that I followed throughout the project:

|  |  |  |
| --- | --- | --- |
| Project Deliverable | Delivery Date | Submission Type |
| Project Proposal | 09/02/15 | Moodle |
| Project Plan | 16/02/15 | Moodle |
| Requirements Specification | 02/03/15 | Moodle |
| Analysis & Design | 16/03/15 | Moodle |
| Project Test Plans | 06/04/15 | Moodle |
| Final Software & Documentation Upload | 20/05/15 | Moodle |
| Final Hard Copies Documentation | 20/05/15 | Bound Hard Copy |
| Project Presentation | 23/05/15 | PowerPoint & Demo |

## 1.6 Technical Details

At this early stage, I believed it would be likely that the project would involve the development of a website incorporating some form of data storage so that the user could keep track of their personal user profiles. In terms of data storage, either a MySQL database or HTML5 local storage could be among the choices on offer. In the end, I opted for server-side MySQL and PHP. Other languages used were CSS and JavaScript. A summary of the prospective technical details is outlined in sections below:

**Web Development:** HTML5, CSS, JavaScript, PHP, MySQL (database)

**Applications:** Microsoft Visual Studio, Notepad++, GitHub, EasyPHP (Testing)

# 2. Requirements Specification

## 2.1 Requirements

### 2.1.1 Purpose

The purpose of this section is to set out the requirements for the development of **Zenlike**, a website based on the principal of giving the user power over the organisation of their fitness, diet and relaxation in a balanced and sustainable way.

The intended clients are people who are requiring guidance in organising their schedules in an easy way to allow for all of the healthy aspects of life that their bodies demand. Specific target audiences include those wishing to lose some weight, build a more sustainable and effective exercise regime and those who wish to weave some meditative practices into their lives. Evidently, the shear amount of choice of fitness plans and diets on the market at this time can be of confusion to many people. One of the primary aims of this app will be to help sort the wood from the trees in that respect.

Several academic articles over recent times have proven the linkages between healthy foods and exercise as a beneficial aspect of daily life. The effect of exercise on brain activity is proven from clearly defined studies of the chemicals released during physical activity (Hunsberger, et al., 2007). It is also clear from some recent studies that practicing mindfulness for a consistent amount of time per week can improve a person’s mood (Goyal, Singh, & Sibinga, 2014). Helping people to strike the balance between all of these beneficial activities, and recognising the positive attributes of all of them is something that I believe is important. This is what gave me the idea to develop a website that would assist people in organising their healthy lifestyles in an easy way.

### 2.1.2 Project Scope

The scope of the project is to develop a web-based application incorporating a planner and easily accessible information on meals, exercise types and relaxing activities. The system shall have a database to store all of this information and will be manipulated through the website GUI by the user. The website administrators will also have the ability to add new items to the database in order to develop the knowledgebase of the site.

### 2.1.3 Definitions, Abbreviations

**MyHub:** Is the central PHP webpage and it has three main sections: Weekly Planner, MyEvents and the links to the pages with choices of meals, exercises and relaxation events.

**MyEvents:** Is a section of the Planner that displays the user’s choice of events in a stylised CSS text box. They can be saved onto the calendar.

## 2.2 User Requirements Definition

In this section, the set of objectives and requirements for the system will be outlined from the customer’s perspective. It is clear that any system that is designed must be very user-friendly and allow the client to easily organise their **Zenlike** events according to their own personal choices from the elements that the app will offer. It is clear that how all of these things balance out and indicate the success of the plan to the user will need to be determined by the code of the finished software package.

In more concrete terms, the main requirements for the user of this product are the following:

* To choose from a list of meals according to preferences, place the meal on the planner
* To choose exercises from a list, and place them on the planner
* To choose relaxation events from a list, and place them on the planner
* To show the user how this combination of exercise and diet would impact on the net intake of calories and vitamins and minerals
* To allow administrators of the website to add new meals, exercises and relaxation events
* To survey the vital statistics of the user, such as height weight, gender and age

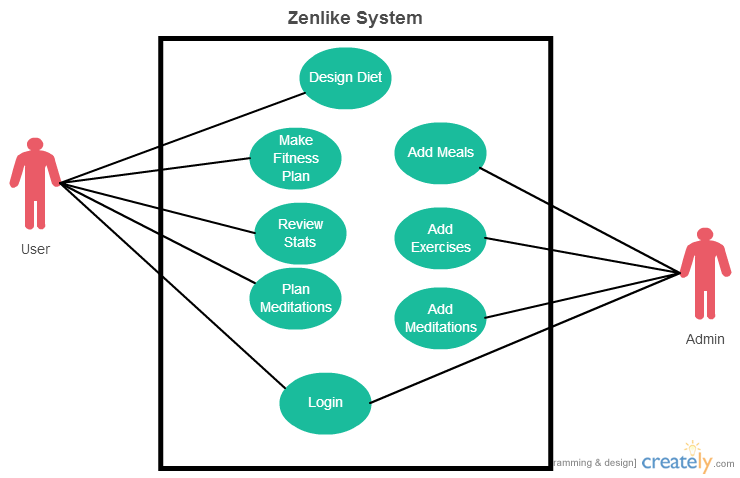
## 2.3 Requirements Specification

My main principal at the outset was the system needed to be one that could be easily learnt within a few minutes. The layout of the GUI and smooth running of the underlying code would be essential to ensuring this. During the testing phase, it was also necessary to ensure that as many errors as possible are routed out. This section will describe in detail both the functional and non-functional requirements of the system.

### 2.3.1 Functional Requirements

This section will describe the functional requirements of the Zenlike system. The functional requirements describe “the system function in detail, its inputs and outputs, exceptions” (Sommerville, 2006, p. 120). The use case model diagram below gives an overview of what the system seeks to achieve at a highly conceptual level.

#### Use Case Diagram



The following section will have a detailed description for each use case roughly in order of importance.

#### 2.3.2 Use Case Description – “Make Fitness Plan”

**Scope**

The scope of this use case is to allow the **User** actor to design a fitness plan using a planner calendar.

**Description**

This use case describes the ability the User will have to choose from a list of exercise types, according to how many calories that they burn. The user will be able to put these exercises onto their planner calendar.

**Flow Description**

The User actor will be presented with a screen that lists exercise types, perhaps with pictures and a brief description of the exercise. The user will click on the exercise “button”/”window”. Clicking this will save this exercise to the user’s list of chosen events. The user will then click on the calendar (in an easily accessible area of the screen) and will drag and drop a CSS designed box with the exercise onto a chosen time on the user’s calendar. The drop of this CSS box will be saved and indicated on the time choses by the user.

**Precondition**

User will be on a webpage that will act as a central hub for co-ordinating all of their Zenlike activities. It will have links to the Zenlike calendar, and the database of meals, exercises and relaxations.

**Activation**

This use case starts when the **User** actor clicks on “Choose Exercises”. This will take them to a webpage which lists the exercises in styled HTML sections.

**Main flow**

1. The system presents the user with the Exercises PHP page.
2. The User clicks on an exercise name to see a picture of the exercise (optional), how many calories burned, and a description. An area of text named something like “Add to Events” can be clicked if the user wants this event.
3. The system registers the user’s choice of this event by making a database entry in the Exercise events table.
4. The User can choose more events if required

**Alternate flow**

1. The system already has that event saved to the Events of the User
2. The User receives a message stating that he/she has already chosen this event.
3. The use case continues at position 2 of the main flow

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

The system presents the calendar with the events saved successfully in the time slots.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 2.3.3 Use Case Description – “Design Diet”

**Scope**

The scope of this use case is to allow the **User** actor to design a diet from a bank of healthy meals using a planner calendar.

**Description**

This use case describes the ability the User will have to choose from a list of meals, according to how many calories that they contain and their vitamin/mineral content etc. The user will be able to put these meals onto their planner calendar.

**Flow Description**

The User actor will be presented with a screen that lists meals, perhaps with pictures and a brief description of the meal. The user will click on the meal “button”/”window”. Clicking this will save this meal to the user’s list of chosen events. The user will then click on the calendar (in an easily accessible area of the screen) and will drag and drop a CSS designed box with the meal onto a chosen time on the user’s calendar. The drop of this CSS box will be saved and indicated on the time chosen by the user.

**Precondition**

User will be on a webpage that will act as a central hub for co-ordinating all of their Zenlike activities. It will have links to the Zenlike calendar, and the database of meals, exercises and relaxations.

**Activation**

This use case starts when the **User** actor clicks on “Choose Meals”. This will take them to a webpage which lists the meals in styled HTML sections.

**Main flow**

1. The system presents the user with the Meals PHP page.
2. The User clicks on a meal name to see a picture of the meal (optional), how many calories it contains, vitamin/mineral content and a description. An area of text named something like “Add to Events” can be clicked if the user wants this event.
3. The system registers the user’s choice of this event by making a database entry in the Meals events table.
4. The User can choose more events if required

**Alternate flow**

1. The system already has that event saved to the MyEvents of the User
2. The User receives a message stating that he/she has already chosen this event.
3. The use case continues at position 2 of the main flow

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

The system presents the calendar with the events saved successfully in the time slots.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 2.3.4 Use Case Description – “Plan Meditations”

**Scope**

The scope of this use case is to allow the **User** actor to plan meditations from a bank of relaxing/meditative events using a planner calendar.

**Description**

This use case describes the ability the User will have to choose from a list of meditations according to their lifestyle preferences. The user will be able to put these events onto their planner calendar.

**Flow Description**

The User actor will be presented with a screen that lists relaxing things to do, perhaps with pictures and a brief description of the event. The user will click on the meditation “button”/”window”. Clicking this will save this meditation to the user’s list of chosen events. The user will then click on the calendar (in an easily accessible area of the screen) and will drag and drop a CSS designed box with the meditation onto a chosen time on the user’s calendar. The drop of this CSS box will be saved and indicated on the time chosen by the user.

**Precondition**

User will be on a webpage that will act as a central hub for co-ordinating all of their Zenlike activities. It will have links to the Zenlike calendar, and the database of meals, exercises and relaxations.

**Activation**

This use case starts when the **User** actor clicks on “Choose Relaxing Events”. This will take them to a webpage which lists the relaxations in styled HTML sections.

**Main flow**

1. The system presents the user with the Relax PHP page.
2. The User clicks on an event name to see a picture of the event (optional). An area of text named something like “Add to Events” can be clicked if the user wants this event.
3. The system registers the user’s choice of this event by making a database entry in the Meditations events table.
4. The User can choose more events if required

**Alternate flow**

1. The system already has that event saved to the MyEvents of the User
2. The User receives a message stating that he/she has already chosen this event.
3. The use case continues at position 2 of the main flow

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

The system presents the calendar with the events saved successfully in the time slots.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 2.3.5 Use Case Description – “Add Meals, Exercises, Meditations”

**Scope**

The scope of this use case is to allow the **Admin** actor to login as a user with admin privileges and easily add new meals, exercises and mediations to the bank of events.

**Flow Description**

The Admin actor will fill in fields and optionally add a picture in order to allow more events for the user to choose from

**Precondition**

Logged in as an Admin user. Ability to “Add” and “Edit” the events in each of the three events pages.

**Activation**

This use case starts when the **Admin** actor clicks on Add or Edit

**Main flow**

1. The Admin clicks Add event
2. Fills out Name, Description and calorie/mineral details etc.
3. The System saves these new details to database
4. The Admin checks to see these events appear correctly without error.

**Alternate flow**

A1

The exercise type is duplicate of one already there.

**Exceptional flow**

E1:

1. The system loses contact with the MySQL database.
2. The Admin receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues after database connectivity restored.

**Termination**

The events saved successfully on the events pages

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 2.3.6 Use Case Description – “Review Stats”

**Scope**

The scope of this use case is to allow the **User** actor to review how many calories they have consumed and/or burned off in a given week, and to review vitamin/mineral intake.

**Description**

This use case describes the ability the User will have to view stats of their calorie count given the meals/exercises etc. they have chosen.

**Flow Description**

The User actor will be presented with a stat box that contains the above described details

**Precondition**

The events have been successfully placed on the calendar

**Activation**

This use case starts when the **User** actor has started placing events on their calendar.

**Main flow**

1. The User has at least one event on his/her calendar.
2. The system feeds info into a stat table.
3. This info is outputted into a display on the MyHub page.

**Alternate flow**

n/a

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

If the user chooses to remove his/her events from the calendar.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 2.3.7 Use Case Description – “Login”

This will allow the User and Admin to see their own personalised MyHub page that fulfils all of the use cases as appropriate above. It allows the functionality that a User can save events to their calendar and return at a later time to view/amend as required.

### 2.4 Non-functional requirements

Non-functional requirements are “quality of service” and “non-behavioural requirements” that determine the aspects of the system’s running that do not relate to items such as use cases or functional requirements, basically determining how the system controls itself and interacts with other systems (Stellman & Greene, 2005, p. 113).

The non-functional requirements for Zenlike essentially encapsulate that the system should give responses in good time. For example, a drag-and-drop event should not take more than 5 seconds to complete. Lag should kept to a minimum. This will be ensured during the testing phase.

The website should also have excellent cross-browser functionality and be fully mobile responsive.

# 3. Analysis and Design

Having completed the requirements for Zenlike it was now time to turn to the next phase of planning. For each requirement it was necessary to implement them as functions on the website.

The website on the client side will be written using HTML5, CSS3 and JavaScript. The original wireframe of the website was laid out as follows:

## C:\Users\Mary\Desktop\Cloud Project\Zenlike\zenlikehub_wireframe.jpg

Whilst this changed throughout the actual development process, it gives some sort of an indication of how the requirements were considered for implementation at the front-end. There is HTML sections for the calendar, the events and an area where the user can choose what events he/she desires.

The desired functionality is to achieve an interactive weekly planner that the end user can populate with the events that suit their daily lives. There will need to be many events to choose from. Some sort of save button, check box or radio button will need to be beside each event so that they can be saved into the MyEvents module that is located on the calendar page.

The initial site map for the Zenlike website was analysed as having at least the following components:

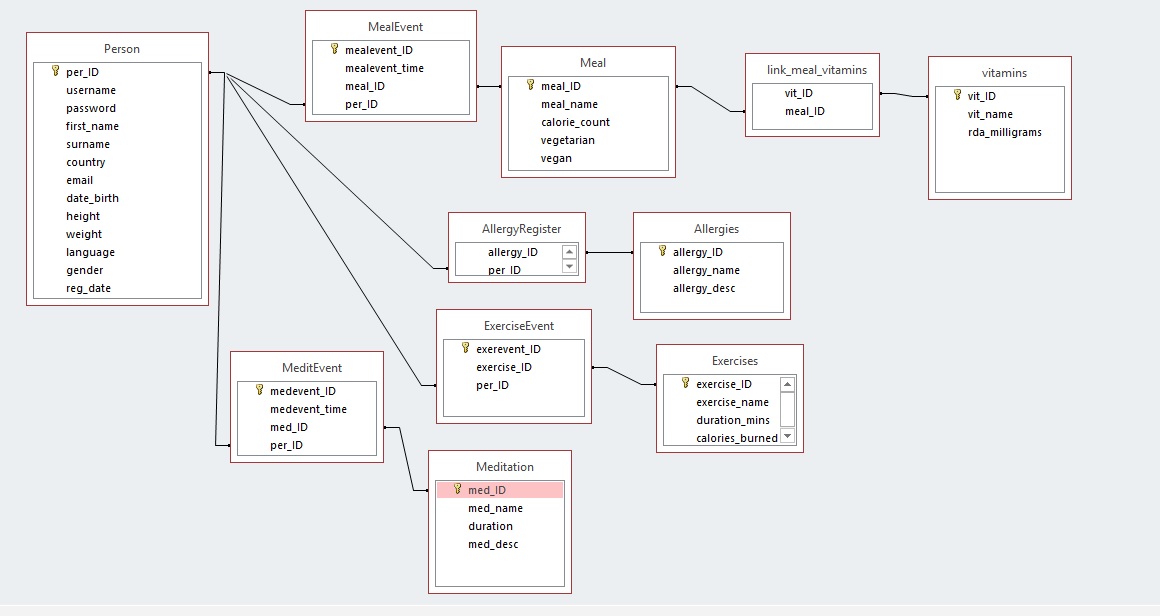
* index.html = will be a standard welcome page
* myhub.php = will contain the calendar and events module
* events.php = will contain the entries for all of the meals, exercises and relaxations
* guide.html = a guide on helping the user to user the website

## 3.1 Entity Relationship Diagram

On the following page is an entity relationship diagram that hopes to outline how the relational database for Zenlike was organised.

With this diagram, one can begin to see how a PHP and MySQL setup may interact with a HTML/CSS front end.

My initial idea was to have a draggable JQuery object that when dropped onto the calendar will create an entry in the database. However, this proved difficult to implement in practice and given the time constraints. The direction that I opted for was to have to have a save button beside each event.



## 3.2 Colours

Given the theme of the website I decided to opt for the following colour palette for my website design.

 -> Main Green: Hex #368D45

 -> Blue: Hex #B84F46

 -> Golden Brown: Hex #B87D46

 -> Darkish Red: Hex #B84F46

Various lighter and darker shades of the above colours were used during the development process if deemed to add to the website but mainly these four colours were used for most areas.

I believe this combination of colours perfectly encapsulates the overall feel of Zenlike as being a calming place away from worries and stresses, a simple yet elegant interface for managing lives around busy schedules.

## 3.3 Responsive Design

It is essential that the Zenlike website not only works on desktops but also on mobile and tablet devices also. To that end different types of screens sizes will have to be catered for and elements of my web application such as the calendar will need to be easily manipulated on a mobile screen. Media queries in CSS will be used as well as utilising percentage scales for most width parameters in the style sheet. The website needs to work on mobiles and tablets from both Apple OS and Android devices as well as in the main browsers Chrome, Firefox, Safari, IE and Opera.

# 4. Development Phase

## 4.1 Writing the Code

Once I had all of the design elements in place, I commenced writing the code for my project. Since I had no previous experience in writing a website in PHP I decided to consult some texts on the subject. The elements that required PHP were the following:

* Login page
* Register page
* Weekly calendar/planner
* Exercises listing page
* Meals listing page
* Relaxations listing page

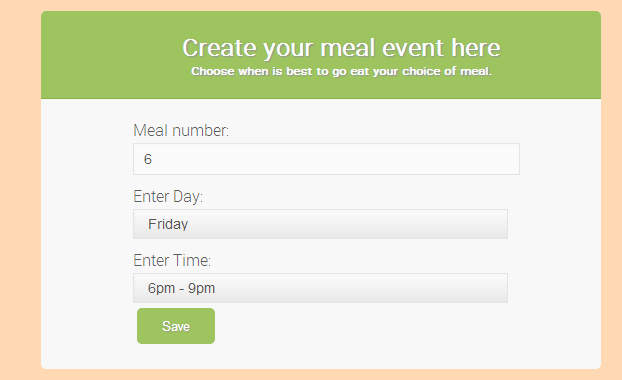
For the purposes of learning PHP, I studied and utilised (with minor adaptations) the code for registrations and logins used in the textbook *PHP & MySQL In Easy Steps* (McGrath, 2012). The code offered a very professional looking front-end registration process implementing basic security that I could then adapt and link up with the database table for users (called Persons in this project). It was also very useful in assisting me in learning how PHP operations work for the coding process on the rest of the website.

Having completed a suitable registration and login process that achieved successful entries into the MySQL database, I set about coding the pages that list all of the events that the user can participate in. PHP sessions were used to ensure that once logged in, any pages navigated to would recognise the logged in user and use their database ID to call back the data specific to that user.

For example, the page called **choosemeals.php** is a webpage that essentially first checks the user is logged in and runs an SQL SELECT \* query to present all of the meals to the user. The other two listings pages function in exactly the same manner, except that they are calling from their respective table in the database.

Once the user decides which event they want to add to their calendar, they are then taken to another page called **createMealEvent.php.** This askes the user for a time and date that suits then and takes this along with the meal choice and the user’s session ID and adds them to an SQL statement.

An example of creating a new event is provided in the pictures below showing the frontend view and code behind it respectively:



User enters meal time and clicks save



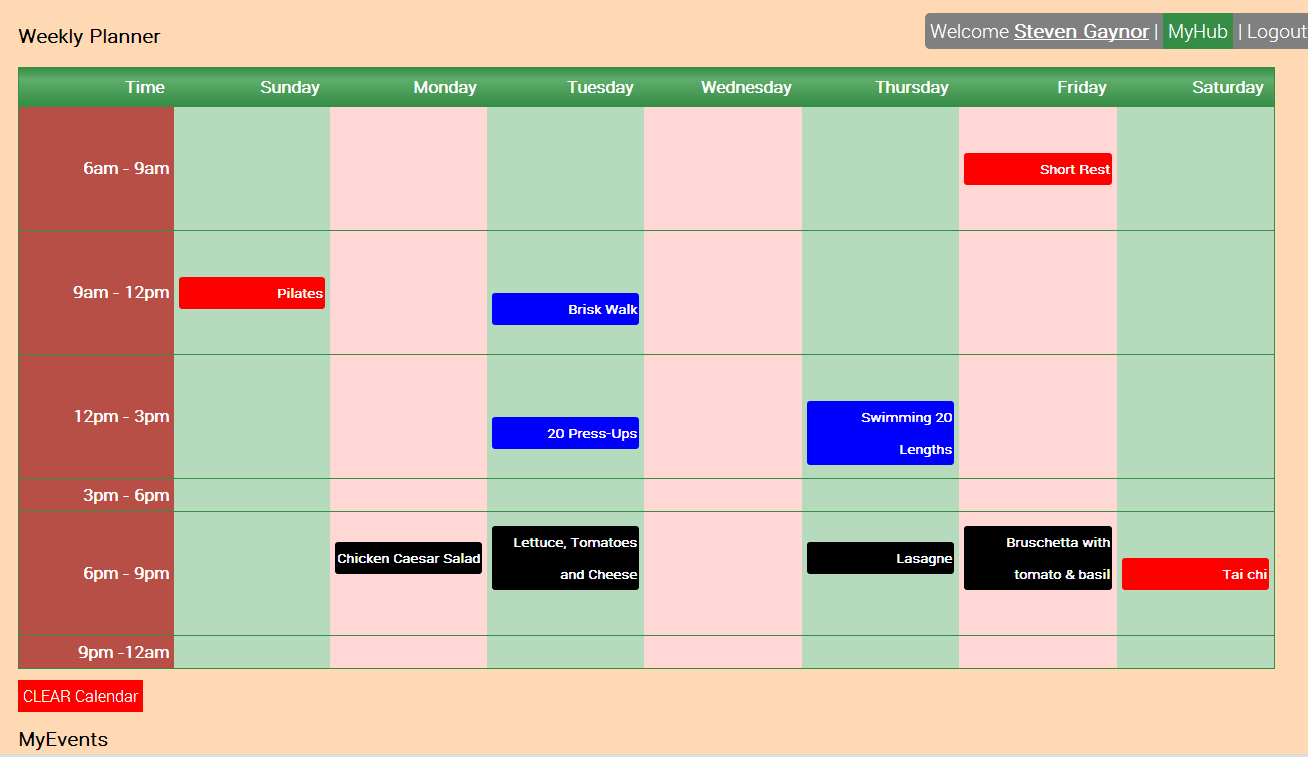
PHP code for creating a new Meal Event

The more complex aspect of the coding process was the creation of the page called **myhub.php** as this page contains the weekly calendar. The solution I came up with was to have an SQL query for each day and time slot in the calendar, using the User’s ID from the PHP session.

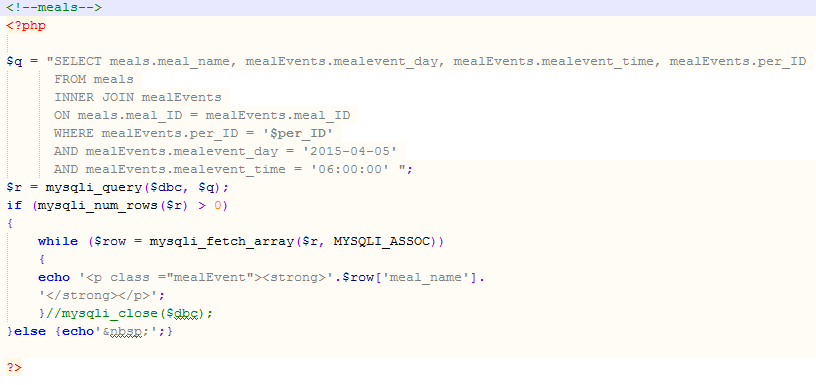
Conditional if statements where utilised to check if any entries under the data time are there for the user, and if not then a simple blank space is output.

The user can place one exercise, one meal and one relaxation in each time slot.

An example of a sample calendar is given below:

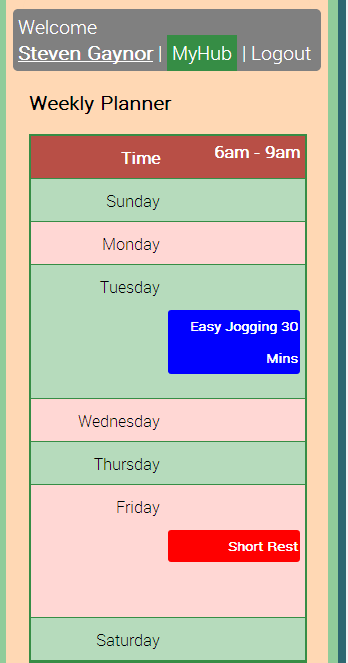
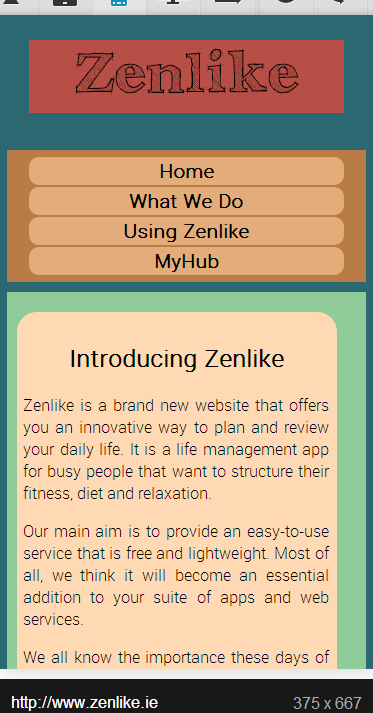


And below is a code snippet from behind one of the calendar slots. It takes data from two of the database tables (**meals** and **mealEvents**) using an inner join on the “per\_ID” foreign key:



The Clear Calendar button offers a Delete function, and the ability is offered to the user to start again with a new calendar if they wish to do so. This simply deletes all events associated with their ID in the database.

The important attribute of having a responsive design has also been implemented using CSS mobile queries. All of the website’s pages are fully responsive depending on the size of screen. An example of the mobile version of Zenlike is shown in the pics below:



## 4.2 Testing Process

Once I had completed the process of coding the process I began a period of testing the website. This involved ensuring that all of the database queries worked correctly and that the requirements outlined at the outset where fulfilled.

I have managed to fulfil most of the requirements but some of the additional goals such as reviewing stats and allowing the user to indicate allergies had to be jettisoned due to time constraints.

The testing process was successful after a few minor bugs where ironed out. All along, I had been using the PHP validator (<http://phpcodechecker.com>) to check my work, so the code was pretty much glitch free when it came to the deployment stage.

# 5. Conclusions

In conclusion, the objectives set out from the beginning of this project have for the most part been fulfilled. At the finality, there is a dynamic database driven website that offers the user the functionalities that I set out in the analysis and design phase. The process was interesting as I learned much about PHP coding and had a chance to build on my knowledge of database design and development, which I found to be a rewarding process.

The process was a steep learning curve, but one that I enjoyed the process of undertaking and completing with a satisfactory and elegant end product.

The completed project is hosted online at <http://www.zenlike.ie>

# Bibliography

Goyal, M., Singh, S., & Sibinga, E. M. (2014). Meditation Programs for Psychological Stress and Well-being: A Systematic Review and Meta-analysis. *JAMA Intern Med., 174*(3), 357-368.

Hunsberger, J. G., Newton, S. S., Bennett, A. H., Duman, C. H., Russell, D. S., Salton, S. R., & Duman, R. S. (2007). Antidepressant actions of the exercise-regulated gene VGF. *Nature Medicine, 13*, 1476 - 1482.

McGrath, M. (2012). *PHP & MySQL In Easy Steps.* Leamington Spa: In Easy Steps Limited.

Sommerville, I. (2006). *Software Engineering* (8th ed.). Boston: Addison-Wesley.

Stellman, A., & Greene, J. (2005). *Applied Software Project Management.* O'Reilly Media.