

Task 2: Developing the System Prototype

From the examination of the sources and assets gathered, I have considered my options.

As the user requirements are clear and, due to legal and ethical implications of using the identified assets, all of the images are to be sourced from free imaging providers or providers that allow use for non-commercial use: "License: Non-commercial Use". Some of these providers require you to set up an account to download the images and some ask you to reference the source of the image.

Care was taken when selecting the images to use due to the legal and ethical implications of the assets.

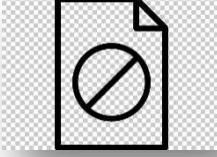
Great care was taken when selecting the external links to the health and fitness and food external links. All of the recipes were from a reputable source and promoted healthy eating and most identified the nutritional value of the meal. They were easy to follow with clear images of the prepared dishes.

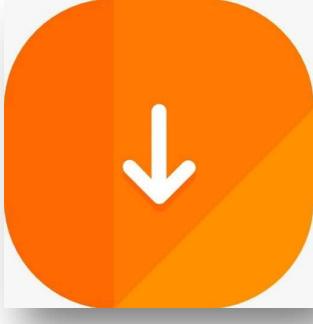
The health and exercise video and information links were sponsored by medical and personal trainers and supply website help and guidance as customers will be using these videos unsupervised by health and medical experts. All of the links hosted fitness videos of a very professional quality and level. From the research of possible code snippets, the only code that I am going to modify is "chat room and blog example code" by using bootstrap, which is excellent for events and layouts.

Legal and ethical consideration are important during the development of this artefact. The dilemma can be displaying people who may be overweight, which reinforces that overweight means unhappy, eating lots of food and no exercise. And while this may be true to some extent, the different ways people view overweight is shocking, and this is most prevalent in the way the media portrays celebrities. Being thin is therefore healthy. So, it is important that ethical and moral issues are thought of before selecting the images and videos.

There is a close relationship between law and ethics but not everything that is legal is ethical. Frequently law attempts only to set the minimum acceptable standard. The aspirations of ethical practice are higher ... It can never be appropriate to defend proposed practice solely on the basis that it is legal.

Assets selected and rationale

Image	Source	Rationale
	https://www.freepik.com/free-icon/search_788138.htm	Users see this search image more than the other option and it is also uses fewer colours.
	https://icons8.com/icons/set/settings	Will use the traditional settings icon as all users know what it is.
	https://imgbin.com/png/wfLgfRPc/computer-icons-mobile-phones-png	This is my option for "no access to members' area". Users may find the other option confusing and it has too many colours.

	https://imgbin.com/png/Jm6mwH7T/a-logo-png	Decided to use this icon as it's free for non-commercial use as the other option needs an extended license. It also stands out.
	https://www.flaticon.com/free-icon/facebook-logo-button_69407	Used this social media logo as users will be able to identify it quickly.
	https://www.stickpng.com/img/icons-logos-kemojis/tech-companies/whatswebsite-logo	Used this social media logo as users will be able to identify it quickly.
	https://www.stickpng.com/img/icons-logos-emojis/tech-companies/twitter-logo	Used this social media logo as users will be able to identify it quickly.
	https://www.stickpng.com/fr/img/icones-logos-emojis/societes-de-technologie/logo-youtube-play	Used this social media logo as users will be able to identify it quickly.

	https://www.flaticon.com/free-icon/instagram-logo_69366	Used this social media logo as users will be able to identify it quickly.
---	---	---

Contact us example code	This is the only code that I am going to modify. I am going to develop my code for the website to ensure that it efficient and robust.
bootstrap	

Links to external sites

https://www.bbcgoodfood.com/recipes/collection/healthy-dinner	BBC good fool link	Decided to link to these external resources, they are all verified by the medical profession and have excellent reviews on social media.
https://lovefoodhatewaste.com/	Love food healthy recipes	
https://www.ketocustomplan.com/	Kenco diet plan	
https://www.foodsavvy.org.uk/	Foodsavvy healthy recipes	
https://www.delish.com/cooking/	Healthy recipes	
https://ifoodreal.com/clean-eating-recipes-dinners/	Healthy recipes	
https://bitesofwellness.com/	Healthy recipes	
https://www.everydayhealth.com/	Health and fitness websites	
https://www.health.com/fitness	Health and fitness websites	

https://www.health.harvard.edu/topics/exercise-and-fitness	Health and fitness websites	each of the recipe.
https://www.pinterest.com/reachyourpeak/health-fitness-topics/	Health and fitness websites	They are reliable sources of information to link to.

Blogs:

https://www.quora.com/What-are-your-top-5-favorite-health-and-fitness-topics	Fitness blog	Will provide further options for customers to select their own topics.
https://www.wix.com/	Wix blog	These are the top 3 blogs in the sector and have been established for some time and have a good following and lots of posts.
https://bitesofwellness.com/blog/	Bites of Wellness	
https://blog.feedspot.com/uk_fitness_blogs/	Top 10 blogs	

Prototype for the proposed digital system

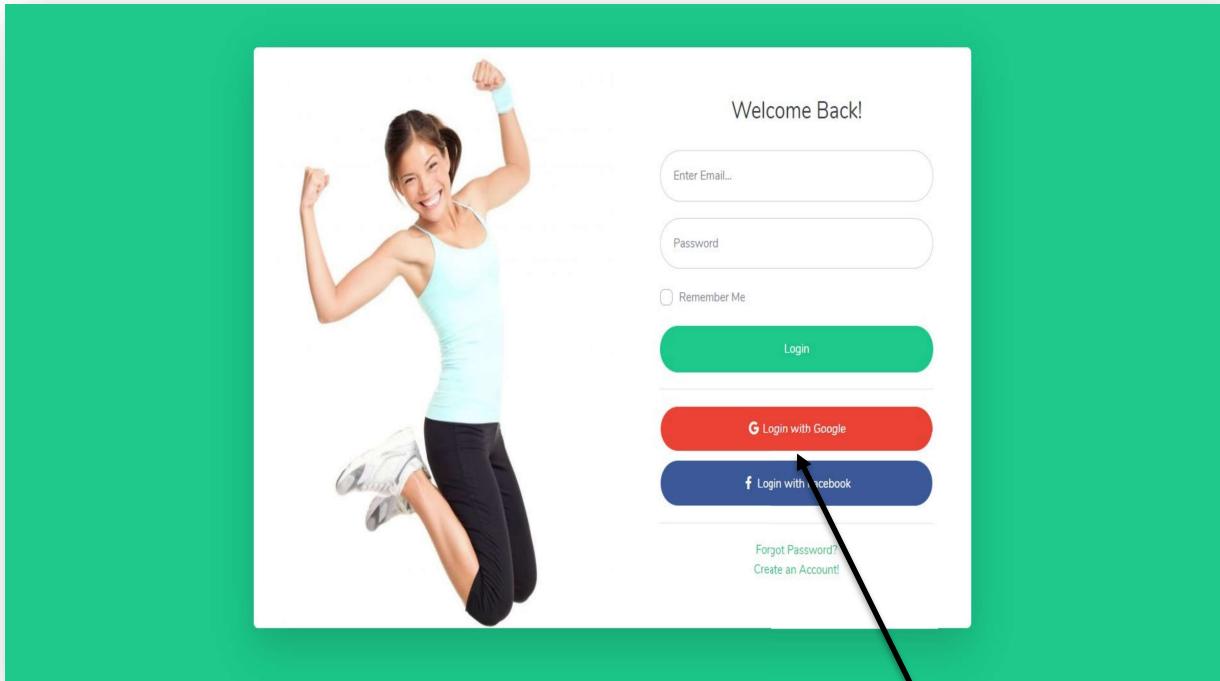
ToKa Fitness system prototype has commissioned my software development company to develop a digital system. ToKa Fitness offers personal training sessions and advice on fitness training and healthy living to its customers and would like a website for their specific requirements.

ToKa Fitness specific requirements are that the proposed solution:

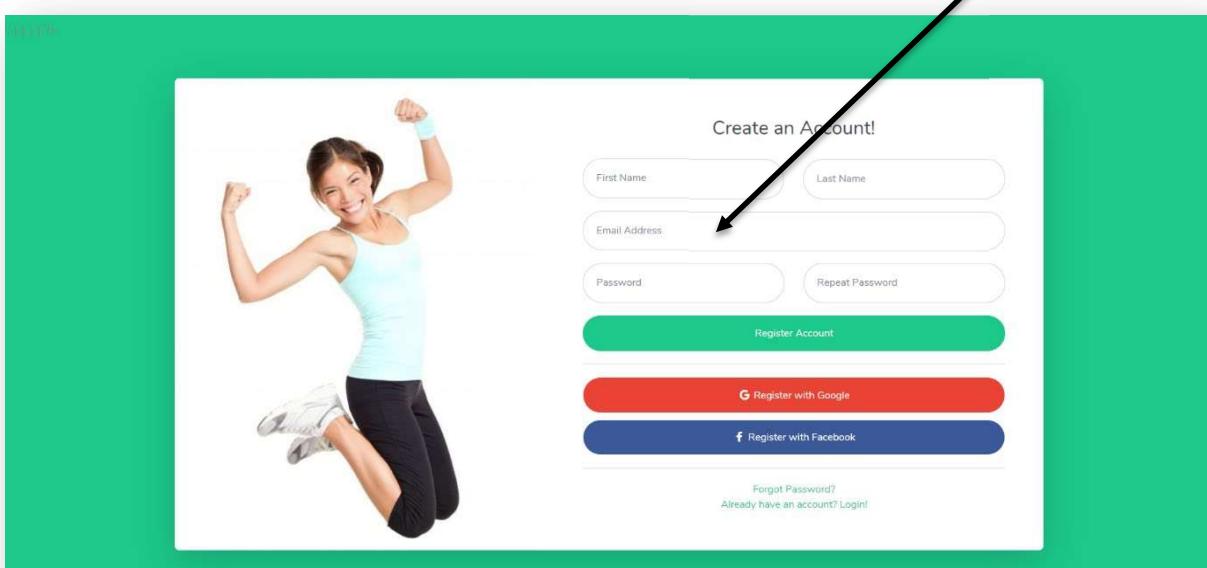
- has free access with some accessibility to services
- customer section for paid content to access full services
- accessibility features for users with sight loss
- link to 'social media' features
- ability to customisable workout and eating plans

- access to various graphs to view stats on:
 - calories
 - fat
 - weight
 - food
 - comparisons and logs for day, week and month.

Login page



This is the login screen; you can use created username and password or if you originally registered using Google or Facebook you can use that option as well. This is to create an easier user experience. Below is the create account page.



This is the payment page, allow clients to pay for the members sections, which gives them access to various extra parts that are not available to standard users.

Client information table



Food diary is to capture how they monitor their food intake. Evidence suggest that viewing what they eat daily will help with keeping their diet in place.

Food Diary

History

Date	Food	Weight (g)	Calories	Protein (g)	Carbohydrates (g)	Fat (g)
19th December 2019	Chicken	500	975	147.75	0	38.6

Generate Report

This allows users to monitor their diet and they can add the various food for each day, and it will work out the calories and fat content. As the user enters more the history will grow and give them average calories per week. Can generate a report for daily, monthly and yearly intake.

Food Diary

History

Date	Food	Weight (g)	Calories	Protein (g)	Carbohydrates (g)	Fat (g)
19th December 2019	Chicken	500	975	147.75	0	38.6

Generate Report

Food Videos

The runner's diet

The weight lifter's diet

Above is a video that can be accessed by members only, below is the data that can be generated in a weekly or monthly report.

Fitness Diary

Date	Exercise	Calories burnt	Time
20th December 2019	Run	300	
20th December 2019	Swim	420	1h
19th December 2019	Walk	142	30m
18th December 2019	Run	300	45m
17th December 2019	Swim	420	1h
17th December 2019	Walk	142	30m
16th December 2019	Run	300	45m
	Swim	420	1h
	Walk	142	30m

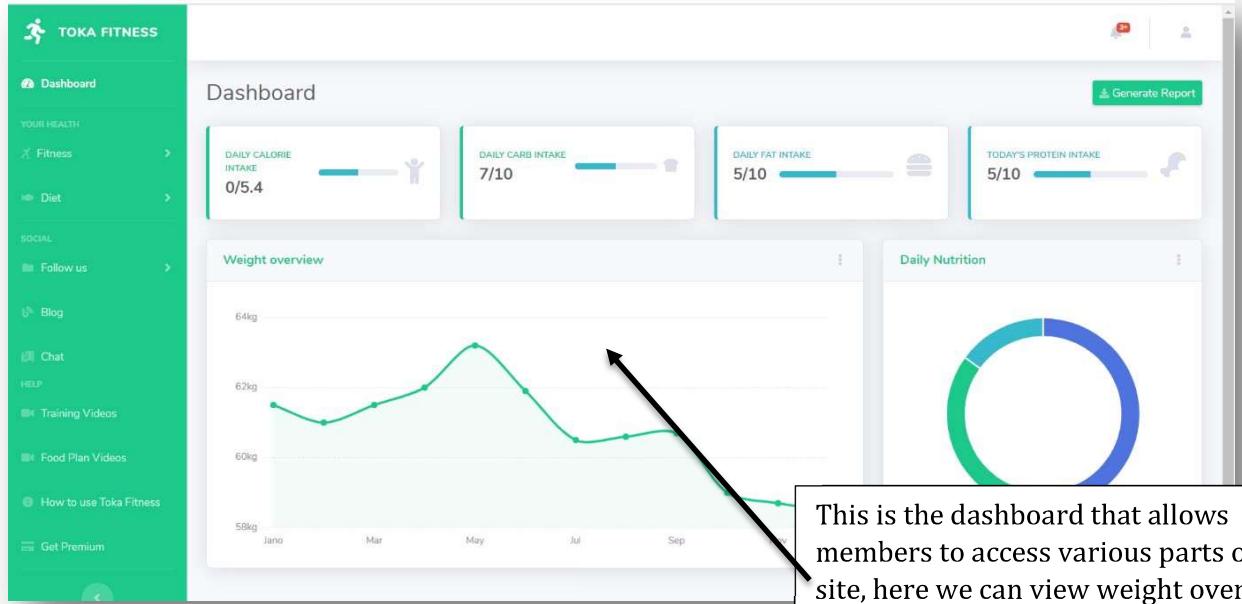
ALERTS CENTER

December 12, 2019 A new monthly report is ready to download!

December 7, 2019 You have achieved your monthly goal!

Show All Alerts

Here members can run a report for monthly stats that can be downloaded. Also have the option to check they have reached their goal for the month.



This is the dashboard that allows members to access various parts of the site, here we can view weight overview for the months. This was selected as it is better and more motivational when they can see a decline over time. If it was day to day it wouldn't show significant changes and, in some cases, may show weight had been added on.

This is the blog page which allows has member and non-member access. The member can see both and return messages to other members.

Also includes popular posts and links to Facebook and Twitter accounts.

This page is used for diet tracking, the user inputs food eaten and the amount in calories. Once submitted it will keep a record of the different amount of food eaten each day. A monthly report is generated that then can be compared to various elements such a protein, carb intake and calories.

The fitness tracker page relies on adding exercise data, this could be for the four most common options. The button at the top allows you to import the data from a fitness tracker.

Website code

PHP code for member page

```
<?php
include ('config.php');
session_destroy();
session_start();
if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];
    $query = $connection->prepare("SELECT * FROM users WHERE email = :email");
    $query->bindParam(":email", $email, PDO::PARAM_STR);
    $query->execute();

    $result = $query->fetch(PDO::FETCH_ASSOC);

    if (!$result) {
        echo '<script>alert("Invalid username and password combination")';
    } else {
        if (password_verify($password, $result['password'])) {
            $_SESSION['user_id'] = $result['id'];
            $_SESSION['first_name'] = $result['firstname'];
            $_SESSION['last_name'] = $result['lastname'];
            header("Location:index.php");
            die();
        } else {
            echo '<script>alert("Invalid username and password combination")';
        }
    }
}
?>
```

This is the PHP code for the login. Links to include another page. Uses a nested if statement

Session set up to do the following:

- makes user data available across the whole website
- makes a temporary file in a server temporary directory that saves session data
- atomically destroys when user close the browser

Super global variable with session data after session started

Select the user details

Binds the parameter to PHP variable in SQL statement

Produces a message if password invalid

```

1 <?php
2
3 include('config.php');
4 session_start();
5
6 if (isset($_POST['register'])) {
7
8     $firstname = $_POST['firstname'];
9     $lastname = $_POST['lastname'];
10    $email = $_POST['email'];
11    $password = $_POST['password'];
12    $repeatpassword = $_POST['repeatpassword'];
13    if($password !== $repeatpassword) {
14        echo '<h1><font color="white"><center>Password and repeat password do not match!</center></font></h1>';
15    } else {
16        $password_hash = password_hash($password, PASSWORD_BCRYPT);
17
18        $query = $connection->prepare("SELECT * FROM users WHERE EMAIL=:email");
19        $query->bindParam(":email", $email, PDO::PARAM_STR);
20        $query->execute();
21
22        if ($query->rowCount() > 0) {
23            echo '<h1><font color="white"><center>The email address is already registered!</center></font></h1>';
24        }
25
26        if ($query->rowCount() == 0) {
27            $query = $connection->prepare("INSERT INTO users(FIRSTNAME, LASTNAME,PASSWORD,EMAIL) VALUES (:firstname, :lastname,:password_hash,:email)");
28            $query->bindParam("firstname", $firstname, PDO::PARAM_STR);
29            $query->bindParam("lastname", $lastname, PDO::PARAM_STR);
30            $query->bindParam("password_hash", $password_hash, PDO::PARAM_STR);
31            $query->bindParam("email", $email, PDO::PARAM_STR);
32            $result = $query->execute();
33
34            if ($result) {
35                echo '<h1><font color="white"><center>Your registration was successful!</center></font></h1>';
36            } else {
37                echo '<h1><font color="white"><center>Something went wrong!</center></font></h1>';
38            }
39        }
40    }
41 }
42
43 ?>
44 <!DOCTYPE html>
45 <html lang="en">
46

```

To collect the data from the form, use the PHP super global variable

Used bindParam with the SQL statement

```

1 <?rAddress: session_start();
2
3 if(!isset($_SESSION['user_id'])){
4     header('Location: login.php');
5     exit;
6 }
7
8
9
10 if(!$_SESSION['premium']){
11     echo 'You need to be a premium member to access this part of the website!';
12 } else {
13 ?>

```

Asset function is used to check whether there is a value entered. If there is no value, it echoes a message back to the user to go to the login in page. Then it checks if the value entered is equal to the value in the database in this case user_id.

If userID is valid it will allow the user to access the training video. In other words, they are a paying customer. Echo a message if they are not.

```

268      <!-- End of Page Wrapper -->
269
270      <!-- Scroll to Top Button-->
271      <a class="scroll-to-top rounded" href="#page-top">
272          <i class="fas fa-angle-up"></i>
273      </a>
274
275      <!-- Logout Modal-->
276      <div class="modal fade" id="logoutModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
277          <div class="modal-dialog" role="document">
278              <div class="modal-content">
279                  <div class="modal-header">
280                      <h5 class="modal-title" id="exampleModalLabel">Ready to Leave?</h5>
281                      <button class="close" type="button" data-dismiss="modal" aria-label="Close">
282                          <span aria-hidden="true">×</span>
283                      </button>
284                  </div>
285                  <div class="modal-body">Select "Logout" below if you are ready to end your current session.</div>
286                  <div class="modal-footer">
287                      <button class="btn btn-secondary" type="button" data-dismiss="modal">Cancel</button>
288                      <a class="btn btn-primary" href="login.php">Logout</a>
289                  </div>
290              </div>
291          </div>
292      </div>
293
294      <!-- Bootstrap core JavaScript-->
295      <script src="vendor/jquery/jquery.min.js"></script>
296      <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
297
298      <!-- Core plugin JavaScript-->
299      <script src="vendor/jquery-easing/jquery.easing.min.js"></script>
300
301      <!-- Custom scripts for all pages-->
302      <script src="js/sb-admin-2.min.js"></script>
303
304      <!-- Page level plugins -->
305      <script src="vendor/chart.js/Chart.min.js"></script>
306
307      <!-- Page level custom scripts -->
308      <script src="js/demo/chart-area-demo.js"></script>
309      <script src="js/demo/chart-pie-demo.js"></script>
310
311  </body>
312
313 </html>
314

```

Used bootstrap JS with some of the functionality on the pages. Menu, graphs and login screen.

```

1  * {
2     margin: 0;
3     padding: 0;
4 }
5
6 html, body, #navbar {
7     min-height: 100%;
8     height: 100%;
9 }
10
11 body {
12     background: rgb(38,38,38);
13     font-family: 'Open Sans', sans-serif;
14     font-weight: 300;
15     margin: 0;
16 }
17
18 h1 {
19     font-family: 'Quicksand', sans-serif;
20     font-weight: 500;
21 }
22
23 ::-webkit-scrollbar {
24     width: 16px;
25 }
26
27 ::-webkit-scrollbar-thumb:hover {
28     background: rgb(0,176,240);
29 }
30
31 ::-webkit-scrollbar-track {
32     background: rgb(13,13,13);
33 }
34
35 ::-webkit-scrollbar-thumb {
36     border-radius: 25px;
37     border-style: solid;
38     border-color: rgb(13,13,13);
39     border-width: 4px;
40     background: rgb(0,112,192);
41 }

```

These styles apply to all elements (except h1 and scrollbar style).

The scrollbar styling only works on select browsers like Chrome.

The meta viewport tag lets me control the content on different sized screens.

Linked jQuery to the website, so it can be used in the JavaScript code (e.g. to style elements).

```
1 <?php
2
3 /* === Clean input data === */
4 function testInput($data){
5     $data=trim($data);
6     $data=stripslashes($data);
7     $data=htmlentities($data);
8     return $data;
9 }
10
11 /* === checks and validates the email; passed === */
12 /* === Param 1: email to be checked === */
13 /* === Param 2: bool - if we should check in DB for the email's existence === */
14 /* === Param 3: bool - if we should check in DB for a copy === */
15 /* === returns: bool - false on error, otherwise true === */
16
17 function validEmail($email, $checkInDbForExistence, $checkInDbForCopy) {
18
19     $atIndex = strrpos($email, "@");
20
21     if (is_bool($atIndex) && !$atIndex)
22         return false;
23
24     $domain = substr($email, $atIndex+1);
25     $local = substr($email, 0, $atIndex);
26     $localLen = strlen($local);
27     $domainLen = strlen($domain);
28
29     //If local part length exceeded
30     if ($localLen < 1 || $localLen > 64)
31         return false;
32
33     //If domain part length exceeded
34     if ($domainLen < 1 || $domainLen > 255)
35         return false;
36
37     //If local part starts or ends with '.'
38     if ($local[0] == '.' || $local[$localLen-1] == '.')
39         return false;
40
41     //If local part has two consecutive dots
42     if (preg_match('/\\.\\./', $local))
43         return false;
44
45     // character not valid in domain part
46     if (!preg_match('/^([A-Za-z0-9\\-\\.])+$/i', $domain))
47         return false;
48
49     // domain part has two consecutive dots
50     if (preg_match('/\\.\\./', $domain))
51         return false;
```

The function validates the email to ensure that the user enters the correct email address. The \$email is the parameter that is passed into the function from the registration form.

Returns position of part of the string to determine the length and it checks to make sure the value is greater than 1 and less than 64. It also checks the domain is after the @ sign.

Cookie Policy

Cookies are important to the proper functioning of a site. To improve your experience, the cookie function has been created to remember login details and provide secure login, collect statistics to optimise site functionality and deliver content tailored to your interests. Click Agree and Proceed to accept cookies and go directly to the site or click on More Information to see detailed descriptions of the types of cookie and choose whether to accept certain cookies while on the site.

```
118
119
120 /* === checks and validates the password passed === */
121 /* === Param 1: $password to be checked === */
122 /* === returns: string on error, else null === */
123
124 function checkPwdErrors($password, $passwordVerification){
125
126     $password = testInput($password);
127     if (strlen($password) <= '6')
128         return "Your Password Must Contain At Least 6 Characters!";
129
130     // For security reasons we want the password to contain at least 1 number, this
131     // through the password field to make sure there is a number in the password
132     if(!preg_match("#[0-9]+#", $password))
133         return "Your Password Must Contain At Least 1 Number!";
134
135     // For security reasons we want the password to contain at least 1 capital letter,
136     // through the password field to make sure there is a capital letter in the password
137     if(!preg_match("#[A-Z]+#", $password))
138         return "Your Password Must Contain At Least 1 Capital Letter!";
139
140     // For security reasons we want the password to contain at least 1 lowercase letter,
141     // through the password field to make sure there is a lowercase letter in the password
142     if(!preg_match("#[a-z]+#", $password))
143         return "Your Password Must Contain At Least 1 Lowercase Letter!";
144
145     if($password !== $passwordVerification)
146         return "The entered password do not match!";
147
148     //No errors found, return null
149     return null;
150 }
```

Comments add above each statement. The purpose of this is to check when creating a new password it has the correct format. For security it was decided that it should be made up of uppercase, lowercase, and a number.

```
1 <?php
2 session_start();
3
4 if (!isset($_SESSION['user_id'])) {
5     header('Location: login.php');
6     exit;
7 }
8 ?>
```

```
12  */
13 (function( factory ){
14   if ( typeof define === 'function' && define.amd ) {
15     // AMD
16     define( ['jquery', 'datatables.net'], function ( $ ) {
17       return factory( $, window, document );
18     });
19   }
20 else if ( typeof exports === 'object' ) {
21   // CommonJS
22   module.exports = function (root, $) {
23     if ( ! root ) {
24       root = window;
25     }
26
27     if ( ! $ || ! $.fn.dataTable ) {
28       // Require DataTables, which attaches to jQuery, including
29       // jQuery if needed and have a $ property so we can access the
30       // jQuery object that is used
31       $ = require('datatables.net')(root, $).$;
32     }
33
34     return factory( $, root, root.document );
35   };
36 }
37 else {
38   // Browser
39   factory( jQuery, window, document );
40 }
41 })(function( $, window, document, undefined ) {
42 'use strict';
43 var DataTable = $.fn.DataTable;
```

```

1  (function () {
2    Number.prototype.toHex = function () {
3      var ret = ((this<0?0x8:0)+((this >> 28) & 0x7)).toString(16) + (this & 0xffffffff).toString(16);
4      while (ret.length < 8) ret = '0'+ret;
5      return ret;
6    };
7    Object.hashCode = function hashCode(o, l) {
8      l = l || 2;
9      var i, c, r = [];
10     for (i=0; i<l; i++)
11       r.push(i*268803292);
12     function stringify(o) {
13       var i,r;
14       if (o === null) return 'n';
15       if (o === true) return 't';
16       if (o === false) return 'f';
17       if (o instanceof Date) return 'd:'+(o+0);
18       i=typeof o;
19       if (i === 'string') return 's:' + o.replace(/([\\\";])/g, '\\$1');
20       if (i === 'number') return 'n:' + o;
21       if (o instanceof Function) return 'm:' + o.toString().replace(/([\\\";])/g, '\\$1');
22       if (o instanceof Array) {
23         r=[];
24         for (i=0; i<o.length; i++)
25           r.push(stringify(o[i]));
26         return 'a:' + r.join(',');
27       }
28       r=[];
29       for (i in o) {
30         r.push(i+':'+stringify(o[i]));
31       }
32       return 'o:' + r.join(',');
33     }
34     o = stringify(o);
35     for (i=0; i<o.length; i++) {
36       for (c=0; c<r.length; c++) {
37         r[c] = (r[c] << 13)-(r[c] >> 19);
38         r[c] += o.charCodeAt(i) << (r[c] % 24);

```

Hashing Function, using JS
allows the password to scramble
to increase security.

C:\Users\19741\Documents\Year 13 CS\Unit 3 Projects\Jack Male\Jack_Male\Project DB - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X

main.py inventory2020.py diet_statistics.php package-lock.json training_videos.php blog.php php.xml workspace.xml modules.xml fitness_tracker.php ESP_Task_V4_21-07-2020.py ProjectDB

```

1 CREATE TABLE [import].[ExerciseLog]
2 (
3   [ExerciseLogID] INT NOT NULL CONSTRAINT PK_ExerciseLogID PRIMARY KEY,
4   [ExerciseID] SMALLINT NOT NULL,
5   [ActivityDate] DATETIME2 NOT NULL,
6   [ExerciseCount] TINYINT NOT NULL,
7   [GymUserId] INT NOT NULL,
8   [FirstInserted] DATETIME2 NOT NULL DEFAULT getdate(),
9   CONSTRAINT [FK_ExerciseLog_Exercise] FOREIGN KEY ([ExerciseId]) REFERENCES [import].[Exercise]([ExerciseId]),
10  CONSTRAINT [FK_ExerciseLog_GymUser] FOREIGN KEY ([GymUserId]) REFERENCES [import].[GymUser]([GymUserId])
11 )
12

```

```
1 <?php
2 session_start();
3
4 require_once "setupDb.php";
5
6 $password = $passwordError = $oldPassword = $oldPasswordError = $confirmPass = $confirmPassError =
7
8 function filterInput($data) {
9     $data = trim($data);
10    $data = stripslashes($data);
11    $data = htmlspecialchars($data);
12    return $data;
13}
14
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16    $oldPassword = filterInput($_POST["oldPassword"]);
17    $password = filterInput($_POST["newPassword"]);
18    $confirmPass = filterInput($_POST["passConfirm"]);
19
20    if (empty($oldPassword)) {
21        $oldPasswordError = "Please enter your old password";
22        $_SESSION["passError"] = $oldPasswordError;
23    }
24    elseif(empty($password)) {
25        $passwordError = "Please enter your new password";
26        $_SESSION["passError"] = $passwordError;
27    }
28    elseif (empty($confirmPass)) {
29        $confirmPassError = "Please confirm your new password";
30        $_SESSION["passError"] = $confirmPassError;
31    }
32    else {
33        if ($password == $confirmPass) {
34            //SQL update statement to change user information
35            $sql = "SELECT passwordHash FROM userinfo WHERE email
36
37            if ($stmt = $connection->prepare($sql)) {
38                //binds variables to parameters
39                $stmt->bind_param("s", $param_email);
40
41                //set parameters
42                $param_email = $_SESSION["email"];
43
44                if ($stmt->execute()) {
45                    $stmt->store_result();
46
47                    $stmt->bind_result($passwordHash);
48                    if ($stmt->fetch()) {
49                        if (password_verify($oldPassword, $passwordHash)) {
50                            //SQL statement below updates data on the database
51                            $sql = "UPDATE userinfo SET passwordHash = ? WHERE email = ?";
52
53                            if ($stmt->execute()) {
54                                echo "Your password has been updated!";
55                            }
56                        }
57                    }
58                }
59            }
60        }
61    }
62
63    if ($passwordError != "") {
64        echo "Old Password Error: " . $oldPasswordError;
65    }
66    if ($passwordError != "") {
67        echo "New Password Error: " . $passwordError;
68    }
69    if ($confirmPassError != "") {
70        echo "Confirm Password Error: " . $confirmPassError;
71    }
72
73    if ($oldPasswordError == "" && $passwordError == "" && $confirmPassError == "") {
74        echo "Your password has been updated!";
75    }
76
77    $stmt->close();
78    $connection->close();
79
80    exit();
81}
```

This function allows the user to reset their password, checks the person's existing email and binds the two together.

These 3 if statements check if the user entered a value in the input boxes.

This SQL statement selects the current password hash of the user from the database. This is done to compare to the user's old password input (if the input = password hash, then the program continues).

```

52
53     if ($stmt = $connection->prepare($sql)) {
54         //binds variables to parameters
55         $stmt->bind_param("ss", $param_passwordHash, $param_email);
56
57         //parameters are set
58         //converts new password into hash
59         $param_passwordHash = password_hash($password, PASSWORD_DEFAULT);
60         $param_email = $_SESSION["email"];
61
62         if ($stmt->execute()) {
63             //changes password characters into asterisks
64             $password = str_repeat("*", strlen($password));
65             $_SESSION["password"] = $password;
66             $_SESSION["passError"] = "";
67         }
68     }
69     $stmt->close();
70 }
71 else {
72     $_SESSION["passError"] = "Old password is not correct";
73 }
$connection->close();
75
76
77 else {
78     //If the SQL statement didn't work, the resulting phrase will be outputted (in the profile page using session variable)
79     $_SESSION["passError"] = "Sorry, something went wrong, please try again";
80 }
81
82 $stmt->close();
83
84 else {
85     $confirmPassError = "Passwords do not match";
86     $_SESSION["passError"] = $confirmPassError;
87 }
88
89 $connection->close();
90 }
91
92 //returns to profile page
93 header("location: profile.php");
94 exit;
95 ?>

```

Testing

Test data	Expected outcome	Reasons for test	Actual outcome
email: a@a password: a	Login normally	Check if valid inputs work	As expected, the user is logged in
Test data Empty inputs	No login	Check if erroneous inputs not in database won't work	As expected, user is not logged in
Email: ab@abc Password: 1111	No login	Check if valid inputs won't work if not in existing database	As expected, user is not logged in
Empty inputs	No login	Check if invalid inputs won't work	As expected, user is not logged in due to lack of inputs

Sign up

Password: 4444 confirm password: 4444			
Title: Mr First name: qqq Last name: sss Email: oooo@a Phone no: abc Password: sss	Account is not made	To check if website recognises invalid phone number (not only being numbers)	As expected, no account is made

Title: Mr First name: qqq Last name: sss Email: oooo@a Phone no: abc Password: sss Confirm password: sss2	Account is not made	To check if website checks if password = confirmed password	As expected, no account is made
Title: Mr First name: qqq Last name: sss Email: oooa Password: sss Confirm password: sss2	Account is not made	To check if website checks validity of email (syntax)	As expected, account is not made

Dashboard

Element to be tested	Expected result	Test data	Expected result from test data	Actual result
Header tab navigation system	When user clicks on Buttons system directs user to right webpage.	n/a	n/a	System successfully directs user to requested webpage.

Sign up validation	<p>When user inputs data, the system checks the data and outputs an error if the input isn't inappropriate for the field.</p> <p>First name and last name must only contain letters and not numbers or special characters.</p> <p>The email field must contain the "@" and ":".</p> <p>There must be at</p>	<p>First name: Chris 555</p> <p>Last name: Jones</p> <p>Email: ChrisJones@gmail.com (email already registered)</p>	<p>Error pop-up and form cannot be submitted</p>	<p>System directs user to an error page if the email is not unique. However, doesn't validate other values such as first name, last name, email syntax, postcode and password and submits the data into the database.</p>
Dashboard	View dashboard	Imported data from user input	Viewed dashboard and all graphs	No issues and all graphs could be seen.
Graphs	View graphs, bar, pie charts	Imported data	Viewed graphs with relevant statistics	No issues

Video	View video	YouTube videos	Select and view video for fitness and diet	No issues
Fitness diary	View tabular data	Date: December Exercise: run, swim and walk Calories: range Distance:	View data in tabular format with range of dates, exercises, calories	No issues
Diet tracker	Inputs various food and weight in grams	Chicken 400 grams	Submit data for tracking	No issues

Test Results

Outcome for dashboard

The outcome was a successful display of the dashboard and graphs displaying the correct data as per design.

Detailed description: This screenshot shows the Toka Fitness dashboard. On the left is a green sidebar with navigation links. The main area has three horizontal bars at the top: Daily Calorie Intake (0/5.4), Daily Carb Intake (7/10), and Daily Fat Intake (5/10). Below these is a line graph titled 'Weight overview' showing weight from January to May. To the right is a donut chart titled 'Daily Nutrition' showing proportions of Carbohydrates, Fat, and Protein.

Date	Exercise	Calories burnt
20th December 2019	Run	300
20th December 2019	Swim	420
19th December 2019	Walk	142
18th December 2019	Run	300
17th December 2019	Swim	420
17th December 2019	Walk	142
16th December 2019	Run	300
15th December 2019	Swim	420
14th December 2019	Walk	142

Test outcome fitness diary

Test successful the fitness diary for days of December were displayed in tabular format and provided the correct data for calories burnt.

Please check the wording

The user can select the monthly report; this can be printed out, based on the data seen in the history above.

Detailed description: This screenshot shows the Toka Fitness Fitness Diary. The left sidebar has a 'History' link. The main area displays a table of exercises from December 14th to 20th, 2019. The top right corner shows an 'ALERTS CENTER' with two notifications: one about a new monthly report ready to download and another about achieving a monthly goal. A URL at the bottom left is: file:///C:/Users/19741/Documents/Revisor%20work/T%20Levels/GSEMs_AgainstTokafit/YoKaFit/fitness_diary.php#.

Date	Exercise	Calories burnt
20th December 2019	Run	300
20th December 2019	Swim	420
19th December 2019	Walk	142
18th December 2019	Run	300
17th December 2019	Swim	420
17th December 2019	Walk	142
16th December 2019	Run	300
15th December 2019	Swim	420
14th December 2019	Walk	142

Test outcome diet tracker

The screenshot shows the Toka Fitness mobile application interface. On the left is a green sidebar with various navigation options: Dashboard, YOUR HEALTH (Fitness, Diet), SOCIAL (Follow us, Blog, Chat, HELP), TRAINING (Training Videos, Food Plan Videos), and HOW TO USE (How to use Toka Fitness, Get Premium). The main content area is titled "Diet Tracker" and contains a form titled "Add exercise data". The form has fields for "Food" (set to "Chicken") and "Amount (grams)" (set to "400"), with a "Submit" button below. A black arrow points from the text box in the "Food" field towards a callout box on the right.

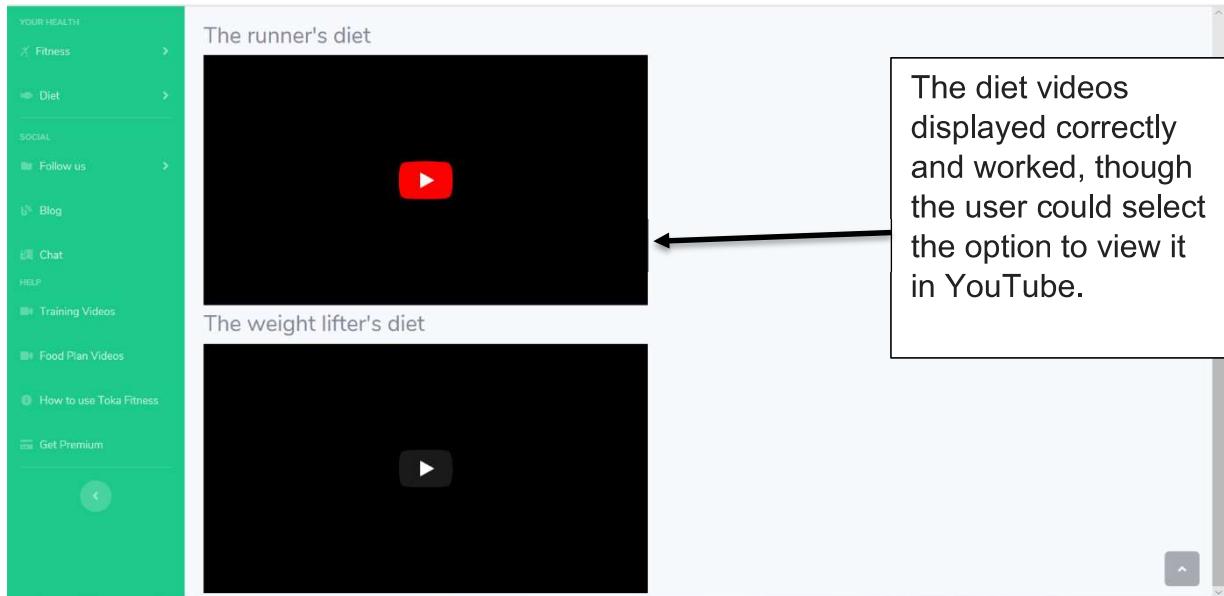
Diet tracker has worked as user may eat various foods and it was better to enter the food and weight.

Test outcome training video

The screenshot shows the Toka Fitness mobile application interface. The sidebar on the left is identical to the previous screenshot. The main content area displays two video thumbnails. The top thumbnail is titled "How to gain muscle weight fast" and the bottom one is titled "How to run fast". Each thumbnail features a large red play button in the center. A black arrow points from the text box in the "Food" field of the Diet Tracker screenshot towards the "How to run fast" video thumbnail.

The training videos displayed correctly and worked, though the user could select the option to view it in YouTube.

Test outcome diet videos



Profile info editing

Test data	Expected outcome	Reasons for test	Actual outcome
No inputs	Website asks for inputs	Check if lack of inputs is detected	As expected, error message is displayed
Valid inputs (e.g. abc for first name)	Website accepts input and info is	Check if valid inputs are accepted	As expected, error message is displayed
Old password is incorrect	Website tells user the input is wrong	Check if incorrect data is detected	As expected, error message is displayed

Old password is correct New password = ccc Confirm password = asdf	Website tells user passwords don't match	Check if code works as it should	As expected, the user is notified of the exact problem
--	--	----------------------------------	--

Unit to test sign-in assumption

Unit to test Sign-in

Assumption The webpage displayed the sign-in input screen and waited for user's action.

Input user ID, user password, user information

Expect output

Register successfully with proper input, register unsuccessfully with improper input, and make alter the same time.

Pass Function fits the requirement.

Fail and probable error

(1) Register still success with improper input. Error related to input checking.

(2) Register success with proper input but no related data shown in database. Error related to connection with database.

Syntax Issues

The screenshot shows a code editor with a PHP script. There are two red circles highlighting parts of the code:

- A red circle highlights the line `$query = $connection->prepare("SELECT * FROM users WHERE EMAIL = :email");`. An arrow points from this circle to a callout box containing the following text:
- A red circle highlights the line `} else { echo "<h1><center>Something went wrong!</center></h1>"; } }`. An arrow points from this circle to another callout box containing the following text:

Few issues when trying to run the code, added a \$ to the if which made it variable, removed this and program worked.

```
21     if ($query->rowCount() > 0) {
22         echo '<h1><font color="white"><center>The email address is already registered!</center></font>
23     }
24 }
```

```

1 <?php
2
3 include('config.php');
4 session_start();
5
6 if (isset($_POST['register'])) {
7
8     $firstname = $_POST['firstname'];
9     $lastname = $_POST['lastname'];
10    $email = $_POST['email'];
11    $password = $_POST['password'];
12    $repeatpassword = $_POST['repeatpassword'];
13    if($password !== $repeatpassword) {
14        echo '<h1><font color="white"><center>Password and repeat password do not match!</center></font></h1>';
15    } else {
16        $password_hash = password_hash($password, PASSWORD_BCRYPT);
17
18        $query = $connection->prepare("SELECT * FROM users WHERE EMAIL=:email");
19        $query->bindParam("email", $email, PDO::PARAM_STR);
20        $query->execute();
21
22        if ($query->rowCount() > 0) {
23            echo '<h1><font color="white"><center>The email address is already taken!</center></font></h1>';
24        }
25
26        if ($query->rowCount() == 0) {
27            $query = $connection->prepare("INSERT INTO users(FIRSTNAME, LASTNAME, PASSWORD_HASH, EMAIL) VALUES (:firstname, :lastname, :password_hash, :email)");
28            $query->bindParam("firstname", $firstname, PDO::PARAM_STR);
29            $query->bindParam("lastname", $lastname, PDO::PARAM_STR);
30            $query->bindParam("password_hash", $password_hash, PDO::PARAM_STR);
31            $query->bindParam("email", $email, PDO::PARAM_STR);
32            $result = $query->execute();
33
34            if ($result) {
35                echo '<h1><font color="white"><center>Your registration was successful!</center></font></h1>';
36            } else {
37                echo '<h1><font color="white"><center>Something went wrong!</center></font></h1>';
38            }
39        }
40    }
41 }

```

Another problem was the semicolon at the end of the statement. This was corrected and loaded the login page.

Another problem that was hard to detect was missed curly brackets. Added the missing brackets and the code interpreted fine.

```

24
25
26        if ($query->rowCount() == 0) {
27            $query = $connection->prepare("INSERT INTO users(FIRSTNAME, LASTNAME, PASSWORD_HASH, EMAIL) VALUES (:firstname, :lastname, :password_hash, :email)");
28            $query->bindParam("firstname", $firstname, PDO::PARAM_STR);
29            $query->bindParam("lastname", $lastname, PDO::PARAM_STR);
30            $query->bindParam("password_hash", $password_hash, PDO::PARAM_STR);
31            $query->bindParam("email", $email, PDO::PARAM_STR);
32            $result = $query->execute();
33
34            if ($result) {
35                echo '<h1><font color="white"><center>Your registration was successful!</center></font></h1>';
36            } else {
37                echo '<h1><font color="white"><center>Something went wrong!</center></font></h1>';
38            }
39        }
40    }
41 }
?>

```